

Oris accounting Web API

It is necessary to [\[LogIn\]](#) to the system first of all and get token. Token is passed to all methods of web API as a parameter. In case testing ask the company "Oris" to provide you with the [\[LogIn\]](#) parameters. Otherwise ask server IT administrator.

Method list

Accounting

- .. Chart of accounts
 - [\[Account\]](#) *Adds, Updates or Deletes* account.
 - [\[AccountsList\]](#) *Gets list of accounts.*
 - [\[GetDebitorCreditorAccountByParent\]](#) *Gets or Adds debtor/creditor account.*
 - [\[GetItemAccountByParent\]](#) *Gets or Adds Item account.*
 - [\[GetProgenyOA_AccountsIDs\]](#) *Gets all sub accounts IDs for parent account.*
- .. Entries
 - [\[Transaction\]](#) *Adds, Updates or Deletes* accounting transaction.
 - [\[GetTransactionByID\]](#) *Gets one accounting transaction and its entries.*
 - [\[EntriesList\]](#) *Gets list of accounting entries.*
 - [\[TransactionSetTaxDocument\]](#) *Sets revenue service Tax Document number for the transaction.*
 - [\[TransactionSetVAT\]](#) *Sets revenue service VAT number for the transaction.*
 - [\[TransactionSetWaybill\]](#) *Sets revenue service Way bill number for the transaction.*
- .. Start balance
 - [\[SetStartBalance\]](#) *Sets account start balance.*
 - [\[StartBalancesGetStartDate\]](#) *Gets start balance date.*
 - [\[StartBalancesSetStartDate\]](#) *Sets start balance date.*
 - [\[StartBalancesList\]](#) *Gets start balances list of accounts.*
- .. Managerial transactions
 - Bank transfers
 - [\[BankTransfer\]](#) *Adds, Updates or Deletes* bank transfer.
 - [\[GetBankTransferByID\]](#) *Gets one bank transfer and its accounting entries.*
 - [\[BankTransfersList\]](#) *Gets list of bank transfer.*
 - Bank transfers (foreign currency)
 - [\[BankTransferForeign\]](#) *Adds, Updates or Deletes* bank transfer foreign.
 - [\[GetBankTransferForeignByID\]](#) *Gets Bank Transfer Foreign and its accounting entries.*
 - [\[BankTransferForeignsList\]](#) *Gets list of bank transfer foreign.*
- Cash income orders
 - [\[CashIncomeOrder\]](#) *Adds, Updates or Deletes* cash income order.
 - [\[GetCashIncomeOrderByID\]](#) *Gets one cash income order and its accounting entries.*
 - [\[CashIncomeOrdersList\]](#) *Gets list of cash income order.*
- Cash outlay orders
 - [\[CashoutlayOrder\]](#) *Adds, Updates or Deletes* cash outlay order.
 - [\[GetCashoutlayOrderByID\]](#) *Gets one cash outlay order and its accounting entries.*
 - [\[CashoutlayOrdersList\]](#) *Gets list of cash outlay order.*
- .. Results
 - [\[BalancesList\]](#) *Gets balances by chart of accounts.*
 - [\[TurnoverList\]](#) *Gets turnover by chart of accounts.*
 - [\[AccountBalancesList\]](#) *Gets all balances for one account.*
 - [\[AccountStatementList\]](#) *Gets account statement.*
 - [\[AccountStatementDetailedList\]](#) *Gets account statement detailed.*

- ... [JournalOrderList] [Gets account journal order.](#)
- ... [CashflowsList] [Gets cashflow.](#)
- ... [MemorialList] [Gets memorial.](#)
- ... [DebitorCreditorBalancesList] [Gets debitor/creditor balances.](#)
- ... [DebitorCreditorTurnoverList] [Gets debitor/creditor turnover.](#)
- ... [TrialBalanceList] [Gets trial balance.](#)
- ... [ProfitAndLossList] [Gets profit and loss.](#)
- ... [TransactionsList] [Gets list of transactions.](#)
- ... [DualEntriesList] [Gets list of entries in dual entries style.](#)

Managerial accounting

- .. Proforma invoices
 - ... [ProformaInvoice] [Adds, Updates or Deletes](#) proforma invoice.
 - ... [GetProformaInvoiceByID] [Gets proforma invoice and its item list.](#)
 - ... [ProformaInvoicesList] [Gets list of proforma invoices.](#)
- .. Accept operations
 - ... [AcceptOperation] [Adds, Updates or Deletes](#) accept operation.
 - ... [GetAcceptOperationByID] [Gets accept operation and its item and expencis list.](#)
 - ... [AcceptOperationsList] [Gets list of accept operation.](#)
- .. Transfer operations
 - ... [TransferOperation] [Adds, Updates or Deletes](#) transfer operation.
 - ... [GetTransferOperationByID] [Gets transfer operation and its item list.](#)
 - ... [TransferOperationsList] [Gets list of transfer operation.](#)
- .. Supply operations
 - ... [SupplyOperation] [Adds, Updates or Deletes](#) supply operation.
 - ... [GetSupplyOperationByID] [Gets supply operation and its item list.](#)
 - ... [SupplyOperationsList] [Gets list of supply operation.](#)
- .. Write off operations
 - ... [WriteOffOperation] [Adds, Updates or Deletes](#) writeOff operation.
 - ... [GetWriteOffOperationByID] [Gets writeOff operation and its item list.](#)
 - ... [WriteOffOperationsList] [Gets list of writeOff operation.](#)
- .. Return to supplier operations
 - ... [ReturnToSupplierOperation] [Adds, Updates or Deletes](#) return to supplier operation.
 - ... [GetReturnToSupplierOperationByID] [Gets return to supplier operation and its item list.](#)
 - ... [ReturnToSupplierOperationsList] [Gets list of return to supplier operation.](#)
- .. Return from customer operations
 - ... [ReturnFromCustomer] [Adds, Updates or Deletes](#) return from customer operation.
 - ... [GetReturnFromCustomerByID] [Gets return from customer operation and its item list.](#)
 - ... [ReturnFromCustomersList] [Gets list of return from customer operation.](#)
- .. Inventory
 - ... [Inventory] [Adds, Updates or Deletes](#) inventory operation.
 - ... [GetInventoryByID] [Gets inventory operation and its item list.](#)
 - ... [InventoriesList] [Gets list of inventory operation.](#)
- .. Orders to suppliers
 - ... [OrderToSupplier] [Adds, Updates or Deletes](#) order to supplier operation.
 - ... [GetOrderToSupplierByID] [Gets order to supplier operation and its item list.](#)
 - ... [OrderToSuppliersList] [Gets list of order to supplier operation.](#)
- .. Transfer Orders
 - ... [OrderOfTransfer] [Adds, Updates or Deletes](#) order of transfer operation.
 - ... [GetOrderOfTransferByID] [Gets order of transfer operation and its item list.](#)
 - ... [OrderOfTransfersList] [Gets list of order of transfer operation.](#)
- .. Orders from buyers
 - ... [OrderFromByuer] [Adds, Updates or Deletes](#) order from buyer operation.
 - ... [GetOrderFromByuerByID] [Gets order from buyer operation and its item list.](#)
 - ... [OrderFromByuersList] [Gets list of order from buyer operation.](#)
- .. Results
 - ... [GetItemsBalancesList] [Gets items balances list.](#)

- ... [GetItemsItemsTurnoversList] [Gets items turnover list.](#)
- ... [GetItemsItemsTurnoverByPeriodList] [Gets items turnover by period list.](#)

Salary

- ... [Employee] [Adds](#), [Updates](#) or [Deletes](#) employee.
- ... [EmployeesList] [Gets list of employee.](#)
- ... [EmployeeDepartment] [Adds](#), [Updates](#) or [Deletes](#) employee department.
- ... [EmployeeDepartmentsList] [Gets list of employee departments.](#)
- ... [EmployeePositionsList] [Gets list of employee positions.](#)
- ... [GetDataTableEmployeeMonthAccrueds] [Gets list of month accrued.](#)
- ... [GetDataTableEmployeeResults] [Gets list of.](#)
- ... [GetDataTableEmployeeYearAccrueds] [Gets list of.](#)
- ... [GetDataTableReportNumbers] [Gets list of.](#)
- ... [GetDataTableReportResult] [Gets list of.](#)
- ... [GetEmployeeDepartmentByID] [Gets one employee department and its related positions list.](#)
- .. Archive
- ... [GetDataTableSalaryResultArchive] [Gets salary results list.](#)
- ... [GetDataTableSalaryAccruedsArchives] [Gets salary accrued list.](#)

Fixed assets

- ... [FixedAsset] [Adds](#), [Updates](#) or [Deletes](#) fixed asset.
- ... [GetFixedAssetsList] [Gets list of fixed assets.](#)
- ... [GetFixedAssetsStatementsList] [Gets list of fixed asset statement.](#)
- ... [GetFixedAssetsStatementsDetailedList] [Gets list of fixed asset statement detailed.](#)

Manufacturing

- ... [ManufacturingDesign] [Adds](#), [Updates](#) or [Deletes](#) manufacturing design.
- ... [GetManufacturingDesignByID] [Gets one manufacturing design and its related materials list.](#)
- ... [ManufacturingDesignsList] [Gets list of manufacturing design.](#)
- ... [CostDriver] [Adds](#), [Updates](#) or [Deletes](#) cost driver.
- ... [GetCostDriverByID] [Gets one cost driver and its related products list.](#)
- ... [CostDriversList] [Gets list of cost driver.](#)
- ... [ManufacturingPlanning] [Adds](#), [Updates](#) or [Deletes](#) manufacturing planning.
- ... [GetManufacturingPlanningByID] [Gets one manufacturing planning and its related materials list.](#)
- ... [ManufacturingPlanningsList] [Gets list of manufacturing planning.](#)
- ... [ManufacturingExecution] [Adds](#), [Updates](#) or [Deletes](#) manufacturing execution.
- ... [GetManufacturingExecutionByID] [Gets one manufacturing execution and its related materials list.](#)
- ... [ManufacturingExecutionsList] [Gets list of manufacturing execution.](#)

Related tables

- .. Debitor/Creditor (organization or person)
- ... [DebitorCreditor] [Adds](#), [Updates](#) or [Deletes](#) debitor/creditor.
- ... [GetDebitorCreditorByID] [Gets one debitor/creditor and its bank accounts, addresses, phones and legal identifiers lists.](#)
- ... [DebitorCreditorsList] [Gets list of debitor/creditors.](#)
- ... Organization
- ... [Organization] [Adds](#), [Updates](#) or [Deletes](#) organization.
- ... [GetOrganizationByID] [Gets one organization and its bank accounts, addresses and phones lists.](#)
- ... [OrganizationsList] [Gets list of organizations.](#)
- ... Person
- ... [Person] [Adds](#), [Updates](#) or [Deletes](#) person.

. **[GetPersonByID]** [Gets one person and its bank accounts, addresses, phones and legal identifiers lists.](#)
 **[PersonsList]** [Gets list of persons.](#)
 .. Item (commodity, fixed asset or service)
 **[Item]** [Adds, Updates or Deletes](#) item (commodity).
 **[GetItemByID]** [Gets one item \(commodity\) and its related units list.](#)
 **[ItemsList]** [Gets list of items \(commodity, fixed asset or service\).](#)
 **[Service]** [Adds, Updates or Deletes](#) item (Service).
 **[ServicesList]** [Gets list of services.](#)
 .. Stock unit
 **[StockUnit]** [Adds, Updates or Deletes](#) stock unit.
 **[StockUnitsList]** [Gets list of stock units.](#)
 .. Currency
 **[Currency]** [Adds, Updates or Deletes](#) currency.
 **[CurrenciesList]** [Gets list of currencies and rates for a specified date.](#)
 .. Currency rate
 **[CurrencyRate]** [Adds, Updates or Deletes](#) currency rate.
 **[CurrencyRatesList]** [Gets list of currencies and rates for a specified period.](#)
 .. Cost centre
 **[CostCentre]** [Adds, Updates or Deletes](#) cost centre.
 **[CostCentresList]** [Gets list of cost centres.](#)
 .. Cost unit
 **[CostUnit]** [Adds, Updates or Deletes](#) cost unit.
 **[CostUnitsList]** [Gets list of cost units.](#)
 .. Comment
 **[Comment]** [Adds, Updates or Deletes](#) comment.
 **[CommentsList]** [Gets list of comments.](#)
 .. Transaction type
 **[TransactionType]** [Adds or Updates](#) transaction type.
 **[TransactionTypesList]** [Gets list of transaction types.](#)
 .. Transactions link
 **[TransactionLink]** [Adds, Updates or Deletes](#) transaction link.
 **[GetTransactionLinkByID]** [Gets list of related and depended IDs for a single transaction.](#)
 .. Manufacturer
 **[Manufacturer]** [Adds, Updates or Deletes](#) manufacturer.
 **[ManufacturersList]** [Gets list of manufacturers.](#)
 .. City
 **[City]** [Adds, Updates or Deletes](#) city.
 **[CitiesList]** [Gets list of cities.](#)
 .. Country
 **[Country]** [Adds, Updates or Deletes](#) country.
 **[CountriesList]** [Gets list of countries.](#)
 .. Bank
 **[Bank]** [Adds, Updates or Deletes](#) bank.
 **[BanksList]** [Gets list of banks.](#)
 .. Vehicle
 **[Vehicle]** [Adds, Updates or Deletes](#) vehicle.
 **[VehiclesList]** [Gets list of Vehiclevs.](#)
 .. Sale price type
 **[SalePriceType]** [Adds, Updates or Deletes](#) sale price type.
 **[SalePriceTypesList]** [Gets list of sale price types.](#)
 .. Status
 **[Status]** [Adds, Updates or Deletes](#) status.
 **[StatusesList]** [Gets list of status.](#)

Additional tables

.. Debitor/Creditor

- ... **[DebitorCredtorGroup]** Adds, Updates or Deletes debitor/credtor group.
- ... **[DebitorCredtorGroupsList]** Gets list of debitor/credtor groups.
- .. Item
 - ... **[ItemGroup]** Adds, Updates or Deletes item group.
 - ... **[ItemGroupsList]** Gets list of item groups.
- .. Cost centre
 - ... **[CostCentreGroup]** Adds, Updates or Deletes cost centre group.
 - ... **[CostCentreGroupsList]** Gets list of cost centre groups.
- .. Bank transfer
 - ... **[BankTransferTaxDepartCodes]** Adds, Updates or Deletes tax department code.
 - ... **[BankTransfersTaxDepartCodesList]** Gets list of tax department codes.
 - ... **[BankTransferTransitCodes]** Adds, Updates or Deletes transit code.
 - ... **[BankTransfersTransitCodesList]** Gets list of transit codes.
 - ... **[BankTransferTreasuryCode]** Adds, Updates or Deletes treasury code.
 - ... **[BankTransferTreasuryCodesList]** Gets list of treasury codes.
- .. Cashflow
 - ... **[CashflowAccount]** Adds, Updates or Deletes cashflow account.
 - ... **[CashflowAccountsList]** Gets list of cashflow accounts.
- .. Cost centre
 - ... **[CostCentreGroup]** Adds, Updates or Deletes cost centre group.
 - ... **[CostCentreGroupsList]** Gets list of cost centre groups.
- .. Fixed assets
 - ... **[FixedAssetsFormulasList]** Gets list of fixed assets formulas.
- .. Profit and loss
 - ... **[ProfitAndLossCategory]** Adds, Updates or Deletes profit and loss category.
 - ... **[ProfitAndLossCategoriesList]** Gets list of profit and loss categories.
- .. Salary
 - ... **[GetDataTableSalaryAccounts]** Gets list of salary accounts.
 - ... **[SalaryAccount]** Adds, Updates or Deletes salary account.
 - ... **[SalaryAccountsList]** Gets list of salary accounts.
 - ... **[SalaryAccruedField]** Adds, Updates or Deletes salary accrued field.
 - ... **[SalaryAccruedFieldsList]** Gets list of salary accrued fields.
 - ... **[SalaryResultField]** Adds, Updates or Deletes salary result field.
 - ... **[SalaryResultFieldsList]** Gets list of salary result fields.
- .. Trial balance
 - ... **[TrialBalanceAccountsRelation]** Adds, Updates or Deletes accounts relation for trial balance.
 - ... **[TrialBalanceAccountsRelationsList]** Gets list of accounts relations for trial balance.
 - ... **[TrialBalanceActiveAccount]** Adds, Updates or Deletes active account for trial balance.
 - ... **[TrialBalanceActiveAccountsList]** Gets list of trial balance active accounts for trial balance.
 - ... **[TrialBalanceCategory]** Adds, Updates or Deletes trial balance category for trial balance.
 - ... **[TrialBalanceCategoriesList]** Gets list of trial balance categories for trial balance.
 - ... **[TrialBalancePassiveAccount]** Adds, Updates or Deletes trial balance apssive for trial balance.
 - ... **[TrialBalancePassiveAccountsList]** Gets list of trial balance passive accounts for trial balance..

LogIn

POST/api/LogIn

In case testing ask the company "Oris" to provide you with the [LogIn] parameters. Otherwise ask server IT administrator. Returns token string.

Schema

```
{
  "user": "string",
```



```

"password": "string",
"language": "string",
"databaseName": "string",
"databaseUserName": "string",
"databaseUserPassword": "string",
"databaselsLocal": true,
"localDatabasePath": "string",
"sqlServer": "string",
"useWindowsAuthentication": true
}

```

Field description

"user": Oris Accounting user name
"password": Oris Accounting password
"language": Language for error description: English (Default) = "en-EN", Georgian = "ka-GE", Russian = "ru-RU"
"databaseName": Database name
"databaseUserName": Database user name
"databaseUserPassword": Database user password
"databaselsLocal": If compact Edition database = true, for MS SQL server database = false
"localDatabasePath": If [DatabaselsLocal]=true, the path to the local database file located on the server
"sqlServer": MS SQL Server name
"useWindowsAuthentication": Use windows authentication for MS SQL Server

Examples

1. Log in with user name "Administrator" with password = "admin1234". Rest of the parameters will be taken from the server configuration file **C:\OAWebAPI\OAWebAPI.config**.

Request body:

```

{
  "user": "Administrator",
  "password": "admin1234"
}

```

Server response Code(200):

```

{
  "token":
  "dyQMHU2beL+SzJKntVQDjr22qib7hCZBkQ+YDmnyv6xNN+bdO4HlBx/nx/guHwQyJlzKIB65P/1PR6EFWTLwoP/jvv6
  1IEoEQwcOn1xS0Hz6YOLfL2BlN8Hhzzwu+SjJWopE0Kt7gYPPCGVCPBlpRd8sqgwWbKFqwg5329F5pxjkqx1kyJg0O
  Y97EnayMeBxpADsPZpyFi5J8bjeoaedqKpuRIWSRGD2B79yakC+kH91XafC4FwdeWcf6DID99OPF5iLvu1INNM6Tt5
  71hiEdYiSSECKC1JM6J12sSCzVG09/eycX5cLzPLrbaATDZ5AoVAfoMkLkEYExmf4b9/GU+Qz4J4KkKF7RVF1rV5N
  AslJKCSuwGm5V/d4PK+AjnXeGEAHapwTYU4t59GU356GhuC9q4iKqdvGlqmf7tsQuF8Gaz3ufCYbnmY+DL1nBE1j"
}

```

Server response Code(400):

```

[
  {
    "errorNumber": 51107,
    "description": "Error SQL database connection!",
    "comment": "Can not connect to database.[NDatabasea]\r\nContact your SQL server administrator.\r\n[4060]
    Cannot open database \"NDatabasea\" requested by the login. The login failed.[18456] Login failed for user 'sa'.",
    "command": "",
    "errorType": "ErrorType_Permission",
    "source": "OA_BusinessObject.dll",
    "class": "DataBase",
    "entryPoint": "IDatabaseConnection",
    "resourceDLL_ForTranslation": "OrisAccounting",
    "errorPointer": 0
  }
]

```

2. Log in to SQL server – “.\ORIS”, SQL database – “MyDatabase” for user name - “Administrator” with password = “admin1234” and database user name - “sa” and database password – “admin987654”. Language for error description – Georgian.

Request body:

```

{
  "user": "Administrator",

```

```

"password": "admin1234",
"language": "ka-GE",
"databaseName": "MyDatabase",
"databaseUserName": "Administrator",
"databaseUserPassword": "admin987654",
"sqlServer": ".\\ORIS"
}

```

3. Log in to SQL CE (Local data base) located in “C:\ORIS Accounting 7.0\Data\Default.sdf” file, for user name - “Administrator” with password = “admin0”. Language for error description – English.

Request body:

```

{
"user": "Administrator",
"password": "admin0",
"language": "en-EN",
"databaseIsLocal": true,
"localDatabasePath": "C:\\ORIS Accounting 7.0\\Data\\Default.sdf"
}

```

Accounting - Chart of accounts

Account

POST/api/Account

Adds new account. Returns OK with ID value or Error description Json.

Schema

```

{
"token": "string",
"accountCategoryInt": 0,
"accountComment": "string",
"accountEnableMinusStockBalance": true,
"accountIsClosed": true,
"accountName": "string",
"accountNonbalance": true,
"accountNumber": "string",
"accountTypeInt": 0,
"accountWriteOffMethodInt": 0,
"oA_DebitorCreditorID": 0,
"oA_ItemsID": 0,
"parentAccountNumber": "string"
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"accountCategoryInt": Financial = 0, Item = 1, DebitorCreditor = 2. Default=0

"accountComment":

"accountEnableMinusStockBalance":

"accountIsClosed":

"accountName": Sets account name if [AccountCategoryInt]=0.

"accountNonbalance":

"accountNumber":

"accountTypeInt": AccountTypeInt: Active = 0, Passive = 1. Default gets from parent account

"accountWriteOffMethodInt": AccountWriteOffMethodInt: None = 0, FIFO = 1, LIFO = 2, Avarage = 3. Default=0

"oA_DebitorCreditorID": If specified than account category will be 'DebitorCreditor' and [Name] will be formed from [Code]+[Name] of DebitorCreditor object. See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value).

"oA_ItemsID": If specified than account category will be 'Item' and [Name] will be formed from [Code]+[Name] of Item object. See [ItemsList] method (use [OA_ItemsID] value).

"parentAccountNumber": Parent account to form tree structure. Specify an empty value "" to specify the root account

Examples

1. Add new cash account - "Cashier N1" with specified number - "1110 1", under the parent account - "1110" and comment - "Special account for cashier 1."

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "accountComment": " Special account for cashier 1.",
  "accountName": " Cashier N1",
  "accountNumber": "1110 1",
  "parentAccountNumber": "1110"
}
```

Server response Code(200):

```
{
  "id": 215,
  "idFieldName": "OA_AccountsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51103,
    "description": "Parent Account transaction detected!",
    "comment": "Account [1110].\r\nFor adding sub account, under account which has entry, you can do following:\r\n1. For account [1110] change account number on [1110-0].\r\n2. For account [1100] add new sub account [1110].\r\n3. For account [1110-0] set parent [1110].\r\n4. After this, for account [1110] it is possible to add new sub account.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Account",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

2. Add new bank account under the parent account - "1210". New account name - "Bank of Georgia"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "accountName": "Bank of Georgia",
  "parentAccountNumber": "1210"
}
```

3. Add new debtor/creditor (organization or person) account under the parent account - "1410".
OA_DebitorCreditorID – "10"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_DebitorCreditorID": 10,
  "parentAccountNumber": "1410"
}
```

4. Add new Item (commodity, fixed asset or service) account under the parent account - "1610".
OA_ItemsID – "2"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ItemsID": 2,
  "parentAccountNumber": "1610"
}
```

PUT/api/Account

Updates account. Returns OK with ID value or Error description Json.

Schema

```
{
  "token": "string",
  "oA_AccountsID": 0,
  "accountCategoryInt": 0,
  "accountComment": "string",
  "accountEnableMinusStockBalance": true,
  "accountIsClosed": true,
  "accountName": "string",
  "accountNonbalance": true,
  "accountNumber": "string",
  "accountTypeInt": 0,
  "accountWriteOffMethodInt": 0,
  "oA_DebitorCreditorID": 0,
  "oA_ItemsID": 0,
  "parentAccountNumber": "string"
}
```

Field description

[Token] and [OA_AccountsID] is necessary. Specify only the fields you want to edit.

"token": Set [token] string from [LogIn] response value

"oA_AccountsID": See [AccountsList] method (use [OA_AccountsID] value).

"accountCategoryInt": Financial = 0, Item = 1, DebitorCreditor = 2. Default=0

"accountComment":

"accountEnableMinusStockBalance":

"accountIsClosed":

"accountName": Sets account name if [AccountCategoryInt]=0.

"accountNonbalance":

"accountNumber":

"accountTypeInt": AccountTypeInt: Active = 0, Passive = 1. Default gets from parent account

"accountWriteOffMethodInt": AccountWriteOffMethodInt: None = 0, FIFO = 1, LIFO = 2, Avarage = 3. Default=0

"oA_DebitorCreditorID": If specified than account category will be 'DebitorCreditor' and [Name] will be formed from [Code]+[Name] of DebitorCreditor object. See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value).

"oA_ItemsID": If specified than account category will be 'Item' and [Name] will be formed from [Code]+[Name] of Item object. See [ItemsList] method (use [OA_ItemsID] value).

"parentAccountNumber": Parent account to form tree structure. Specify an empty value "" to specify the root account

Examples

1. Update cash account - "Cashier N1" with OA_AccountsID – "215" and specify comment - "Special account for cashier number 1."

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_AccountsID": 215,
  "accountComment": " Special accout for cashier number 1."
}
```

Server response Code(200):

```
{
  "id": 215,
  "idFieldName": "OA_AccountsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_AccountsID = 215",
    "command": "",
    "errorType": "ErrorType_Error",
  }
]
```

```

"source": "OA_BusinessObject.dll",
"class": "Account",
"entryPoint": "GetProperties",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

2. Update Item account with OA_AccountsID – “219”, set the write-off method as FIFO and allow a minus stock balance.

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"oA_AccountsID ": 219,
"accountEnableMinusStockBalance": true,
"accountWriteOffMethodInt": 1
}

```

3. Update Item account with “OA_AccountsID” – 219 and close it.

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"oA_AccountsID": 219,
"accountIsClosed": true
}

```

DELETE/api/Account

Returns OK with ID value or Error description Json.

Schema

```

{
"token": "string",
"oA_AccountsID": 0
}

```

Field description

[Token] and [OA_AccountsID] is necessary. Specify only the fields you want to edit.

***token**: Set [token] string from [LogIn] response value

***oA_AccountsID**: See [AccountsList] method (use [OA_AccountsID] value).

Examples

1. Delete account with “OA_AccountsID” – 219.

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"oA_AccountsID": 219
}

```

Server response Code(200):

```

{
"id": 219,
"idFieldName": "OA_AccountsID"
}

```

Server response Code(400):

```

[
{
"errorNumber": 51109,
"description": "ORIS_Tools internal error!",
"comment": "OA_AccountsID=219 Account [3110 1]\nExist relation to another table.\nRelated table [OA_Entries].\n\n[547] The DELETE statement conflicted with the REFERENCE constraint

```

"FK_OA_Entries_OA_Accounts\". The conflict occurred in database \"NDatabaseEN\", table \"dbo.OA_Entries\", column 'OA_AccountsID'.[3621] The statement has been terminated.",

```
"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "Account",
"entryPoint": "RealDelete",
"resourceDLL_ForTranslation": "ORIS_Administrate",
"errorPointer": 0
}
]
```

AccountsList

POST/api/AccountsList

Gets chart of accounts. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

[Token] is necessary.

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1. Example: get account by ID=123 - "[OA_AccountsID]=123" or get accounts whose numbers start with 11 and level is 2 - "[Account] LIKE N'11%' AND [Level]=2"

"sort": Use SQL ORDER BY string with field names. Example: "[Account],[Name]"

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get first 2 accounts from database

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 2,
      "PageCount": 0,
      "PageRowCount": 2,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_AccountsID": 1,
      "Account": "1000",
      "ParentID": 0,
      "Name": "Carrent assets",
      "Level": 0,
      "Comment": ""
    }
  ]
}
```

```

"Close": false,
"Leaf": false,
"Enable Minus Stock Balance": false,
"Type": 0,
"WriteOff method": 0,
"Nonbalance": false,
"Category": 0
},
{
"OA_AccountsID": 10,
"Account": "1100",
"ParentID": 1,
"Name": "Cash on Hand",
"Level": 1,
"Comment": "",
"Close": false,
"Leaf": false,
"Enable Minus Stock Balance": false,
"Type": 0,
"WriteOff method": 0,
"Nonbalance": false,
"Category": 0,
"Parent account": "1000"
}
]
}

```

2. Get the 2nd page from the entire list of accounts, 15 records per page

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"topRecordsCount": 15,
"pageNumber": 2
}

```

3. Get all field names for further use in the filter

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"topRecordsCount": 0
}

```

4. Get a list of accounts whose account numbers start with "11" and level is 2

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"filter": "[Account] LIKE N'11%' AND [Level]=2"
}

```

5. Get a list of debtor/creditors (organization or person) accounts under "1410" account and sort it by account name

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"filter": "[OA_DebitorCreditorID]>0 AND [Parent account] = N'1410'",
"sort": "[Name]"
}

```

6. Get a list of items (commodity, fixed asset or service) accounts under "1610" account and sort it by account number

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"filter": "[OA_ItemsID]>0 AND [Parent account] = N'1610'",
"sort": "[Name]"
}

```

GetDebtorCreditorAccountByParent

POST/api/GetDebtorCreditorAccountByParent

Gets or Adds DebtorCreditor account by parent account. Returns OA_AccountsID value.

Schema

```
{
  "token": "string",
  "parentAccount": "string",
  "debtorCreditorID": 0,
  "addNewAccountIfNotExists": true,
  "raiseError": true
}
```

Field description

Token and ParentAccount is necessary.

"token": Set [token] string from [LogIn] response value,

"parentAccount": Parent account number. Example: "3110".

"debtorCreditorID": OA_DebtorCreditorID value. See [DebtorCreditorsList] method,

"addNewAccountIfNotExists": Add debtor/creditor new account if not exists,

"raiseError": Fills error description

Examples

1. Get debtor/creditors OA_AccountsID (organization or person) under "3110 - Accounts payable – trade" account using "debtorCreditorID": - 1 "Organization"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "parentAccount": "3110",
  "debtorCreditorID": 1,
  "raiseError": true
}
```

Server response Code(200):

```
{
  "id": 224,
  "idFieldName": "OA_AccountsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51188,
    "description": "Account does not exist under parent account!",
    "comment": "",
    "command": "WHERE [ParentID] = 119 AND [OA_DebtorCreditorID] = 1",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Account",
    "entryPoint": "GetAccountIDByParent",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

2. Add debtor/creditors (organization or person) accounts under "1410 - Accounts receivable - trade" account using "debtorCreditorID"- 2 "Organization2"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "parentAccount": "1410",

```

```
"debitorCreditorID": 2,  
"addNewAccountIfNotExists": true  
}
```

GetItemAccountByParent

POST/api/GetItemAccountByParent

Gets or Adds Item account by parent account. Returns OA_AccountsID value.

Schema

```
{  
  "token": "string",  
  "parentAccount": "string",  
  "oA_ItemsID": 0,  
  "addNewAccountIfNotExists": true,  
  "raiseError": true  
}
```

Field description

Token and ParentAccount is necessary.

"token": Set [token] string from [LogIn] response value,
"parentAccount": Parent account number. Example: "1610".
"oA_ItemsID": OA_ItemsID value. See [ItemsList] method,
"addNewAccountIfNotExists": Add item new account if not exists,
"raiseError": Fills error description

Examples

1. Get item OA_AccountsID under "1610 - Merchandise inventory" account using "oA_ItemsID": - 1
Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "parentAccount": "1610",  
  "oA_ItemsID": 1,  
  "raiseError": true  
}
```

Server response Code(200):

```
{  
  "id": 230,  
  "idFieldName": "OA_AccountsID"  
}
```

Server response Code(400):

```
[  
  {  
    "errorNumber": 51188,  
    "description": "Account does not exist under parent account!",  
    "comment": "",  
    "command": "WHERE [ParentID] = 69 AND [OA_ItemsID] = 5",  
    "errorType": "ErrorType_Error",  
    "source": "OA_BusinessObject.dll",  
    "class": "Account",  
    "entryPoint": "GetAccountIDByParent",  
    "resourceDLL_ForTranslation": "ORIS_Accounting",  
    "errorPointer": 0  
  }  
]
```

2. Add items accounts under "1610" account using "oA_ItemsID" - 5
Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",
```



```
"parentAccount": "1610",
"oA_ItemsID": 5,
"addNewAccountIfNotExists": true
}
```

GetProgenyOA_AccountsIDs

POST/api/GetProgenyOA_AccountsIDs

Gets all sub accounts IDs (OA_AccountsID) for parent account. Returns comma separated ID values.

Schema

```
{
  "token": "string",
  "parentID": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value,
"parentID": See [AccountsList] method (use [OA_AccountsID] value)

Examples

1. Get a list of accounts under "1610" "parentID" using "OA_AccountsID" - 56

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "parentID": 56
}
```

Server response Code(200):

```
{
  "id": "56,217,258",
  "idFieldName": "OA_AccountsID"
}
```

Accounting - Entries

Transaction

POST/api/Transaction

Adds new accounting transaction. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "transactionComment": "string",
  "transactionDate": "2023-04-25T10:29:30.008Z",
  "transactionDelay": true,
  "transactionDocumentalConfirm": true,
  "transactionDocumentNumber": "string",
  "correctDisbalance": true,
  "transactionEntries": [
    {
      "mainEntry": true,
      "account": "string",
      "debitAmount": 0,
      "creditAmount": 0,
      "currency": "string",
    }
  ]
}
```

```

    "debitQuantity": 0,
    "creditQuantity": 0,
    "unit": "string",
    "costCentre": "string",
    "costUnit": "string",
    "amountCalculationType": 0,
    "comment": "string",
    "cashFlow": "string",
    "memorial": 0,
    "rate": 0,
    "isRateFixed": true,
    "debitEquivalent": 0,
    "creditEquivalent": 0,
    "rate2": 0,
    "debitEquivalent2": 0,
    "creditEquivalent2": 0,
    "stockRegisterRate": 0,
    "debitStock": 0,
    "creditStock": 0,
    "internalTurnover": true,
    "actionInCaseDisbalance": 0,
    "oA_AccountsID": 0,
    "oA_CashflowCategoriesID": 0,
    "oA_CurrenciesID": 0,
    "oA_CostCentreID": 0,
    "oA_CostUnitID": 0,
    "oA_StockUnitsID": 0,
    "relatedOA_EntriesID": 0
  }
]
}

```

Field description

[Token] and [TransactionDate] is necessary. Specify only those fields that you want to fill.

"**token**": Set [token] string from [LogIn] response value,

"**transactionComment**":

"**transactionDate**":

"**transactionDelay**":

"**transactionDocumentalConfirm**":

"**transactionDocumentNumber**":

"**correctDisbalance**": If specified True, checks disbalance and reflects it the main entry. If not, then only disbalance is checked. Default = True

"**transactionEntries**":

"**mainEntry**": Set true if entry is main. Default=false.

"**account**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_AccountsID] property specified

"**debitAmount**": Debit amount

"**creditAmount**": Credit amount

"**currency**": See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified. If omitted, gets first register currency from parameters

"**debitQuantity**": Debit quantity for the item account type

"**creditQuantity**": Credit quantity for the item account type

"**unit**": See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified

"**costCentre**": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters

"**costUnit**": See [CostUnitsList] method (use [Name] value). Can be omitted if [OA_CostUnitID] property specified. If omitted, gets cost unit from parameters

"**amountCalculationType**": CreditAmountCalculation = 0, DebitAmountCalculation = 1, None = 2. Default=0

"**comment**":

"**cashFlow**": See [CashflowCategoriesList] method (use [Abridgment] value). Can be omitted if [OA_CashflowCategoriesID] property specified

"**memorial**":

"**rate**": Will be filled by rate from currency rate on transaction date. See [CurrencyRate] method

"**isRateFixed**": Sets a fixed rate flag. If true, then equivalent values will not be recalculated when the exchange rate changes

"**debitEquivalent**": If it is not specified, it will be filled with [DebitAmount] multiplied by the currency rate on the date of the transaction

"**creditEquivalent**": If it is not specified, it will be filled with [CreditAmount] multiplied by the currency rate on the date of the transaction

"**rate2**": Will be filled by rate from currency rate on transaction date. See [CurrencyRate] method

"**debitEquivalent2**": If it is not specified, it will be filled with [DebitAmount] multiplied by the currency rate2 on the date of the transaction

"**creditEquivalent2**": If it is not specified, it will be filled with [CreditAmount] multiplied by the currency rate2 on the date of the transaction

"**stockRegisterRate**": Will be filled by register rate for Item method properties if transaction entry unit is different from Item register unit

"**debitStock**": Calculates from [DebitQuantity]*[StockRegisterRate]

"**creditStock**": Calculates from [CreditQuantity]*[StockRegisterRate]

"**internalTurnover**": If set to true - this entry will not be included in turnover reports

"**actionInCaseDisbalance**": Use it in case [MainEntry] = true. Add disbalance to main entry = 0, Distribute disbalance to non main entries = 1. Default=0

"**oA_AccountsID**": See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [account] property specified

"**oA_CashflowCategoriesID**": See [CashflowCategoriesList] method (use [OA_CashflowCategoriesID] value). Can be omitted if [CashFlow] property specified

"**oA_CurrenciesID**": See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currency] property specified. If ommite, gets firts register currency from parameters

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If ommite, gets cost centre from parameters

"**oA_CostUnitID**": See [CostUnitsList] method (use [OA_CostUnitID] value). Can be omitted if [CostUnit] property specified. If ommite, gets cost unit from parameters

"**oA_StockUnitsID**": See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified

"**relatedOA_EntriesID**": Use this property to increase the cost of items in the FIFO/LIFO invoice list. First, define an invoice for the goods receipt operation. Then use the [GetTransactionByID] method to determine the [OA_EntriesID] from the goods receipt entries from [transactionEntries]. The accounts in this entry and the related entry must match

Examples

1. Add easy entry, transaction - transfer 200 GEL from bank account "1210 1" Bank of Georgia to cash account "1110 2" Cashier N1.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "transactionDate": "2023-04-26T13:05:16.657Z",
  "transactionEntries": [
    {
      "mainEntry": true,
      "account": "1110 2",
      "debitAmount": 200
    },
    {
      "mainEntry": false,
      "account": "1210 1",
      "creditAmount": 200
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 31,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51038,
    "description": "Account is closed!",
    "comment": "Account [1110 2] Cashier N1\nUncheck close check box for this account and resume or cancel transaction\n OA_AccountisID=220\nOA_TransactionsID [1008]\nDocument number []\nDate[26.04.2023 13:05:16]",
  }
]
```

```

"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "Transaction",
"entryPoint": "PostUpdateActionStage1",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

2. Add transaction in case disbalance - transfer 200 GEL from bank account "1210 1" to cash account "1110 2" 400 GEL

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"transactionDate": "2023-04-26T13:05:16.657Z",
"correctDisbalance": false,
"transactionEntries": [
{
"mainEntry": true,
"account": "1110 2",
"debitAmount": 200
},
{
"mainEntry": false,
"account": "1210 1",
"creditAmount": 400
}
]
}

```

Server response Code(400):

```

[
{
"errorNumber": 51040,
"description": "Disbalance detected!",
"comment": "200 <> 400\nCheck main entry. Index [1]. (Off-balance entry is not involved in determining the balance)",
"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "Transaction",
"entryPoint": "DisbalanceAction",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

3. Add Difficult Transaction - Revenue income using debtor/creditor account 1410 1, income account for supply 6110 1 and payable VAT account 3330 and payment (cash income) using cash account 1110 2 and debtor/creditor account 1410 1

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"transactionDate": "2023-04-26T13:05:16.657Z",
"transactionEntries": [
{
"mainEntry": true,
"account": "1410 1",
"debitAmount": 1200
},
{
"mainEntry": false,
"account": "6110 1",
"creditAmount": 1016.94
}
]
}

```

```

    "mainEntry": false,
    "account": "3330",
    "creditAmount": 183.06
  },
  {
    "mainEntry": true,
    "account": "1110 2",
    "debitAmount": 1200
  },
  {
    "mainEntry": false,
    "account": "1410 1",
    "creditAmount": 1200
  }
]
}.

```

4. Add transaction in currency – Billing for services 200 EUR using debtor/creditor account 3110 4 and expense account 7411

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "transactionEntries": [
    {
      "mainEntry": true,
      "account": "3110 4",
      "currency": "EUR"
    },
    {
      "mainEntry": false,
      "account": "7411",
      "debitAmount": 200,
      "currency": "EUR"
    }
  ]
}

```

5. Add accept the goods transaction using debtor/creditor account 3110 4, item account 1610 1 – amount 200 GEL, quantity – 10 piece. item account 1610 2 – amount 100 GEL, quantity – 5 piece and payed VAT account 3340 amount 45.76 GEL.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "transactionEntries": [
    {
      "mainEntry": true,
      "account": "3110 1"
    },
    {
      "mainEntry": false,
      "account": "1610 1",
      "debitAmount": 200,
      "debitQuantity": 10
    },
    {
      "mainEntry": false,
      "account": "1610 2",
      "debitAmount": 100,
      "debitQuantity": 5
    },
    {
      "mainEntry": false,
      "account": "3340",
      "debitAmount": 45.76
    }
  ]
}

```

```
}
```

PUT/api/Transaction

Updates accounting transaction. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "transactionComment": "string",
  "transactionDate": "2023-04-25T12:34:46.232Z",
  "transactionDelay": true,
  "transactionDocumentalConfirm": true,
  "transactionDocumentNumber": "string",
  "clearEntries": true,
  "correctDisbalance": true,
  "transactionEntries": [
    {
      "oA_EntriesID": 0,
      "deleteRow": true,
      "mainEntry": true,
      "account": "string",
      "debitAmount": 0,
      "creditAmount": 0,
      "currency": "string",
      "debitQuantity": 0,
      "creditQuantity": 0,
      "unit": "string",
      "costCentre": "string",
      "costUnit": "string",
      "amountCalculationType": 0,
      "comment": "string",
      "cashFlow": "string",
      "memorial": 0,
      "rate": 0,
      "isRateFixed": true,
      "debitEquivalent": 0,
      "creditEquivalent": 0,
      "rate2": 0,
      "debitEquivalent2": 0,
      "creditEquivalent2": 0,
      "stockRegisterRate": 0,
      "debitStock": 0,
      "creditStock": 0,
      "internalTurnover": true,
      "actionInCaseDisbalance": 0,
      "oA_AccountsID": 0,
      "oA_CashflowCategoriesID": 0,
      "oA_CurrenciesID": 0,
      "oA_CostCentreID": 0,
      "oA_CostUnitID": 0,
      "oA_StockUnitsID": 0,
      "relatedOA_EntriesID": 0
    }
  ]
}
```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

"transactionComment":

"**transactionDate**":
"transactionDelay":
"transactionDocumentalConfirm":
"transactionDocumentNumber":
"clearEntries": Clears all entries for the transaction
"correctDisbalance": If specified True, checks disbalance and reflects it the main entry. If not, then only disbalance is checked. Default = True

"transactionEntries": Set OA_EntriesID=0 to add a new entry specification. Specify the OA_EntriesID value to update information about entry fields. To delete existing entry, specify the value OA_EntriesID and set "DeleteRow"=true
"oA_EntriesID": Specify an existing OA_EntriesID to update or remove related item information, or 0 to add a new related item
"deleteRow": If true, deletes entry information for specified OA_EntriesID value
"mainEntry": Set true if entry is main. Default=false
"account": See [AccountsList] method (use [Account] value)
"debitAmount":
"creditAmount":
"currency": See [CurrenciesList] method (use [Currency] value)
"debitQuantity":
"creditQuantity":
"unit": See [StockUnitsList] method (use [Unit] value)
"costCentre": See [CostCentresList] method (use [Name] value)
"costUnit": See [CostUnitsList] method (use [Name] value)
"amountCalculationType": CreditAmountCalculation = 0, DebitAmountCalculation = 1, None = 2. Default=0
"comment":
"cashFlow": See [CashflowCategoriesList] method (use [Abridgment] value)
"memorial":
"rate": Will be filled by rate from currency rate table on transaction date. See [CurrencyRate] method
"isRateFixed": Sets a fixed rate flag
"debitEquivalent":
"creditEquivalent":
"rate2": Will be filled by rate from currency rate table on transaction date. See [CurrencyRate] method
"debitEquivalent2":
"creditEquivalent2":
"stockRegisterRate": Will be filled by register rate for Item method properties if transaction entry unit is different from Item register unit
"debitStock": Calculates from [DebitQuantity]*[StockRegisterRate]
"creditStock": Calculates from [CreditQuantity]*[StockRegisterRate]
"internalTurnover": If set to true - this entry will not be included in turnover reports
"actionInCaseDisbalance": Use it in case [MainEntry] = true. Add disbalance to main entry = 0, Distribute disbalance to non main entries = 1. Default=0
"oA_AccountsID": See [AccountsList] method (use [OA_AccountsID] value)
"oA_CashflowCategoriesID": See [CashflowCategoriesList] method (use [OA_CashflowCategoriesID] value)
"oA_CurrenciesID": See [CurrenciesList] method (use [OA_CurrenciesID] value)
"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value)
"oA_CostUnitID": See [CostUnitsList] method (use [OA_CostUnitID] value)
"oA_StockUnitsID": See [StockUnitsList] method (use [OA_StockUnitsID] value)
"relatedOA_EntriesID": Use this property to increase the cost of items in the FIFO/LIFO invoice list. First, define an invoice for the goods receipt operation. Then use the [GetTransactionByID] method to determine the [OA_EntriesID] from the goods receipt entries from [transactionEntries]. The accounts in this entry and the related entry must match

Examples

1. Update transaction using "oA_TransactionsID": - 34 accept the goods transaction change amount in item account 1610 1 – amount 300 GEL and item account 3340 2 – amount 400 GEL and payed VAT account 3340 amount 106.78 GEL.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 34,
  "transactionDate": "2023-04-27T06:57:02.270Z",
  "transactionDocumentNumber": "123",
  "clearEntries": true,
  "transactionEntries": [
    {
      "mainEntry": true,
```

```

    "account": "3110 1"
  },
  {
    "mainEntry": false,
    "account": "1610 1",
    "debitAmount": 300,
    "debitQuantity": 10
  },
  {
    "mainEntry": false,
    "account": "1610 2",
    "debitAmount": 400,
    "debitQuantity": 5
  },
  {
    "mainEntry": false,
    "account": "3340",
    "debitAmount": 106.78
  }
]
}

```

Server response Code(200):

```

{
  "id": 34,
  "idFieldName": "OA_TransactionsID"
}

```

Server response Code(400):

```

[
  {
    "errorNumber": 51067,
    "description": "Can not change/delete data when transaction has been documental confirmed!",
    "comment": "For change/delete data first remove documental confirm.\nOA_TransactionsID= 34",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "CheckDocumentalConfirm",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

2. Update transaction in case disbalance - transfer 400 GEL from bank account "1210 1" to cash account "1110 2" 600 GEL

Request body:

```

{
  "token": "... paste here [Login] response value ...",
  "oA_TransactionsID": 5052,
  "transactionDate": "2023-04-26T13:05:16.657Z",
  "transactionDocumentNumber": "23",
  "clearEntries": true,
  "correctDisbalance": false,
  "transactionEntries": [
    {
      "mainEntry": true,
      "account": "1110 2",
      "debitAmount": 400
    },
    {
      "mainEntry": false,
      "account": "1210 1",
      "creditAmount": 600
    }
  ]
}

```

```
}
```

Server response Code(400):

```
[  
{  
  "errorNumber": 51040,  
  "description": "Disbalance detected!",  
  "comment": "200 <> 400\nCheck main entry. Index [1]. (Off-balance entry is not involved in determining the  
balance)\nOA_TransactionsID [5052]\nDocument number []\nDate[01.06.2023 12:19:31]",  
  "command": "",  
  "errorType": "ErrorType_Error",  
  "source": "OA_BusinessObject.dll",  
  "class": "Transaction",  
  "entryPoint": "DisbalanceAction",  
  "resourceDLL_ForTranslation": "ORIS_Accounting",  
  "errorPointer": 0  
}  
]
```

DELETE/api/Transaction

Returns OK or Error description Json.

Schema

```
{  
  "token": "string",  
  "oA_TransactionsID": 0  
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

Examples

1. Delete transaction with oA_TransactionsID – "31".
Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "oA_AccountsID": 31  
}
```

Server response Code(200):

```
{  
  "id": 31,  
  "idFieldName": "OA_TransactionsID"  
}
```

Server response Code(400):

```
[  
{  
  "errorNumber": 51051,  
  "description": "Record does not exist or read prohibited for user!",  
  "comment": "OA_TransactionsID = 31",  
  "command": "",  
  "errorType": "ErrorType_Error",  
  "source": "OA_BusinessObject.dll",  
  "class": "Transaction",  
  "entryPoint": "GetProperties",  
  "resourceDLL_ForTranslation": "ORIS_Accounting",  
  "errorPointer": 0  
}  
]
```

GetTransactionByID

POST/api/GetTransactionByID

Gets transaction and its accounting entries by OA_TransactionsID value.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID": See [TransactionsList] method (use [OA_TransactionsID] value)

Examples

1. Get transaction and its accounting entries using oA_TransactionsID – "31".

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 31
}
```

Server response Code(200):

```
{
  "oA_TransactionsID": 30,
  "amount": 200,
  "autoFillAmountAndOA_CurrenciesID": true,
  "hasEntries": true,
  "oA_CurrenciesID": 1,
  "transactionComment": "",
  "transactionDate": "2023-04-26T13:05:16",
  "transactionDelay": false,
  "transactionDocumentalConfirm": false,
  "transactionDocumentNumber": "",
  "transactionType": "",
  "transactionTypeGUID": "",
  "transactionEntries": [
    {
      "oA_EntriesID": 113,
      "oA_TransactionsID": 30,
      "account": "1110 2",
      "accountName": "Cashier N1",
      "debitAmount": 200,
      "creditAmount": 0,
      "currency": "GEL",
      "debitQuantity": 0,
      "creditQuantity": 0,
      "unit": "",
      "costCentre": "General cost center",
      "costUnit": "Total cost unit",
      "amountCalculationType": 0,
      "comment": "",
      "cashFlow": "",
      "memorial": 0,
      "rate": 1,
      "debitEquivalent": 200,
      "creditEquivalent": 0,
      "rate2": 0.39926535,
      "debitEquivalent2": 79.85307,
    }
  ]
}
```

```

"creditEquivalent2": 0,
"stockRegisterRate": 0,
"debitStock": 0,
"creditStock": 0,
"debitStockAvailable": 0,
"creditStockAvailable": 0,
"isRateFixed": false,
"internalTurnover": false,
"actionInCaseDisbalance": 0,
"sequenceNumber": 0,
"accountNonBalance": false,
"oA_AccountsID": 220,
"mainEntry": true,
"mainEntryID": 0,
"oA_CashflowCategoriesID": 0,
"oA_CurrenciesID": 1,
"oA_CostCentreID": 1,
"oA_CostUnitID": 1,
"oA_StockUnitsID": 0,
"accountLeaf": true,
"accountClose": false,
"accountType": 0,
"oA_DebitorCreditorID": 0,
"isPerson": false,
"oA_ItemsID": 0,
"itemType": 0,
"oA_StockUnitsIDRegisterUnit": 0,
"registerUnit": "",
"hasADoubleRegistrationUnit": false,
"accountCategory": 0,
"accountWriteOffmethod": 0,
"relatedOA_EntriesID": 0,
"relatedSequenceNumber": 0,
"transferWithinCostCentreGroup": false,
"oA_CostCentreGroupID": 0,
"corrOA_AccountsID": 0,
"correspondentAccount": "",
"isReverseEntry": false,
"relatedOA_TransactionsID": 0,
"relatedDocumentNumber": ""
},
{
"oA_EntriesID": 114,
"oA_TransactionsID": 30,
"account": "1210 1",
"accountName": "Bank of Georgia",
"debitAmount": 0,
"creditAmount": 200,
"currency": "GEL",
"debitQuantity": 0,
"creditQuantity": 0,
"unit": "",
"costCentre": "General cost center",
"costUnit": "Total cost unit",
"amountCalculationType": 0,
"comment": "",
"cashFlow": "",
"memorial": 0,
"rate": 1,
"debitEquivalent": 0,
"creditEquivalent": 200,
"rate2": 0.39926535,
"debitEquivalent2": 0,
"creditEquivalent2": 79.85307,
"stockRegisterRate": 0,
"debitStock": 0,
"creditStock": 0,

```

```

"debitStockAvailable": 0,
"creditStockAvailable": 0,
"isRateFixed": false,
"internalTurnover": false,
"actionInCaseDisbalance": 0,
"sequenceNumber": 0,
"accountNonBalance": false,
"oA_AccountsID": 216,
"mainEntry": false,
"mainEntryID": 113,
"oA_CashflowCategoriesID": 0,
"oA_CurrenciesID": 1,
"oA_CostCentreID": 1,
"oA_CostUnitID": 1,
"oA_StockUnitsID": 0,
"accountLeaf": true,
"accountClose": false,
"accountType": 0,
"oA_DebitorCreditorID": 0,
"isPerson": false,
"oA_ItemsID": 0,
"itemType": 0,
"oA_StockUnitsIDRegisterUnit": 0,
"registerUnit": "",
"hasADoubleRegistrationUnit": false,
"accountCategory": 0,
"accountWriteOffmethod": 0,
"relatedOA_EntriesID": 0,
"relatedSequenceNumber": 0,
"transferWithinCostCentreGroup": false,
"oA_CostCentreGroupID": 0,
"corrOA_AccountsID": 220,
"correspondentAccount": "1110 2",
"isReverseEntry": false,
"relatedOA_TransactionsID": 0,
"relatedDocumentNumber": ""
}
]
}

```

Server response Code(400):

```

[
{
"errorNumber": 51051,
"description": "Record does not exist or read prohibited for user!",
"comment": "OA_TransactionsID = 31",
"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "Transaction",
"entryPoint": "GetProperties",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

EntriesList

POST/api/EntriesList

Gets list of accounting entries. Returns [Info] and [Result] tables.

Schema

```

{
"token": "string",

```



```

"diapazonStartDate": "2023-04-27T07:09:45.314Z",
"diapazonEndDate": "2023-04-27T07:09:45.314Z",
"topRecordsCount": 0,
"filter": "string",
"sort": "string",
"pageNumber": 0
}

```

Field description

[Token],[DiapazonStartDate] and [DiapazonEndDate] is necessary.

"token": Set [token] string from [LogIn] response value

"diapazonStartDate": Diapazon start date

"diapazonEndDate": Diapazon end date (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01", "DiapazonEndDate":"2011-01-01"

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1. Example: get entries where debit amount more than 100 - "[Debit amount]>100" or get entries where account numbers start with 11 and correspondent account = 3110 1 "[Account] LIKE N'11% AND [Correspondent account]=N'3110 1"

"sort": Use SQL ORDER BY string with field names. Example: "[Transaction date],[Account]"

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get first transaction from database in specified diapazon

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "diapazonStartDate": "2023-04-26T07:09:45.314Z",
  "diapazonEndDate": "2023-04-27T07:09:45.314Z",
  "topRecordsCount": 1
}

```

Server response Code(200):

```

{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 0,
      "PageRowCount": 1,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "Delay": false,
      "Documental confirm": false,
      "Transaction date": "2023-04-26T13:05:16",
      "Currency": "GEL",
      "OA_TransactionTypesID": null,
      "Transaction type": null,
      "Document number": "",
      "Account": "1110 2",
      "OA_ItemsID": null,
      "Item type": null,
      "OA_StockUnitsIDRegisterUnit": null,
      "Register unit": null,
      "OA_DebitorCreditorID": null,
      "Is person": null,
      "Account name": " Cashier N1",
      "Debit amount": 200.0,
      "Credit amount": 0.0,
      "Cost centre": "General cost center",
      "Debit quantity": 0.0,
      "Credit quantity": 0.0,
      "Unit": null,
    }
  ]
}

```

```

"Amount calculation type": null,
"Cost unit": "Total cost unit",
"Transaction comment": "",
"Comment": null,
"Cash flow": null,
"Memorial": null,
"Rate": 1.00000000,
"Debit equivalent": 200.0,
"Credit equivalent": 0.0,
"Rate2": 0.39926535,
"Debit equivalent2": 79.85307,
"Credit equivalent2": 0.0,
"Debit stock": 0.0,
"Credit stock": 0.0,
"Debit stock available": 0.0,
"Credit stock available": 0.0,
"Is rate fixed": false,
"Internal turnover": false,
"Action in case disbalance": 0,
"Main entry": true,
"SequenceNumber": 1,
"OA_UsersID": 1,
>User": "sa",
"OA_DepartmentsID": 1,
"Department": "Administrators",
>Create date": null,
>Last update date": null,
"OA_EntriesID": 113,
"OA_AccountsID": 220,
"OA_CostCentreID": 1,
"OA_CostUnitID": 1,
"OA_CashflowCategoriesID": null,
"OA_CurrenciesID": 1,
"OA_StockUnitsID": null,
"OA_TransactionsID": 30,
"Is start balance": false,
"Related OA_EntriesID": null,
"Related OA_TransactionsID": null,
"Related document number": null,
"Related sequence number": null,
"CorrAccountID": null,
"Correspondent account": null,
"Correspondent OA_ItemsID": null,
"Correspondent OA_DebitorCreditorID": null,
"Correspondent account name": null,
"Transfer within cost centre group": false,
"OA_Accounts Category": 0,
"Account nonbalance": false,
"Category": 0,
"Is reverse entry": false,
"OA_SalaryResultsID": null,
"RS Waybill number": null,
"RS VAT number": null,
"Record related binary documents": null,
"RS Tax document number": null,
"Correspondent OA_CashflowCategoriesID": null,
"Correspondent account cash flow": null
}
]
}

```

Server response Code(400):

```

[
{
"errorNumber": 51123,
"description": "Period start date must be more than start date!",

```

```

"comment": "Period start date 26.04.2020\nStart date 03.05.2020",
"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "Entries",
"entryPoint": "ListRefresh",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

2. Get the 2nd page from the entire list of transactions, 15 records per page

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"topRecordsCount": 15,
"pageNumber": 2
}

```

3. Get a list of transactions whose account numbers start with 11 and sort it by date

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"filter": "[Account] LIKE N'11%",
"sort": "[Transaction date]"
}

```

TransactionSetTaxDocument

POST/api/TransactionSetTaxDocument

Sets revenue service Tax Document ID and number for the transaction. Returns true or False.

Schema

```

{
"token": "string",
"oA_TransactionsID": 0,
"rS_TaxDocument_ID": 0,
"rS_TaxDocument_Number": "string"
}

```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

"rS_TaxDocument_ID": Sets Tax Document ID value recieved from revenue service

"rS_TaxDocument_Number": Sets Tax Document number recieved from revenue service

Examples

1. Sets revenue service Tax Document ID and number for the transaction by "oA_TransactionsID": - 34

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"oA_TransactionsID": 34,
"rS_TaxDocument_ID": 1,
"rS_TaxDocument_Number": "123456789"
}

```

Server response Code(200):

```

{
"id": 34,
"idFieldName": "OA_TransactionsID"
}

```

```
}
```

TransactionSetVAT

POST/api/TransactionSetVAT

Sets revenue service VAT ID and number for the transaction. Returns true or False.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "rS_VAT_ID": 0,
  "rS_VAT_Number": "string"
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"**token**": Set [token] string from [LogIn] response value

"**oA_TransactionsID**":

"**rS_VAT_ID**": Sets VAT ID recieved from revenue service

"**rS_VAT_Number**": Sets VAT number recieved from revenue service

Examples

1. Sets revenue service VAT ID and number for the transaction

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 34,
  "rS_VAT_ID": 203024188,
  "rS_VAT_Number": "gs-72 4164071"
}
```

Server response Code(200):

```
{
  "id": 34,
  "idFieldName": "OA_TransactionsID"
}
```

TransactionSetWaybill

POST/api/TransactionSetWaybill

Sets revenue service Way bill ID and number for the transaction. Returns true or False.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "rS_Waybill_ID": 0,
  "rS_Waybill_Number": "string"
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"**token**": Set [token] string from [LogIn] response value

"**oA_TransactionsID**":

"rS_Waybill_ID": Sets waybill ID recieved from revenue service
"rS_Waybill_Number": Sets waybill number recieved from revenue service

Examples

1. Sets revenue service VAT ID and number for the transaction

Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "oA_TransactionsID": 34,  
  "rS_Waybill_ID": 721476227,  
  "rS_Waybill_Number": "0686916171"  
}
```

Server response Code(200):

```
{  
  "id": 34,  
  "idFieldName": "OA_TransactionsID"  
}
```

Accounting – Start balance

SetStartBalance

POST/api/SetStartBalance

Sets start balance data.

Schema

```
{  
  "token": "string",  
  "account": "string",  
  "oA_CurrenciesID": 0,  
  "oA_CostCentreID": 0,  
  "oA_CostUnitID": 0,  
  "debitStartBalance": 0,  
  "creditStartBalance": 0,  
  "isRateFixed": true,  
  "debitStartEquivalent": 0,  
  "creditStartEquivalent": 0,  
  "debitStartEquivalent2": 0,  
  "creditStartEquivalent2": 0,  
  "startStockBalance": 0,  
  "startStockAvailableBalance": 0,  
  "stockUnit": "string",  
  "oA_EntriesID": 0  
}
```

Field description

[Token],[Account],[OA_CurrenciesID],[OA_CostCentreID] and [OA_CostUnitID] is necessary.
Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"account": See [AccountsList] method (use [Account] value)

"oA_CurrenciesID": See [CurrenciesList] method (use [OA_CurrenciesID] value)

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value)

"oA_CostUnitID": See [CostUnitsList] method (use [OA_CostUnitID] value)

"debitStartBalance": Start debit balance

"creditStartBalance": startn credit balance

"isRateFixed": If true, then equivalent values will not be recalculated when the exchange rate changes

"debitStartEquivalent": If omitted, it will be filled with [DebitStartBalance] multiplied by the currency rate at the balance start date

"**creditStartEquivalent**": If omitted, it will be filled with [CreditStartBalance] multiplied by the currency rate at the balance start date

"**debitStartEquivalent2**": If omitted, it will be filled with [DebitStartBalance] multiplied by the currency rate2 at the balance start date

"**creditStartEquivalent2**": If omitted, it will be filled with [CreditStartBalance] multiplied by the currency rate2 at the balance start date

"**startStockBalance**": The start stock balance for the account of item type

"**startStockAvailableBalance**": The start available stock balance for the account of item type. If omitted, it will be filled from [StartStockBalance]

"**stockUnit**": See [StockUnitsList] method (use [Unit] value). If omitted, gets account related item register minimum unit

"**oA_EntriesID**": Specify if you want to modify an existing start balance entry. Otherwise, the start balance will be determined by account, currency, cost center and cost unit

Examples

1. Set start balance, number - "1110 1 debit amount 100 GEL, General cost centre, Total cost unit

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "account": "1110 1",
  "debitStartBalance": 100,
  "oA_CurrenciesID": 1,
  "oA_CostCentreID": 1,
  "oA_CostUnitID": 1
}
```

Server response Code(200):

```
{
  "id": 142,
  "idFieldName": "OA_EntriesID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51038,
    "description": "Account is closed!",
    "comment": "Account [1110 1]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "StartBalances",
    "entryPoint": "SetStartStockUnit",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

2. Set start balance in currency, number - "1210 1 debit amount 100 USD, New cost center, Newcost unit

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "account": "1210 1",
  "debitStartBalance": 100,
  "oA_CurrenciesID": 2,
  "oA_CostCentreID": 2,
  "oA_CostUnitID": 2
}
```

StartBalancesGetStartDate

POST/api/StartBalancesGetStartDate

Gets start balances date.

Schema

```
{
  "token": "string"
}
```

Field description

"**token**": Set [token] string from [LogIn] response value

Examples

1. Get start balances date

Request body:

```
{
  "token": "... paste here [LogIn] response value ..."
}
```

Server response Code(200):

```
"2020-05-03T00:00:00"
```

StartBalancesSetStartDate

POST/api/StartBalancesSetStartDate

Sets start balance date.

Schema

```
{
  "token": "string",
  "startDate": "2023-04-28T12:11:06.847Z",
  "recalculateEquivalent": true,
  "recalculateEquivalent2": true
}
```

Field description

"**token**": Set [token] string from [LogIn] response value

"**startDate**":

"**recalculateEquivalent**":

"**recalculateEquivalent2**":

Examples

1. Set start balances date - "2023-12-31"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "startDate": "2023-12-31",
  "recalculateEquivalent": true,
  "recalculateEquivalent2": true
}
```

Server response Code(200):

```
{
  "id": 1,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51024,
    "description": "Start date must be less than transaction date!",
    "comment": "Detect transaction with date [22.04.2023 00:00:00] which is less or equal than [31.12.2023 00:00:00]",
  }
]
```

```

"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "StartBalances",
"entryPoint": "SetStartDate",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

StartBalncesList

POST/api/StartBalncesList

Gets list of start balances. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "currenciesIDs": "string",
  "costCentreIDs": "string",
  "costUnitIDs": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"currenciesIDs": Comma separated currency IDs or "" - All currencies. See [CurrenciesList] method (use [OA_CurrenciesID] value)

"costCentreIDs": Comma separated cost centre IDs or "" - All cost centres. See [CostCentresList] method (use [OA_CostCentreID] value). Default=""

"costUnitIDs": Comma separated cost unit IDs or "" - All cost units. See [CostUnitsList] method (use [OA_CostUnitID] value). Default=""

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property: [TopRecordsCount]=1. Example: get account by ID=123 - "[OA_AccountsID]=123"

"sort": Use SQL ORDER BY string with field names. Example: "[Account]"

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get list of all start balances that starts with 11 in General cost center, currency –GEL ,total cost unit and sort by start debit amount

Request body:

```

{
  "token": "... paste here [LogIn] response value ..."
  "currenciesIDs": "1",
  "costCentreIDs": "1",
  "costUnitIDs": "1",
  "filter": "[Account] LIKE N'11%'AND UPPER([ViewAllStartBalances].[Currency])=N'GEL'",
  "sort": "[Start debit balance]"
}

```

Server response Code(200):

```

{
  "Info": [
    {
      "RowCount": 3,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ]
}

```

```

}
],
"Result": [
{
  "OA_EntriesID": 148,
  "Account": "1110 2",
  "Account name": " Cashier N1",
  "Start debit balance": 100.0,
  "Start credit balance": 0.0,
  "Currency": "GEL",
  "Start stock": 0.00000000,
  "Start stock available": 0.00000000,
  "Start debit equivalent": 100.0,
  "Start credit equivalent": 0.0,
  "Start debit equivalent2": 36.908541,
  "Start credit equivalent2": 0.0,
  "Is rate fixed": false,
  "Cost centre": "General cost center",
  "Cost unit": "Total cost unit",
  "OA_AccountsID": 220,
  "OA_CurrenciesID": 1,
  "OA_TransactionsID": 1,
  "OA_CostCentreID": 1,
  "OA_CostUnitID": 1,
  "IsStartBalance": true,
  "Start quantity": 0.00000000,
  "Stock register rate": 0.00000000,
  "Category": 0,
  "Is reverse entry": false,
  "Type": 0
},
{
  "OA_EntriesID": 146,
  "Account": "1110 3",
  "Account name": " Cashier N1",
  "Start debit balance": 1000.0,
  "Start credit balance": 0.0,
  "Currency": "GEL",
  "Start stock": 0.00000000,
  "Start stock available": 0.00000000,
  "Start debit equivalent": 1000.0,
  "Start credit equivalent": 0.0,
  "Start debit equivalent2": 369.08541,
  "Start credit equivalent2": 0.0,
  "Is rate fixed": false,
  "Cost centre": "General cost center",
  "Cost unit": "Total cost unit",
  "OA_AccountsID": 221,
  "OA_CurrenciesID": 1,
  "OA_TransactionsID": 1,
  "OA_CostCentreID": 1,
  "OA_CostUnitID": 1,
  "IsStartBalance": true,
  "Start quantity": 0.00000000,
  "Stock register rate": 0.00000000,
  "Category": 0,
  "Is reverse entry": false,
  "Type": 0
},
{
  "OA_EntriesID": 1096,
  "Account": "1110 1",
  "Account name": " Cashier N1",
  "Start debit balance": 1000.0,
  "Start credit balance": 0.0,
  "Currency": "GEL",
  "Start stock": 0.00000000,

```

```

    "Start stock available": 0.00000000,
    "Start debit equivalent": 1000.0,
    "Start credit equivalent": 0.0,
    "Start debit equivalent2": 369.08541,
    "Start credit equivalent2": 0.0,
    "Is rate fixed": false,
    "Cost centre": "General cost center",
    "Cost unit": "Total cost unit",
    "OA_AccountsID": 265,
    "OA_CurrenciesID": 1,
    "OA_TransactionsID": 1,
    "OA_CostCentreID": 1,
    "OA_CostUnitID": 1,
    "IsStartBalance": true,
    "Start quantity": 0.00000000,
    "Stock register rate": 0.00000000,
    "Category": 0,
    "Is reverse entry": false,
    "Type": 0
  }
}
}

```

2. Get list of all start balances for further use in the filter

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0
}

```

3. Get list of all start balances in General cost center, currency –GEL and total cost unit

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "currenciesIDs": "1",
  "costCentreIDs": "1",
  "costUnitIDs": "1",
  "topRecordsCount": 0
}

```

4. Get a list of Fixed Assets (commodity, fixed asset or service) start balances using item type in filter and sort by name

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[Item type] = 1",
  "sort": "[Account name]"
}

```

Accounting - Menegerial transactions

BankTransfer

POST/api/BankTransfer

Adds new Bank Transfer. Returns ID value or Error descriprion Json.

Schema

```

{
  "token": "string",
  "additionalInfo": "string",

```

```

"amount": 0,
"bankAccount": "string",
"bankAccountName": "string",
"bankAccountsID": 0,
"comment": "string",
"costCentreName": "string",
"documentDate": "2023-04-25T08:07:03.587Z",
"documentNumber": "string",
"oA_CostCentreID": 0,
"oA_StatusID": 0,
"payer": "string",
"payerBankAccount": "string",
"payerBankCode": "string",
"payerBankID": 0,
"payerBankName": "string",
"payerCode": "string",
"receiver": "string",
"receiverBankAccount": "string",
"receiverBankCode": "string",
"receiverBankID": 0,
"receiverBankName": "string",
"receiverCode": "string",
"receiverID": 0,
"receiverParentAccount": "string",
"receiverParentAccountName": "string",
"receiverParentAccountsID": 0,
"status": "string",
"taxPayerCode": "string",
"taxPayerName": "string",
"treasuryCode": "string",
"createAccountingEntries": true
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

“**token**”: Set [token] string from [LogIn] response value

“**additionalInfo**”:

“**amount**”:

“**bankAccount**”: Bank account from chart of accounts. See [AccountsList] method (use [Account] value). Can be omitted if [BankAccountsID] property specified

“**bankAccountName**”:

“**bankAccountsID**”: Bank account ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [BankAccount] property specified

“**comment**”:

“**costCentreName**”: See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified

“**documentDate**”: Sets operation date. If not specified - sets current date.

“**documentNumber**”: Sets the transaction document number. If not specified, sets a unique number in the current year.

“**oA_CostCentreID**”: See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreName] property specified

“**oA_StatusID**”: See [StatusesList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified

“**payerBankAccount**”: See [Owner] method (use [account] value from [OwnerBankAccounts]).

“**payerBankCode**”: See [Owner] method (use [BankCode] value from [OwnerBankAccounts]).

“**payerBankID**”: See [Owner] method (use [OA_OwnerBankAccountsID] value from [OwnerBankAccounts]).

“**payerBankName**”: See [Owner] method (use [BankName] value from [OwnerBankAccounts]).

“**payerCode**”:

“**receiver**”:

“**receiverBankAccount**”:

“**receiverBankCode**”: See [BanksList] method (use [Code] value). Can be omitted if [ReceiverBankID] property specified

“**receiverBankID**”: See [BanksList] method (use [OA_BanksID] value). Can be omitted if [ReceiverCode] property specified

“**receiverBankName**”:

“**receiverCode**“: See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [ReceiverID] property specified

“**receiverID**“: See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [ReceiverCode] property specified

“**receiverParentAccount**“: See [AccountsList] method (use [Account] value). Can be omitted if [ReceiverParentAccountsID] property specified. If both are omitted, gets the account from the parameters.

“**receiverParentAccountName**“:

“**receiverParentAccountsID**“: See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [ReceiverParentAccount] property specified. If both are omitted, gets the account from the parameters.

“**Status**“: See [StatusesList] method (use [Name] value). Can be omitted if [OA_StatusID] property specified

“**taxPayerCode**“:

“**taxPayerName**“:

“**treasuryCode**“:

“**createAccountingEntries**“: Creates accounting entries using managerial information. Default=false

Examples

1. Add new Bank Transfer - with specified amount - 23 GEL, bankAccountsID – 216 Bank of Georgia 1210 1 and receiverParentAccountsID – 119 Accounts payable – trade 3110

Request body:

```
{
  "token": "... paste here [LogIn] response value ..."
  "amount": 23,
  "bankAccountsID": 216,
  "documentDate": "2023-06-01T11:47:32.905Z",
  "receiverParentAccountsID": 119
}
```

Server response Code(200):

```
{
  "id": 5056,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51262,
    "description": "Bank account required!",
    "comment": "OA_TransactionsID=0 [16]\n",
    "command": "",
    "errorType": "ErrorType_Permission",
    "source": "OA_BusinessObject.dll",
    "class": "BankTransfers",
    "entryPoint": "CheckInputsForTransaction",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/BankTransfer

Updates Bank Transfer. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "additionalInfo": "string",
  "amount": 0,
  "bankAccount": "string",
  "bankAccountName": "string",
  "bankAccountsID": 0,
  "comment": "string",
  "costCentreName": "string",
}
```

```

"documentDate": "2023-04-26T07:11:59.542Z",
"documentNumber": "string",
"oA_CostCentreID": 0,
"oA_StatusID": 0,
"payer": "string",
"payerBankAccount": "string",
"payerBankCode": "string",
"payerBankID": 0,
"payerBankName": "string",
"payerCode": "string",
"receiver": "string",
"receiverBankAccount": "string",
"receiverBankCode": "string",
"receiverBankID": 0,
"receiverBankName": "string",
"receiverCode": "string",
"receiverID": 0,
"receiverParentAccount": "string",
"receiverParentAccountName": "string",
"receiverParentAccountsID": 0,
"status": "string",
"taxPayerCode": "string",
"taxPayerName": "string",
"treasuryCode": "string",
"createAccountingEntries": true
}

```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID": See [BankTransfersList] method (use [OA_TransactionsID] value).

"additionalInfo":

"amount":

"bankAccount": Bank account from chart of accounts. See [AccountsList] method (use [Account] value).

"bankAccountName":

"bankAccountsID": Bank account ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"Comment":

"costCentreName": See [CostCentresList] method (use [Name] value).

"documentDate": Sets operation date.

"documentNumber":

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value).

"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value).

"Payer":

"payerBankAccount": See [Owner] method (use [Account] value from [OwnerBankAccounts]).

"payerBankCode": See [Owner] method (use [BankCode] value from [OwnerBankAccounts]).

"payerBankID": See [Owner] method (use [OA_OwnerBankAccountsID] value from [OwnerBankAccounts]).

"payerBankName": See [Owner] method (use [BankName] value from [OwnerBankAccounts]).

"payerCode":

"receiver":

"receiverBankAccount":

"receiverBankCode": See [BanksList] method (use [Code] value).

"receiverBankID": See [BanksList] method (use [OA_BanksID] value).

"receiverBankName":

"receiverCode": See [DebitorCreditorsList] method (use [Code] value).

"receiverID": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value).

"receiverParentAccount": See [AccountsList] method (use [Account] value).

"receiverParentAccountName":

"receiverParentAccountsID": See [AccountsList] method (use [OA_AccountsID] value).

"Status": See [StatusesList] method (use [Name] value).

"taxPayerCode":

"taxPayerName":

"treasuryCode":

"createAccountingEntries": Creates accounting entries using managerial information. Default=false

Examples

1. Update Bank Transfer with OA_TransactionsID – 5053 and specify amount – 83 GEL.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 5053,
  "amount": 83
}
```

Server response Code(200):

```
{
  "id": 5053,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 0",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/BankTransfer

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value
"oA_TransactionsID":

Examples

1. Delete Bank Transfer with OA_TransactionsID – 5081.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 5081
}
```

Server response Code(200):

```
{
  "id": 5081,
  "idFieldName": "OA_TransactionsID"
}
```


Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 5081",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "BankTransfers",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

GetBankTransferByID

POST/api/GetBankTransferByID

Gets Bank Transfer and its accounting entries by OA_TransactionsID value.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

Examples

1. Get Bank Transfer and its accounting entries by OA_TransactionsID – 5083.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 5083
}
```

Server response Code(200):

```
{
  "oA_TransactionsID": 5083,
  "additionalInfo": "",
  "amount": 99,
  "bankAccount": "1210",
  "bankAccountName": "Bank - national currency (resident banks)",
  "bankAccountsID": 48,
  "comment": "",
  "costCentreName": "",
  "documentalConfirm": false,
  "documentDate": "2023-06-02T13:34:16.387",
  "documentNumber": 16,
  "oA_CostCentreID": 0,
  "oA_StatusID": 0,
  "oA_TransactionTypesID": 1,
  "payer": "Organisation",
  "payerBankAccount": "",
  "payerBankCode": "TBCBGE22",
  "payerBankID": 1,
  "payerBankName": "TBC Bank",
  "payerCode": "2",
}
```

```

"payerID": 0,
"receiver": "Organization2",
"receiverBankAccount": "2222",
"receiverBankCode": "BNLNGE22",
"receiverBankID": 1,
"receiverBankName": "National Bank Of Georgia",
"receiverCode": "2",
"receiverID": 2,
"receiverParentAccount": "3110",
"receiverParentAccountName": "Accounts payable - trade",
"receiverParentAccountsID": 119,
"reflectInformationInEntries": false,
"status": "",
"taxPayerCode": "",
"taxPayerName": "",
"treasuryCode": "",
"transactionEntries": []
}

```

Server response Code(400):

```

[
{
  "errorNumber": 51051,
  "description": "Record does not exist or read prohibited for user!",
  "comment": "OA_TransactionsID = 50837",
  "command": "",
  "errorType": "ErrorType_Error",
  "source": "OA_BusinessObject.dll",
  "class": "Transaction",
  "entryPoint": "GetProperties",
  "resourceDLL_ForTranslation": "ORIS_Accounting",
  "errorPointer": 0
}
]

```

BankTransfersList

POST/api/BankTransfersList

Gets list of bank transfer. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

[Token] is necessary.

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1. Example: get transfers by document number =3 and receiver bank code=BNLNGE221 - "[Document number]=N'3' AND [Receiver bank code]=N'BNLNGE221"

"sort": Use SQL ORDER BY string with field names. Example: "[Receiver]"

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get list of bank transfer which document number starts with 1, sort by date decreasing

Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "topRecordsCount": 1,
  "filter": "UPPER([ViewOA_BankTransfers].[Document number]) LIKE N'%1%'",
  "sort": "[Document date] DESC",
  "pageNumber": 2
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 6,
      "PageCount": 6,
      "PageRowCount": 1,
      "PageNumber": 2
    }
  ],
  "Result": [
    {
      "OA_TransactionsID": 5083,
      "Comment": "",
      "Documental confirm": false,
      "Delay": false,
      "Transaction date": "2023-06-02T13:34:16.387",
      "Create date": null,
      "Last update date": null,
      "OA_UsersID": 1,
      "OA_TransactionTypesID": 1,
      "OA_DepartmentsID": 1,
      "Payer": "Organisation",
      "PayerBankID": 1,
      "Payer bank account": "",
      "Tax payer code": "",
      "Tax payer name": "",
      "ReceiverID": 2,
      "Receiver": "Organization2",
      "ReceiverBankID": 1,
      "Receiver bank account": "2222",
      "Amount": 99.00000000,
      "Additional information": "",
      "Payer bank code": "TBCBGE22",
      "Payer bank name": "TBC Bank",
      "Receiver bank code": "BNLNGE22",
      "Receiver bank name": "National Bank Of Georgia",
      "Treasury code": "",
      "Document number": "16",
      "User": "sa",
      "Department": "Administrators",
      "Document date": "2023-06-02T13:34:16.387",
      "Payer code": "2",
      "Receiver code": "2",
      "BankAccountsID": 48,
      "ReceiverParentAccountsID": 119,
      "OA_CostCentreID": null,
      "Bank account": "1210",
      "Receiver parent account": "3110",
      "Cost centre": null,
      "Has entries": false,
      "Record related binary documents": null,
      "OA_StatusID": null,
      "Status": null,
      "Numeric document number": 16,
      "NUM_ROW": 6
    }
  ]
}
```

BankTransferForeign

POST/api/BankTransferForeign

Adds new Bank Transfer foreign. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "additionalInfo": "string",
  "amount": 0,
  "bankAccount": "string",
  "bankAccountName": "string",
  "bankAccountsID": 0,
  "beneficiaryBankSWIFT": "string",
  "city": "string",
  "comment": "string",
  "costCentreName": "string",
  "countryCode": "string",
  "currency": "string",
  "currencyID": 0,
  "debitAmountAccount": "string",
  "debitTransferAccount": "string",
  "documentDate": "2023-04-26T12:30:01.227Z",
  "documentNumber": "string",
  "intermediateBankName": "string",
  "intermediateBankServiceType": "string",
  "intermediateBankSWIFT": "string",
  "oA_CostCentreID": 0,
  "oA_StatusID": 0,
  "payer": "string",
  "payerAddress": "string",
  "payerBankCode": "string",
  "payerBankID": 0,
  "payerBankName": "string",
  "payerCode": "string",
  "receiver": "string",
  "receiverAddress": "string",
  "receiverBankAccount": "string",
  "receiverBankCode": "string",
  "receiverBankID": 0,
  "receiverBankName": "string",
  "receiverID": 0,
  "receiverParentAccount": "string",
  "receiverParentAccountName": "string",
  "receiverParentAccountsID": 0,
  "registrationCountryCode": "string",
  "reportingValue": "string",
  "status": "string",
  "taxPayerCode": "string",
  "taxPayerName": "string",
  "treasuryCode": "string",
  "createAccountingEntries": true
}
```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"additionalInfo":

"amount":

"bankAccount": Bank account from chart of accounts. See [AccountsList] method (use [Account] value). Can be omitted if [BankAccountsID] property specified. If both are omitted, gets the account from the parameters.

"bankAccountName":

"bankAccountsID": Bank account ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [BankAccount] property specified. If both are omitted, gets the account from the parameters.
"beneficiaryBankSWIFT":
"city":
"comment":
"costCentreName": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If both are omitted, gets the Cost Centre from the parameters.
"countryCode":
"currency": See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified
"currencyID": See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currency] property specified
"debitAmountAccount":
"debitTransferAccount":
"documentDate": Sets operation date. If not specified - sets current date.
"documentNumber": Sets the transaction document number. If not specified, sets a unique number in the current year.
"intermediateBankName":
"intermediateBankServiceType":
"intermediateBankSWIFT":
"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreName] property specified. If both are omitted, gets the Cost Centre from the parameters.
"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified
"payer":
"payerAddress":
"payerBankCode":
"payerBankID": See [Owner] method (use [OA_OwnerBankAccountsID] value from [OwnerBankAccounts]).
"payerBankName":
"payerCode":
"receiver":
"receiverAddress":
"receiverBankAccount":
"receiverBankCode": See [BanksList] method (use [Code] value). Can be omitted if [ReceiverBankID] property specified. If both are omitted, gets the account from the parameters.
"receiverBankID": See [BanksList] method (use [OA_BanksID] value). Can be omitted if [ReceiverCode] property specified
"receiverBankName":
"receiverID": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [ReceiverCode] property specified
"receiverParentAccount": See [AccountsList] method (use [Account] value). Can be omitted if [ReceiverParentAccountsID] property specified. If both are omitted, gets the account from the parameters.
"receiverParentAccountName":
"receiverParentAccountsID": See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [ReceiverParentAccount] property specified. If both are omitted, gets the account from the parameters.
"registrationCountryCode":
"reportingValue":
"status": See [StatusesList] method (use [Name] value). Can be omitted if [OA_StatusID] property specified.
"taxPayerCode":
"taxPayerName":
"treasuryCode":
"createAccountingEntries": Creates accounting entries using managerial information. Default=false

Examples

1. Add new Bank Transfer foreign - with specified amount – 12 USD, bankAccountsID – 49 Bank - foreign currency (resident banks), currencyID – 2, documentNumber – 15, OA_CostCentreID – 1 General cost center, payerBankID – 8 TBC Bank, receiverBankID – 14 Bank of Georgia, receiverID – 258 and receiverParentAccountsID – 119 Accounts payable – trade 3110.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "amount": 123,
  "bankAccountsID": 49,
  "currencyID": 2,
  "documentDate": "2023-06-05T11:48:32.779Z",
  "documentNumber": "15",
  "oA_CostCentreID": 1,

```

```
"payerBankID": 8,  
"receiverBankID": 14,  
"receiverID": 258,  
"receiverParentAccountsID": 119  
}
```

Server response Code(200):

```
{  
  "id": 5103,  
  "idFieldName": "OA_TransactionsID"  
}
```

Server response Code(400):

```
[  
  {  
    "errorNumber": 51262,  
    "description": "Bank account required!",  
    "comment": "OA_TransactionsID=0 [15]\n",  
    "command": "",  
    "errorType": "ErrorType_Permission",  
    "source": "OA_BusinessObject.dll",  
    "class": "BankTransferForeign",  
    "entryPoint": "CheckInputsForTransaction",  
    "resourceDLL_ForTranslation": "ORIS_Accounting",  
    "errorPointer": 0  
  }  
]
```

PUT/api/BankTransferForeign

Updates Bank Transfer foreign. Returns OK or Error description Json.

Schema

```
{  
  "token": "string",  
  "oA_TransactionsID": 0,  
  "additionalInfo": "string",  
  "amount": 0,  
  "bankAccount": "string",  
  "bankAccountName": "string",  
  "bankAccountsID": 0,  
  "beneficiaryBankSWIFT": "string",  
  "city": "string",  
  "comment": "string",  
  "costCentreName": "string",  
  "countryCode": "string",  
  "currency": "string",  
  "currencyID": 0,  
  "debitAmountAccount": "string",  
  "debitTransferAccount": "string",  
  "documentDate": "2023-04-26T13:09:59.716Z",  
  "documentNumber": "string",  
  "intermediateBankName": "string",  
  "intermediateBankServiceType": "string",  
  "intermediateBankSWIFT": "string",  
  "oA_CostCentreID": 0,  
  "oA_StatusID": 0,  
  "payer": "string",  
  "payerAddress": "string",  
  "payerBankCode": "string",  
  "payerBankID": 0,  
  "payerBankName": "string",  
  "payerCode": "string",  
  "receiver": "string",  
  "receiverAddress": "string",  
}
```

```

"receiverBankAccount": "string",
"receiverBankCode": "string",
"receiverBankID": 0,
"receiverBankName": "string",
"receiverID": 0,
"receiverParentAccount": "string",
"receiverParentAccountName": "string",
"receiverParentAccountsID": 0,
"registrationCountryCode": "string",
"reportingValue": "string",
"status": "string",
"taxPayerCode": "string",
"taxPayerName": "string",
"treasuryCode": "string",
"createAccountingEntries": true
}

```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID": See [BankTransferForeignsList] method (use [OA_TransactionsID] value).

"additionalInfo":

"amount":

"bankAccount": Bank account from chart of accounts. See [AccountsList] method (use [Account] value).

"bankAccountName":

"bankAccountsID": Bank account ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"beneficiaryBankSWIFT":

"city":

"comment":

"costCentreName": See [CostCentresList] method (use [Name] value).

"countryCode":

"currency": See [CurrenciesList] method (use [Currency] value).

"currencyID": See [CurrenciesList] method (use [OA_CurrenciesID] value).

"debitAmountAccount":

"debitTransferAccount":

"documentDate": Sets operation date.

"documentNumber":

"intermediateBankName":

"intermediateBankServiceType":

"intermediateBankSWIFT":

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value).

"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value).

"payer":

"payerAddress":

"payerBankCode":

"payerBankID": See [Owner] method (use [OA_OwnerBankAccountsID] value from [OwnerBankAccounts]).

"payerBankName": See [Owner] method (use [bankName] value from [OwnerBankAccounts]).

"payerCode": See [Owner] method (use [BankCode] value from [OwnerBankAccounts]).

"receiver":

"receiverAddress":

"receiverBankAccount":

"receiverBankCode": See [BanksList] method (use [Code] value).

"receiverBankID": See [BanksList] method (use [OA_BanksID] value).

"receiverBankName":

"receiverID": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value).

"receiverParentAccount": See [AccountsList] method (use [Account] value).

"receiverParentAccountName":

"receiverParentAccountsID": See [AccountsList] method (use [OA_AccountsID] value).

"registrationCountryCode":

"reportingValue":

"status": See [StatusesList] method (use [Name] value).

"taxPayerCode":

"taxPayerName":

"treasuryCode":

"createAccountingEntries": Creates accounting entries using managerial information. Default=false

Examples

1. Update Bank Transfer foreign with OA_TransactionsID – "5102", specify amount – "300", specify comment - "change currency" and specify currencyID – "3".

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 5102,
  "amount": 300,
  "comment": "change currency",
  "currencyID": 3
}
```

Server response Code(200):

```
{
  "id": 5102,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 0",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/BankTransferForeign

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

Examples

1. Delete Bank Transfer foreign with OA_TransactionsID – "5091".

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 5091
}
```

Server response Code(200):

```
{
  "id": 5091,
}
```



```
"idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
{
  "errorNumber": 51051,
  "description": "Record does not exist or read prohibited for user!",
  "comment": "OA_TransactionsID = 5091",
  "command": "",
  "errorType": "ErrorType_Error",
  "source": "OA_BusinessObject.dll",
  "class": "BankTransferForeign",
  "entryPoint": "GetProperties",
  "resourceDLL_ForTranslation": "ORIS_Accounting",
  "errorPointer": 0
}
]
```

GetBankTransferForeignByID

POST/api/GetBankTransferForeignByID

Gets Bank Transfer Foreign and its accounting entries by OA_TransactionsID value.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"**token**": Set [token] string from [LogIn] response value
"**oA_TransactionsID**":

Examples

1. Get Bank Transfer Foreign and its accounting entries by OA_TransactionsID – 5103.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 5103
}
```

Server response Code(200):

```
{
  "oA_TransactionsID": 5103,
  "additionalInfo": "",
  "amount": 123,
  "bankAccount": "1220",
  "bankAccountName": "Bank - foreign currency (resident banks)",
  "bankAccountsID": 49,
  "beneficiaryBankSWIFT": "",
  "city": "",
  "comment": "",
  "costCentreName": "General cost center",
  "countryCode": "",
  "currency": "USD",
  "currencyID": 2,
  "debitAmountAccount": "",
  "debitTransferAccount": "",
  "documentalConfirm": false,
}
```

```
"documentDate": "2023-06-05T11:48:32.78",
"documentNumber": 11,
"intermediateBankName": "",
"intermediateBankServiceType": "",
"intermediateBankSWIFT": "",
"oA_CostCentreID": 1,
"oA_StatusID": 0,
"oA_TransactionTypesID": 2,
"payer": "Organisation",
"payerAddress": "Al. Kazbegi Ave.",
"payerBankCode": "TBCBGE22",
"payerBankID": 8,
"payerBankName": "TBC Bank",
"payerCode": "2",
"payerID": 0,
"receiver": "",
"receiverAddress": "",
"receiverBankAccount": "",
"receiverBankCode": "",
"receiverBankID": 14,
"receiverBankName": "",
"receiverID": 258,
"receiverParentAccount": "3110",
"receiverParentAccountName": "Accounts payable - trade",
"receiverParentAccountsID": 119,
"reflectInformationInEntries": false,
"registrationCountryCode": "",
"reportingValue": "",
"status": "",
"taxPayerCode": "",
"taxPayerName": "",
"treasuryCode": "",
"transactionEntries": []
}
```

Server response Code(400):

```
[
{
  "errorNumber": 51051,
  "description": "Record does not exist or read prohibited for user!",
  "comment": "OA_TransactionsID = 51038",
  "command": "",
  "errorType": "ErrorType_Error",
  "source": "OA_BusinessObject.dll",
  "class": "Transaction",
  "entryPoint": "GetProperties",
  "resourceDLL_ForTranslation": "ORIS_Accounting",
  "errorPointer": 0
}
]
```

BankTransferForeignsList

POST/api/BankTransferForeignsList

Gets list of bank transfer foreign. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

[Token] is necessary.

"**token**": Set [token] string from [Login] response value

"**topRecordsCount**": The number of records to accept. If you specify 0 you will get all records. Default=0

"**filter**": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1. Example: get transfers by document number =3 and receiver bank code=BNLNGE221

"[Document number]=N'3' AND [Receiver bank code]=N'BNLNGE221"

"**sort**": Use SQL ORDER BY string with field names. Example: "[Receiver]"

"**pageNumber**": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get list of bank transfer foreign.

Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "topRecordsCount": 1,
  "filter": "UPPER([ViewOA_BankTransferForeign].[Currency]) LIKE N'%EUR'",
  "sort": "[Document number] DESC",
  "pageNumber": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 3,
      "PageCount": 3,
      "PageRowCount": 1,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_TransactionsID": 5085,
      "Create date": null,
      "Payer": "Organisation",
      "PayerBankID": 1,
      "Payer bank account": null,
      "Receiver": "",
      "ReceiverID": null,
      "ReceiverBankID": null,
      "Amount": 333.00000000,
      "Payer bank code": "TBCBGE22",
      "Receiver bank code": "",
      "Receiver bank name": "",
      "Receiver bank account": "",
      "Payer bank name": "TBC Bank",
      "Receiver address": "",
      "Receiver bank SWIFT": "",
      "Intermediate bank name": "",
      "Intermediate bank SWIFT": "",
      "Intermediate bank service type": "",
      "Debit amount account": "",
      "Debit transfers account": "",
      "OA_TransactionTypesID": 2,
      "Receiver account": null,
      "Comment": "",
      "Document date": "2023-06-05T13:04:27",
      "Payer code": "2",
      "Currency": "EUR",
      "Delay": false,
      "Documental confirm": false,
      "Transaction date": "2023-06-05T13:04:27",
      "Last update date": null,
    }
  ]
}
```

```

    "User": "sa",
    "Department": "Administrators",
    "Document number": "2",
    "Payer address": "Al. Kazbegi Ave.",
    "BankAccountsID": null,
    "ReceiverParentAccountsID": 119,
    "OA_CostCentreID": null,
    "Bank account": null,
    "Receiver parent account": "3110",
    "Cost centre": null,
    "Has entries": false,
    "OA_UsersID": 1,
    "OA_DepartmentsID": 1,
    "Record related binary documents": null,
    "Registration country code": "",
    "Country code": "",
    "City": "",
    "Reporting value": "",
    "OA_StatusID": null,
    "Status": null,
    "Numeric document number": 2,
    "NUM_ROW": 2
  }
]
}

```

CashIncomeOrder

POST/api/CashIncomeOrder

Adds new Cash Income Order. Returns ID value or Error description Json.

Schema

```

{
  "token": "string",
  "amount": 0,
  "appendix": "string",
  "base": "string",
  "cashAccountInOtherCurrency": "string",
  "cashAccountInOtherCurrencyID": 0,
  "cashAccountInRegisterCurrency": "string",
  "cashAccountInRegisterCurrencyID": 0,
  "cashAccountNameInOtherCurrency": "string",
  "cashAccountNameInRegisterCurrency": "string",
  "cashflowCategories": "string",
  "cashflowCategoriesName": "string",
  "cashierID": 0,
  "corrAccount": "string",
  "costCentreName": "string",
  "currency": "string",
  "documentDate": "2023-04-28T11:45:30.610Z",
  "documentNumber": "string",
  "memorial": 0,
  "oA_AccountsID": 0,
  "oA_CashflowCategoriesID": 0,
  "oA_CostCentreID": 0,
  "oA_CurrenciesID": 0,
  "oA_StatusID": 0,
  "payer": "string",
  "payerID": 0,
  "purposeCode": "string",
  "status": "string",
  "createAccountingEntries": true
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"**token**": Set [token] string from [LogIn] response value

"**amount**":

"**appendix**":

"**base**":

"**cashAccountInOtherCurrency**": Cash account in other currency from chart of accounts. See [AccountsList] method (use [Account] value). Can be omitted if [CashAccountInOtherCurrencyID] property specified. If both are omitted, gets the account from the parameters.

"**cashAccountInOtherCurrencyID**": Cash account ID in other currency from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [CashAccountInOtherCurrency] property specified. If both are omitted, gets the account from the parameters.

"**cashAccountInRegisterCurrency**": Cash account in register currency from chart of accounts. See [AccountsList] method (use [Account] value). Can be omitted if [CashAccountInRegisterCurrencyID] property specified. If both are omitted, gets the account from the parameters.

"**cashAccountInRegisterCurrencyID**": Cash account in register currency from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [CashAccountInRegisterCurrency] property specified. If both are omitted, gets the account from the parameters.

"**cashAccountNameInOtherCurrency**":

"**cashAccountNameInRegisterCurrency**":

"**cashflowCategories**": See [CashflowCategoriesList] method (use [Abridgment] value). Can be omitted if [OA_CashflowCategoriesID] property specified

"**cashflowCategoriesName**":

"**cashierID**": See [EmployeeList] method (use [OA_EmployeeID] value).

"**corrAccount**": Correspondent account from chart of accounts. See [AccountsList] method (use [Account] value). Can be omitted if [OA_AccountsID] property specified

"**costCentreName**": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified

"**currency**": See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified.

"**documentDate**": Sets operation date. If not specified - sets current date.

"**documentNumber**": Sets the transaction document number. If not specified, sets a unique number in the current year.

"**memorial**":

"**oA_AccountsID**": Correspondent account from chart of accounts. See [AccountsList] method (use [Account] value). Can be omitted if [CorrAccount] property specified

"**oA_CashflowCategoriesID**": See [CashflowCategoriesList] method (use [OA_CashflowCategoriesID] value). Can be omitted if [CashflowCategories] property specified

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreName] property specified

"**oA_CurrenciesID**": See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currency] property specified

"**oA_StatusID**": See [StatusList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified

"**payer**": See [DebitorCreditorList] method (use [Code] value). Can be omitted if [PayerID] property specified

"**payerID**": See [DebitorCreditorList] method (use [OA_DebitorCreditorID] value). Can be omitted if [Payer] property specified

"**purposeCode**":

"**status**": See [StatusList] method (use [Name] value). Can be omitted if [OA_StatusID] property specified

"**createAccountingEntries**": Creates accounting entries using managerial information. Default=false

Examples

1. Add new Cash Income Order - with specified amount – 555 GEL, cashAccountInOtherCurrency – 1110 1, corrAccount – "1110 2", currency – "EUR" and documentNumber – "15".

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "amount": 555,
  "cashAccountInOtherCurrency": "1110 1",
  "corrAccount": "1110 2",
  "currency": "EUR",
  "documentNumber": "15"
}
```

Server response Code(200):

```
{
```

```
"id": 5111,  
"idFieldName": "OA_TransactionsID"  
}
```

Server response Code(400):

```
[  
{  
  "errorNumber": 51190,  
  "description": "Property not found!",  
  "comment": "OA_TransactionsID=0 [string]\n Status=[string]",  
  "command": "",  
  "errorType": "ErrorType_Error",  
  "source": "OA_BusinessObject.dll",  
  "class": "CashIncomeOrder",  
  "entryPoint": "set property",  
  "resourceDLL_ForTranslation": "ORIS_Accounting",  
  "errorPointer": 0  
}  
]
```

PUT/api/CashIncomeOrder

Updates Cash Income Order. Returns OK or Error description Json.

Schema

```
{  
  "token": "string",  
  "oA_TransactionsID": 0,  
  "amount": 0,  
  "appendix": "string",  
  "base": "string",  
  "cashAccountInOtherCurrency": "string",  
  "cashAccountInOtherCurrencyID": 0,  
  "cashAccountInRegisterCurrency": "string",  
  "cashAccountInRegisterCurrencyID": 0,  
  "cashAccountNameInOtherCurrency": "string",  
  "cashAccountNameInRegisterCurrency": "string",  
  "cashflowCategories": "string",  
  "cashflowCategoriesName": "string",  
  "cashierID": 0,  
  "corrAccount": "string",  
  "costCentreName": "string",  
  "currency": "string",  
  "documentDate": "2023-04-28T12:15:10.117Z",  
  "documentNumber": "string",  
  "memorial": 0,  
  "oA_AccountsID": 0,  
  "oA_CashflowCategoriesID": 0,  
  "oA_CostCentreID": 0,  
  "oA_CurrenciesID": 0,  
  "oA_StatusID": 0,  
  "payer": "string",  
  "payerID": 0,  
  "purposeCode": "string",  
  "status": "string",  
  "createAccountingEntries": true  
}
```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID": See [CashIncomeOrdersList] method (use [OA_TransactionsID] value).

"amount":

"**appendix**":

"**base**":

"**cashAccountInOtherCurrency**": Cash account in other currency from chart of accounts. See [AccountsList] method (use [Account] value).

"**cashAccountInOtherCurrencyID**": Cash account ID in other currency from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**cashAccountInRegisterCurrency**": Cash account in register currency from chart of accounts. See [AccountsList] method (use [Account] value).

"**cashAccountInRegisterCurrencyID**": Cash account in register currency from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**cashAccountNameInOtherCurrency**":

"**cashAccountNameInRegisterCurrency**":

"**cashflowCategories**": See [CashflowCategoriesList] method (use [Abridgment] value).

"**cashflowCategoriesName**":

"**cashierID**": See [EmployeeList] method (use [OA_EmployeeID] value).

"**corrAccount**": Correspondent account from chart of accounts. See [AccountsList] method (use [Account] value).

"**costCentreName**": See [CostCentresList] method (use [Name] value).

"**currency**": See [CurrenciesList] method (use [Currency] value).

"**documentDate**": Sets operation date.

"**documentNumber**":

"**memorial**":

"**oA_AccountsID**": Correspondent account from chart of accounts. See [AccountsList] method (use [Account] value).

"**oA_CashflowCategoriesID**": See [CashflowCategoriesList] method (use [OA_CashflowCategoriesID] value).

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value).

"**oA_CurrenciesID**": See [CurrenciesList] method (use [OA_CurrenciesID] value).

"**oA_StatusID**": See [StatusList] method (use [OA_StatusID] value).

"**payer**": See [DebitorCreditorList] method (use [Code] value).

"**payerID**": See [DebitorCreditorList] method (use [OA_DebitorCreditorID] value).

"**purposeCode**":

"**status**": See [StatusList] method (use [Name] value).

"**createAccountingEntries**": Creates accounting entries using managerial information. Default=false

Examples

1. Update Cash Income Order – with OA_TransactionsID – 5117, amount - 270 and currency – USD.

Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "oA_TransactionsID": 5117,
  "amount": 270,
  "currency": "USD"
}
```

Server response Code(200):

```
{
  "id": 5117,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51190,
    "description": "Property not found!",
    "comment": "OA_TransactionsID=5117 [string]\n Status=[string]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "CashIncomeOrder",
    "entryPoint": "set property",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  },
  {
    "errorNumber": 51026,
    "description": "Currency required!",
    "comment": " Transaction date [06.06.2023 07:52:51] \nOA_TransactionsID [5117] Entry number [1]",
  }
]
```

```

"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "Transaction",
"entryPoint": "SetOA_CurrenciesID",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

DELETE/api/CashIncomeOrder

Returns OK or Error description Json.

Schema

```

{
  "token": "string",
  "oA_TransactionsID": 0
}

```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

Examples

1. Delete Cash Income Order - with specified OA_TransactionsID – 5109.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 5109
}

```

Server response Code(200):

```

{
  "id": 5109,
  "idFieldName": "OA_TransactionsID"
}

```

Server response Code(400):

```

[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 5109",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

GetCashIncomeOrderByID

POST/api/GetCashIncomeOrderByID

Gets Cash Income Order and its accounting entries by OA_TransactionsID value.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

Examples

1. Get Cash Income Order and its accounting entries by OA_TransactionsID – 5110.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 5110
}
```

Server response Code(200):

```
{
  "oA_TransactionsID": 5110,
  "amount": 333,
  "appendix": "",
  "base": "",
  "cashAccountInOtherCurrency": "1110 1",
  "cashAccountInOtherCurrencyID": 265,
  "cashAccountInRegisterCurrency": "1110",
  "cashAccountInRegisterCurrencyID": 46,
  "cashAccountNameInOtherCurrency": "Cashier N1",
  "cashAccountNameInRegisterCurrency": "Cash on hand national currency",
  "cashflowCategories": null,
  "cashflowCategoriesName": null,
  "cashier": "",
  "cashierID": 0,
  "corrAccount": "1110 2",
  "corrAccountName": "Cashier N1",
  "costCentreName": "",
  "currency": "EUR",
  "documentalConfirm": false,
  "documentDate": "2023-06-06T11:48:23",
  "documentNumber": 4,
  "memorial": 0,
  "oA_AccountsID": 220,
  "oA_CashflowCategoriesID": 0,
  "oA_CostCentreID": 0,
  "oA_CurrenciesID": 3,
  "oA_StatusID": 0,
  "oA_TransactionTypesID": 0,
  "payer": "",
  "payerID": 0,
  "purposeCode": "",
  "reflectInformationInEntries": true,
  "status": "",
  "transactionEntries": []
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
  }
]
```

```

"comment": "OA_TransactionsID = 51101",
"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "Transaction",
"entryPoint": "GetProperties",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

CashIncomeOrdersList

POST/api/CashIncomeOrdersList

Gets list of Cash Income Orders. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

[Token] is necessary.

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property: [TopRecordsCount]=1. Example: get transfers by document number =3 and currencu=USD - "[Document number]=N'3' AND [Currency]=N'USD'"

"sort": Use SQL ORDER BY string with field names. Example: "[Payer]"

"pageNumber" records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get list of Cash Income Orders.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "filter": "[ViewOA_CashIncomeOrders].[Has entries]=1",
  "sort": "[Amount] DESC",
  "pageNumber": 2
}

```

Server response Code(200):

```

{
  "Info": [
    {
      "RowCount": 2,
      "PageCount": 2,
      "PageRowCount": 1,
      "PageNumber": 2
    }
  ],
  "Result": [
    {
      "OA_TransactionsID": 5107,
      "Date": "2023-06-06T11:34:30",
      "OA_CashflowCategoriesID": null,
      "Analytical code": null,

```

```

"Appendix": "",
"PayerID": null,
"OA_CurrenciesID": 1,
"Currency": "GEL",
"Document number": "1",
"Payer": "",
"Purpose code": "",
"TransactionReceiverCashAccountInRegisterCurrencyID": 265,
"Transaction receiver cash account in register currency": "1110 1",
"TransactionReceiverCashAccountInOtherCurrencyID": 47,
"Transaction receiver cash account in other currency": "1120",
"OA_TransactionTypesID": 4,
"OA_CostCentreID": 1,
"Cost centre": "General cost center",
"Amount": 15.00000000,
"Delay": false,
"Documental confirm": false,
"Transaction date": "2023-06-06T11:34:30",
"Create date": null,
"Last update date": null,
"User": "sa",
"Department": "Administrators",
"Base": "",
"Has entries": true,
"OA_UsersID": 1,
"OA_DepartmentsID": 1,
"Corresponding account": "1110 2",
"Corresponding account name": " Cashier N1",
"OA_AccountsID": 220,
"OA_ItemsID": null,
"OA_DebitorCreditorID": null,
"Record related binary documents": null,
"OA_StatusID": null,
>Status": null,
"OA_DebitorCreditorIDCashier": null,
"Cashier": null,
"Memorial": null,
"Numeric document number": 1,
"NUM_ROW": 1
}
]
}

```

CashOutLayOrder

POST/api/CashOutLayOrder

Adds new Cash Out Lay Order. Returns ID value or Error description Json.

Schema

```

{
  "token": "string",
  "amount": 0,
  "appendix": "string",
  "base": "string",
  "cashAccountInOtherCurrency": "string",
  "cashAccountInOtherCurrencyID": 0,
  "cashAccountInRegisterCurrency": "string",
  "cashAccountInRegisterCurrencyID": 0,
  "cashAccountNameInOtherCurrency": "string",
  "cashAccountNameInRegisterCurrency": "string",
  "cashflowCategories": "string",
  "cashflowCategoriesName": "string",
  "cashierID": 0,
  "corrAccount": "string",

```

```

"costCentreName": "string",
"currency": "string",
"dateAndPlace": "string",
"document": "string",
"documentDate": "2023-04-28T14:00:07.859Z",
"documentNumber": "string",
"memorial": 0,
"oA_AccountsID": 0,
"oA_CashflowCategoriesID": 0,
"oA_CostCentreID": 0,
"oA_CurrenciesID": 0,
"oA_StatusID": 0,
"purposeCode": "string",
"receiver": "string",
"receiverID": 0,
"status": "string",
"createAccountingEntries": true
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"amount":

"appendix":

"base":

"cashAccountInOtherCurrency": Cash account in other currency from chart of accounts. See [AccountsList] method (use [Account] value). Can be omitted if [CashAccountInOtherCurrencyID] property specified

"cashAccountInOtherCurrencyID": Cash account ID in other currency from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [CashAccountInOtherCurrency] property specified

"cashAccountInRegisterCurrency": Cash account in register currency from chart of accounts. See [AccountsList] method (use [Account] value). Can be omitted if [CashAccountInRegisterCurrencyID] property specified

"cashAccountInRegisterCurrencyID": Cash account in register currency from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [CashAccountInRegisterCurrency] property specified

"cashAccountNameInOtherCurrency":

"cashAccountNameInRegisterCurrency":

"cashflowCategories": See [CashflowCategoriesList] method (use [Abridgment] value). Can be omitted if [OA_CashflowCategoriesID] property specified

"cashflowCategoriesName":

"cashierID": See [EmployeeList] method (use [OA_EmployeeID] value).

"corrAccount": Correspondent account from chart of accounts. See [AccountsList] method (use [Account] value). Can be omitted if [OA_AccountsID] property specified

"costCentreName": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified

"currency": See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified.

"dateAndPlace":

"document":

"documentDate": Sets operation date. If not specified - sets current date.

"documentNumber": Sets the transaction document number. If not specified, sets a unique number in the current year.

"memorial":

"oA_AccountsID": Correspondent account from chart of accounts. See [AccountsList] method (use [Account] value). Can be omitted if [CorrAccount] property specified

"oA_CashflowCategoriesID": See [CashflowCategoriesList] method (use [OA_CashflowCategoriesID] value). Can be omitted if [CashflowCategories] property specified

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreName] property specified

"oA_CurrenciesID": See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currency] property specified

"oA_StatusID": See [StatusList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified

"purposeCode":

"receiver": See [DebitorCreditorList] method (use [Code] value). Can be omitted if [ReceiverID] property specified

"receiverID": See [DebitorCreditorList] method (use [OA_DebitorCreditorID] value). Can be omitted if [Receiver] property specified

"status": See [StatusList] method (use [Name] value). Can be omitted if [OA_StatusID] property specified

"createAccountingEntries": Creates accounting entries using managerial information. Default=false

Examples

1. Add new Cash Out Lay Order - with specified amount – “5”, cashAccountInOtherCurrency – “1110 1” and currency – “USD”.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "amount": 5,
  "cashAccountInOtherCurrency": "1110 1",
  "currency": "USD"
}
```

Server response Code(200):

```
{
  "id": 5125,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51190,
    "description": "Property not found!",
    "comment": "OA_TransactionsID=0 [4]\n Status=[string]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "CashOutLayOrder",
    "entryPoint": "set property",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  },
  {
    "errorNumber": 51262,
    "description": "Bank account required!",
    "comment": "OA_TransactionsID=0 [4]\n",
    "command": "",
    "errorType": "ErrorType_Permission",
    "source": "OA_BusinessObject.dll",
    "class": "CashOutLayOrder",
    "entryPoint": "CheckInputsForTransaction",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/CashOutLayOrder

Updates Cash Out Lay Order. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "amount": 0,
  "appendix": "string",
  "base": "string",
  "cashAccountInOtherCurrency": "string",
  "cashAccountInOtherCurrencyID": 0,
  "cashAccountInRegisterCurrency": "string",
  "cashAccountInRegisterCurrencyID": 0,
  "cashAccountNameInOtherCurrency": "string",
  "cashAccountNameInRegisterCurrency": "string",
  "cashflowCategories": "string",
}
```

```

"cashflowCategoriesName": "string",
"cashierID": 0,
"corrAccount": "string",
"costCentreName": "string",
"currency": "string",
"dateAndPlace": "string",
"document": "string",
"documentDate": "2023-05-01T11:42:07.413Z",
"documentNumber": "string",
"memorial": 0,
"oA_AccountsID": 0,
"oA_CashflowCategoriesID": 0,
"oA_CostCentreID": 0,
"oA_CurrenciesID": 0,
"oA_StatusID": 0,
"purposeCode": "string",
"receiver": "string",
"receiverID": 0,
"status": "string",
"createAccountingEntries": true
}

```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID": See [CashoutlayOrdersList] method (use [OA_TransactionsID] value).

"amount":

"appendix":

"base":

"cashAccountInOtherCurrency": Cash account in other currency from chart of accounts. See [AccountsList] method (use [Account] value).

"cashAccountInOtherCurrencyID": Cash account ID in other currency from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"cashAccountInRegisterCurrency": Cash account in register currency from chart of accounts. See [AccountsList] method (use [Account] value).

"cashAccountInRegisterCurrencyID": Cash account in register currency from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"cashAccountNameInOtherCurrency":

"cashAccountNameInRegisterCurrency":

"cashflowCategories": See [CashflowCategoriesList] method (use [Abridgment] value).

"cashflowCategoriesName":

"cashierID": See [EmployeeList] method (use [OA_EmployeeID] value).

"corrAccount": Correspondent account from chart of accounts. See [AccountsList] method (use [Account] value).

"costCentreName": See [CostCentresList] method (use [Name] value).

"currency": See [CurrenciesList] method (use [Currency] value).

"dateAndPlace":

"document":

"documentDate": Sets operation date.

"documentNumber":

"memorial":

"oA_AccountsID": Correspondent account from chart of accounts. See [AccountsList] method (use [Account] value).

"oA_CashflowCategoriesID": See [CashflowCategoriesList] method (use [OA_CashflowCategoriesID] value).

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value).

"oA_CurrenciesID": See [CurrenciesList] method (use [OA_CurrenciesID] value).

"oA_StatusID": See [StatusList] method (use [OA_StatusID] value).

"purposeCode":

"receiver": See [DebitorCreditorList] method (use [Code] value).

"receiverID": See [DebitorCreditorList] method (use [OA_DebitorCreditorID] value).

"status": See [StatusList] method (use [Name] value).

"createAccountingEntries": Creates accounting entries using managerial information. Default=false

Examples

1. Update Cash Out Lay Order - with OA_TransactionsID - 5126, amount – 456, costCentreName – New cost center and currency – GEL.

Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "oA_TransactionsID": 5126,
  "amount": 456,
  "costCentreName": "New cost center",
  "currency": "GEL"
}
```

Server response Code(200):

```
{
  "id": 5126,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 51267",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/CashOutLayOrder

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [Login] response value

"oA_TransactionsID":

Examples

1. Delete Cash Out Lay Order - with specified OA_TransactionsID – 5126.

Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "oA_TransactionsID": 5126
}
```

Server response Code(200):

```
{
  "id": 5126,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
```

```
"errorNumber": 51051,
"description": "Record does not exist or read prohibited for user!",
"comment": "OA_TransactionsID = 5126",
"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "Transaction",
"entryPoint": "GetProperties",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]
```

GetCashOutLayOrderByID

POST/api/GetCashOutLayOrderByID

Gets Cash Out Lay Order and its accounting entries by OA_TransactionsID value.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

Examples

1. Get Cash Out Lay Order and its accounting entries by OA_TransactionsID – 5128.

```
Request body:
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 5128
}
```

```
Server response Code(200):
{
  "oA_TransactionsID": 5128,
  "amount": 23,
  "appendix": "",
  "base": "",
  "cashAccountInOtherCurrency": "1120",
  "cashAccountInOtherCurrencyID": 47,
  "cashAccountInRegisterCurrency": "1110 1",
  "cashAccountInRegisterCurrencyID": 265,
  "cashAccountNameInOtherCurrency": "Cash on hand foreign currency",
  "cashAccountNameInRegisterCurrency": " Cashier N1 ",
  "cashflowCategories": null,
  "cashflowCategoriesName": null,
  "cashier": "",
  "cashierID": 0,
  "corrAccount": "1110 2",
  "corrAccountName": " Cashier N1",
  "costCentreName": "General cost center",
  "currency": "EUR",
  "dateAndPlace": "",
  "document": "",
  "documentalConfirm": false,
  "documentDate": "2023-06-06T14:58:06",
  "documentNumber": 3,
}
```



```
"memorial": 0,
"oA_AccountsID": 220,
"oA_CashflowCategoriesID": 0,
"oA_CostCentreID": 1,
"oA_CurrenciesID": 3,
"oA_StatusID": 0,
"oA_TransactionTypesID": 0,
"purposeCode": "",
"receiver": "",
"receiverID": 0,
"reflectInformationInEntries": true,
"status": "",
"transactionEntries": [
  {
    "oA_EntriesID": 4212,
    "oA_TransactionsID": 5128,
    "account": "1120",
    "accountName": "Cash on hand foreign currency",
    "debitAmount": 0,
    "creditAmount": 23,
    "currency": "EUR",
    "debitQuantity": 0,
    "creditQuantity": 0,
    "unit": "",
    "costCentre": "General cost center",
    "costUnit": "Total cost unit",
    "amountCalculationType": 0,
    "comment": "",
    "cashFlow": "",
    "memorial": 0,
    "rate": 2.8023,
    "debitEquivalent": 0,
    "creditEquivalent": 64.4529,
    "rate2": 1.06921286,
    "debitEquivalent2": 0,
    "creditEquivalent2": 24.59189578,
    "stockRegisterRate": 0,
    "debitStock": 0,
    "creditStock": 0,
    "debitStockAvailable": 0,
    "creditStockAvailable": 0,
    "isRateFixed": false,
    "internalTurnover": false,
    "actionInCaseDisbalance": 0,
    "sequenceNumber": 0,
    "accountNonBalance": false,
    "oA_AccountsID": 47,
    "mainEntry": true,
    "mainEntryID": 0,
    "oA_CashflowCategoriesID": 0,
    "oA_CurrenciesID": 3,
    "oA_CostCentreID": 1,
    "oA_CostUnitID": 1,
    "oA_StockUnitsID": 0,
    "accountLeaf": true,
    "accountClose": false,
    "accountType": 0,
    "oA_DebitorCreditorID": 0,
    "isPerson": false,
    "oA_ItemsID": 0,
    "itemType": 0,
    "oA_StockUnitsIDRegisterUnit": 0,
    "registerUnit": "",
    "hasADoubleRegistrationUnit": false,
    "accountCategory": 0,
    "accountWriteOffmethod": 0,
    "relatedOA_EntriesID": 0,
```

```
"relatedSequenceNumber": 0,
"transferWithinCostCentreGroup": false,
"oA_CostCentreGroupID": 0,
"corrOA_AccountsID": 0,
"correspondentAccount": "",
"isReverseEntry": false,
"relatedOA_TransactionsID": 0,
"relatedDocumentNumber": ""
},
{
  "oA_EntriesID": 4213,
  "oA_TransactionsID": 5128,
  "account": "1110 2",
  "accountName": " Cashier N1",
  "debitAmount": 23,
  "creditAmount": 0,
  "currency": "EUR",
  "debitQuantity": 0,
  "creditQuantity": 0,
  "unit": "",
  "costCentre": "General cost center",
  "costUnit": "Total cost unit",
  "amountCalculationType": 0,
  "comment": "",
  "cashFlow": "",
  "memorial": 0,
  "rate": 2.8023,
  "debitEquivalent": 64.4529,
  "creditEquivalent": 0,
  "rate2": 1.06921286,
  "debitEquivalent2": 24.59189578,
  "creditEquivalent2": 0,
  "stockRegisterRate": 0,
  "debitStock": 0,
  "creditStock": 0,
  "debitStockAvailable": 0,
  "creditStockAvailable": 0,
  "isRateFixed": false,
  "internalTurnover": false,
  "actionInCaseDisbalance": 0,
  "sequenceNumber": 0,
  "accountNonBalance": false,
  "oA_AccountsID": 220,
  "mainEntry": false,
  "mainEntryID": 4212,
  "oA_CashflowCategoriesID": 0,
  "oA_CurrenciesID": 3,
  "oA_CostCentreID": 1,
  "oA_CostUnitID": 1,
  "oA_StockUnitsID": 0,
  "accountLeaf": true,
  "accountClose": false,
  "accountType": 0,
  "oA_DebitorCreditorID": 0,
  "isPerson": false,
  "oA_ItemsID": 0,
  "itemType": 0,
  "oA_StockUnitsIDRegisterUnit": 0,
  "registerUnit": "",
  "hasADoubleRegistrationUnit": false,
  "accountCategory": 0,
  "accountWriteOffmethod": 0,
  "relatedOA_EntriesID": 0,
  "relatedSequenceNumber": 0,
  "transferWithinCostCentreGroup": false,
  "oA_CostCentreGroupID": 0,
  "corrOA_AccountsID": 47,
```

```

    "correspondentAccount": "1120",
    "isReverseEntry": false,
    "relatedOA_TransactionsID": 0,
    "relatedDocumentNumber": ""
  }
]
}

```

Server response Code(400):

```

[
{
  "errorNumber": 51051,
  "description": "Record does not exist or read prohibited for user!",
  "comment": "OA_TransactionsID = 512828",
  "command": "",
  "errorType": "ErrorType_Error",
  "source": "OA_BusinessObject.dll",
  "class": "Transaction",
  "entryPoint": "GetProperties",
  "resourceDLL_ForTranslation": "ORIS_Accounting",
  "errorPointer": 0
}
]

```

CashOutLayOrdersList

POST/api/CashOutLayOrdersList

Gets list of Out Lay Orders. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

[Token] is necessary.

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property: [TopRecordsCount]=1. Example: get transfers by document number =3 and currencu=USD - "[Document number]=N'3' AND [Currency]=N'USD'"

"sort": Use SQL ORDER BY string with field names. Example: "[Receiver]"

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get list of Out Lay Orders.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "filter": "[ViewOA_CashOutlayOrders].[Amount]>=23",
  "sort": "[Amount]",
  "pageNumber": 1
}

```

Server response Code(200):

```

{
  "Info": [

```

```

{
  "RowCount": 2,
  "PageCount": 2,
  "PageRowCount": 1,
  "PageNumber": 1
}
],
"Result": [
{
  "OA_TransactionsID": 5128,
  "Date": "2023-06-06T14:58:06",
  "OA_CashflowCategoriesID": null,
  "Analytical code": null,
  "Appendix": "",
  "ReceiverID": null,
  "Receiver": "",
  "OA_CurrenciesID": 3,
  "Currency": "EUR",
  "Document number": "3",
  "Purpose code": "",
  "TransactionReceiverCashAccountInRegisterCurrencyID": 265,
  "Transaction receiver cash account in register currency": "1110 1",
  "TransactionReceiverCashAccountInOtherCurrencyID": 47,
  "Transaction receiver cash account in other currency": "1120",
  "OA_TransactionTypesID": 3,
  "OA_CostCentreID": 1,
  "Cost centre": "General cost center",
  "Amount": 23.00000000,
  "Delay": false,
  "Documental confirm": false,
  "Transaction date": "2023-06-06T14:58:06",
  "Create date": null,
  "Last update date": null,
  "User": "sa",
  "Department": "Administrators",
  "Document": "",
  "Date and place": "",
  "Base": "",
  "Has entries": true,
  "OA_UsersID": 1,
  "OA_DepartmentsID": 1,
  "Corresponding account": "1110 2",
  "Corresponding account name": " Cashier N1",
  "OA_AccountsID": 220,
  "OA_ItemsID": null,
  "OA_DebitorCreditorID": null,
  "Record related binary documents": null,
  "OA_StatusID": null,
  "Status": null,
  "OA_DebitorCreditorIDCashier": null,
  "Cashier": null,
  "Memorial": null,
  "Numeric document number": 3,
  "NUM_ROW": 2
}
]
}

```

Accounting – Results

BalancesList

POST/api/BalancesList

Gets balances by chart of accounts. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "currenciesIDs": "string",
  "costCentreIDs": "string",
  "costUnitIDs": "string",
  "date": "2023-05-04T13:05:57.688Z",
  "showZeroResults": true,
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

[Token] and [Date] is necessary.

"token": Set [token] string from [LogIn] response value

"currenciesIDs": Currency ID or "0" - Consolidate in register currency or "-1" - Consolidate in second register currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Default="1", if license is [Singlecurrency] or ="0", otherwise.

"costCentreIDs": Comma separated cost centre IDs or "" - All cost centres. See [CostCentresList] method (use [OA_CostCentreID] value). Default=""

"costUnitIDs": Comma separated cost unit IDs or "" - All cost units. See [CostUnitsList] method (use [OA_CostUnitID] value). Default=""

"date": Date of balance (including date).

"showZeroResults": If false - removes rows with **zero** values from the list of results. Default=false.

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property: [TopRecordsCount]=1. Example: get account by ID=123 - "[OA_AccountsID]=123" or get accounts whose numbers start with 11 and level is 2 - "[Account] LIKE N'11%' AND [Level]=2"

"sort": Use SQL ORDER BY string with field names. Example: "[Account],[Name]"

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get balances by date 2021-12-31 by chart of accounts.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "date": "2021-12-31T08:47:31.358Z",
  "topRecordsCount": 2,
  "pageNumber": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 331,
      "PageCount": 166,
      "PageRowCount": 2,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_AccountsID": 1,
      "Account": "1000",
      "ParentID": 0,
      "Name": "Carrent assets",
      "Level": 0,
      "Comment": "",
      "Close": false,
    }
  ]
}
```

```

    "Leaf": false,
    "Enable Minus Stock Balance": false,
    "Type": 0,
    "WriteOff method": 0,
    "Nonbalance": false,
    "Category": 0,
    "ParentIDs": "46,10,48,11,12,56,59,13,14,69,70,73,15,16,17,18,1",
    "HasResult": true,
    "IsRoot": true,
    "Debit balance": 0.0,
    "Credit balance": 0.0,
    "Debit equivalent": 101470.94,
    "Credit equivalent": 0.0,
    "Debit equivalent2": 37451.443492,
    "Credit equivalent2": 0.0,
    "NUM_ROW": 1
  }
]
}

```

TurnoverList

POST/api/TurnoverList

Gets turnover by chart of accounts. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "currenciesIDs": "string",
  "costCentreIDs": "string",
  "costUnitIDs": "string",
  "diapazonStartDate": "2023-05-10T12:21:14.393Z",
  "diapazonEndDate": "2023-05-10T12:21:14.393Z",
  "showZeroResults": true,
  "includeInternalTurns": true,
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

[Token],[DiapazonStartDate] and [DiapazonEndDate] is necessary.

"token": Set [token] string from [LogIn] response value

"currenciesIDs": Currency ID or "0" - Consolidate in register currency or "-1" - Consolidate in second register currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Default="1", if license is [Singlecurrency] or ="0", otherwise.

"costCentreIDs": Comma separated cost centre IDs or "" - All cost centres. See [CostCentresList] method (use [OA_CostCentreID] value). Default=""

"costUnitIDs": Comma separated cost unit IDs or "" - All cost units. See [CostUnitsList] method (use [OA_CostUnitID] value). Default=""

"diapazonStartDate": Start date of turnover.

"diapazonEndDate": "End date of turnover (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01", "DiapazonEndDate":"2011-01-01"

"showZeroResults": If false - removes rows with zero values from the list of results. Default=false.

"includeInternalTurns":

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1. Example: get account by ID=123 - "[OA_AccountsID]=123" or get accounts whose numbers start with 11 and level is 2 - "[Account] LIKE N'11%' AND [Level]=2"

"sort": Use SQL ORDER BY string with field names. Example: "[Account],[Name]"

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get turnover by chart of accounts.

Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "diapazonStartDate": "2021-01-01T13:24:45.332Z",
  "diapazonEndDate": "2022-01-01T13:24:45.332Z",
  "topRecordsCount": 2,
  "filter": "UPPER([ViewTurnover].[Account]) LIKE N'%2%'",
  "sort": "[Level] DESC",
  "pageNumber": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 176,
      "PageCount": 88,
      "PageRowCount": 2,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_AccountsID": 216,
      "Account": "1210 1",
      "ParentID": 48,
      "Name": "Bank of Georgia",
      "Level": 3,
      "Comment": "",
      "Close": false,
      "Leaf": true,
      "Enable Minus Stock Balance": false,
      "Type": 0,
      "WriteOff method": 0,
      "Nonbalance": false,
      "Category": 0,
      "Parent account": "1210",
      "HasResult": true,
      "IsRoot": true,
      "Start debit balance": 0.0,
      "Start credit balance": 0.0,
      "Debit balance": 0.0,
      "Credit balance": 0.0,
      "Start stock balance": 0.0,
      "Start debit equivalent": 270.94,
      "Start credit equivalent": 0.0,
      "Debit equivalent": 270.94,
      "Credit equivalent": 0.0,
      "Start debit equivalent2": 100.0,
      "Start credit equivalent2": 0.0,
      "Debit equivalent2": 100.0,
      "Credit equivalent2": 0.0,
      "Difference of turnover": 0.0,
      "Difference of turnover equivalent": 0.0,
      "Difference of turnover equivalent2": 0.0,
      "NUM_ROW": 91
    }
  ]
}
```

AccountBalancesList

POST/api/AccountBalancesList

Gets one account balances. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "date": "2023-05-04T12:24:22.799Z",
  "account": "string",
  "showZeroResults": true,
  "topRecordsCount": 0,
  "reportType": 0,
  "showResultsInSaleUnit": true
}
```

Field description

[Token],[Date] and [Account] is necessary.

"**token**": Set [token] string from [LogIn] response value

"**date**": Date of balance (including date).

"**account**": The account for which you want to receive a statement. See [AccountsList] method (use [Account] value).

"**showZeroResults**": If false - removes rows with **zero** values from the list of results. Default=false.

"**topRecordsCount**": The number of records to accept. If you specify 0 you will get all records. Default=0

"**reportType**": Report type. 0-Expanded, 1-Compressed. Default=0.

"**showResultsInSaleUnit**": Show stock balances in item sale unit. Default=False.

Examples

1. Get "1210 1" account balances

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "date": "2021-12-31T12:04:24.288Z",
  "account": "1210 1",
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_AccountsID": 216,
      "Account": "1210 1",
      "ParentID": 48,
      "Name": "Bank of Georgia",
      "Close": false,
      "Leaf": true,
      "Level": 3,
      "Type": 0,
      "WriteOffMethod": 0,
      "Nonbalance": false,
      "Category": 0,
      "OA_CostCentreID": 2,
      "OA_CostUnitID": 2,
      "OA_CurrenciesID": 2,
      "Cost centre": "New cost center",
      "Cost unit": "New cost unit",
      "Debit balance": 100.0,
      "Currency": "USD",
      "Stock balance": 0.0,
    }
  ]
}
```



```

    "Debit equivalent": 270.94,
    "Debit equivalent2": 100.0,
    "HasResult": true
  }
]
}

```

AccountStatementList

POST/api/AccountStatementList

Gets account statement. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "currenciesIDs": "string",
  "costCentreIDs": "string",
  "costUnitIDs": "string",
  "diapazonStartDate": "2023-05-04T12:51:04.408Z",
  "diapazonEndDate": "2023-05-04T12:51:04.408Z",
  "account": "string",
  "topRecordsCount": 0,
  "pageNumber": 0
}

```

Field description

[Token],[DiapazonStartDate],[DiapazonEndDate] and [Account] is necessary.

"token": Set [token] string from [LogIn] response value

"currenciesIDs": Currency ID or "0" - Consolidate in register currency or "-1" - Consolidate in second register currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Default="1", if license is [Singlecurrency] or ="0", otherwise.

"costCentreIDs": Comma separated cost centre IDs or "" - All cost centres. See [CostCentresList] method (use [OA_CostCentreID] value). Default=""

"costUnitIDs": Comma separated cost unit IDs or "" - All cost units. See [CostUnitsList] method (use [OA_CostUnitID] value). Default=""

"diapazonStartDate": Diapazon start date.

"diapazonEndDate": Diapazon end date (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01", "DiapazonEndDate":"2011-01-01"

"account": The account for which you want to receive a statement. See [AccountsList] method (use [Account] value).

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get account statement

Request body:

Server response Code(200):

AccountStatementDetailedList

POST/api/AccountStatementDetailedList

Gets account statement detailed. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "currenciesIDs": "string",
  "costCentreIDs": "string",

```

```

"costUnitIDs": "string",
"diapazonStartDate": "2023-05-04T12:41:21.695Z",
"diapazonEndDate": "2023-05-04T12:41:21.695Z",
"account": "string",
"topRecordsCount": 0,
"pageNumber": 0
}

```

Field description

[Token],[DiapazonStartDate],[DiapazonEndDate] and [Account] is necessary.

"token": Set [token] string from [LogIn] response value

"currenciesIDs": Currency ID or "0" - Consolidate in register currency or "-1" - Consolidate in second register currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Default="1", if license is [Singlecurrency] or ="0", otherwise.

"costCentreIDs": Comma separated cost centre IDs or "" - All cost centres. See [CostCentresList] method (use [OA_CostCentreID] value). Default=""

"costUnitIDs": Comma separated cost unit IDs or "" - All cost units. See [CostUnitsList] method (use [OA_CostUnitID] value). Default=""

"diapazonStartDate": Diapazon start date.

"diapazonEndDate": Diapazon end date (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01", "DiapazonEndDate":"2011-01-01"

"account": The account for which you want to receive a statement. See [AccountsList] method (use [Account] value).

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get "1110 1" account statement detailed.

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"diapazonStartDate": "2021-01-01T12:28:33.948Z",
"diapazonEndDate": "2022-01-01T12:28:33.948Z",
"account": "1110 1"
}

```

Server response Code(200):

```

{
"Info": [
{
"RowCount": 0,
"PageCount": 0,
"PageRowCount": 0,
"PageNumber": 0
}
],
"Result": [
{
"Correspondent account": "Start",
"Delay": false,
"Documental confirm": false,
"Main entry": false,
"Debit balance": 0.0,
"Credit balance": 0.0,
"Debit equivalent balance": 1000.0,
"Credit equivalent balance": 0.0,
"Debit equivalent2 balance": 369.08541,
"Credit equivalent2 balance": 0.0,
"Stock balance": 0.0,
"SortType": 0
},
{
"Correspondent account": "Final",
"Delay": false,
"Documental confirm": false,
"Debit amount": 0.0,

```

```

    "Credit amount": 0.0,
    "Debit equivalent": 0.0,
    "Credit equivalent": 0.0,
    "Debit equivalent2": 0.0,
    "Credit equivalent2": 0.0,
    "Debit stock": 0.0,
    "Credit stock": 0.0,
    "Main entry": false,
    "Credit balance": 0.0,
    "Debit equivalent balance": 1000.0,
    "Debit equivalent2 balance": 369.08541,
    "SortType": 2
  }
]
}

```

JournalOrderList

POST/api/JournalOrderList

Gets account journal order. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "currenciesIDs": "string",
  "costCentreIDs": "string",
  "costUnitIDs": "string",
  "diapazonStartDate": "2023-05-10T09:06:49.782Z",
  "diapazonEndDate": "2023-05-10T09:06:49.783Z",
  "account": "string",
  "topRecordsCount": 0,
  "pageNumber": 0
}

```

Field description

[Token],[DiapazonStartDate] and [DiapazonEndDate] is necessary.

"token": Set [token] string from [LogIn] response value

"currenciesIDs": Currency ID or "0" - Consolidate in register currency or "-1" - Consolidate in second register currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Default="1", if license is [Singlecurrency] or ="0", otherwise.

"costCentreIDs": Comma separated cost centre IDs or "" - All cost centres. See [CostCentresList] method (use [OA_CostCentreID] value). Default=""

"costUnitIDs": Comma separated cost unit IDs or "" - All cost units. See [CostUnitsList] method (use [OA_CostUnitID] value). Default=""

"diapazonStartDate":

"diapazonEndDate": Diapazon end date (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01", "DiapazonEndDate":"2011-01-01"

"account": The account for which you want to receive a journal order. See [AccountsList] method (use [Account] value).

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get account journal order.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "diapazonStartDate": "2023-05-01T11:13:13.984Z",
  "diapazonEndDate": "2023-06-01T11:13:13.984Z",
  "account": "1410 2",
  "topRecordsCount": 1,
  "pageNumber": 1
}

```

}

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 0,
      "PageCount": 0,
      "PageRowCount": 1,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "Range": "01.05.2023",
      "CorrAccountID": 258,
      "Correspondent account": "Start",
      "Level": 0,
      "OA_ItemsID": null,
      "OA_DebitorCreditorID": null,
      "Category": null,
      "Correspondent account name": null,
      "ParentID": 0,
      "Debit amount": null,
      "Credit amount": null,
      "Debit equivalent": null,
      "Credit equivalent": null,
      "Debit equivalent2": null,
      "Credit equivalent2": null,
      "Debit stock": null,
      "Credit stock": null,
      "Debit balance": 0.0,
      "Credit balance": 0.0,
      "Debit equivalent balance": 100.0,
      "Credit equivalent balance": 0.0,
      "Debit equivalent2 balance": 36.908541,
      "Credit equivalent2 balance": 0.0,
      "Stock balance": 0.0,
      "DateFrom": null,
      "DateTo": null,
      "SortType": 0,
      "ID": 1
    },
    {
      "Range": "",
      "CorrAccountID": 2,
      "Correspondent account": "2000",
      "Level": 0,
      "OA_ItemsID": null,
      "OA_DebitorCreditorID": null,
      "Category": null,
      "Correspondent account name": "Long - Term Assets",
      "ParentID": 0,
      "Debit amount": 0.0,
      "Credit amount": 20.0,
      "Debit equivalent": 0.0,
      "Credit equivalent": 20.0,
      "Debit equivalent2": 0.0,
      "Credit equivalent2": 8.0547724,
      "Debit stock": 0.0,
      "Credit stock": 0.0,
      "Debit balance": 0.0,
      "Credit balance": 0.0,
      "Debit equivalent balance": 0.0,
      "Credit equivalent balance": 0.0,
      "Debit equivalent2 balance": 0.0,
    }
  ]
}
```

```
"Credit equivalent2 balance": 0.0,
"Stock balance": 0.0,
"DateFrom": "2023-05-01T00:00:00",
"DateTo": "2023-06-01T00:00:00",
"SortType": 1,
"ID": 2
},
{
  "Range": null,
  "CorrAccountID": 19,
  "Correspondent account": "2100",
  "Level": 1,
  "OA_ItemsID": null,
  "OA_DebitorCreditorID": null,
  "Category": null,
  "Correspondent account name": "Fixed assets",
  "ParentID": 2,
  "Debit amount": 0.0,
  "Credit amount": 20.0,
  "Debit equivalent": 0.0,
  "Credit equivalent": 20.0,
  "Debit equivalent2": 0.0,
  "Credit equivalent2": 8.0547724,
  "Debit stock": 0.0,
  "Credit stock": 0.0,
  "Debit balance": 0.0,
  "Credit balance": 0.0,
  "Debit equivalent balance": 0.0,
  "Credit equivalent balance": 0.0,
  "Debit equivalent2 balance": 0.0,
  "Credit equivalent2 balance": 0.0,
  "Stock balance": 0.0,
  "DateFrom": "2023-05-01T00:00:00",
  "DateTo": "2023-06-01T00:00:00",
  "SortType": 1,
  "ID": 3
},
{
  "Range": null,
  "CorrAccountID": 82,
  "Correspondent account": "2110",
  "Level": 2,
  "OA_ItemsID": null,
  "OA_DebitorCreditorID": null,
  "Category": null,
  "Correspondent account name": "Land",
  "ParentID": 3,
  "Debit amount": 0.0,
  "Credit amount": 20.0,
  "Debit equivalent": 0.0,
  "Credit equivalent": 20.0,
  "Debit equivalent2": 0.0,
  "Credit equivalent2": 8.0547724,
  "Debit stock": 0.0,
  "Credit stock": 0.0,
  "Debit balance": 0.0,
  "Credit balance": 0.0,
  "Debit equivalent balance": 0.0,
  "Credit equivalent balance": 0.0,
  "Debit equivalent2 balance": 0.0,
  "Credit equivalent2 balance": 0.0,
  "Stock balance": 0.0,
  "DateFrom": "2023-05-01T00:00:00",
  "DateTo": "2023-06-01T00:00:00",
  "SortType": 1,
  "ID": 4
},
}
```

```

{
  "Range": null,
  "CorrAccountID": 1232,
  "Correspondent account": "2110 1010",
  "Level": 3,
  "OA_ItemsID": 1010,
  "OA_DebitorCreditorID": null,
  "Category": 1,
  "Correspondent account name": "21 FixedAsset",
  "ParentID": 4,
  "Debit amount": 0.0,
  "Credit amount": 20.0,
  "Debit equivalent": 0.0,
  "Credit equivalent": 20.0,
  "Debit equivalent2": 0.0,
  "Credit equivalent2": 8.0547724,
  "Debit stock": 0.0,
  "Credit stock": 0.0,
  "Debit balance": 0.0,
  "Credit balance": 0.0,
  "Debit equivalent balance": 0.0,
  "Credit equivalent balance": 0.0,
  "Debit equivalent2 balance": 0.0,
  "Credit equivalent2 balance": 0.0,
  "Stock balance": 0.0,
  "DateFrom": "2023-05-01T00:00:00",
  "DateTo": "2023-06-01T00:00:00",
  "SortType": 1,
  "ID": 5
},
{
  "Range": null,
  "CorrAccountID": null,
  "Correspondent account": "Sum",
  "Level": 0,
  "OA_ItemsID": null,
  "OA_DebitorCreditorID": null,
  "Category": null,
  "Correspondent account name": null,
  "ParentID": 0,
  "Debit amount": 0.0,
  "Credit amount": 20.0,
  "Debit equivalent": 0.0,
  "Credit equivalent": 20.0,
  "Debit equivalent2": 0.0,
  "Credit equivalent2": 8.0547724,
  "Debit stock": 0.0,
  "Credit stock": 0.0,
  "Debit balance": 0.0,
  "Credit balance": 20.0,
  "Debit equivalent balance": 80.0,
  "Credit equivalent balance": 0.0,
  "Debit equivalent2 balance": 28.853768600000002,
  "Credit equivalent2 balance": 0.0,
  "Stock balance": 0.0,
  "DateFrom": "2023-05-01T00:00:00",
  "DateTo": "2023-06-01T00:00:00",
  "SortType": 1,
  "ID": 6
},
{
  "Range": "01.06.2023",
  "CorrAccountID": null,
  "Correspondent account": "Total",
  "Level": 0,
  "OA_ItemsID": null,
  "OA_DebitorCreditorID": null,

```

```

"Category": null,
"Correspondent account name": null,
"ParentID": 0,
"Debit amount": 0.0,
"Credit amount": 20.0,
"Debit equivalent": 0.0,
"Credit equivalent": 20.0,
"Debit equivalent2": 0.0,
"Credit equivalent2": 8.0547724,
"Debit stock": 0.0,
"Credit stock": 0.0,
"Debit balance": 0.0,
"Credit balance": 0.0,
"Debit equivalent balance": 80.0,
"Credit equivalent balance": 0.0,
"Debit equivalent2 balance": 28.8537686,
"Credit equivalent2 balance": 0.0,
"Stock balance": null,
"DateFrom": null,
"DateTo": null,
"SortType": 2,
"ID": 7
}
]
}

```

CashflowsList

POST/api/CashflowsList

Gets cashflow. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "currenciesIDs": "string",
  "costCentreIDs": "string",
  "costUnitIDs": "string",
  "diapazonStartDate": "2023-05-08T07:39:17.623Z",
  "diapazonEndDate": "2023-05-08T07:39:17.623Z",
  "showZeroResults": true,
  "reportType": 0,
  "topRecordsCount": 0,
  "pageNumber": 0
}

```

Field description

[Token],[DiapazonStartDate] and [DiapazonEndDate] is necessary.

"token": Set [token] string from [LogIn] response value

"currenciesIDs": Currency ID or "0" - Consolidate in register currency or "-1" - Consolidate in second register currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Default="1", if license is [Singlecurrency] or ="0", otherwise.

"costCentreIDs": Comma separated cost centre IDs or "" - All cost centres. See [CostCentresList] method (use [OA_CostCentreID] value). Default=""

"costUnitIDs": Comma separated cost unit IDs or "" - All cost units. See [CostUnitsList] method (use [OA_CostUnitID] value). Default=""

"diapazonStartDate": Diapazon start date.

"diapazonEndDate": Diapazon end date (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01", "DiapazonEndDate":"2011-01-01"

"showZeroResults": If false - removes rows with **zero** values from the list of results. Default=false.

"reportType": Report type: Expended = 0, Compressed = 1

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get cashflow.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "diapazonStartDate": "2021-01-01T11:46:57.445Z",
  "diapazonEndDate": "2022-01-01T11:46:57.445Z"
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 19,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "Level": 0,
      "Leaf": false,
      "OA_CashflowCategoriesID": 0,
      "Abridgment": "",
      "ParentID": 0,
      "Comment": "Start",
      "Alternative comment": "",
      "ID": 1,
      "SortType": 0,
      "Report type": 0,
      "Total": 101370.94
    },
    {
      "Level": 0,
      "Leaf": false,
      "OA_CashflowCategoriesID": 0,
      "Abridgment": "",
      "ParentID": 0,
      "Comment": "Final",
      "Alternative comment": "",
      "ID": 44,
      "SortType": 7,
      "Report type": 0,
      "Total": 101370.94
    }
  ]
}
```

MemorialList

POST/api/MemorialList

Gets memorial. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "currenciesIDs": "string",
  "costCentrelDs": "string",
  "costUnitIDs": "string",
  "diapazonStartDate": "2023-05-10T11:05:36.227Z",
  "diapazonEndDate": "2023-05-10T11:05:36.227Z",
  "memorial": 0,
}
```



```
"level": 0,
"topRecordsCount": 0,
"pageNumber": 0
}
```

Field description

[Token],[DiapazonStartDate] and [DiapazonEndDate] is necessary.

"**token**": Set [token] string from [LogIn] response value

"**currenciesIDs**": Currency ID or "0" - Consolidate in register currency or "-1" - Consolidate in second register currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Default="1", if license is [Singlecurrency] or ="0", otherwise.

"**costCentreIDs**": Comma separated cost centre IDs or "" - All cost centres. See [CostCentresList] method (use [OA_CostCentreID] value). Default=""

"**costUnitIDs**": Comma separated cost unit IDs or "" - All cost units. See [CostUnitsList] method (use [OA_CostUnitID] value). Default=""

"**diapazonStartDate**": Diapazon start date.

"**diapazonEndDate**": Diapazon end date (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01", "DiapazonEndDate":"2011-01-01"

"**memorial**": The number for which you want to receive a memorial.

"**level**": Account level for which you want to receive a memorial.

"**topRecordsCount**": The number of records to accept. If you specify 0 you will get all records. Default=0

"**pageNumber**": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get memorial.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "diapazonStartDate": "2021-01-01T13:25:12.868Z",
  "diapazonEndDate": "2023-05-31T13:25:12.868Z",
  "memorial": 2
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "DebitAccountID": 1,
      "CreditAccountID": 1,
      "Debit": "1000 - Carrent assets",
      "Credit": "1000 - Carrent assets",
      "Amount": 1000.0,
      "Memorial Level": 0,
      "Memorial": 2
    }
  ]
}
```

DebitorCreditorBalancesList

POST/api/DebitorCreditorBalancesList

Gets debitor/creditor balances. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "currenciesIDs": "string",
  "costCentreIDs": "string",
  "costUnitIDs": "string",
  "date": "2023-05-08T12:14:46.398Z",
  "showZeroResults": true,
  "parentAccounts": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

[Token] and [Date] is necessary.

"token": Set [token] string from [LogIn] response value

"currenciesIDs": Currency ID or "0" - Consolidate in register currency or "-1" - Consolidate in second register currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Default="1", if license is [Singlecurrency] or ="0", otherwise.

"costCentreIDs": Comma separated cost centre IDs or "" - All cost centres. See [CostCentresList] method (use [OA_CostCentreID] value). Default=""

"costUnitIDs": Comma separated cost unit IDs or "" - All cost units. See [CostUnitsList] method (use [OA_CostUnitID] value). Default=""

"date": Date of balance (including date).

"showZeroResults": If false - removes rows with zero values from the list of results. Default=false.

"parentAccounts": Comma separated parent accounts for which you want to receive a balances.

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property: [TopRecordsCount]=1.

"sort": Use SQL ORDER BY string with field names.

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get debtor/creditor balances.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "date": "2021-12-31T12:13:28.810Z"
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 16,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_DebitorCreditorID": 2,
      "Code": "2",
      "Name": "Person333",
      "Is person": true,
      "Address": "",
      "Zip": "",
      "Comment": "",
      "Clock number": "",
      "Is foreigner": false,
      "VAT type": 0,
      "Debit balance": 100.0,
      "HasResult": true
    }
  ]
}
```

```
}  
]  
}
```

DebitorCreditorTurnoverList

POST/api/DebitorCreditorTurnoverList

Gets debtor/creditor turnover. Returns [Info] and [Result] tables.

Schema

```
{  
  "token": "string",  
  "currenciesIDs": "string",  
  "costCentreIDs": "string",  
  "costUnitIDs": "string",  
  "diapazonStartDate": "2023-05-08T12:58:27.651Z",  
  "diapazonEndDate": "2023-05-08T12:58:27.651Z",  
  "showZeroResults": true,  
  "compressed": true,  
  "topRecordsCount": 0,  
  "filter": "string",  
  "sort": "string",  
  "pageNumber": 0  
}
```

Field description

[Token],[DiapazonStartDate] and [DiapazonEndDate] is necessary.

"token": Set [token] string from [LogIn] response value

"currenciesIDs": Currency ID or "0" - Consolidate in register currency or "-1" - Consolidate in second register currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Default="1", if license is [Singlecurrency] or ="0", otherwise.

"costCentreIDs": Comma separated cost centre IDs or "" - All cost centres. See [CostCentresList] method (use [OA_CostCentreID] value). Default=""

"costUnitIDs": Comma separated cost unit IDs or "" - All cost units. See [CostUnitsList] method (use [OA_CostUnitID] value). Default=""

"diapazonStartDate": Diapazon start date.

"diapazonEndDate": Diapazon end date (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01", "DiapazonEndDate":"2011-01-01"

"showZeroResults": If false - removes rows with zero values from the list of results. Default=false.

"compressed": Compressed debit and credit turnover.

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property: [TopRecordsCount]=1.

"sort": Use SQL ORDER BY string with field names.

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get debtor/creditor turnover.

Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "diapazonStartDate": "2021-01-01T12:55:52.683Z",  
  "diapazonEndDate": "2022-01-01T12:55:52.683Z"  
}
```

Server response Code(200):

```
{  
  "Info": [  
    {  
      "RowCount": 16,  
      "PageCount": 0,  
      "PageRowCount": 0,  
    }  
  ]  
}
```

```

    "PageNumber": 0
  }
],
"Result": [
  {
    "OA_DebitorCreditorID": 2,
    "Code": "2",
    "Name": "Person333",
    "Is person": true,
    "Address": "",
    "Zip": "",
    "Comment": "",
    "Clock number": "",
    "Is foreigner": false,
    "VAT type": 0,
    "HasResult": true,
    "Start balance": 100.0,
    "Balance": 100.0
  }
]
}

```

TrialBalanceList

POST/api/TrialBalanceList

Gets trial balance. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "currenciesIDs": "string",
  "costCentreIDs": "string",
  "costUnitIDs": "string",
  "date": "2023-05-10T12:01:13.826Z",
  "showZeroResults": true,
  "topRecordsCount": 0,
  "pageNumber": 0
}

```

Field description

[Token] and [Date] is necessary.

"token": Set [token] string from [LogIn] response value

"currenciesIDs": Currency ID or "0" - Consolidate in register currency or "-1" - Consolidate in second register currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Default="1", if license is [Singlecurrency] or ="0", otherwise.

"costCentreIDs": Comma separated cost centre IDs or "" - All cost centres. See [CostCentresList] method (use [OA_CostCentreID] value). Default=""

"costUnitIDs": Comma separated cost unit IDs or "" - All cost units. See [CostUnitsList] method (use [OA_CostUnitID] value). Default=""

"date": Date of balance (including date).

"showZeroResults": If false - removes rows with **zero** values from the list of results. Default=false.

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get trial balance.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "date": "2021-12-31T13:37:24.932Z",
  "topRecordsCount": 1,
  "pageNumber": 3
}

```

```

}

    Server response Code(200):
{
  "Info": [
    {
      "RowCount": 40,
      "PageCount": 40,
      "PageRowCount": 1,
      "PageNumber": 3
    }
  ],
  "Result": [
    {
      "ID": 1,
      "Account1": "1200",
      "Comment1": "Cash in Bank",
      "Balance1": 270.94,
      "Separator": "",
      "Account2": "Total",
      "Comment2": "Passives",
      "Balance2": 0.0
    },
    {
      "ID": 2,
      "Account1": "Sum",
      "Comment1": "(No group name)",
      "Balance1": 270.94,
      "Separator": "",
      "Account2": "",
      "Comment2": "",
      "Balance2": null
    },
    {
      "ID": 3,
      "Account1": "Total",
      "Comment1": "Actives",
      "Balance1": 270.94,
      "Separator": "",
      "Account2": "",
      "Comment2": "",
      "Balance2": null
    }
  ]
}

```

ProfitAndLossList

POST/api/ProfitAndLossList

Gets profit and loss. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "currenciesIDs": "string",
  "costCentreIDs": "string",
  "costUnitIDs": "string",
  "diapazonStartDate": "2023-05-10T11:41:09.127Z",
  "diapazonEndDate": "2023-05-10T11:41:09.127Z",
  "showZeroResults": true,
  "topRecordsCount": 0,
  "pageNumber": 0
}

```

Field description

[Token],[DiapazonStartDate] and [DiapazonEndDate] is necessary.

"**token**": Set [token] string from [LogIn] response value

"**currenciesIDs**": Currency ID or "0" - Consolidate in register currency or "-1" - Consolidate in second register currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Default="1", if license is [Singlecurrency] or ="0", otherwise.

"**costCentreIDs**": Comma separated cost centre IDs or "" - All cost centres. See [CostCentresList] method (use [OA_CostCentreID] value). Default=""

"**costUnitIDs**": Comma separated cost unit IDs or "" - All cost units. See [CostUnitsList] method (use [OA_CostUnitID] value). Default=""

"**diapazonStartDate**": Diapazon start date.

"**diapazonEndDate**": Diapazon end date (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01", "DiapazonEndDate":"2011-01-01"

"**showZeroResults**": If false - removes rows with **zero** values from the list of results. Default=false.

"**topRecordsCount**": The number of records to accept. If you specify 0 you will get all records. Default=0

"**pageNumber**": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get profit and loss.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "diapazonStartDate": "2021-01-01T13:00:48.236Z",
  "diapazonEndDate": "2023-05-26T13:00:48.236Z",
  "topRecordsCount": 1,
  "pageNumber": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 0,
      "PageCount": 0,
      "PageRowCount": 1,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_ProfitAndLossCategoriesID": 1,
      "Name": "Receipts",
      "Leaf": false,
      "ParentID": 0,
      "Level": 0,
      "OA_AccountsID": null,
      "Account": null,
      "Profit and loss category": "Receipts",
      "Parent category": null,
      "HasResult": true,
      "Total": 932.19
    },
    {
      "OA_ProfitAndLossCategoriesID": 2,
      "Name": "Income from sales",
      "Leaf": true,
      "ParentID": 1,
      "Level": 1,
      "OA_AccountsID": 167,
      "Account": "6110",
      "Profit and loss category": "Income from sales",
      "Parent category": "Receipts",
      "HasResult": true,
      "Total": 932.19
    }
  ]
}
```

```

{
  "OA_ProfitAndLossCategoriesID": 3,
  "Name": "Cost of product sold",
  "Leaf": false,
  "ParentID": 0,
  "Level": 0,
  "OA_AccountsID": null,
  "Account": null,
  "Profit and loss category": "Cost of product sold",
  "Parent category": null,
  "HasResult": true,
  "Total": -5.934
},
{
  "OA_ProfitAndLossCategoriesID": 5,
  "Name": "Acquisition/purchase",
  "Leaf": true,
  "ParentID": 3,
  "Level": 1,
  "OA_AccountsID": 180,
  "Account": "7210",
  "Profit and loss category": "Acquisition/purchase",
  "Parent category": "Cost of product sold",
  "HasResult": true,
  "Total": -255.424
},
{
  "OA_ProfitAndLossCategoriesID": 6,
  "Name": "Returns/discounts",
  "Leaf": true,
  "ParentID": 3,
  "Level": 1,
  "OA_AccountsID": 181,
  "Account": "7220",
  "Profit and loss category": "Returns/discounts",
  "Parent category": "Cost of product sold",
  "HasResult": true,
  "Total": 249.49
},
{
  "OA_ProfitAndLossCategoriesID": 9,
  "Name": "Administration expenses",
  "Leaf": false,
  "ParentID": 0,
  "Level": 0,
  "OA_AccountsID": null,
  "Account": null,
  "Profit and loss category": "Administration expenses",
  "Parent category": null,
  "HasResult": true,
  "Total": -13600.86
},
{
  "OA_ProfitAndLossCategoriesID": 10,
  "Name": "General and administrative expenses",
  "Leaf": true,
  "ParentID": 9,
  "Level": 1,
  "OA_AccountsID": 41,
  "Account": "7400",
  "Profit and loss category": "General and administrative expenses",
  "Parent category": "Administration expenses",
  "HasResult": true,
  "Total": -13600.86
}
]
}

```

TransactionsList

POST/api/TransactionsList

Gets list of transactions. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "diapazonStartDate": "2023-05-10T11:52:43.087Z",
  "diapazonEndDate": "2023-05-10T11:52:43.087Z",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

[Token],[DiapazonStartDate] and [DiapazonEndDate] is necessary.

"token": Set [token] string from [LogIn] response value

"diapazonStartDate": Diapazon start date.

"diapazonEndDate": Diapazon end date (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01", "DiapazonEndDate":"2011-01-01"

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1. Example: get transactions where Delay is false and OA_TransactionTypesID =16 - "[Delay]=0 AND [OA_TransactionTypesID]=16"

"sort": Use SQL ORDER BY string with field names. Example: "[Amount]"

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get list of transactions.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "diapazonStartDate": "2023-05-16T13:18:19.141Z",
  "diapazonEndDate": "2023-05-20T13:18:19.141Z",
  "topRecordsCount": 2,
  "filter": "UPPER([ViewOA_Transactions].[Document number]) LIKE N'%1%'",
  "sort": "[Transaction date] DESC",
  "pageNumber": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 0,
      "PageCount": 0,
      "PageRowCount": 2,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_TransactionsID": 2012,
      "Delay": false,
      "Documental confirm": false,
      "Transaction type": "Transfer operation",
      "Transaction date": "2023-05-19T17:45:15",
      "Document number": "1",
      "Comment": ""
    }
  ]
}
```



```

    "User": "sa",
    "Department": "Administrators",
    "OA_UsersID": 1,
    "OA_TransactionTypesID": 12,
    "OA_DepartmentsID": 1,
    "Amount": 0.00000000,
    "OA_CurrenciesID": 1,
    "Currency": "GEL",
    "Is start balance": false,
    "Has entries": true,
    "Related transactions": 0,
    "Have link": 0
  },
  {
    "OA_TransactionsID": 2004,
    "Delay": false,
    "Documental confirm": false,
    "Transaction type": "Manufacturing execution",
    "Transaction date": "2023-05-17T15:07:18",
    "Document number": "1",
    "Comment": "",
    "Create date": "2023-05-17T15:07:18",
    "Last update date": "2023-05-25T16:52:40",
    "User": "sa",
    "Department": "Administrators",
    "OA_UsersID": 1,
    "OA_TransactionTypesID": 22,
    "OA_DepartmentsID": 1,
    "Is start balance": false,
    "Has entries": false,
    "Related transactions": 0,
    "Have link": 0
  }
]
}

```

DualEntriesList

POST/api/DualEntriesList

Gets list of entries in dual entries style. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "diapazonStartDate": "2023-05-08T13:10:43.212Z",
  "diapazonEndDate": "2023-05-08T13:10:43.212Z",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

[Token],[DiapazonStartDate] and [DiapazonEndDate] is necessary.

"token":

"diapazonStartDate": Diapazon start date.

"diapazonEndDate": Diapazon end date (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01", "DiapazonEndDate":"2011-01-01"

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1.

"sort": Use SQL ORDER BY string with field names.

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get list of entries in dual entries style.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "diapazonStartDate": "2023-05-01T12:40:37.755Z",
  "diapazonEndDate": "2023-06-01T12:40:37.756Z",
  "topRecordsCount": 1,
  "filter": "UPPER([ViewOA_DualEntries].[Debit account]) LIKE N'%1410%'",
  "pageNumber": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 2,
      "PageCount": 2,
      "PageRowCount": 1,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_EntriesID": 4065,
      "DebitOA_AccountsID": 217,
      "Debit account": "1410 1",
      "Debit category": 2,
      "DebitOA_ItemsID": null,
      "Debit item type": null,
      "DebitOA_DebitorCreditorID": 1,
      "Debit is person": false,
      "Debit account name": "1 Organization",
      "DebitOA_CostCentreID": 1,
      "Debit cost centre": "General cost center",
      "DebitOA_CostUnitID": 1,
      "Debit cost unit": "Total cost unit",
      "Debit comment": null,
      "CreditOA_AccountID": 211,
      "Credit account": "6110 1",
      "Credit category": 0,
      "CreditOA_ItemsID": null,
      "Credit item type": null,
      "CreditOA_DebitorCreditorID": null,
      "Credit is person": null,
      "Credit account name": "Income for taxable sale",
      "CreditOA_CostCentreID": 1,
      "Credit cost centre": "General cost center",
      "CreditOA_CostUnitID": 1,
      "Credit cost unit": "Total cost unit",
      "Credit comment": null,
      "Amount": 8.470000,
      "Equivalent": 8.470000,
      "Equivalent2": 3.267873,
      "Debit quantity": 0.0,
      "Debit unit": null,
      "Debit stock": 0.0,
      "Credit quantity": 0.0,
      "Credit unit": null,
      "Credit stock": 0.0,
      "OA_CurrenciesID": 1,
      "Currency": "GEL",
      "OA_TransactionsID": 5050,
      "Delay": false,
      "Documental confirm": false,
    }
  ]
}
```

```
"Transaction date": "2023-05-31T15:11:41",
"Is start balance": false,
"OA_TransactionTypesID": 11,
"Transaction type": "Supply operation",
"Transaction comment": "",
"Document number": "3",
"RS Waybill number": null,
"RS VAT number": null,
"RS Tax document number": null,
"OA_UsersID": 1,
>User": "sa",
>Create date": null,
>Last update date": null,
>NUM_ROW": 1
}
]
}
```

Menegerial accounting

ProformalInvoice

POST/api/ProformalInvoice

Adds new Proforma Invoice. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "costCentreName": "string",
  "currency": "string",
  "documentComment": "string",
  "documentDate": "2023-05-04T10:17:20.696Z",
  "documentNumber": "string",
  "isRateFixed": true,
  "oA_CostCentreID": 0,
  "oA_CurrenciesID": 0,
  "oA_SalePriceTypesID": 0,
  "oA_StatusID": 0,
  "rate": 0,
  "receiverAddress": "string",
  "receiverBankAccount": "string",
  "receiverBankCode": "string",
  "receiverBankID": 0,
  "receiverCode": "string",
  "receiverID": 0,
  "salePriceType": "string",
  "status": "string",
  "supplierAddress": "string",
  "supplierBankAccount": "string",
  "supplierBankCode": "string",
  "supplierBankID": 0,
  "supplierBankName": "string",
  "supplierCode": "string",
  "proformalInvoiceItems": [
    {
      "code": "string",
      "oA_ItemsID": 0,
      "quantity": 0,
      "unit": "string",
      "oA_StockUnitsID": 0,
      "price": 0,
      "amount": 0,
      "vatRate": 0,
    }
  ]
}
```

```

    "vat": 0,
    "exciseRate": 0,
    "excise": 0,
    "quantityInRegisterUnit": 0,
    "costCentre": "string",
    "oA_CostCentreID": 0
  }
]
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"costCentreName": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"currency": See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified. If omitted, gets first register currency from parameters.

"documentComment":

"documentDate": Sets operation date. If not specified - sets current date.

"documentNumber":

"isRateFixed":

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreName] property specified. If omitted, gets cost centre from parameters.

"oA_CurrenciesID": See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currency] property specified. If omitted, gets first register currency from parameters.

"oA_SalePriceTypesID": See [SalePriceTypesList] method (use [oA_SalePriceTypesID] value). Can be omitted if [SalePriceType] property specified.

"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified.

"rate": Sets the currency rate. If not specified, gets the rate by [DocumentDate]. See [CurrencyRatesList]

"receiverAddress":

"receiverBankAccount":

"receiverBankCode":

"receiverBankID": See [BanksList] method (use [OA_BanksID] value).

"receiverCode":

"receiverID": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID]).

"salePriceType": See [SalePriceTypesList] method (use [description] value). Can be omitted if [OA_SalePriceTypesID] property specified.

"status": See [StatusesList] method (use [Name] value). Can be omitted if [OA_StatusID] property specified.

"supplierAddress": See [GetDebitorCreditorByID] method (use [Address] value from [DebitorCreditorAddresses]).

"supplierBankAccount": See [Owner] method (use [Account] value from [OwnerBankAccounts]).

"supplierBankCode": See [Owner] method (use [BankCode] value from [OwnerBankAccounts]).

"supplierBankID": See [Owner] method (use [OA_OwnerBankAccountsID] value from [OwnerBankAccounts]).

"supplierBankName":

"supplierCode": See [Owner] method (use [Code] value from [OwnerBankAccounts]).

"proformaInvoiceItems": ProformaInvoiceItemAdd

"code": See [ItemsList] method (use [Code] value). Can be omitted if [OA_ItemsID] property specified.

"oA_ItemsID": See [ItemsList] method (use [OA_ItemsID] value). Can be omitted if [Code] property specified.

"quantity":

"unit": See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified. If both are omitted, gets the Item register minimum unit.

"oA_StockUnitsID": See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified. If both are omitted, gets the OA_StockUnitsID for Item register minimum unit.

"price": If omitted, calculated as [Amount]/[Quantity]

"amount": If omitted, calculated as [Quantity]*[Price]

"vatRate": If not specified, depends on [vatType] values of [Item], [Receiver] and [owner] (See [GetItemByID], [GetDebitorCreditorByID] and [Owner] methods)

"vat": If omitted, calculated as [Amount]*[VATRate]/100

"exciseRate": If not specified, gets from [exciseRate] value of [Item] object (See [GetItemByID] method)

"excise": If omitted, calculated as [Quantity]*[ExciseRate]

"quantityInRegisterUnit": Required if [hasDoubleRegisterUnit] is true for the [Item] object

"costCentre": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters.

Examples

1. Add proforma invoice.

Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "receiverBankID": 1,
  "receiverCode": "1",
  "receiverID": 2,
  "proformaInvoiceItems": [
    {
      "code": "1",
      "oA_ItemsID": 1,
      "oA_StockUnitsID": 1,
      "price": 5
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 2005,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51111,
    "description": "Item does not exist!",
    "comment": "[11] Record is not in table or user have no permissions to read this record.\n Entry number [1]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "ProformaInvoices",
    "entryPoint": "SetItem",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/ProformaInvoice

Updates Proforma Invoice. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "costCentreName": "string",
  "currency": "string",
  "documentComment": "string",
  "documentDate": "2023-05-04T10:58:18.652Z",
  "documentNumber": "string",
  "isRateFixed": true,
  "oA_CostCentreID": 0,
  "oA_CurrenciesID": 0,
  "oA_SalePriceTypesID": 0,
  "oA_StatusID": 0,
  "rate": 0,
  "receiverAddress": "string",
  "receiverBankAccount": "string",
  "receiverBankCode": "string",
  "receiverBankID": 0,
  "receiverCode": "string",
}
```

```

"receiverID": 0,
"salePriceType": "string",
"status": "string",
"supplierAddress": "string",
"supplierBankAccount": "string",
"supplierBankCode": "string",
"supplierBankID": 0,
"supplierBankName": "string",
"supplierCode": "string",
"clearItems": true,
"proformaInvoiceItems": [
  {
    "oA_ManualEntriesID": 0,
    "deleteRow": true,
    "code": "string",
    "oA_ItemsID": 0,
    "quantity": 0,
    "unit": "string",
    "oA_StockUnitsID": 0,
    "price": 0,
    "amount": 0,
    "vatRate": 0,
    "vat": 0,
    "exciseRate": 0,
    "excise": 0,
    "quantityInRegisterUnit": 0,
    "costCentre": "string",
    "oA_CostCentreID": 0
  }
]
}

```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

"costCentreName": See [CostCentresList] method (use [Name] value).

"currency": See [CurrenciesList] method (use [Currency] value).

"documentComment":

"documentDate": Sets operation date.

"documentNumber":

"isRateFixed":

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value).

"oA_CurrenciesID": See [CurrenciesList] method (use [OA_CurrenciesID] value).

"oA_SalePriceTypesID": See [SalePriceTypesList] method (use [oA_SalePriceTypesID] value).

"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value).

"rate": Sets the currency rate.

"receiverAddress":

"receiverBankAccount":

"receiverBankCode":

"receiverBankID": See [BanksList] method (use [OA_BanksID] value).

"receiverCode":

"receiverID": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID]).

"salePriceType": See [SalePriceTypesList] method (use [description] value).

"status": See [StatusesList] method (use [Name] value).

"supplierAddress": See [GetDebitorCreditorByID] method (use [Address] value from [DebitorCreditorAddresses]).

"supplierBankAccount": See [Owner] method (use [Account] value from [OwnerBankAccounts]).

"supplierBankCode": See [Owner] method (use [BankCode] value from [OwnerBankAccounts]).

"supplierBankID": See [Owner] method (use [OA_OwnerBankAccountsID] value from [OwnerBankAccounts]).

"supplierBankName":

"supplierCode": "string", See [Owner] method (use [Code] value from [OwnerBankAccounts]).

"clearItems": Clears all items for the operation.

"**proformaInvoiceItems**": Set OA_ManagerialEntriesID=0 to add a new Item specification. Specify the OA_ManagerialEntriesID value to update information about Item fields. To delete existing row, specify the value OA_ManagerialEntriesID and set "DeleteRow"=true.

"**oA_ManagerialEntriesID**": Specify an existing OA_ManagerialEntriesID to update or remove Item information, or 0 to add a new Item.

"**deleteRow**": If true, deletes Item information for specified OA_ManagerialEntriesID value.

"**code**": See [ItemsList] method (use [Code] value).

"**oA_ItemsID**": See [ItemsList] method (use [OA_ItemsID] value).

"**quantity**":

"**unit**": See [StockUnitsList] method (use [OA_StockUnitsID] value).

"**oA_StockUnitsID**": See [StockUnitsList] method (use [OA_StockUnitsID] value).

"**price**":

"**amount**":

"**vatRate**":

"**vat**":

"**exciseRate**":

"**excise**":

"**quantityInRegisterUnit**":

"**costCentre**": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters.

Examples

1. Update Proforma Invoice of "receiverID" - 2

Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "oA_TransactionsID": 2005,
  "receiverBankID": 1,
  "receiverCode": "1",
  "receiverID": 2,
  "proformaInvoiceItems": [
    {
      "code": "2",
      "oA_ItemsID": 1,
      "oA_StockUnitsID": 1,
      "price": 5
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 2005,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51111,
    "description": "Item does not exist!",
    "comment": "[222] Record is not in table or user have no permissions to read this record.\nOA_TransactionsID [2005] Entry number [4]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "ProformaInvoices",
    "entryPoint": "SetItem",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/ProformaInvoice
Returns OK or Error description Json.

Schema

```
{  
  "token": "string",  
  "oA_TransactionsID": 0  
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"**token**": Set [token] string from [LogIn] response value
"**oA_TransactionsID**":

Examples

1. Delete Proforma Invoice

Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "oA_TransactionsID": 2005  
}
```

Server response Code(200):

```
{  
  "id": 2005,  
  "idFieldName": "OA_TransactionsID"  
}
```

Server response Code(400):

```
[  
  {  
    "errorNumber": 51051,  
    "description": "Record does not exist or read prohibited for user!",  
    "comment": "OA_TransactionsID = 205",  
    "command": "",  
    "errorType": "ErrorType_Error",  
    "source": "OA_BusinessObject.dll",  
    "class": "ProformaInvoices",  
    "entryPoint": "GetProperties",  
    "resourceDLL_ForTranslation": "ORIS_Accounting",  
    "errorPointer": 0  
  }  
]
```

GetProformaInvoiceByID

POST/api/GetProformaInvoiceByID

Gets Proforma Invoice and its Items by OA_TransactionsID value

Schema

```
{  
  "token": "string",  
  "oA_TransactionsID": 0  
}
```

Field description

[Token] and [OA_TransactionsID] is necessary

"**token**": Set [token] string from [LogIn] response value

"OA_TransactionsID": See [TransactionsList] method (use [OA_TransactionsID] value)

Examples

1. Get Proforma Invoice and its Items by "OA_TransactionsID" -2005

Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "oA_TransactionsID": 2005
}
```

Server response Code(200):

```
{
  "oA_TransactionsID": 2005,
  "amount": 100,
  "costCentreName": "General cost center",
  "currency": "GEL",
  "debitorCreditorVATType": 0,
  "documentComment": "",
  "documentDate": "2023-05-10T00:00:00",
  "documentNumber": 1,
  "excise": 0,
  "isRateFixed": false,
  "oA_CostCentreID": 1,
  "oA_CurrenciesID": 1,
  "oA_SalePriceTypesID": 0,
  "oA_StatusID": 0,
  "oA_TransactionTypesID": 5,
  "ownerVATType": 0,
  "rate": 1,
  "receiverAddress": "",
  "receiverBankAccount": "",
  "receiverBankCode": "",
  "receiverBankID": 0,
  "receiverBankName": "",
  "receiverCode": "1",
  "receiverID": 1,
  "receiverName": "Organization",
  "salePriceChangingLastReason": 0,
  "salePriceType": "",
  "status": "",
  "supplier": "",
  "supplierAddress": "",
  "supplierBankAccount": "",
  "supplierBankCode": "",
  "supplierBankID": 0,
  "supplierBankName": "",
  "supplierCode": "2",
  "supplierID": 0,
  "transactionDocumentNumber": "1",
  "vat": 15.25,
  "proformaInvoiceItems": [
    {
      "oA_ManagementEntriesID": 2004,
      "oA_TransactionsID": 2005,
      "code": "2003",
      "item": "item",
      "oA_ItemsID": 2003,
      "quantity": 1,
      "unit": "Piece",
      "oA_StockUnitsID": 1,
      "price": 100,
      "amount": 100,
      "vatRate": 18,
      "vat": 15.25,
      "exciseRate": 0,
      "excise": 0,
    }
  ]
}
```

```

"quantityInRegisterUnit": 0,
"itemType": 0,
"itemVATType": 0,
"itemSalePrice": 0,
"itemSalePriceCurrency": "",
"itemSalePriceOA_CurrenciesID": 0,
"itemSalePriceUnit": "Piece",
"itemSalePriceOA_StockUnitsID": 1,
"itemSalePriceInSaleCurrencyAndUnit": 0,
"hasADoubleRegistrationUnit": false,
"oA_StockUnitsIDRegisterUnit": 1,
"registerUnit": "Piece",
"costCentre": "",
"oA_CostCentreID": 0,
"sequenceNumber": 1
}
]
}

```

Server response Code(400):

```

[
{
"errorNumber": 51051,
"description": "Record does not exist or read prohibited for user!",
"comment": "OA_TransactionsID = 206",
"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "ProformaInvoices",
"entryPoint": "GetProperties",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

ProformaInvoicesList

POST/api/ProformaInvoicesList

Gets Proforma Invoices. Returns [Info] and [Result] tables

Schema

```

{
"token": "string",
"topRecordsCount": 0,
"filter": "string",
"sort": "string",
"pageNumber": 0
}

```

Field description

"**token**": Set [token] string from [LogIn] response value

"**topRecordsCount**": The number of records to accept. If you specify 0 you will get all records. Default=0

"**filter**": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"**sort**": Use SQL ORDER BY string with field names.

"**pageNumber**": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get list from Currency – GEL

Request body:

```

{
"token": "... paste here [LogIn] response value ...",

```

```
"topRecordsCount": 1,
"filter": "UPPER([ViewOA_ProformaInvoices].[Currency]) LIKE N'GEL%'",
"sort": "[Amount],[Currency]",
"pageNumber": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 1,
      "PageRowCount": 1,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "Document date": "2023-05-29T00:00:00",
      "ReceiverID": null,
      "Receiver code": "",
      "ReceiverBankID": 1,
      "Receiver bank account": "",
      "Receiver bank code": "",
      "Receiver bank name": "",
      "Amount": 0.00000000,
      "Supplier": "",
      "SupplierBankID": null,
      "Supplier bank account": "",
      "Supplier bank code": "",
      "Supplier bank name": "",
      "Receiver address": "",
      "Supplier address": "",
      "OA_TransactionsID": 5011,
      "OA_TransactionTypesID": 5,
      "Comment": "",
      "Receiver name": "",
      "Document number": "2",
      "Has entries": false,
      "VAT": 0.00000000,
      "Excise": 0.00000000,
      "Rate": 1.00000000,
      "Is rate fixed": false,
      "OA_CurrenciesID": 1,
      "Currency": "GEL",
      "OA_CostCentreID": 1,
      "Cost centre": "General cost center",
      "Record related binary documents": null,
      "Owner VAT type": 0,
      "DebitorCreditor VAT type": 0,
      "Supplier code": "2",
      "OA_SalePriceTypesID": null,
      "Price type": null,
      "OA_StatusID": null,
      "Status": null,
      "Numeric document number": 2,
      "NUM_ROW": 1
    }
  ]
}
```

AcceptOperation

POST/api/AcceptOperation

Adds new Accept Operation. Returns ID value or Error description Json

Schema

```
{
  "token": "string",
  "costCentreName": "string",
  "currency": "string",
  "documentComment": "string",
  "documentDate": "2023-06-08T12:30:54.370Z",
  "documentNumber": "string",
  "dueDate": "2023-06-08T12:30:54.370Z",
  "incomesAccountForFreeAccept": "string",
  "isRateFixed": true,
  "oA_CostCentreID": 0,
  "oA_CurrenciesID": 0,
  "oA_IncomesAccountForFreeAcceptID": 0,
  "oA_ParentAccountForSellerID": 0,
  "oA_PayedExciseAccountID": 0,
  "oA_PayedVATAccountID": 0,
  "oA_StatusID": 0,
  "parentAccountForSeller": "string",
  "payedExciseAccount": "string",
  "payedVATAccount": "string",
  "rate": 0,
  "receiverAddress": "string",
  "receiverAddressesID": 0,
  "rS_TaxDocument_ID": 0,
  "rS_TaxDocument_Number": "string",
  "rS_VAT_ID": 0,
  "rS_VAT_Number": "string",
  "rS_Waybill_ID": 0,
  "rS_Waybill_Number": "string",
  "sellerCode": "string",
  "sellerID": 0,
  "status": "string",
  "supplierAddress": "string",
  "supplierAddressID": 0,
  "supplierCode": "string",
  "supplierID": 0,
  "createAccountingEntries": true,
  "acceptOperationItems": [
    {
      "code": "string",
      "oA_ItemsID": 0,
      "quantity": 0,
      "unit": "string",
      "oA_StockUnitsID": 0,
      "price": 0,
      "amount": 0,
      "vatRate": 0,
      "vat": 0,
      "exciseRate": 0,
      "excise": 0,
      "quantityInRegisterUnit": 0,
      "supplierItemCode": "string",
      "supplierItemName": "string",
      "supplierItemUnit": "string",
      "isInvoiceExpense": true,
      "oA_AccountsID": 0,
      "registerParentAccount": "string",
      "supplierItemVATType": 0,
      "freeAccept": true,
      "costCentre": "string",
      "oA_CostCentreID": 0
    }
  ],
  "acceptOperationInvoiceExpensesSharingItems": [
    {
```

```

    "oA_EntriesID": 0,
    "expenseAmount": 0,
    "managerialEntrySequenceNumber": 0
  }
]
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

“token”: Set [token] string from [LogIn] response value

“CostCentreName”: See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If ommitte, gets cost centre from parameters.

“Currency”: See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified. If ommitte, gets firts register currency from parameters.

“documentComment”

“documentDate”: Sets operation date. If not specified - sets current date.

“DocumentNumber”: Sets the transaction document number. If not specified, sets a unique number in the current year.

“dueDate”

“incomesAccountForFreeAccept”: See [AccountsList] method (use [Account] value). Can be omitted if [OA_IncomesAccountForFreeAcceptID] property specified. If both are omitted, gets the account from the parameters.

“isRateFixed”: Sets a fixed rate flag. If true, then equivalent values in accounting entries of this operation will not be recalculated when the exchange rate changes.

“OA_CostCentreID”: See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreName] property specified. If ommitte, gets cost centre from parameters.

“OA_CurrenciesID”: See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currency] property specified. If ommitte, gets firts register currency from parameters.

“OA_IncomesAccountForFreeAcceptID”: ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [incomesAccountForFreeAccept] property specified. If both are omitted, gets the account from the parameters.

“OA_ParentAccountForSellerID”: ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [parentAccountForSeller] property specified. If both are omitted, gets the account from the parameters.

“OA_PayedExciseAccountID”: ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [payedExciseAccount] property specified. If both are omitted, gets the account from the parameters.

“OA_PayedVATAccountID”: ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [payedVATAccount] property specified. If both are omitted, gets the account from the parameters.

“OA_StatusID”: See [StatusesList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified.

“parentAccountForSeller”: See [AccountsList] method (use [Account] value). Can be omitted if [OA_ParentAccountForSellerID] property specified. If both are omitted, gets the account from the parameters.

PayedExciseAccount: See [AccountsList] method (use [Account] value). Can be omitted if [OA_PayedExciseAccountID] property specified. If both are omitted, gets the account from the parameters.

“payedVATAccount”: See [AccountsList] method (use [Account] value). Can be omitted if [OA_PayedVATAccountID] property specified. If both are omitted, gets the account from the parameters.

“Rate”: Sets the currency rate. If not specified, gets the rate by [DocumentDate]. See [CurrencyRatesList]

“ReceiverAddress”: See [Owner] method (use [Address] value). Can be omitted if [ReceiverAddressesID] property specified.

“receiverAddressesID”: See [Owner] method (use [OA_OwnerAddressesID] value). Can be omitted if [ReceiverAddress] property specified.

“RS_TaxDocument_ID”: Sets Tax Document ID value recieved from revenue service

“RS_TaxDocument_Number”: Sets Tax Document number recieved from revenue service

“RS_VAT_ID”: Sets VAT ID recieved from revenue service

“RS_VAT_Number”: Sets VAT number recieved from revenue service

“RS_Waybill_ID”: Sets waybill ID recieved from revenue service

“RS_Waybill_Number”: Sets waybill number recieved from revenue service

“sellerCode”: See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [SellerID] property specified.

“SellerID”: See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [SellerCode] property specified.

“Status”: See [StatusesList] method (use [Name] value). Can be omitted if [OA_StatusID] property specified.

“SupplierAddress”: See [GetDebitorCreditorByID] method (use [Address] value from [DebitorCreditorAddresses]).

“supplierAddressID”: See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorAddressesID] value from [DebitorCreditorAddresses]).

“SupplierCode”: See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [SupplierID] property specified.

“supplierID”: See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [SupplierCode] property specified.

“**CreateAccountingEntries**”: Creates accounting entries using managerial information. Default=false

“**acceptOperationItems**”: AcceptOperationItemAdd

“**Code**”: See [ItemsList] method (use [Code] value). Can be omitted if [OA_ItemsID] property specified.

“**OA_ItemsID**”: See [ItemsList] method (use [OA_ItemsID] value). Can be omitted if [Code] property specified.

“**Quantity**”

“**Unit**”: See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified. If both are omitted, gets the Item register minimum unit.

“**OA_StockUnitsID**”: See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified. If both are omitted, gets the OA_StockUnitsID for Item register minimum unit.

“**price**”: If omitted, calculated as [Amount]/[Quantity]

“**Amount**”: If omitted, calculated as [Quantity]*[Price]

“**vatRate**”: If not specified, depends on [vatType] values of [Item], [Buyer] and [owner] (See [GetItemByID], [GetDebtorCreditorByID] and [Owner] methods)

“**Vat**”: If omitted, calculated as [Amount]*[VATRate]/100

“**exciseRate**”: If not specified, gets from [exciseRate] value of [Item] object (See [GetItemByID] method)

“**excise**”: If omitted, calculated as [Quantity]*[ExciseRate]

“**quantityInRegisterUnit**”: Required if [hasDoubleRegisterUnit] is true for the [Item] object

“**supplierItemCode**”

“**supplierItemName**”

“**supplierItemUnit**”

“**isInvoiceExpense**”: Set to true if [Item] is a service and you want to allocate the amount value to bill items as expenses. Fill [AcceptOperationInvoiceExpensesSharingItems] for which records the expense amount should be shared.

“**OA_AccountsID**”: If not specified, gets from [oA_AccountsID] value of [Item] object (See [GetItemByID] method)

“**registerParentAccount**”: If not specified, gets account by [OA_AccountsID] of [Item] object (See [GetItemByID] method)

“**supplierItemVATType**”: Standard = 0, ZeroRate = 1, Untaxed = 2

“**freeAccept**”

“**costCentre**”: See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

“**OA_CostCentreID**”: See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters.

“**acceptOperationInvoiceExpensesSharingItems**”: AcceptOperationInvoiceExpensesSharingItemAdd

“**OA_EntriesID**”: Specify the value of the accounting entry ID that belongs to the accept or other invoice, and the account type in entry is the [item] (commodity or fixed asset, not a service). For get an invoice transaction see [GetTransactionByID] method and use [OA_EntriesID] value from [TransactionEntries].

“**expenseAmount**”: Expense amount

“**managerialEntrySequenceNumber**”: Specify the SequenceNumber of the managerial entry that belongs to the current accept operation, and the item is commodity or fixed asset, not a service.

Examples

1. Adds new Accept Operation

Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "sellerID": 1,
  "createAccountingEntries": true,
  "acceptOperationItems": [
    {
      "oA_ItemsID": 1,
      "quantity": 5,
      "price": 1,
      "registerParentAccount": "1610"
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 190,
  "idFieldName": "Oa_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51038,
    "description": "Account is closed!",
    "comment": "Account [1610 1] 1 \nUncheck close check box for this account and resume or cancel transaction\nOA_AccountisID=262\nOA_TransactionsID [2010]\nDocument number [30]\nDate[10.05.2023 00:00:00]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "PostUpdateActionStagel",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

2. Adds new Accept Operation with sharing expense cost.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "sellerID": 1,
  "createAccountingEntries": true,
  "acceptOperationItems": [
    {
      "oA_ItemsID": 1,
      "quantity": 5,
      "price": 100,
      "registerParentAccount": "1610"
    },
    {
      "oA_ItemsID": 4005,
      "unit": "Service",
      "amount": 100,
      "isInvoiceExpense": true
    }
  ],
  "acceptOperationInvoiceExpensesSharingItems": [
    {
      "oA_EntriesID": 4211,
      "expenseAmount": 84.75,
      "managerialEntrySequenceNumber": 1
    }
  ]
}
```

Put/api/AcceptOperation

Updates Existing Accept Operation. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "costCentreName": "string",
  "currency": "string",
  "documentComment": "string",
  "documentDate": "2023-04-28T10:53:22.337Z",
  "documentNumber": "string",
  "dueDate": "2023-04-28T10:53:22.337Z",
  "incomesAccountForFreeAccept": "string",
  "isRateFixed": true,
  "oA_CostCentreID": 0,
  "oA_CurrenciesID": 0,
  "oA_IncomesAccountForFreeAcceptID": 0,
  "oA_ParentAccountForSellerID": 0,
}
```

```

"oA_PayedExciseAccountID": 0,
"oA_PayedVATAccountID": 0,
"oA_StatusID": 0,
"parentAccountForSeller": "string",
"payedExciseAccount": "string",
"payedVATAccount": "string",
"rate": 0,
"receiverAddress": "string",
"receiverAddressesID": 0,
"rS_TaxDocument_ID": 0,
"rS_TaxDocument_Number": "string",
"rS_VAT_ID": 0,
"rS_VAT_Number": "string",
"rS_Waybill_ID": 0,
"rS_Waybill_Number": "string",
"sellerCode": "string",
"sellerID": 0,
"status": "string",
"supplierAddress": "string",
"supplierAddressID": 0,
"supplierCode": "string",
"supplierID": 0,
"createAccountingEntries": true,
"clearItems": true,
"clearInvExpenses": true,
"acceptOperationItems": [
  {
    "oA_ManualEntriesID": 0,
    "deleteRow": true,
    "code": "string",
    "oA_ItemsID": 0,
    "quantity": 0,
    "unit": "string",
    "oA_StockUnitsID": 0,
    "price": 0,
    "amount": 0,
    "vatRate": 0,
    "vat": 0,
    "exciseRate": 0,
    "excise": 0,
    "quantityInRegisterUnit": 0,
    "supplierItemCode": "string",
    "supplierItemName": "string",
    "supplierItemUnit": "string",
    "isInvoiceExpense": true,
    "oA_AccountsID": 0,
    "registerParentAccount": "string",
    "supplierItemVATType": 0,
    "freeAccept": true,
    "costCentre": "string",
    "oA_CostCentreID": 0
  }
],
"acceptOperationInvoiceExpensesSharingItems": [
  {
    "oA_InvoiceExpensesItemsID": 0,
    "deleteRow": true,
    "oA_EntriesID": 0,
    "expenseAmount": 0,
    "managerialEntrySequenceNumber": 0
  }
]
}

```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change.

"**token**": Set [token] string from [LogIn] response value
"**OA_TransactionsID**": ,
"**costCentreName**": See [CostCentresList] method (use [Name] value).,
"**currency**": See [CurrenciesList] method (use [Currency] value).
"**documentComment**"
"**documentDate**": Sets operation date.
"**documentNumber**"
"**dueDate**"
"**incomesAccountForFreeAccept**": See [AccountsList] method (use [Account] value).
"**isRateFixed**": Sets a fixed rate flag. If true, then equivalent values in accounting entries of this operation will not be recalculated when the exchange rate changes.
"**OA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value).
"**OA_CurrenciesID**": See [CurrenciesList] method (use [OA_CurrenciesID] value).
"**OA_IncomesAccountForFreeAcceptID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).
"**OA_ParentAccountForSellerID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).
"**OA_PayedExciseAccountID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).
"**OA_PayedVATAccountID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).
"**OA_StatusID**": See [StatusesList] method (use [OA_StatusID] value).
"**parentAccountForSeller**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_ParentAccountForSellerID] property specified. If both are omitted, gets the account from the parameters.
"**payedExciseAccount**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_PayedExciseAccountID] property specified. If both are omitted, gets the account from the parameters.
"**payedVATAccount**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_PayedVATAccountID] property specified. If both are omitted, gets the account from the parameters.
"**rate**": Sets the currency rate.
"**receiverAddress**": See [Owner] method (use [Address] value or [address] value from [OwnerAddresses]).
"**receiverAddressesID**": See [Owner] method (use [OA_OwnerAddressesID] value from [OwnerAddresses]).
"**rs_TaxDocument_ID**": Sets Tax Document ID value recieved from revenue service
"**rs_TaxDocument_Number**": Sets Tax Document number recieved from revenue service
"**rs_VAT_ID**": Sets VAT ID recieved from revenue service
"**rs_VAT_Number**": Sets VAT number recieved from revenue service
"**rs_Waybill_ID**": Sets waybill ID recieved from revenue service
"**rs_Waybill_Number**": Sets waybill number recieved from revenue service
"**sellerCode**": See [DebitorCreditorsList] method (use [Code] value).
"**sellerID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value)
"**status**": See [StatusesList] method (use [Name] value)
"**supplierAddress**": See [GetDebitorCreditorByID] method (use [Address] value from [DebitorCreditorAddresses]).
"**supplierAddressID**": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorAddressesID] value from [DebitorCreditorAddresses]).
"**supplierCode**": See [DebitorCreditorsList] method (use [Code] value).
"**supplierID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value)
"**createAccountingEntries**": Creates accounting entries using managerial information. Default=false
"**clearItems**": Clears all items for the operation.
"**clearInvExpenses**": Clears all Invoice Expenses for the operation.

"**acceptOperationItems**": Set OA_ManagerialEntriesID=0 to add a new Item specification. Specify the OA_ManagerialEntriesID value to update information about Item fields. To delete existing row, specify the value OA_ManagerialEntriesID and set "DeleteRow"=true
"**OA_ManagerialEntriesID**": Specify an existing OA_ManagerialEntriesID to update or remove Item information, or 0 to add a new Item.
"**deleteRow**": If true, deletes Item information for specified OA_ManagerialEntriesID value.
"**code**": See [ItemsList] method (use [Code] value).
"**OA_ItemsID**": See [ItemsList] method (use [OA_ItemsID] value).
"**quantity**":
"**unit**": See [StockUnitsList] method (use [Unit] value).
"**OA_StockUnitsID**": See [StockUnitsList] method (use [OA_StockUnitsID] value).
"**price**":
"**amount**":
"**vatRate**":
"**vat**":
"**exciseRate**":
"**excise**":
"**quantityInRegisterUnit**":
"**supplierItemCode**":

"supplierItemName":

"supplierItemUnit":

"isInvoiceExpense": Set to true if [Item] is a service and you want to allocate the amount value to bill items as expenses. Fill [AcceptOperationInvoiceExpensesSharingItems] for which records the expense amount should be shared.

"OA_AccountsID":

"registerParentAccount":

"supplierItemVATType": Standard = 0, ZeroRate = 1, Untaxed = 2

"freeAccept":

"costCentre": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If ommite, gets cost centre from parameters.

"OA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If ommite, gets cost centre from parameters.

"acceptOperationInvoiceExpensesSharingItems": Set OA_InvoiceExpensesItemsID=0 to add a new Invoice Expenses specification. Specify the OA_InvoiceExpensesItemsID value to update information about Invoice Expenses fields. To delete existing row, specify the value OA_InvoiceExpensesItemsID and set "DeleteRow"=true.

"OA_InvoiceExpensesItemsID": Specify an existing OA_InvoiceExpensesItemsID to update or remove Invoice Expenses information, or 0 to add a new Invoice Expense.

"deleteRow": If true, deletes Invoice Expenses information for specified OA_InvoiceExpensesItemsID value.

"OA_EntriesID": Specify the value of the accounting entry ID that belongs to the accept or other invoice, and the account type in entry is the [item] (commodity or fixed asset, not a service). For get an invoice transaction see [GetTransactionByID] method and use [OA_EntriesID] value from [TransactionEntry].

"expenseAmount": Expense amount

"managerialEntrySequenceNumber": Specify the SequenceNumber of the managerial entry that belongs to the current accept operation, and the item is commodity or fixed asset, not a service.

Examples

1. Update quantity with "OA_TransactionID" - 40

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 40,
  "sellerID": 1,
  "createAccountingEntries": true,
  "acceptOperationItems": [
    {
      "oA_ItemsID": 1,
      "quantity": 5,
      "price": 3,
      "registerParentAccount": "1610"
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 40,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51067,
    "description": "Can not change/delete data when transaction has been documental confirmed!",
    "comment": "For change/delete data first remove documental confirm.\noA_TransactionsID= 40",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "CheckDocumentalConfirm",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
],
```

]

2. **Update** AcceptOperationInvoiceExpensesSharingItem with registerParentAccount "1620"
Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "oA_TransactionsID":5130,
  "sellerID": 1,
  "createAccountingEntries": true,
  "acceptOperationItems": [
    {
      "oA_ItemsID": 1,
      "quantity": 5,
      "price": 100,
      "registerParentAccount":"1620"
    },
    {
      "oA_ItemsID": 4005,
      "unit": "Service",
      "amount": 100,
      "isInvoiceExpense": true
    }
  ],
  "acceptOperationInvoiceExpensesSharingItems": [
    {
      "oA_EntriesID": 4211,
      "expenseAmount": 84.75,
      "managerialEntrySequenceNumber": 1
    }
  ]
}
```

DELETE/api/AcceptOperation

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [Login] response value
"oA_TransactionsID":

Examples

1. Delete account with "OA_TransactionsID" - 40.
Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "oA_TransactionsID": 40
}
```

Server response Code(200):

```
{
  "id": 40,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

[

```
{
  "errorNumber": 51067,
  "description": "Can not change/delete data when transaction has been documental confirmed!",
  "comment": "For change/delete data first remove documental confirm.\nOA_TransactionsID= 40",
  "command": "",
  "errorType": "ErrorType_Error",
  "source": "OA_BusinessObject.dll",
  "class": "Transaction",
  "entryPoint": "CheckDocumentalConfirm",
  "resourceDLL_ForTranslation": "ORIS_Accounting",
  "errorPointer": 0
}
```

2. Delete Accept Operation using "oA_TransactionsID" - 3350
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 3350
}
```

GetAcceptOperationByID

POST/api/GetAcceptOperationByID

Gets Accept Operation and its Items and Invoice Expenses by OA_TransactionsID value.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"OA_TransactionsID": See [TransactionsList] method (use [OA_TransactionsID] value)

Examples

1. Get Accept Operation and its Items by "OA_TransactionsID" - 2007
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 2007
}
```

Server response Code(200):

```
{
  "oA_TransactionsID": 2007,
  "amount": 5,
  "costCentreName": "General cost center",
  "currency": "GEL",
  "debitorCreditorVATType": 0,
  "documentalConfirm": true,
  "documentComment": "",
  "documentDate": "2023-05-19T00:00:00",
  "documentNumber": 27,
  "dueDate": "2023-06-08T00:00:00",
  "excise": 0,
  "incomesAccountForFreeAccept": "8190",
  "isDueDateNULL": true,
}
```

```
"isRateFixed": false,
"oA_CostCentreID": 1,
"oA_CurrenciesID": 1,
"oA_IncomesAccountForFreeAcceptID": 205,
"oA_ParentAccountForSellerID": 119,
"oA_PayedExciseAccountID": 134,
"oA_PayedVATAccountID": 133,
"oA_StatusID": 0,
"oA_TransactionTypesID": 16,
"ownerVATType": 0,
"rate": 0,
"receiver": "",
"receiverAddress": "",
"receiverAddressesID": 0,
"redistributableInvoiceExpense": 0,
"reflectInformationInEntries": false,
"rS_TaxDocument_ID": 0,
"rS_TaxDocument_Number": "",
"rS_VAT_ID": 0,
"rS_VAT_Number": "",
"rS_Waybill_ID": 0,
"rS_Waybill_Number": "",
"salePriceChangingLastReason": 0,
"sellerCode": "1",
"sellerID": 1,
"sellerName": "Organization",
"status": "",
"supplierAddress": "",
"supplierAddressID": 0,
"supplierCode": "1",
"supplierID": 1,
"supplierName": "Organization",
"vat": 0.76,
"acceptOperationItems": [
  {
    "oA_ManualEntriesID": 2005,
    "oA_TransactionsID": 2007,
    "code": "1",
    "item": "item2",
    "oA_ItemsID": 1,
    "quantity": 5,
    "unit": "Piece",
    "oA_StockUnitsID": 1,
    "price": 1,
    "amount": 5,
    "vatRate": 18,
    "vat": 0.76,
    "exciseRate": 0,
    "excise": 0,
    "quantityInRegisterUnit": 0,
    "oA_EntriesID": 2017,
    "supplierItemCode": "",
    "supplierItemName": "",
    "supplierItemUnit": "",
    "isInvoiceExpense": false,
    "oA_AccountsID": 69,
    "registerParentAccount": "1610",
    "itemType": 0,
    "itemVATType": 0,
    "supplierItemVATType": 0,
    "freeAccept": false,
    "hasADoubleRegistrationUnit": false,
    "oA_StockUnitsIDRegisterUnit": 1,
    "registerUnit": "Piece",
    "costCentre": "",
    "oA_CostCentreID": 0,
    "sequenceNumber": 1
  }
]
```

```
    }
  ],
  "acceptOperationInvoiceExpensesSharingItems": []
}
```

Server response Code(400):

```
[
{
  "errorNumber": 51051,
  "description": "Record does not exist or read prohibited for user!",
  "comment": "OA_TransactionsID = 2007",
  "command": "",
  "errorType": "ErrorType_Error",
  "source": "OA_BusinessObject.dll",
  "class": "Transaction",
  "entryPoint": "GetProperties",
  "resourceDLL_ForTranslation": "ORIS_Accounting",
  "errorPointer": 0
}
]
```

AcceptOperationsList

POST/api /AcceptOperationsList

Gets Accept Operations. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": "string",

"sort": Use SQL ORDER BY string with field names.

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get first 2 accounts from database

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2,
  "filter": "UPPER([ViewOA_AcceptOperations].[Seller]) LIKE N'%ORG%'",
  "sort": "[Document date],[Document number]",
  "pageNumber": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 1,
      "PageRowCount": 2,
      "PageNumber": 1
    }
  ]
}
```

```

],
"Result": [
{
  "OA_TransactionsID": 40,
  "Document date": "2023-05-10T00:00:00",
  "Document number": "27",
  "SellerID": 1,
  "Seller code": "1",
  "Seller": "Organization",
  "SupplierID": 1,
  "Supplier code": "1",
  "Supplier": "Organization",
  "Supplier address": "",
  "Receiver": "",
  "Receiver address": "",
  "OA_CostCentreID": 1,
  "Cost centre": "General cost center",
  "Comment": "",
  "OA_TransactionTypesID": 16,
  "OA_ParentAccountForSellerID": 119,
  "Parent account for seller": "3110",
  "OA_PayedVATAccountID": 133,
  "Paid VAT account": "3340",
  "OA_CurrenciesID": 1,
  "Currency": "GEL",
  "Rate": 1.00000000,
  "Is rate fixed": false,
  "Redistributable invoice expense": 0.00000000,
  "Delay": true,
  "Documental confirm": true,
  "Amount": 5.00000000,
  "Has entries": true,
  "VAT": 0.76000000,
  "Excise": 0.00000000,
  "OA_UsersID": 1,
  "OA_DepartmentsID": 1,
  "Owner VAT type": 0,
  "DebitorCreditor VAT type": 0,
  "OA_IncomeAccountForFreeSupplyID": 205,
  "Incomes account for free accept": "8190",
  "OA_PayedExciseAccountID": 134,
  "Paid excise account": "3350",
  "Numeric document number": 27,
  "NUM_ROW": 1
}
]
}

```

TransferOperation

POST/api/TransferOperation

Adds new Transfer Operation. Returns ID value or Error description Json.

Schema

```

{
  "token": "string",
  "costCentreFrom": "string",
  "costCentreTo": "string",
  "documentComment": "string",
  "documentDate": "2023-05-05T12:16:16.018Z",
  "documentNumber": "string",
  "oA_CostCentreID": 0,
  "oA_SalePriceTypesID": 0,
  "oA_StatusID": 0,

```

```

"rS_TaxDocument_ID": 0,
"rS_TaxDocument_Number": "string",
"rS_Waybill_ID": 0,
"rS_Waybill_Number": "string",
"salePriceType": "string",
"status": "string",
"toOA_CostCentreID": 0,
"createAccountingEntries": true,
"transferOperationItems": [
  {
    "code": "string",
    "oA_ItemsID": 0,
    "quantity": 0,
    "unit": "string",
    "oA_StockUnitsID": 0,
    "price": 0,
    "amount": 0,
    "costCentreFrom": "string",
    "oA_CostCentreID": 0,
    "costCentreTo": "string",
    "toOA_CostCentreID": 0,
    "quantityInRegisterUnit": 0,
    "registerParentAccount": "string",
    "oA_AccountsID": 0,
    "parentAccountForDestinationItem": "string",
    "oA_AccountsIDTo": 0,
    "rsItemCode": "string",
    "rsItemName": "string",
    "rsItemUnit": "string",
    "rsItemVATType": 0
  }
]
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"costCentreFrom": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If ommite, gets cost centre from parameters. "

"costCentreTo": See [CostCentresList] method (use [Name] value). Can be omitted if [ToOA_CostCentreID] property specified. If ommite, gets cost centre from parameters.

"documentComment":

"documentDate": Sets operation date. If not specified - sets current date.

"documentNumber": Sets the transaction document number. If not specified, sets a unique number in the current year.

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreFrom] property specified. If ommite, gets cost centre from parameters.

"oA_SalePriceTypesID": See [SalePriceTypesList] method (use [oA_SalePriceTypesID] value). Can be omitted if [SalePriceType] property specified.

"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified.

"rS_TaxDocument_ID": Sets Tax Document ID value recieved from revenue service

"rS_TaxDocument_Number": Sets Tax Document number recieved from revenue service

"rS_Waybill_ID": Sets waybill ID recieved from revenue service

"rS_Waybill_Number": Sets waybill number recieved from revenue service

"salePriceType": See [SalePriceTypesList] method (use [description] value). Can be omitted if [OA_SalePriceTypesID] property specified.

"status": See [StatusesList] method (use [Name] value). Can be omitted if [OA_StatusID] property specified.

"toOA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreIDTo] value). Can be omitted if [CostCentre] property specified. If ommite, gets cost centre from parameters.

"createAccountingEntries": Creates accounting entries using managerial information. Default=false

"transferOperationItems": TransferOperationItemAdd

"code": See [ItemsList] method (use [Code] value). Can be omitted if [OA_ItemsID] property specified.

"oA_ItemsID": See [ItemsList] method (use [OA_ItemsID] value). Can be omitted if [Code] property specified.

"quantity":

"unit": See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified. If both are omitted, gets the Item register minimum unit.

"oA_StockUnitsID": See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified. If both are omitted, gets the OA_StockUnitsID for Item register minimum unit.

"price": If omitted, calculated as [Amount]/[Quantity]

"amount": If omitted, calculated as [Quantity]*[Price]

"costCentreFrom": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters.

"costCentreTo": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"toOA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters.

"quantityInRegisterUnit": Required if [hasDoubleRegisterUnit] is true for the [Item] object

"registerParentAccount": If not specified, gets account by [OA_AccountsID] of [Item] object (See [GetItemByID] method)

"oA_AccountsID": If not specified, gets from [oA_AccountsID] value of [Item] object (See [GetItemByID] method)

"parentAccountForDestinationItem":

"oA_AccountsIDTo":

"rsItemCode":

"rsItemName":

"rsItemUnit":

"rsItemVATType": Standard = 0, ZeroRate = 1, Untaxed = 2

Examples

1. Add new Transfer Operation.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "costCentreFrom": "General cost center",
  "costCentreTo": "New cost center",
  "oA_CostCentreID": 1,
  "toOA_CostCentreID": 2,
  "createAccountingEntries": true,
  "transferOperationItems": [
    {
      "code": "1",
      "oA_ItemsID": 1,
      "quantity": 5,
      "oA_StockUnitsID": 1,
      "costCentreFrom": "General cost center",
      "oA_CostCentreID": 1,
      "costCentreTo": "New cost center",
      "toOA_CostCentreID": 2,
      "registerParentAccount": "1610",
      "oA_AccountsID": 1,
      "parentAccountForDestinationItem": "1620",
      "oA_AccountsIDTo": 2
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 2012,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51053,
    "description": "OA_CostCentreID does not exist!",
    "comment": "[22] Record is not in table or user have no permissions to read this record.\n Entry number [1]",
    "command": ""
  }
]
```

```

"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "TransferOperation",
"entryPoint": "SetToOA_CostCentreID",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

PUT/api/TransferOperation

Updates Transfer Operation. Returns OK or Error description Json.

Schema

```

{
  "token": "string",
  "oA_TransactionsID": 0,
  "costCentreFrom": "string",
  "costCentreTo": "string",
  "documentComment": "string",
  "documentDate": "2023-05-05T13:08:44.243Z",
  "documentNumber": "string",
  "oA_CostCentreID": 0,
  "oA_SalePriceTypesID": 0,
  "oA_StatusID": 0,
  "rS_TaxDocument_ID": 0,
  "rS_TaxDocument_Number": "string",
  "rS_Waybill_ID": 0,
  "rS_Waybill_Number": "string",
  "salePriceType": "string",
  "status": "string",
  "toOA_CostCentreID": 0,
  "createAccountingEntries": true,
  "clearItems": true,
  "transferOperationItems": [
    {
      "oA_ManagementEntriesID": 0,
      "deleteRow": true,
      "code": "string",
      "oA_ItemsID": 0,
      "quantity": 0,
      "unit": "string",
      "oA_StockUnitsID": 0,
      "price": 0,
      "amount": 0,
      "costCentreFrom": "string",
      "oA_CostCentreID": 0,
      "costCentreTo": "string",
      "toOA_CostCentreID": 0,
      "quantityInRegisterUnit": 0,
      "registerParentAccount": "string",
      "oA_AccountsID": 0,
      "parentAccountForDestinationItem": "string",
      "oA_AccountsIDTo": 0,
      "rsItemCode": "string",
      "rsItemName": "string",
      "rsItemUnit": "string",
      "rsItemVATType": 0
    }
  ]
}

```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change

"token": Set [token] string from [LogIn] response value
 "oA_TransactionsID":
 "costCentreFrom": See [CostCentresList] method (use [Name] value).
 "costCentreTo": See [CostCentresList] method (use [Name] value).
 "documentComment":
 "documentDate": Sets operation date.
 "documentNumber":
 "oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value).
 "oA_SalePriceTypesID": See [SalePriceTypesList] method (use [oA_SalePriceTypesID] value).
 "oA_StatusID": See [StatusesList] method (use [OA_StatusID] value).
 "rS_TaxDocument_ID": Sets Tax Document ID value recieved from revenue service
 "rS_TaxDocument_Number": Sets Tax Document number recieved from revenue service
 "rS_Waybill_ID": Sets waybill ID recieved from revenue service
 "rS_Waybill_Number": Sets waybill number recieved from revenue service
 "salePriceType": See [SalePriceTypesList] method (use [description] value).
 "status": See [StatusesList] method (use [Name] value).
 "toOA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreIDTo] value).
 "createAccountingEntries": Creates accounting entries using managerial information. Default=false
 "clearItems": Clears all items for the operation.

"transferOperationItems": Set OA_ManagerialEntriesID=0 to add a new Item specification. Specify the OA_ManagerialEntriesID value to update information about Item fields. To delete existing row, specify the value OA_ManagerialEntriesID and set "DeleteRow=true".
 "oA_ManagerialEntriesID": Specify an existing OA_ManagerialEntriesID to update or remove Item information, or 0 to add a new Item.
 "deleteRow": If true, deletes Item information for specified OA_ManagerialEntriesID value.
 "code": See [ItemsList] method (use [Code] value).
 "oA_ItemsID": See [ItemsList] method (use [OA_ItemsID] value).
 "quantity":
 "unit": See [StockUnitsList] method (use [Unit] value).
 "oA_StockUnitsID": See [StockUnitsList] method (use [OA_StockUnitsID] value).
 "price":
 "amount":
 "costCentreFrom": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If ommite, gets cost centre from parameters.
 "oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If ommite, gets cost centre from parameters.
 "costCentreTo": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If ommite, gets cost centre from parameters.
 "toOA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If ommite, gets cost centre from parameters.
 "quantityInRegisterUnit":
 "registerParentAccount":
 "oA_AccountsID":
 "parentAccountForDestinationItem":
 "oA_AccountsIDTo":
 "rsItemCode":
 "rsItemName":
 "rsItemUnit":
 "rsItemVATType": Standard = 0, ZeroRate = 1, Untaxed = 2

Examples

1. Update Transfer Operation. Change parentAccountForDestinationItem "1640".

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 2012,
  "costCentreFrom": "General cost center",
  "costCentreTo": "New cost center",
  "oA_CostCentreID": 1,
  "toOA_CostCentreID": 2,
  "createAccountingEntries": true,
  "transferOperationItems": [
    {
      "code": "1",
```

```

"oA_ItemsID": 1,
"quantity": 5,
"oA_StockUnitsID": 1,
"costCentreFrom": "General cost center",
"oA_CostCentreID": 1,
"costCentreTo": "New cost center",
"toOA_CostCentreID": 2,
"registerParentAccount": "1610",
"oA_AccountsID": 1,
"parentAccountForDestinationItem": "1640",
"oA_AccountsIDTo": 2
}
]
}

```

Server response Code(200):

```

{
  "id": 2012,
  "idFieldName": "OA_TransactionsID"
}

```

Server response Code(400):

```

[
  {
    "errorNumber": 51065,
    "description": "Wrong field value!",
    "comment": "[OA_StockUnitsID] Account [] is not a item maintenance! Transaction date [19.05.2023 17:45:15]
\nOA_TransactionsID [2012] Entry number [10]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "CheckStockAccont",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

DELETE/api/TransferOperation
Returns OK or Error description Json.

Schema

```

{
  "token": "string",
  "oA_TransactionsID": 0
}

```

Field description

[Token] and [OA_TransactionsID] is necessary.

"**token**": Set [token] string from [LogIn] response value
"**oA_TransactionsID**":

Examples

1. Delete TransferOperation.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 2012
}

```

Server response Code(200):

```

{

```

```
"id": 2012,
"idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
{
  "errorNumber": 51067,
  "description": "Can not change/delete data when transaction has been documental confirmed!",
  "comment": "For change/delete data first remove documental confirm.\nOA_TransactionsID= 5072",
  "command": "",
  "errorType": "ErrorType_Error",
  "source": "OA_BusinessObject.dll",
  "class": "Transaction",
  "entryPoint": "CheckDocumentalConfirm",
  "resourceDLL_ForTranslation": "ORIS_Accounting",
  "errorPointer": 0
}
]
```

GetTransferOperationByID

POST/api/GetTransferOperationByID

Gets Transfer Operation and its Items by OA_TransactionsID value.

Schema

```
{
  "token": "string",
  "OA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"OA_TransactionsID": See [TransactionsList] method (use [OA_TransactionsID] value)

Examples

1. Get Transfer Operation by "OA_TransactionsID" - 2012

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 2012
}
```

Server response Code(200):

```
{
  "oA_TransactionsID": 2012,
  "amount": 0,
  "costCentreFrom": "General cost center",
  "costCentreTo": "New cost center",
  "documentalConfirm": false,
  "documentComment": "",
  "documentDate": "2023-05-10T17:45:15",
  "documentNumber": 1,
  "oA_CostCentreID": 1,
  "oA_SalePriceTypesID": 0,
  "oA_StatusID": 0,
  "oA_TransactionTypesID": 12,
  "reflectInformationInEntries": false,
  "rS_TaxDocument_ID": 0,
  "rS_TaxDocument_Number": "",
  "rS_Waybill_ID": 0,
}
```

```

"rS_Waybill_Number": "",
"salePriceChangingLastReason": 0,
"salePriceType": null,
"status": "",
"toOA_CostCentreID": 2,
"transferOperationItems": [
  {
    "oA_ManualEntriesID": 2009,
    "oA_TransactionsID": 2012,
    "code": "1",
    "item": "item",
    "oA_ItemsID": 2,
    "quantity": 1,
    "unit": "Piece",
    "oA_StockUnitsID": 1,
    "price": 0,
    "amount": 0,
    "costCentreFrom": "",
    "oA_CostCentreID": 0,
    "costCentreTo": "",
    "toOA_CostCentreID": 0,
    "quantityInRegisterUnit": 0,
    "oA_EntriesID": 2028,
    "registerParentAccount": "1610",
    "oA_AccountsID": 69,
    "parentAccountForDestinationItem": "",
    "oA_AccountsIDTo": 0,
    "rsItemCode": "",
    "rsItemName": "",
    "rsItemUnit": "",
    "errorPointer": 0,
    "rsItemVATType": 0,
    "hasADoubleRegistrationUnit": false,
    "itemType": 0,
    "itemVATType": 0,
    "registerUnit": "Piece",
    "oA_StockUnitsIDRegisterUnit": 1,
    "sequenceNumber": 1
  }
]
}

```

Server response Code(400):

```

[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 214",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

TransferOperationsList

POST/api/TransferOperationsList

Gets Transfer Operations. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"**token**": Set [token] string from [LogIn] response value

"**topRecordsCount**": The number of records to accept. If you specify 0 you will get all records. Default=0

"**filter**": Use SQL WHERE string by field names. Get field names by launch this method with property: [TopRecordsCount]=1.

"**sort**": Use SQL ORDER BY string with field names.

"**pageNumber**": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Gets list of Transfer Operations.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "filter": "[ViewOA_TransferOperations].[OA_TransactionsID]>201",
  "sort": "[Document date],[Has entries],[Delay]",
  "pageNumber": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 1,
      "PageRowCount": 1,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_TransactionsID": 2012
      "Document date": "2023-06-02T14:35:16",
      "OA_TransactionTypesID": 12,
      "Comment": "",
      "Document number": "1",
      "OA_CostCentreID": 1,
      "Cost centre from": "General cost center",
      "ToOA_CostCentreID": 2,
      "Cost centre to": "New cost center",
      "Delay": false,
      "Documental confirm": true,
      "Has entries": true,
      "RS_Waybill_ID": null,
      "RS Waybill number": null,
      "OA_UsersID": 1,
      "User": "sa",
      "OA_DepartmentsID": 1,
      "Department": "Administrators",
      "Record related binary documents": null,
      "RS_TaxDocument_ID": null,
      "RS Tax document number": null,
      "OA_SalePriceTypesID": null,
      "Price type": null,
      "Amount": 0.00000000,
      "OA_StatusID": null,
      "Status": null,
    }
  ]
}
```

```
"Numeric document number": 1,  
"NUM_ROW": 1  
}  
]  
}
```

SupplyOperation

POST/api/SupplyOperation

Adds new Supply Operation. Returns ID value or Error description Json

Schema

```
{  
  "token": "string",  
  "buyerCode": "string",  
  "buyerID": 0,  
  "costCentreName": "string",  
  "currency": "string",  
  "documentComment": "string",  
  "documentDate": "2023-05-05T10:05:41.569Z",  
  "documentNumber": "string",  
  "dueDate": "2023-05-05T10:05:41.569Z",  
  "expenseAccountForFixedAssetWriteOff": "string",  
  "expenseAccountForFreeSupply": "string",  
  "expenseForWriteOffAccount": "string",  
  "incomesAccountForSupply": "string",  
  "incomesAccountForUntaxed": "string",  
  "incomesAccountForZeroVAT": "string",  
  "isRateFixed": true,  
  "oA_CostCentreID": 0,  
  "oA_CurrenciesID": 0,  
  "oA_ExpenseAccountForFixedAssetWriteOffID": 0,  
  "oA_ExpenseAccountForFreeSupplyID": 0,  
  "oA_ExpenseForWriteOffAccountsID": 0,  
  "oA_IncomesAccountForSupplyID": 0,  
  "oA_IncomesAccountForUntaxedID": 0,  
  "oA_IncomesAccountForZeroVATID": 0,  
  "oA_ParentForBuyerAccountsID": 0,  
  "oA_PayableExciseAccountsID": 0,  
  "oA_PayableVATAccountsID": 0,  
  "oA_SalePriceTypesID": 0,  
  "oA_StatusID": 0,  
  "parentForBuyerAccount": "string",  
  "payableExciseAccount": "string",  
  "payableVATAccount": "string",  
  "rate": 0,  
  "receiverAddress": "string",  
  "receiverAddressesID": 0,  
  "receiverCode": "string",  
  "receiverID": 0,  
  "rS_TaxDocument_ID": 0,  
  "rS_TaxDocument_Number": "string",  
  "rS_VAT_ID": 0,  
  "rS_VAT_Number": "string",  
  "rS_Waybill_ID": 0,  
  "rS_Waybill_Number": "string",  
  "salePriceType": "string",  
  "status": "string",  
  "supplierAddress": "string",  
  "supplierAddressesID": 0,  
  "createAccountingEntries": true,  
  "supplyOperationItems": [  
    {  
      "code": "string",
```



```

"oA_ItemsID": 0,
"quantity": 0,
"unit": "string",
"oA_StockUnitsID": 0,
"price": 0,
"amount": 0,
"vatRate": 0,
"vat": 0,
"exciseRate": 0,
"excise": 0,
"quantityInRegisterUnit": 0,
"rsItemCode": "string",
"rsItemName": "string",
"rsItemUnit": "string",
"oA_AccountsID": 0,
"registerParentAccount": "string",
"rsItemVATType": 0,
"freeSupply": true,
"itemSalePrice": 0,
"oA_CostCentreID": 0,
"costCentre": "string",
"oA_AccountsIDExpenseAccount": 0,
"itemWriteOffExpenseAccount": "string"
}
]
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"**token**": Set [token] string from [LogIn] response value

"**buyerCode**": See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [BuyerID] property specified.

"**buyerID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [BuyerCode] property specified.

"**costCentreName**": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"**currency**": See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified. If omitted, gets first register currency from parameters.

"**documentComment**":

"**documentDate**": Sets operation date. If not specified - sets current date.

"**documentNumber**":

"**dueDate**":

"**expenseAccountForFixedAssetWriteOff**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_ExpenseAccountForFixedAssetWriteOffID] property specified. If both are omitted, gets the account from the parameters.

"**expenseAccountForFreeSupply**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_ExpenseAccountForFreeSupplyID] property specified. If both are omitted, gets the account from the parameters.

"**expenseForWriteOffAccount**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_ExpenseForWriteOffAccountsID] property specified. If both are omitted, gets the account from the parameters.

"**incomesAccountForSupply**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_IncomesAccountForSupplyID] property specified. If both are omitted, gets the account from the parameters.

"**incomesAccountForUntaxed**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_IncomesAccountForUntaxedID] property specified. If both are omitted, gets the account from the parameters.

"**incomesAccountForZeroVAT**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_IncomesAccountForZeroVATID] property specified. If both are omitted, gets the account from the parameters.

"**isRateFixed**": Sets a fixed rate flag. If true, then equivalent values in accounting entries of this operation will not be recalculated when the exchange rate changes.

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreName] property specified. If omitted, gets cost centre from parameters.

"**oA_CurrenciesID**": See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currency] property specified. If omitted, gets first register currency from parameters.

"**oA_ExpenseAccountForFixedAssetWriteOffID**": See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [ExpenseAccountForFixedAssetWriteOff] property specified. If both are omitted, gets the account from the parameters.

"**oA_ExpenseAccountForFreeSupplyID**": See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [ExpenseAccountForFreeSupply] property specified. If both are omitted, gets the account from the parameters.

"oA_ExpenseForWriteOffAccountsID": See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [ExpenseForWriteOffAccount] property specified. If both are omitted, gets the account from the parameters.

"oA_IncomesAccountForSupplyID": See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [IncomesAccountForSupply] property specified. If both are omitted, gets the account from the parameters.

"oA_IncomesAccountForUntaxedID": See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [IncomesAccountForUntaxed] property specified. If both are omitted, gets the account from the parameters.

"oA_IncomesAccountForZeroVATID": See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [IncomesAccountForZeroVAT] property specified. If both are omitted, gets the account from the parameters.

"oA_ParentForBuyerAccountsID": See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [OA_ParentForBuyerAccountsID] property specified. If both are omitted, gets the account from the parameters.

"oA_PayableExciseAccountsID": See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [OA_PayableExciseAccountsID] property specified. If both are omitted, gets the account from the parameters.

"oA_PayableVATAccountsID": See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [OA_PayableVATAccountsID] property specified. If both are omitted, gets the account from the parameters.

"oA_SalePriceTypesID": See [SalePriceTypesList] method (use [oA_SalePriceTypesID] value). Can be omitted if [SalePriceType] property specified.

"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified.

"parentForBuyerAccount": See [AccountsList] method (use [Account] value). Can be omitted if [OA_ParentForBuyerAccountsID] property specified. If both are omitted, gets the account from the parameters.

"payableExciseAccount": See [AccountsList] method (use [Account] value). Can be omitted if [OA_PayableExciseAccountsID] property specified. If both are omitted, gets the account from the parameters.

"payableVATAccount": See [AccountsList] method (use [Account] value). Can be omitted if [OA_PayableVATAccountsID] property specified. If both are omitted, gets the account from the parameters.

"rate": Sets the currency rate. If not specified, gets the rate by [DocumentDate]. See [CurrencyRatesList]

"receiverAddress": See [GetDebitorCreditorByID] method (use [Address] value from [DebitorCreditorAddresses]).

"receiverAddressesID": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorAddressesID] value from [DebitorCreditorAddresses]).

"receiverCode": See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [ReceiverID] property specified.

"receiverID": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [ReceiverCode] property specified.

"rS_TaxDocument_ID": Sets Tax Document ID value recieved from revenue service

"rS_TaxDocument_Number": Sets Tax Document number recieved from revenue service

"rS_VAT_ID": Sets VAT ID recieved from revenue service

"rS_VAT_Number": Sets VAT number recieved from revenue service

"rS_Waybill_ID": Sets waybill ID recieved from revenue service

"rS_Waybill_Number": Sets waybill number recieved from revenue service

"salePriceType": See [SalePriceTypesList] method (use [description] value). Can be omitted if [OA_SalePriceTypesID] property specified.

"status": See [StatusesList] method (use [Name] value). Can be omitted if [OA_StatusID] property specified.

"supplierAddress": See [Owner] method (use [Address] value or [address] value from [OwnerAddresses]).

"supplierAddressesID": See [Owner] method (use [OA_OwnerAddressesID] value from [OwnerAddresses]).

"createAccountingEntries": Creates accounting entries using managerial information. Default=false

"supplyOperationItems": SupplyOperationItemAdd

"code": See [ItemsList] method (use [Code] value). Can be omitted if [OA_ItemsID] property specified.

"oA_ItemsID": See [ItemsList] method (use [OA_ItemsID] value). Can be omitted if [Code] property specified.

"quantity":

"unit": See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified. If both are omitted, gets the Item sale unit.

"oA_StockUnitsID": See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified. If both are omitted, gets the OA_StockUnitsID for Item sale unit.

"price": If omitted, calculated as [Amount]/[Quantity]

"amount": If omitted, calculated as [Quantity]*[Price]

"vatRate": If not specified, depends on [vatType] values of [Item], [Buyer] and [owner] (See [GetItemByID], [GetDebitorCreditorByID] and [Owner] methods)

"vat": If omitted, calculated as [Amount]*[VATRate]/100

"exciseRate": If not specified, gets from [exciseRate] value of [Item] object (See [GetItemByID] method)

"excise": If omitted, calculated as [Quantity]*[ExciseRate]

"quantityInRegisterUnit": Required if [hasDoubleRegisterUnit] is true for the [Item] object

"rsItemCode":

"rsItemName":

"rsItemUnit":

"oA_AccountsID": If not specified, gets from [oA_AccountsID] value of [Item] object (See [GetItemByID] method)

"registerParentAccount": If not specified, gets account by [OA_AccountsID] of [Item] object (See [GetItemByID] method)

"rsItemVATType": Standard = 0, ZeroRate = 1, Untaxed = 2

"freeSupply":

"itemSalePrice":

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters.

"costCentre": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"oA_AccountsIDExpenseAccount":

"itemWriteOffExpenseAccount":

Examples

1. Add new Supply Operation.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "buyerCode": "3110",
  "buyerID": 1,
  "costCentreName": "General cost center",
  "oA_CostCentreID": 1,
  "receiverCode": "1410",
  "receiverID": 2,
  "createAccountingEntries": true,
  "supplyOperationItems": [
    {
      "code": "1",
      "oA_ItemsID": 1,
      "oA_StockUnitsID": 1,
      "price": 5,
      "amount": 10,
      "oA_AccountsID": 1,
      "registerParentAccount": "1610",
      "itemSalePrice": 5,
      "oA_CostCentreID": 1,
      "costCentre": "General cost center",
      "itemWriteOffExpenseAccount": "7210"
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 3008,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51053,
    "description": "OA_CostCentreID does not exist!",
    "comment": "[3] Record is not in table or user have no permissions to read this record.\n Entry number [1]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "SupplyOperation",
    "entryPoint": "SetOA_CostCentreID",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/SupplyOperation

Updates Supply Operation. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "buyerCode": "string",
  "buyerID": 0,
  "costCentreName": "string",
  "currency": "string",
  "documentComment": "string",
  "documentDate": "2023-05-05T11:16:32.255Z",
  "documentNumber": "string",
  "dueDate": "2023-05-05T11:16:32.255Z",
  "expenseAccountForFixedAssetWriteOff": "string",
  "expenseAccountForFreeSupply": "string",
  "expenseForWriteOffAccount": "string",
  "incomesAccountForSupply": "string",
  "incomesAccountForUntaxed": "string",
  "incomesAccountForZeroVAT": "string",
  "isRateFixed": true,
  "oA_CostCentreID": 0,
  "oA_CurrenciesID": 0,
  "oA_ExpenseAccountForFixedAssetWriteOffID": 0,
  "oA_ExpenseAccountForFreeSupplyID": 0,
  "oA_ExpenseForWriteOffAccountsID": 0,
  "oA_IncomesAccountForSupplyID": 0,
  "oA_IncomesAccountForUntaxedID": 0,
  "oA_IncomesAccountForZeroVATID": 0,
  "oA_ParentForBuyerAccountsID": 0,
  "oA_PayableExciseAccountsID": 0,
  "oA_PayableVATAccountsID": 0,
  "oA_SalePriceTypesID": 0,
  "oA_StatusID": 0,
  "parentForBuyerAccount": "string",
  "payableExciseAccount": "string",
  "payableVATAccount": "string",
  "rate": 0,
  "receiverAddress": "string",
  "receiverAddressesID": 0,
  "receiverCode": "string",
  "receiverID": 0,
  "rS_TaxDocument_ID": 0,
  "rS_TaxDocument_Number": "string",
  "rS_VAT_ID": 0,
  "rS_VAT_Number": "string",
  "rS_Waybill_ID": 0,
  "rS_Waybill_Number": "string",
  "salePriceType": "string",
  "status": "string",
  "supplierAddress": "string",
  "supplierAddressesID": 0,
  "createAccountingEntries": true,
  "clearItems": true,
  "supplyOperationItems": [
    {
      "oA_ManualEntriesID": 0,
      "deleteRow": true,
      "code": "string",
      "oA_ItemsID": 0,
      "quantity": 0,
      "unit": "string",
      "oA_StockUnitsID": 0,
      "price": 0,
      "amount": 0,
      "vatRate": 0,
      "vat": 0,
      "exciseRate": 0,
      "excise": 0,
    }
  ]
}
```

```

    "quantityInRegisterUnit": 0,
    "rsItemCode": "string",
    "rsItemName": "string",
    "rsItemUnit": "string",
    "oA_AccountsID": 0,
    "registerParentAccount": "string",
    "rsItemVATType": 0,
    "freeSupply": true,
    "itemSalePrice": 0,
    "oA_CostCentreID": 0,
    "costCentre": "string",
    "oA_AccountsIDExpenseAccount": 0,
    "itemWriteOffExpenseAccount": "string"
  }
]
}

```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change.

"**token**": Set [token] string from [Login] response value

"**oA_TransactionsID**":

"**buyerCode**": See [DebitorCreditorsList] method (use [Code] value).

"**buyerID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value).

"**costCentreName**": See [CostCentresList] method (use [Name] value).

"**currency**": See [CurrenciesList] method (use [Currency] value).

"**documentComment**":

"**documentDate**": Sets operation date.

"**documentNumber**":

"**dueDate**":

"**expenseAccountForFixedAssetWriteOff**": See [AccountsList] method (use [Account] value).

"**expenseAccountForFreeSupply**": See [AccountsList] method (use [Account] value).

"**expenseForWriteOffAccount**": See [AccountsList] method (use [Account] value).

"**incomesAccountForSupply**": See [AccountsList] method (use [Account] value).

"**incomesAccountForUntaxed**": See [AccountsList] method (use [Account] value).

"**incomesAccountForZeroVAT**": See [AccountsList] method (use [Account] value).

"**isRateFixed**": Sets a fixed rate flag. If true, then equivalent values in accounting entries of this operation will not be recalculated when the exchange rate changes.

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value).

"**oA_CurrenciesID**": See [CurrenciesList] method (use [OA_CurrenciesID] value).

"**oA_ExpenseAccountForFixedAssetWriteOffID**": See [AccountsList] method (use [OA_AccountsID] value).

"**oA_ExpenseAccountForFreeSupplyID**": See [AccountsList] method (use [OA_AccountsID] value).

"**oA_ExpenseForWriteOffAccountsID**": See [AccountsList] method (use [OA_AccountsID] value).

"**oA_IncomesAccountForSupplyID**": See [AccountsList] method (use [OA_AccountsID] value).

"**oA_IncomesAccountForUntaxedID**": See [AccountsList] method (use [OA_AccountsID] value).

"**oA_IncomesAccountForZeroVATID**": See [AccountsList] method (use [OA_AccountsID] value).

"**oA_ParentForBuyerAccountsID**": See [AccountsList] method (use [OA_AccountsID] value).

"**oA_PayableExciseAccountsID**": See [AccountsList] method (use [OA_AccountsID] value).

"**oA_PayableVATAccountsID**": See [AccountsList] method (use [OA_AccountsID] value).

"**oA_SalePriceTypesID**": See [SalePriceTypesList] method (use [oA_SalePriceTypesID] value).

"**oA_StatusID**": See [StatusesList] method (use [OA_StatusID] value).

"**parentForBuyerAccount**": See [AccountsList] method (use [Account] value).

"**payableExciseAccount**": See [AccountsList] method (use [Account] value).

"**payableVATAccount**": See [AccountsList] method (use [Account] value).

"**rate**": Sets the currency rate.

"**receiverAddress**": See [GetDebitorCreditorByID] method (use [Address] value from [DebitorCreditorAddresses]).

"**receiverAddressesID**": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorAddressesID] value from [DebitorCreditorAddresses]).

"**receiverCode**": See [DebitorCreditorsList] method (use [Code] value).

"**receiverID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value).

"**rS_TaxDocument_ID**": Sets Tax Document ID value recieved from revenue service

"**rS_TaxDocument_Number**": Sets Tax Document number recieved from revenue service

"**rS_VAT_ID**": Sets VAT ID recieved from revenue service

"**rS_VAT_Number**": Sets VAT number recieved from revenue service

"**rS_Waybill_ID**": Sets waybill ID recieved from revenue service

"**rS_Waybill_Number**": Sets waybill number recieved from revenue service

"**salePriceType**": See [SalePriceTypesList] method (use [description] value).
 "**status**": See [StatusesList] method (use [Name] value).
 "**supplierAddress**": See [Owner] method (use [Address] value or [address] value from [OwnerAddresses]).
 "**supplierAddressesID**": See [Owner] method (use [OA_OwnerAddressesID] value from [OwnerAddresses]).
 "**createAccountingEntries**": Creates accounting entries using managerial information. Default=false
 "**clearItems**": Clears all items for the operation.

"**supplyOperationItems**": Set OA_ManagerialEntriesID=0 to add a new Item specification. Specify the OA_ManagerialEntriesID value to update information about Item fields. To delete existing row, specify the value OA_ManagerialEntriesID and set "DeleteRow"=true.
 "**oA_ManagerialEntriesID**": Specify an existing OA_ManagerialEntriesID to update or remove Item information, or 0 to add a new Item.
 "**deleteRow**": If true, deletes Item information for specified OA_ManagerialEntriesID value.
 "**code**": See [ItemsList] method (use [Code] value).
 "**oA_ItemsID**": See [ItemsList] method (use [OA_ItemsID] value).
 "**quantity**":
 "**unit**": See [StockUnitsList] method (use [Unit] value).
 "**oA_StockUnitsID**": See [StockUnitsList] method (use [OA_StockUnitsID] value).
 "**price**":
 "**amount**":
 "**vatRate**":
 "**vat**":
 "**exciseRate**":
 "**excise**":
 "**quantityInRegisterUnit**":
 "**rsItemCode**":
 "**rsItemName**":
 "**rsItemUnit**":
 "**oA_AccountsID**":
 "**registerParentAccount**":
 "**rsItemVATType**": Standard = 0, ZeroRate = 1, Untaxed = 2
 "**freeSupply**":
 "**itemSalePrice**":
 "**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters.
 "**costCentre**": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.
 "**oA_AccountsIDExpenseAccount**":
 "**itemWriteOffExpenseAccount**":

Examples

1. Update Supply Operation. Change registerParentAccount "1620".

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 3008,
  "buyerCode": "3110",
  "buyerID": 1,
  "receiverCode": "1410",
  "receiverID": 2,
  "createAccountingEntries": true,
  "supplyOperationItems": [
    {
      "code": "1",
      "oA_ItemsID": 1,
      "oA_StockUnitsID": 1,
      "price": 5,
      "amount": 10,
      "oA_AccountsID": 1,
      "registerParentAccount": "1620",
      "itemSalePrice": 5,
      "itemWriteOffExpenseAccount": "7210"
    }
  ]
}
```

Server response Code(200):

```
{
  "id":3008,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51125,
    "description": "Add new sub account under item account is prohibited!",
    "comment": "Account [1610 1] 1610 1",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Account",
    "entryPoint": "PrepareNewProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/SupplyOperation

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID": 0

Examples

1. Delete SupplyOperation.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 3008
}
```

Server response Code(200):

```
{
  "id": 3008,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 308",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "GetProperties",
  }
]
```



```
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]
```

GetSupplyOperationByID

POST/api/GetSupplyOperationByID

Gets Supply Operation and its Items by OA_TransactionsID value.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"OA_TransactionsID": See [TransactionsList] method (use [OA_TransactionsID] value).

Examples

1. Get Supply Operation by "oA_TransactionsID" - 3008

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 3008
}
```

Server response Code(200):

```
{
  "oA_TransactionsID": 3008,
  "amount": 0,
  "buyerCode": "1",
  "buyerID": 1,
  "buyerName": "Organization",
  "costCentreName": "General cost center",
  "currency": "GEL",
  "debitorCreditorVATType": 0,
  "documentalConfirm": false,
  "documentComment": "",
  "documentDate": "2023-05-10T16:34:07",
  "documentNumber": 1,
  "dueDate": "2023-05-10T16:55:34",
  "excise": 0,
  "expenseAccountForFixedAssetWriteOff": "8290",
  "expenseAccountForFreeSupply": "9200",
  "expenseForWriteOffAccount": "7210",
  "incomesAccountForSupply": "6110 1",
  "incomesAccountForUntaxed": "6110 3",
  "incomesAccountForZeroVAT": "6110 2",
  "isDueDateNULL": true,
  "isRateFixed": false,
  "oA_CostCentreID": 1,
  "oA_CurrenciesID": 1,
  "oA_ExpenseAccountForFixedAssetWriteOffID": 208,
  "oA_ExpenseAccountForFreeSupplyID": 45,
  "oA_ExpenseForWriteOffAccountsID": 180,
  "oA_IncomesAccountForSupplyID": 211,
  "oA_IncomesAccountForUntaxedID": 213,
  "oA_IncomesAccountForZeroVATID": 212,
  "oA_ParentForBuyerAccountsID": 56,
```



```
"oA_PayableExciseAccountsID": 134,
"oA_PayableVATAccountsID": 132,
"oA_SalePriceTypesID": 0,
"oA_StatusID": 0,
"oA_TransactionTypesID": 11,
"ownerVATType": 0,
"parentForBuyerAccount": "1410",
"payableExciseAccount": "3350",
"payableVATAccount": "3330",
"rate": 1,
"receiverAddress": "",
"receiverAddressesID": 0,
"receiverCode": "1",
"receiverID": 1,
"receiverName": "Organization",
"reflectInformationInEntries": false,
"rS_TaxDocument_ID": 0,
"rS_TaxDocument_Number": "",
"rS_VAT_ID": 0,
"rS_VAT_Number": "",
"rS_Waybill_ID": 0,
"rS_Waybill_Number": "",
"salePriceChangingLastReason": 0,
"salePriceType": "",
"status": "",
"supplier": "",
"supplierAddress": "",
"supplierAddressesID": 0,
"supplierID": 0,
"vat": 0,
"supplyOperationItems": [
  {
    "oA_ManagementEntriesID": 3005,
    "oA_TransactionsID": 3008,
    "code": "1",
    "item": "",
    "oA_ItemsID": 1,
    "quantity": 1,
    "unit": "Piece",
    "oA_StockUnitsID": 1,
    "price": 0,
    "amount": 0,
    "vatRate": 18,
    "vat": 0,
    "exciseRate": 0,
    "excise": 0,
    "quantityInRegisterUnit": 0,
    "rsItemCode": "",
    "rsItemName": "",
    "rsItemUnit": "",
    "oA_AccountsID": 262,
    "oA_EntriesID": 3035,
    "registerParentAccount": "1610 1",
    "errorPointer": 0,
    "itemVATType": 0,
    "rsItemVATType": 0,
    "freeSupply": false,
    "itemSalePrice": 0,
    "itemSalePriceOA_CurrenciesID": 1,
    "itemSalePriceCurrency": "GEL",
    "itemSalePriceOA_StockUnitsID": 1,
    "itemSalePriceUnit": "Piece",
    "itemSalePriceInSaleCurrencyAndUnit": 0,
    "hasADoubleRegistrationUnit": false,
    "oA_StockUnitsIDRegisterUnit": 1,
    "registerUnit": "Piece",
    "oA_CostCentreID": 0,
```

```

    "costCentre": "",
    "oA_AccountsIDExpenseAccount": 0,
    "itemWriteOffExpenseAccount": "",
    "itemType": 0,
    "sequenceNumber": 1
  }
]
}

```

Server response Code(400):

```

[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 3009",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

SupplyOperationsList

POST/api/SupplyOperationsList

Gets Supply Operations. Returns [Info] and [Result] tables

Schema

```

{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property: [TopRecordsCount]=1.

"sort": Use SQL ORDER BY string with field names.

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get List Supply Operations.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "filter": "UPPER([ViewOA_SupplyOperations].[Supplier address]) LIKE N'%AL. KAZBEGI AVE.%'",
  "sort": "[Receiver name],[Receiver code]",
  "pageNumber": 1
}

```

Server response Code(200):

```

{
  "Info": [
    {

```

```
"RowCount": 1,
"PageCount": 1,
"PageRowCount": 1,
"PageNumber": 1
}
],
"Result": [
{
"OA_TransactionsID": 5050,
"Document date": "2023-05-31T15:11:41",
"Amount": 10.00000000,
"Supplier": "Organization",
"Supplier address": "Al. Kazbegi Ave.",
"OA_TransactionTypesID": 11,
"Comment": "",
"Document number": "3",
"BuyerID": 1,
"Buyer code": "1",
"Buyer name": "Organization",
"OA_CostCentreID": 1,
"Cost centre": "General cost center",
"OA_ParentForBuyerAccountsID": 56,
"Parent account for a buyer": "1410",
"OA_ExpenseForWriteOffAccountsID": 180,
"Expense account for item write off": "7210",
"OA_PayableVATAccountsID": 132,
"Payable VAT account": "3330",
"OA_PayableExciseAccountsID": 134,
"Payable excise account": "3350",
"Receiver address": "",
"OA_IncomesAccountForSupplyID": 211,
"Incomes account for supply": "6110 1",
"Delay": false,
"Documental confirm": false,
"Has entries": true,
"ReceiverAddressID": null,
"SupplierAddressID": null,
"VAT": 1.53000000,
"Excise": 0.00000000,
"RS_Waybill_ID": null,
"RS Waybill number": null,
"RS_VAT_ID": null,
"RS VAT number": null,
"OA_ExpenseAccountForFixedAssetWriteOffID": 208,
"Expense account for fixed asset write off": "8290",
"OA_UsersID": 1,
"OA_DepartmentsID": 1,
"User": "sa",
"Department": "Administrators",
"Rate": 1.00000000,
"Is rate fixed": false,
"OA_CurrenciesID": 1,
"Currency": "GEL",
"Record related binary documents": null,
"Owner VAT type": 0,
"DebitorCreditor VAT type": 0,
"ReceiverID": 2,
"Receiver code": "2",
"Receiver name": "Organization2",
"RS_TaxDocument_ID": null,
"RS Tax document number": null,
"OA_IncomeAccountForZeroVATID": 212,
"Incomes account for zero VAT supply": "6110 2",
"OA_IncomeAccountForUntaxedID": 213,
"Incomes account for untaxed supply": "6110 3",
"OA_ExpenseAccountForFreeSupplyID": 45,
"Expense account for free supply": "9200",
```

```

"OA_SalePriceTypesID": null,
"Price type": null,
"OA_DebitorCreditorGroupID": null,
"Due date": null,
"OA_StatusID": null,
"Status": null,
"Numeric document number": 3,
"NUM_ROW": 1
}
]
}

```

WriteOffOperation

POST/api/WriteOffOperation

Adds new Write Off Operation. Returns ID value or Error description Json.

Schema

```

{
  "token": "string",
  "costCentre": "string",
  "documentComment": "string",
  "documentDate": "2023-05-08T05:14:19.099Z",
  "documentNumber": "string",
  "expenseAccountForFixedAssetWriteOff": "string",
  "expenseForWriteOffAccount": "string",
  "oA_CostCentreID": 0,
  "oA_ExpenseAccountForFixedAssetWriteOffID": 0,
  "oA_ExpenseForWriteOffAccountsID": 0,
  "oA_StatusID": 0,
  "status": "string",
  "createAccountingEntries": true,
  "writeOffOperationItems": [
    {
      "code": "string",
      "oA_ItemsID": 0,
      "quantity": 0,
      "unit": "string",
      "oA_StockUnitsID": 0,
      "quantityInRegisterUnit": 0,
      "oA_AccountsID": 0,
      "registerParentAccount": "string",
      "oA_CostCentreID": 0,
      "costCentre": "string",
      "oA_AccountsIDExpenseAccount": 0,
      "itemWriteOffExpenseAccount": "string"
    }
  ]
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"costCentre": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"documentComment":

"documentDate": Sets operation date. If not specified - sets current date.

"documentNumber": Sets the transaction document number. If not specified, sets a unique number in the current year.

"expenseAccountForFixedAssetWriteOff": See [AccountsList] method (use [Account] value). Can be omitted if [OA_ExpenseAccountForFixedAssetWriteOffID] property specified. If both are omitted, gets the account from the parameters.

"**expenseForWriteOffAccount**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_ExpenseForWriteOffAccountsID] property specified. If both are omitted, gets the account from the parameters.
"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters. "oA_ExpenseAccountForFixedAssetWriteOffID": 0,
"oA_ExpenseForWriteOffAccountsID": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [ExpenseAccountForFixedAssetWriteOff] property specified. If both are omitted, gets the account from the parameters.
"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified.
"status": See [StatusesList] method (use [Name] value). Can be omitted if [OA_StatusID] property specified.
"createAccountingEntries": Creates accounting entries using managerial information. Default=false

"writeOffOperationItems":WriteOffOperationItemAdd

"code": See [ItemsList] method (use [Code] value). Can be omitted if [OA_ItemsID] property specified.

"oA_ItemsID": See [ItemsList] method (use [OA_ItemsID] value). Can be omitted if [Code] property specified.

"quantity":

"unit": See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified. If both are omitted, gets the Item register minimum unit.

"oA_StockUnitsID": See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified. If both are omitted, gets the OA_StockUnitsID for Item register minimum unit.

"quantityInRegisterUnit": Required if [hasDoubleRegisterUnit] is true for the [Item] object

"oA_AccountsID": If not specified, gets from [oA_AccountsID] value of [Item] object (See [GetItemByID] method)

"registerParentAccount": If not specified, gets account by [OA_AccountsID] of [Item] object (See [GetItemByID] method)

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters.

"costCentre": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"oA_AccountsIDExpenseAccount":

"itemWriteOffExpenseAccount":

Examples

1. Add new Write Off Operation.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "expenseForWriteOffAccount": "7210",
  "createAccountingEntries": true,
  "writeOffOperationItems": [
    {
      "code": "1",
      "oA_ItemsID": 1,
      "quantity": 1,
      "registerParentAccount": "1610",
      "itemWriteOffExpenseAccount": "7210"
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 3009,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51038,
    "description": "Account is closed!",
    "comment": "Account [1610 1] 1 item2\nUncheck close check box for this account and resume or cancel transaction\n OA_AccountsID=262\nOA_TransactionsID [5078]\nDocument number [2]\nDate[02.06.2023 16:18:45]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "PostUpdateActionStagel",
  }
]
```

```

"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

PUT/api/WriteOffOperation

Updates Write Off Operation. Returns OK or Error description Json.

Schema

```

{
  "token": "string",
  "oA_TransactionsID": 0,
  "costCentre": "string",
  "documentComment": "string",
  "documentDate": "2023-05-08T06:23:42.284Z",
  "documentNumber": "string",
  "expenseAccountForFixedAssetWriteOff": "string",
  "expenseForWriteOffAccount": "string",
  "oA_CostCentreID": 0,
  "oA_ExpenseAccountForFixedAssetWriteOffID": 0,
  "oA_ExpenseForWriteOffAccountsID": 0,
  "oA_StatusID": 0,
  "status": "string",
  "createAccountingEntries": true,
  "clearItems": true,
  "writeOffOperationItems": [
    {
      "oA_ManualEntriesID": 0,
      "deleteRow": true,
      "code": "string",
      "oA_ItemsID": 0,
      "quantity": 0,
      "unit": "string",
      "oA_StockUnitsID": 0,
      "quantityInRegisterUnit": 0,
      "oA_AccountsID": 0,
      "registerParentAccount": "string",
      "oA_CostCentreID": 0,
      "costCentre": "string",
      "oA_AccountsIDExpenseAccount": 0,
      "itemWriteOffExpenseAccount": "string"
    }
  ]
}

```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

"costCentre": See [CostCentresList] method (use [Name] value).

"documentComment":

"documentDate": Sets operation date.

"documentNumber":

"expenseAccountForFixedAssetWriteOff": See [AccountsList] method (use [Account] value).

"expenseForWriteOffAccount": See [AccountsList] method (use [Account] value).

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value).

"oA_ExpenseAccountForFixedAssetWriteOffID": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"oA_ExpenseForWriteOffAccountsID": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value).

"status": See [StatusesList] method (use [Name] value).

"**createAccountingEntries**": Creates accounting entries using managerial information. Default=false

"**clearItems**": Clears all items for the operation.

"**writeOffOperationItems**": Set OA_ManagerialEntriesID=0 to add a new Item specification. Specify the OA_ManagerialEntriesID value to update information about Item fields. To delete existing row, specify the value OA_ManagerialEntriesID and set "DeleteRow"=true.

"**oA_ManagerialEntriesID**": Specify an existing OA_ManagerialEntriesID to update or remove Item information, or 0 to add a new Item.

"**deleteRow**": If true, deletes Item information for specified OA_ManagerialEntriesID value.

"**code**": See [ItemsList] method (use [Code] value).

"**oA_ItemsID**": See [ItemsList] method (use [OA_ItemsID] value).

"**quantity**":

"**unit**": See [StockUnitsList] method (use [Unit] value).

"**oA_StockUnitsID**": See [StockUnitsList] method (use [OA_StockUnitsID] value).

"**quantityInRegisterUnit**":

"**oA_AccountsID**":

"**registerParentAccount**":

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters.

"**costCentre**": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"**oA_AccountsIDExpenseAccount**":

"**itemWriteOffExpenseAccount**":

Examples

1. Update Write Off Operation "oA_TransactionsID" - 3009, change quantity - 2

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 3009,
  "expenseForWriteOffAccount": "7210",
  "createAccountingEntries": true,
  "writeOffOperationItems": [
    {
      "code": "1",
      "oA_ItemsID": 1,
      "quantity": 2,
      "registerParentAccount": "1610",
      "itemWriteOffExpenseAccount": "7210"
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 3009,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51067,
    "description": "Can not change/delete data when transaction has been documental confirmed!",
    "comment": "For change/delete data first remove documental confirm.\noA_TransactionsID= 3009",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "CheckDocumentalConfirm",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/WriteOffOperation
Returns OK or Error description Json.

Schema

```
{  
  "token": "string",  
  "oA_TransactionsID": 0  
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value
"oA_TransactionsID":

Examples

1. Delete WriteOffOperation.

Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "oA_TransactionsID":3009  
}
```

Server response Code(200):

```
{  
  "id": 3009,  
  "idFieldName": "OA_TransactionsID"  
}
```

Server response Code(400):

```
[  
{  
  "errorNumber": 51067,  
  "description": "Can not change/delete data when transaction has been documental confirmed!",  
  "comment": "For change/delete data first remove documental confirm.\nOA_TransactionsID= 3009",  
  "command": "",  
  "errorType": "ErrorType_Error",  
  "source": "OA_BusinessObject.dll",  
  "class": "Transaction",  
  "entryPoint": "CheckDocumentalConfirm",  
  "resourceDLL_ForTranslation": "ORIS_Accounting",  
  "errorPointer": 0  
}  
]
```

GetWriteOffOperationByID

POST/api/GetWriteOffOperationByID

Gets Write Off Operation and its Items by OA_TransactionsID value.

Schema

```
{  
  "token": "string",  
  "oA_TransactionsID": 0  
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"OA_TransactionsID": See [TransactionsList] method (use [OA_TransactionsID] value)

Examples

1. Get Write Off Operation by "oA_TransactionsID" - 3009

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 3009
}
```

Server response Code(200):

```
{
  "oA_TransactionsID": 3009,
  "costCentre": "General cost center",
  "documentalConfirm": false,
  "documentComment": "",
  "documentDate": "2023-05-10T17:58:56",
  "documentNumber": 1,
  "expenseAccountForFixedAssetWriteOff": "8290",
  "expenseForWriteOffAccount": "7210",
  "oA_CostCentreID": 1,
  "oA_ExpenseAccountForFixedAssetWriteOffID": 208,
  "oA_ExpenseForWriteOffAccountsID": 180,
  "oA_StatusID": 0,
  "oA_TransactionTypesID": 24,
  "reflectInformationInEntries": false,
  "status": "",
  "writeOffOperationItems": [
    {
      "oA_ManagementEntriesID": 3006,
      "oA_TransactionsID": 3009,
      "code": "1",
      "item": "item",
      "oA_ItemsID": 2,
      "quantity": 1,
      "unit": "Piece",
      "oA_StockUnitsID": 1,
      "quantityInRegisterUnit": 0,
      "itemType": 0,
      "oA_EntriesID": 3037,
      "oA_AccountsID": 69,
      "registerParentAccount": "1610",
      "errorPointer": 0,
      "hasADoubleRegistrationUnit": false,
      "oA_StockUnitsIDRegisterUnit": 1,
      "registerUnit": "Piece",
      "oA_CostCentreID": 0,
      "costCentre": "",
      "oA_AccountsIDExpenseAccount": 0,
      "itemWriteOffExpenseAccount": "",
      "sequenceNumber": 1
    }
  ]
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 309",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
  }
]
```

```

    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

WriteOffOperationsList

POST/api/WriteOffOperationsList

Gets WriteOf Operations. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property: [TopRecordsCount]=1.

"sort": Use SQL ORDER BY string with field names.

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get List WriteOf Operations

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "filter": "UPPER([ViewOA_WriteOffOperations].[Cost centre])=N'GENERAL COST CENTER'",
  "sort": "[OA_CostCentreID],[Document date]",
  "pageNumber": 1
}

```

Server response Code(200):

```

{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 1,
      "PageRowCount": 1,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_TransactionsID": 3009,
      "Document date": "2023-06-02T16:34:04",
      "OA_TransactionTypeID": 24,
      "Comment": "",
      "Document number": "2",
      "OA_CostCentreID": 1,
      "Cost centre": "General cost center",
      "Delay": false,
      "Documental confirm": false,
      "Has entries": true,
      "User": "sa",
    }
  ]
}

```

```

"OA_DepartmentsID": 1,
"Department": "Administrators",
"OA_ExpenseForWriteOffAccountsID": 180,
"Expense account for item write off": "7210",
"OA_ExpenseAccountForFixedAssetWriteOffID": 208,
"Expense account for fixed asset write off": "8290",
"Record related binary documents": null,
"OA_StatusID": null,
>Status": null,
"Numeric document number": 2,
"NUM_ROW": 1
}
]
}

```

ReturnToSupplier

POST/api/ReturnToSupplier

Adds new Return To Supplier. Returns ID value or Error description Json.

Schema

```

{
  "token": "string",
  "costCentreName": "string",
  "currency": "string",
  "documentComment": "string",
  "documentDate": "2023-05-05T07:14:49.195Z",
  "documentNumber": "string",
  "dueDate": "2023-05-05T07:14:49.195Z",
  "expenseAccountForFixedAssetWriteOff": "string",
  "expenseForWriteOffAccount": "string",
  "isRateFixed": true,
  "oA_CostCentreID": 0,
  "oA_CurrenciesID": 0,
  "oA_ExpenseAccountForFixedAssetWriteOffID": 0,
  "oA_ExpenseForWriteOffAccountsID": 0,
  "oA_ParentAccountOfSupplierID": 0,
  "oA_PayedExciseAccountsID": 0,
  "oA_PayedVATAccountID": 0,
  "oA_PurchasedItemReturnAccountID": 0,
  "oA_StatusID": 0,
  "parentAccountOfSupplier": "string",
  "payedExciseAccount": "string",
  "payedVATAccount": "string",
  "purchasedItemReturnAccount": "string",
  "rate": 0,
  "receiverAddress": "string",
  "receiverAddressesID": 0,
  "rS_TaxDocument_ID": 0,
  "rS_TaxDocument_Number": "string",
  "rS_VAT_ID": 0,
  "rS_VAT_Number": "string",
  "rS_Waybill_ID": 0,
  "rS_Waybill_Number": "string",
  "sellerCode": "string",
  "sellerID": 0,
  "status": "string",
  "supplierAddress": "string",
  "supplierAddressID": 0,
  "supplierCode": "string",
  "supplierID": 0,
  "createAccountingEntries": true,
  "clearItems": true,
  "returnToSupplierItems": [

```

```

{
  "code": "string",
  "oA_ItemsID": 0,
  "quantity": 0,
  "unit": "string",
  "oA_StockUnitsID": 0,
  "price": 0,
  "amount": 0,
  "vatRate": 0,
  "vat": 0,
  "exciseRate": 0,
  "excise": 0,
  "quantityInRegisterUnit": 0,
  "supplierItemCode": "string",
  "supplierItemName": "string",
  "supplierItemUnit": "string",
  "oA_AccountsID": 0,
  "registerParentAccount": "string",
  "supplierItemVATType": 0,
  "costCentre": "string",
  "oA_CostCentreID": 0,
  "oA_AccountsIDExpenseAccount": 0,
  "itemWriteOffExpenseAccount": "string"
}
]
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"**token**": Set [token] string from [LogIn] response value

"**costCentreName**": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"**currency**": See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified. If omitted, gets first register currency from parameters.

"**documentComment**":

"**documentDate**": Sets operation date. If not specified - sets current date.

"**documentNumber**": Sets the transaction document number. If not specified, sets a unique number in the current year.

"**dueDate**":

"**expenseAccountForFixedAssetWriteOff**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_ExpenseAccountForFixedAssetWriteOffID] property specified. If both are omitted, gets the account from the parameters.

"**expenseForWriteOffAccount**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_ExpenseForWriteOffAccountsID] property specified. If both are omitted, gets the account from the parameters.

"**isRateFixed**": Sets a fixed rate flag. If true, then equivalent values in accounting entries of this operation will not be recalculated when the exchange rate changes.

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreName] property specified. If omitted, gets cost centre from parameters.

"**oA_CurrenciesID**": See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currency] property specified. If omitted, gets first register currency from parameters.

"**oA_ExpenseAccountForFixedAssetWriteOffID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [ExpenseAccountForFixedAssetWriteOff] property specified. If both are omitted, gets the account from the parameters.

"**oA_ExpenseForWriteOffAccountsID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [ExpenseForWriteOffAccount] property specified. If both are omitted, gets the account from the parameters.

"**oA_ParentAccountOfSupplierID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [ParentAccountOfSupplier] property specified. If both are omitted, gets the account from the parameters.

"**oA_PayedExciseAccountsID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [PayedExciseAccount] property specified. If both are omitted, gets the account from the parameters.

"**oA_PayedVATAccountID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [PayedVATAccount] property specified. If both are omitted, gets the account from the parameters.

"**oA_PurchasedItemReturnAccountID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [PurchasedItemReturnAccount] property specified. If both are omitted, gets the account from the parameters.

"**oA_StatusID**": See [StatusesList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified.

"**parentAccountOfSupplier**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_ParentAccountOfSupplierID] property specified. If both are omitted, gets the account from the parameters.

"**payedExciseAccount**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_PayedExciseAccountsID] property specified. If both are omitted, gets the account from the parameters.

"**payedVATAccount**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_PayedVATAccountID] property specified. If both are omitted, gets the account from the parameters.

"**purchasedItemReturnAccount**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_PurchasedItemReturnAccountID] property specified. If both are omitted, gets the account from the parameters.

"**rate**": Sets the currency rate. If not specified, gets the rate by [DocumentDate]. See [CurrencyRatesList]

"**receiverAddress**": See [GetDebitorCreditorByID] method (use [Address] value from [DebitorCreditorAddresses]).

"**receiverAddressesID**": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorAddressesID] value from [DebitorCreditorAddresses]).

"**rS_TaxDocument_ID**": Sets Tax Document ID value recieved from revenue service

"**rS_TaxDocument_Number**": Sets Tax Document number recieved from revenue service

"**rS_VAT_ID**": Sets VAT ID recieved from revenue service

"**rS_VAT_Number**": Sets VAT number recieved from revenue service

"**rS_Waybill_ID**": Sets waybill ID recieved from revenue service

"**rS_Waybill_Number**": Sets waybill number recieved from revenue service

"**sellerCode**": See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [SellerID] property specified.

"**sellerID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [SellerCode] property specified.

"**status**": See [StatusesList] method (use [Name] value). Can be omitted if [OA_StatusID] property specified.

"**supplierAddress**": See [Owner] method (use [Address] value or [address] value from [OwnerAddresses]).

"**supplierAddressesID**": See [Owner] method (use [OA_OwnerAddressesID] value from [OwnerAddresses]).

"**supplierCode**": See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [SupplierID] property specified.

"**supplierID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [SupplierCode] property specified.

"**createAccountingEntries**": Creates accounting entries using managerial information. Default=false

"**clearItems**": Clears all items for the operation.

"**returnToSupplierItems**": ReturnToSupplierItemAdd

"**code**": See [ItemsList] method (use [Code] value). Can be omitted if [OA_ItemsID] property specified. If both are omitted, gets the Item register minimum unit.

"**oA_ItemsID**": See [ItemsList] method (use [OA_ItemsID] value). Can be omitted if [Code] property specified. If both are omitted, gets the OA_StockUnitsID for Item register minimum unit.

"**quantity**":

"**unit**": See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified. If both are omitted, gets the OA_StockUnitsID for Item register minimum unit.

"**oA_StockUnitsID**": See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified. If both are omitted, gets the OA_StockUnitsID for Item register minimum unit.

"**price**": If omitted, calculated as [Amount]/[Quantity]

"**amount**": If omitted, calculated as [Quantity]*[Price]

"**vatRate**": If not specified, depends on [vatType] values of [Item], [Seller] and [owner] (See [GetItemByID], [GetDebitorCreditorByID] and [Owner] methods)

"**vat**": If omitted, calculated as [Amount]*[VATRate]/100

"**exciseRate**": If not specified, gets from [exciseRate] value of [Item] object (See [GetItemByID] method)

"**excise**": If omitted, calculated as [Quantity]*[ExciseRate]

"**quantityInRegisterUnit**": Required if [hasDoubleRegisterUnit] is true for the [Item] object

"**supplierItemCode**":

"**supplierItemName**":

"**supplierItemUnit**":

"**oA_AccountsID**": If not specified, gets from [oA_AccountsID] value of [Item] object (See [GetItemByID] method)

"**registerParentAccount**": If not specified, gets account by [OA_AccountsID] of [Item] object (See [GetItemByID] method)

"**supplierItemVATType**": Standard = 0, ZeroRate = 1, Untaxed = 2

"**costCentre**": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If ommite, gets cost centre from parameters.

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If ommite, gets cost centre from parameters

"**oA_AccountsIDExpenseAccount**":

"**itemWriteOffExpenseAccount**":

Examples

1. Add new Return To Supplier.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "expenseForWriteOffAccount": "7210",
  "oA_ParentAccountOfSupplierID": 1,
  "oA_PayedExciseAccountsID": 2,
  "parentAccountOfSupplier": "1410",
  "purchasedItemReturnAccount": "1610 1",
  "supplierID": 2,
  "createAccountingEntries": true,
  "returnToSupplierItems": [
    {
      "code": "1",
      "oA_ItemsID": 1,
      "quantity": 1,
      "price": 10,
      "oA_AccountsIDExpenseAccount": 1,
      "itemWriteOffExpenseAccount": "7210"
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 3005,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51190,
    "description": "Property not found!",
    "comment": "OA_TransactionsID=0 [2]\n[OA_CostCentreID]=11",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "ReturnToSupplier",
    "entryPoint": "set property",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/ReturnToSupplier

Updates Return To Supplier. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "costCentreName": "string",
  "currency": "string",
  "documentComment": "string",
  "documentDate": "2023-05-05T07:49:51.875Z",
  "documentNumber": "string",
  "dueDate": "2023-05-05T07:49:51.875Z",
  "expenseAccountForFixedAssetWriteOff": "string",
  "expenseForWriteOffAccount": "string",
  "isRateFixed": true,
  "oA_CostCentreID": 0,
}
```

```

"oA_CurrenciesID": 0,
"oA_ExpenseAccountForFixedAssetWriteOffID": 0,
"oA_ExpenseForWriteOffAccountsID": 0,
"oA_ParentAccountOfSupplierID": 0,
"oA_PayedExciseAccountsID": 0,
"oA_PayedVATAccountID": 0,
"oA_PurchasedItemReturnAccountID": 0,
"oA_StatusID": 0,
"parentAccountOfSupplier": "string",
"payedExciseAccount": "string",
"payedVATAccount": "string",
"purchasedItemReturnAccount": "string",
"rate": 0,
"receiverAddress": "string",
"receiverAddressesID": 0,
"rS_TaxDocument_ID": 0,
"rS_TaxDocument_Number": "string",
"rS_VAT_ID": 0,
"rS_VAT_Number": "string",
"rS_Waybill_ID": 0,
"rS_Waybill_Number": "string",
"sellerCode": "string",
"sellerID": 0,
"status": "string",
"supplierAddress": "string",
"supplierAddressID": 0,
"supplierCode": "string",
"supplierID": 0,
"createAccountingEntries": true,
"clearItems": true,
"returnToSupplierItems": [
  {
    "oA_ManualEntriesID": 0,
    "deleteRow": true,
    "code": "string",
    "oA_ItemsID": 0,
    "quantity": 0,
    "unit": "string",
    "oA_StockUnitsID": 0,
    "price": 0,
    "amount": 0,
    "vatRate": 0,
    "vat": 0,
    "exciseRate": 0,
    "excise": 0,
    "quantityInRegisterUnit": 0,
    "supplierItemCode": "string",
    "supplierItemName": "string",
    "supplierItemUnit": "string",
    "oA_AccountsID": 0,
    "registerParentAccount": "string",
    "supplierItemVATType": 0,
    "costCentre": "string",
    "oA_CostCentreID": 0,
    "oA_AccountsIDExpenseAccount": 0,
    "itemWriteOffExpenseAccount": "string"
  }
]
}

```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change.

"**token**": Set [token] string from [LogIn] response value

"**oA_TransactionsID**":

"**costCentreName**": See [CostCentresList] method (use [Name] value).

"**currency**": See [CurrenciesList] method (use [Currency] value).

"**documentComment**":

"**documentDate**": Sets operation date.

"**documentNumber**":

"**dueDate**":

"**expenseAccountForFixedAssetWriteOff**": See [AccountsList] method (use [Account] value).

"**expenseForWriteOffAccount**": See [AccountsList] method (use [Account] value)

"**isRateFixed**": Sets a fixed rate flag. If true, then equivalent values in accounting entries of this operation will not be recalculated when the exchange rate changes.

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value).

"**oA_CurrenciesID**": See [CurrenciesList] method (use [OA_CurrenciesID] value).

"**oA_ExpenseAccountForFixedAssetWriteOffID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**oA_ExpenseForWriteOffAccountsID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**oA_ParentAccountOfSupplierID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**oA_PayedExciseAccountsID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**oA_PayedVATAccountID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**oA_PurchasedItemReturnAccountID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**oA_StatusID**": See [StatusesList] method (use [OA_StatusID] value).

"**parentAccountOfSupplier**": See [AccountsList] method (use [Account] value).

"**payedExciseAccount**": See [AccountsList] method (use [Account] value).

"**payedVATAccount**": See [AccountsList] method (use [Account] value).

"**purchasedItemReturnAccount**": See [AccountsList] method (use [Account] value).

"**rate**": Sets the currency rate.

"**receiverAddress**": See [GetDebitorCreditorByID] method (use [Address] value from [DebitorCreditorAddresses]).

"**receiverAddressesID**": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorAddressesID] value from [DebitorCreditorAddresses]).

"**rS_TaxDocument_ID**": Sets Tax Document ID value recieved from revenue service

"**rS_TaxDocument_Number**": Sets Tax Document number recieved from revenue service

"**rS_VAT_ID**": Sets VAT ID recieved from revenue service

"**rS_VAT_Number**": Sets VAT number recieved from revenue service

"**rS_Waybill_ID**": Sets waybill ID recieved from revenue service

"**rS_Waybill_Number**": Sets waybill number recieved from revenue service

"**sellerCode**": See [DebitorCreditorsList] method (use [Code] value).

"**sellerID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value).

"**status**": See [StatusesList] method (use [Name] value).

"**supplierAddress**": See [Owner] method (use [Address] value or [address] value from [OwnerAddresses]).

"**supplierAddressID**": See [Owner] method (use [OA_OwnerAddressesID] value from [OwnerAddresses]).

"**supplierCode**": See [DebitorCreditorsList] method (use [Code] value).

"**supplierID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value).

"**createAccountingEntries**": Creates accounting entries using managerial information. Default=false

"**clearItems**": Clears all items for the operation.

"**returnToSupplierItems**": Set OA_ManagerialEntriesID=0 to add a new Item specification. Specify the OA_ManagerialEntriesID value to update information about Item fields. To delete existing row, specify the value OA_ManagerialEntriesID and set "DeleteRow=true".

"**oA_ManagerialEntriesID**": Specify an existing OA_ManagerialEntriesID to update or remove Item information, or 0 to add a new Item.

"**deleteRow**": If true, deletes Item information for specified OA_ManagerialEntriesID value.

"**code**": See [ItemsList] method (use [Code] value).

"**oA_ItemsID**": See [ItemsList] method (use [OA_ItemsID] value).

"**quantity**":

"**unit**": See [StockUnitsList] method (use [Unit] value).

"**oA_StockUnitsID**": See [StockUnitsList] method (use [OA_StockUnitsID] value).

"**price**":

"**amount**":

"**vatRate**":

"**vat**":

"**exciseRate**":

"**excise**":

"**quantityInRegisterUnit**":

"**supplierItemCode**":

"**supplierItemName**":

"**supplierItemUnit**":

"oA_AccountsID":
 "registerParentAccount": ,
 "supplierItemVATType": Standard = 0, ZeroRate = 1, Untaxed = 2
 "costCentre": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If ommite, gets cost centre from parameters.
 "oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If ommite, gets cost centre from parameters.
 "oA_AccountsIDExpenseAccount":
 "itemWriteOffExpenseAccount":

Examples

1. Update Return To Supplier Change cod - 2.
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 3005,
  "expenseForWriteOffAccount": "7210",
  "oA_ParentAccountOfSupplierID": 1,
  "oA_PayedExciseAccountsID": 2,
  "parentAccountOfSupplier": "1410",
  "purchasedItemReturnAccount": "1610 1",
  "supplierID": 2,
  "createAccountingEntries": true,
  "returnToSupplierItems": [
    {
      "code": "2",
      "oA_ItemsID": 1,
      "quantity": 1,
      "price": 10,
      "oA_AccountsIDExpenseAccount": 1,
      "itemWriteOffExpenseAccount": "7210"
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 3005,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51111,
    "description": "Item does not exist!",
    "comment": "[5] Record is not in table or user have no permissions to read this record.\nOA_TransactionsID [3005]
Entry number [6]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "ReturnToSupplier",
    "entryPoint": "SetItem",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/ReturnToSupplier

Returns OK or Error description Json.

Schema

```
{
```

```
"token": "string",
"oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value
"oA_TransactionsID":

Examples

1. Delete ReturnToSupplier.
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 3005
}
```

Server response Code(200):

```
{
  "id": 3005,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 504",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

GetReturnToSupplierByID

POST/api/GetReturnToSupplierByID

Gets Return To Supplier and its Items by OA_TransactionsID value.

Schema

```
{
  "token": "string",
  "OA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value
"OA_TransactionsID": See [TransactionsList] method (use [OA_TransactionsID] value)

Examples

1. Get Return To Supplier by "oA_TransactionsID" - 3005
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",

```

```
"oA_TransactionsID": 3005
}
```

Server response Code(200):

```
{
  "oA_TransactionsID": 3005,
  "amount": 200,
  "costCentreName": "General cost center",
  "currency": "GEL",
  "debitorCreditorVATType": 0,
  "documentalConfirm": false,
  "documentComment": "",
  "documentDate": "2023-05-10T15:54:44",
  "documentNumber": 1,
  "dueDate": "2023-05-10T16:04:40",
  "excise": 0,
  "expenseAccountForFixedAssetWriteOff": "8290",
  "expenseForWriteOffAccount": "7210",
  "isDueDateNULL": true,
  "isRateFixed": false,
  "oA_CostCentreID": 1,
  "oA_CurrenciesID": 1,
  "oA_ExpenseAccountForFixedAssetWriteOffID": 208,
  "oA_ExpenseForWriteOffAccountsID": 180,
  "oA_ParentAccountOfSupplierID": 119,
  "oA_PayedExciseAccountsID": 135,
  "oA_PayedVATAccountID": 133,
  "oA_PurchasedItemReturnAccountID": 181,
  "oA_StatusID": 0,
  "oA_TransactionTypesID": 25,
  "ownerVATType": 0,
  "payedExciseAccount": "3360",
  "rate": 1,
  "receiver": "",
  "receiverAddress": "",
  "receiverAddressesID": 0,
  "reflectInformationInEntries": false,
  "rS_TaxDocument_ID": 0,
  "rS_TaxDocument_Number": "",
  "rS_VAT_ID": 0,
  "rS_VAT_Number": "",
  "rS_Waybill_ID": 0,
  "rS_Waybill_Number": "",
  "sellerCode": "1",
  "sellerID": 1,
  "sellerName": "Organization",
  "status": "",
  "supplierAddress": "",
  "supplierAddressID": 0,
  "supplierCode": "1",
  "supplierID": 1,
  "supplierName": "1",
  "vat": 30.51,
  "returnToSupplierItems": [
    {
      "oA_ManagerialEntriesID": 3004,
      "oA_TransactionsID": 3005,
      "code": "1",
      "item": "item",
      "oA_ItemsID": 2,
      "quantity": 2,
      "unit": "Piece",
      "oA_StockUnitsID": 1,
      "price": 100,
      "amount": 200,
      "vatRate": 18,
      "vat": 30.51,
    }
  ]
}
```

```

"exciseRate": 0,
"excise": 0,
"quantityInRegisterUnit": 0,
"supplierItemCode": "",
"supplierItemName": "",
"supplierItemUnit": "",
"sequenceNumber": 1,
"oA_EntriesID": 3023,
"oA_AccountsID": 69,
"registerParentAccount": "1610",
"errorPointer": 0,
"itemType": 0,
"itemVATType": 0,
"supplierItemVATType": 0,
"hasADoubleRegistrationUnit": false,
"oA_StockUnitsIDRegisterUnit": 1,
"registerUnit": "Piece",
"costCentre": "",
"oA_CostCentreID": 0,
"oA_AccountsIDExpenseAccount": 0,
"itemWriteOffExpenseAccount": ""
}
]
}

```

Server response Code(400):

```

[
{
"errorNumber": 51051,
"description": "Record does not exist or read prohibited for user!",
"comment": "OA_TransactionsID = 3006",
"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "Transaction",
"entryPoint": "GetProperties",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

ReturnToSuppliersList

POST/api/ReturnToSuppliersList

Gets .Return To Suppliers Returns [Info] and [Result] tables.

Schema

```

{
"token": "string",
"topRecordsCount": 0,
"filter": "string",
"sort": "string",
"pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1.

"sort": Use SQL ORDER BY string with field names.

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get ReturnToSuppliersList.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "filter": "UPPER([ViewOA_ReturnToSupplierOperations].[Currency]) LIKE N'%GEL%'",
  "sort": "[Currency],[Supplier code]",
  "pageNumber": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 1,
      "PageRowCount": 1,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_TransactionsID": 5042,
      "Document date": "2023-05-30T16:01:51",
      "Document number": "2",
      "SellerID": 1018,
      "Seller code": "12345",
      "Seller": "New organization",
      "SupplierID": 1018,
      "Supplier code": "12345",
      "Supplier": 1018,
      "Supplier address": "Al. Kazbegi Ave.",
      "Receiver": "Organization",
      "Receiver address": "Al. Kazbegi Ave.",
      "OA_CostCentreID": 1,
      "Cost centre": "General cost center",
      "Comment": "",
      "OA_TransactionTypesID": 25,
      "OA_ParentAccountOfSupplierID": 56,
      "Parent account of supplier": "1410",
      "OA_PurchasedItemReturnAccountID": 69,
      "Purchased item return account": "1610",
      "OA_ExpenseForWriteOffAccountsID": 180,
      "Expense account for item write off": "7210",
      "OA_ExpenseAccountForFixedAssetWriteOffID": 208,
      "Expense account for fixed asset write off": "8290",
      "OA_PayedVATAccountID": 133,
      "Payed VAT account": "3340",
      "OA_PayedExciseAccountsID": 2,
      "Payed excise account": "2000",
      "OA_CurrenciesID": 1,
      "Currency": "GEL",
      "Rate": 1.00000000,
      "Is rate fixed": false,
      "Delay": false,
      "Documental confirm": false,
      "Amount": 100.00000000,
      "Has entries": false,
      "SupplierAddressID": null,
      "ReceiverAddressesID": null,
      "VAT": 15.25000000,
      "Excise": 0.00000000,
      "RS_Waybill_ID": null,
      "RS_Waybill_Number": null,
      "RS_VAT_ID": null,
    }
  ]
}
```

```
"RS_VAT_Number": null,
"OA_UsersID": 1,
"OA_DepartmentsID": 1,
"Record related binary documents": null,
"Owner VAT type": 0,
"DebitorCreditor VAT type": 0,
"RS_TaxDocument_ID": null,
"RS Tax document number": null,
"Due date": null,
"OA_StatusID": null,
"Status": null,
"Numeric document number": 2,
"NUM_ROW": 1
}
]
}
```

ReturnFromCustomer

POST/api/ReturnFromCustomer

Adds new Return From Customer. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "buyerCode": "string",
  "buyerID": 0,
  "costCentreName": "string",
  "currency": "string",
  "documentComment": "string",
  "documentDate": "2023-05-04T11:31:57.987Z",
  "documentNumber": "string",
  "dueDate": "2023-05-04T11:31:57.987Z",
  "incomeAccountForEqualize": "string",
  "incomeAccountForUntaxed": "string",
  "incomeAccountForZeroVAT": "string",
  "isRateFixed": true,
  "oA_CostCentreID": 0,
  "oA_CurrenciesID": 0,
  "oA_IncomeAccountForEqualizeID": 0,
  "oA_IncomeAccountForUntaxedID": 0,
  "oA_IncomeAccountForZeroVATID": 0,
  "oA_ParentAccountForBuyerID": 0,
  "oA_PayedExciseAccountID": 0,
  "oA_PayedVATAccountID": 0,
  "oA_ReturnedGoodLossAccountID": 0,
  "oA_StatusID": 0,
  "parentAccountForBuyer": "string",
  "payedExciseAccount": "string",
  "payedVATAccount": "string",
  "rate": 0,
  "receiverAddress": "string",
  "receiverAddressID": 0,
  "receiverCode": "string",
  "receiverID": 0,
  "returnedGoodLossAccount": "string",
  "rS_TaxDocument_ID": 0,
  "rS_TaxDocument_Number": "string",
  "rS_VAT_ID": 0,
  "rS_VAT_Number": "string",
  "rS_Waybill_ID": 0,
  "rS_Waybill_Number": "string",
  "status": "string",
  "supplierAddress": "string",
```

```

"supplierAddressesID": 0,
"createAccountingEntries": true,
"returnFromCustomerItems": [
  {
    "code": "string",
    "oA_ItemsID": 0,
    "quantity": 0,
    "unit": "string",
    "oA_StockUnitsID": 0,
    "price": 0,
    "amount": 0,
    "vatRate": 0,
    "vat": 0,
    "exciseRate": 0,
    "excise": 0,
    "quantityInRegisterUnit": 0,
    "oA_AccountsID": 0,
    "registerParentAccount": "string",
    "rsItemCode": "string",
    "rsItemName": "string",
    "rsItemUnit": "string",
    "rsItemVATType": 0,
    "costCentre": "string",
    "oA_CostCentreID": 0
  }
]
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"buyerCode": See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [BuyerID] property specified.

"buyerID": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [BuyerCode] property specified.

"costCentreName": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"currency": See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified. If omitted, gets first register currency from parameters.

"documentComment":

"documentDate": Sets operation date. If not specified - sets current date.

"documentNumber": Sets the transaction document number. If not specified, sets a unique number in the current year.

"dueDate":

"incomeAccountForEqualize": See [AccountsList] method (use [Account] value). Can be omitted if [OA_IncomeAccountForEqualizeID] property specified. If both are omitted, gets the account from the parameters.

"incomeAccountForUntaxed": See [AccountsList] method (use [Account] value). Can be omitted if [OA_IncomeAccountForUntaxedID] property specified. If both are omitted, gets the account from the parameters.

"incomeAccountForZeroVAT": See [AccountsList] method (use [Account] value). Can be omitted if [OA_IncomeAccountForZeroVATID] property specified. If both are omitted, gets the account from the parameters.

"isRateFixed": Sets a fixed rate flag. If true, then equivalent values in accounting entries of this operation will not be recalculated when the exchange rate changes.

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreName] property specified. If omitted, gets cost centre from parameters.

"oA_CurrenciesID": See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currency] property specified. If omitted, gets first register currency from parameters.

"oA_IncomeAccountForEqualizeID": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [IncomeAccountForEqualize] property specified. If both are omitted, gets the account from the parameters.

"oA_IncomeAccountForUntaxedID": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [IncomeAccountForUntaxed] property specified. If both are omitted, gets the account from the parameters.

"oA_IncomeAccountForZeroVATID": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [IncomeAccountForZeroVAT] property specified. If both are omitted, gets the account from the parameters.

"oA_ParentAccountForBuyerID": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [IncomeAccountForZeroVAT] property specified. If both are omitted, gets the account from the parameters.

"oA_PayedExciseAccountID": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [PayedExciseAccount] property specified. If both are omitted, gets the account from the parameters.

"oA_PayedVATAccountID": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [PayedVATAccount] property specified. If both are omitted, gets the account from the parameters.

"oA_ReturnedGoodLossAccountID": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [ReturnedGoodLossAccount] property specified. If both are omitted, gets the account from the parameters.

"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified.

"parentAccountForBuyer": See [AccountsList] method (use [Account] value). Can be omitted if [OA_ParentAccountForBuyerID] property specified. If both are omitted, gets the account from the parameters.

"payedExciseAccount": See [AccountsList] method (use [Account] value). Can be omitted if [OA_PayedExciseAccountID] property specified. If both are omitted, gets the account from the parameters.

"payedVATAccount": See [AccountsList] method (use [Account] value). Can be omitted if [OA_PayedVATAccountID] property specified. If both are omitted, gets the account from the parameters.

"rate": Sets the currency rate. If not specified, gets the rate by [DocumentDate]. See [CurrencyRatesList]

"receiverAddress": See [GetDebitorCreditorByID] method (use [Address] value from [DebitorCreditorAddresses]).

"receiverAddressID": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorAddressesID] value from [DebitorCreditorAddresses]).

"receiverCode": See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [ReceiverID] property specified.

"receiverID": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [ReceiverCode] property specified.

"returnedGoodLossAccount": See [AccountsList] method (use [Account] value). Can be omitted if [OA_ReturnedGoodLossAccountID] property specified. If both are omitted, gets the account from the parameters.

"rS_TaxDocument_ID": Sets Tax Document ID value recieved from revenue service

"rS_TaxDocument_Number": Sets Tax Document number recieved from revenue service

"rS_VAT_ID": Sets VAT ID recieved from revenue service

"rS_VAT_Number": Sets VAT number recieved from revenue service

"rS_Waybill_ID": Sets waybill ID recieved from revenue service

"rS_Waybill_Number": Sets waybill number recieved from revenue service

"status": See [StatusesList] method (use [Name] value). Can be omitted if [OA_StatusID] property specified.

"supplierAddress": See [Owner] method (use [Address] value or [address] value from [OwnerAddresses]).

"supplierAddressesID": See [Owner] method (use [OA_OwnerAddressesID] value from [OwnerAddresses]).

"createAccountingEntries": Creates accounting entries using managerial information. Default=false

"returnFromCustomerItems": ReturnFromCustomerItemAdd

"code": See [ItemsList] method (use [Code] value). Can be omitted if [OA_ItemsID] property specified.

"oA_ItemsID": See [ItemsList] method (use [OA_ItemsID] value). Can be omitted if [Code] property specified.

"quantity":

"unit": See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified. If both are omitted, gets the Item register minimum unit.

"oA_StockUnitsID": See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified. If both are omitted, gets the OA_StockUnitsID for Item register minimum unit.

"price": If omitted, calculated as [Amount]/[Quantity]

"amount": If omitted, calculated as [Quantity]*[Price]

"vatRate": If not specified, depends on [vatType] values of [Item], [Buyer] and [owner] (See [GetItemByID], [GetDebitorCreditorByID] and [Owner] methods)

"vat": If omitted, calculated as [Amount]*[VATRate]/100

"exciseRate": If not specified, gets from [exciseRate] value of [Item] object (See [GetItemByID] method)

"excise": If omitted, calculated as [Quantity]*[ExciseRate]

"quantityInRegisterUnit": Required if [hasDoubleRegisterUnit] is true for the [Item] object

"oA_AccountsID": If not specified, gets from [oA_AccountsID] value of [Item] object (See [GetItemByID] method)

"registerParentAccount": If not specified, gets account by [OA_AccountsID] of [Item] object (See [GetItemByID] method)

"rsItemCode":

"rsItemName":

"rsItemUnit":

"rsItemVATType": Standard = 0, ZeroRate = 1, Untaxed = 2

"costCentre": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters.

Examples

1. Add new Return From Customer.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "buyerCode": "1",
  "buyerID": 1,
  "parentAccountForBuyer": "1410",
  "payedExciseAccount": "3110",
  "createAccountingEntries": true,
  "returnFromCustomerItems": [
    {
      "code": "1",
      "oA_ItemsID": 1,
      "price": 100,
      "amount": 100,
      "registerParentAccount": "1610"
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 3004,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51190,
    "description": "Property not found!",
    "comment": "OA_TransactionsID=0 [5]\n Buyer=[11]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "ReturnFromCustomer",
    "entryPoint": "set property",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/ReturnFromCustomer

Updates Return From Customer. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "buyerCode": "string",
  "buyerID": 0,
  "costCentreName": "string",
  "currency": "string",
  "documentComment": "string",
  "documentDate": "2023-05-04T12:35:40.594Z",
  "documentNumber": "string",
  "dueDate": "2023-05-04T12:35:40.594Z",
  "incomeAccountForEqualize": "string",
  "incomeAccountForUntaxed": "string",
  "incomeAccountForZeroVAT": "string",
  "isRateFixed": true,
}
```

```

"oA_CostCentreID": 0,
"oA_CurrenciesID": 0,
"oA_IncomeAccountForEqualizeID": 0,
"oA_IncomeAccountForUntaxedID": 0,
"oA_IncomeAccountForZeroVATID": 0,
"oA_ParentAccountForBuyerID": 0,
"oA_PayedExciseAccountID": 0,
"oA_PayedVATAccountID": 0,
"oA_ReturnedGoodLossAccountID": 0,
"oA_StatusID": 0,
"parentAccountForBuyer": "string",
"payedExciseAccount": "string",
"payedVATAccount": "string",
"rate": 0,
"receiverAddress": "string",
"receiverAddressID": 0,
"receiverCode": "string",
"receiverID": 0,
"returnedGoodLossAccount": "string",
"rS_TaxDocument_ID": 0,
"rS_TaxDocument_Number": "string",
"rS_VAT_ID": 0,
"rS_VAT_Number": "string",
"rS_Waybill_ID": 0,
"rS_Waybill_Number": "string",
"status": "string",
"supplierAddress": "string",
"supplierAddressesID": 0,
"createAccountingEntries": true,
"clearItems": true,
"returnFromCustomerItems": [
  {
    "oA_ManualEntriesID": 0,
    "deleteRow": true,
    "code": "string",
    "oA_ItemsID": 0,
    "quantity": 0,
    "unit": "string",
    "oA_StockUnitsID": 0,
    "price": 0,
    "amount": 0,
    "vatRate": 0,
    "vat": 0,
    "exciseRate": 0,
    "excise": 0,
    "quantityInRegisterUnit": 0,
    "oA_AccountsID": 0,
    "registerParentAccount": "string",
    "rsItemCode": "string",
    "rsItemName": "string",
    "rsItemUnit": "string",
    "rsItemVATType": 0,
    "costCentre": "string",
    "oA_CostCentreID": 0
  }
]
}

```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change.

"**token**": Set [token] string from [LogIn] response value

"**oA_TransactionsID**":

"**buyerCode**": See [DebitorCreditorsList] method (use [Code] value).

"**buyerID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value).

"**costCentreName**": See [CostCentresList] method (use [Name] value).

"**currency**": See [CurrenciesList] method (use [Currency] value).

"**documentComment**":

"**documentDate**": Sets operation date.

"**documentNumber**":

"**dueDate**":

"**incomeAccountForEqualize**": See [AccountsList] method (use [Account] value).

"**incomeAccountForUntaxed**": See [AccountsList] method (use [Account] value).

"**incomeAccountForZeroVAT**": See [AccountsList] method (use [Account] value).

"**isRateFixed**": Sets a fixed rate flag. If true, then equivalent values in accounting entries of this operation will not be recalculated when the exchange rate changes.

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value).

"**oA_CurrenciesID**": See [CurrenciesList] method (use [OA_CurrenciesID] value).

"**oA_IncomeAccountForEqualizeID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**oA_IncomeAccountForUntaxedID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**oA_IncomeAccountForZeroVATID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**oA_ParentAccountForBuyerID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**oA_PayedExciseAccountID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**oA_PayedVATAccountID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**oA_ReturnedGoodLossAccountID**": ID from chart of accounts. See [AccountsList] method (use [OA_AccountsID] value).

"**oA_StatusID**": See [StatusesList] method (use [OA_StatusID] value).

"**parentAccountForBuyer**": See [AccountsList] method (use [Account] value).

"**payedExciseAccount**": See [AccountsList] method (use [Account] value).

"**payedVATAccount**": See [AccountsList] method (use [Account] value).

"**rate**": Sets the currency rate.

"**receiverAddress**": See [GetDebitorCreditorByID] method (use [Address] value from [DebitorCreditorAddresses]).

"**receiverAddressID**": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorAddressesID] value from [DebitorCreditorAddresses]).

"**receiverCode**": See [DebitorCreditorsList] method (use [Code] value).

"**receiverID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value).

"**returnedGoodLossAccount**": See [AccountsList] method (use [Account] value).

"**rS_TaxDocument_ID**": Sets Tax Document ID value recieved from revenue service

"**rS_TaxDocument_Number**": Sets Tax Document number recieved from revenue service

"**rS_VAT_ID**": Sets VAT ID recieved from revenue service

"**rS_VAT_Number**": Sets VAT number recieved from revenue service

"**rS_Waybill_ID**": Sets waybill ID recieved from revenue service

"**rS_Waybill_Number**": Sets waybill number recieved from revenue service

"**status**": See [StatusesList] method (use [Name] value).

"**supplierAddress**": See [Owner] method (use [Address] value or [address] value from [OwnerAddresses]).

"**supplierAddressesID**": See [Owner] method (use [OA_OwnerAddressesID] value from [OwnerAddresses]).

"**createAccountingEntries**": Creates accounting entries using managerial information. Default=false

"**clearItems**": Clears all items for the operation.

"**returnFromCustomerItems**": Set OA_ManagerialEntriesID=0 to add a new Item specification. Specify the OA_ManagerialEntriesID value to update information about Item fields. To delete existing row, specify the value OA_ManagerialEntriesID and set "DeleteRow"=true.

"**oA_ManagerialEntriesID**": Specify an existing OA_ManagerialEntriesID to update or remove Item information, or 0 to add a new Item.

"**deleteRow**": If true, deletes Item information for specified OA_ManagerialEntriesID value.

"**code**": See [ItemsList] method (use [Code] value).

"**oA_ItemsID**": See [ItemsList] method (use [OA_ItemsID] value).

"**quantity**":

"**unit**": See [StockUnitsList] method (use [Unit] value).

"**oA_StockUnitsID**": See [StockUnitsList] method (use [OA_StockUnitsID] value).

"**price**":

"**amount**":

"**vatRate**":

"**vat**":

"**exciseRate**":

"**excise**":

"**quantityInRegisterUnit**":

"**oA_AccountsID**":

"**registerParentAccount**":

"rsItemCode":

"rsItemName":

"rsItemUnit":

"rsItemVATType": Standard = 0, ZeroRate = 1, Untaxed = 2

"costCentre": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters.

Examples

1. Update Return From Customer.Change "registerParentAccount" - 1620.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 3004,
  "buyerCode": "1",
  "buyerID": 1,
  "parentAccountForBuyer": "1410",
  "payedExciseAccount": "3110",
  "createAccountingEntries": true,
  "returnFromCustomerItems": [
    {
      "code": "1",
      "oA_ItemsID": 1,
      "price": 100,
      "amount": 100,
      "registerParentAccount": "1620"
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 3004,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51190,
    "description": "Property not found!",
    "comment": "OA_TransactionsID=3004 [2]\n Buyer=[11]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "ReturnFromCustomer",
    "entryPoint": "set property",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/ReturnFromCustomer

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

Examples

1. Delete ReturnFromCustomer
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 3004
}
```

Server response Code(200):

```
{
  "id": 3004,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 503",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Transaction",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

GetReturnFromCustomerByID

POST/api/GetReturnFromCustomerByID

Gets Return From Customer and its Items by OA_TransactionsID value.

Schema

```
{
  "token": "string",
  "OA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"OA_TransactionsID": See [TransactionsList] method (use [OA_TransactionsID] value).

Examples

1. Gets Return From Customer by OA_TransactionsID-3004
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 3004
}
```

Server response Code(200):

```
{
```

```
"oA_TransactionsID": 3004,
"amount": 100,
"buyerCode": "1",
"buyerID": 1,
"buyerName": "Organization",
"costCentreName": "General cost center",
"currency": "GEL",
"debitorCreditorVATType": 0,
"documentalConfirm": false,
"documentComment": "",
"documentDate": "2023-05-10T15:11:05",
"documentNumber": 1,
"dueDate": "2023-05-10T15:45:13",
"excise": 0,
"incomeAccountForUntaxed": "6110 3",
"incomeAccountForZeroVAT": "6110 2",
"isDueDateNULL": true,
"isRateFixed": false,
"oA_CostCentreID": 1,
"oA_CurrenciesID": 1,
"oA_IncomeAccountForEqualizeID": 211,
"oA_IncomeAccountForUntaxedID": 213,
"oA_IncomeAccountForZeroVATID": 212,
"oA_ParentAccountForBuyerID": 56,
"oA_PayedExciseAccountID": 135,
"oA_PayedVATAccountID": 133,
"oA_ReturnedGoodLossAccountID": 181,
"oA_StatusID": 0,
"oA_TransactionTypesID": 26,
"ownerVATType": 0,
"rate": 1,
"receiverAddress": "",
"receiverAddressID": 0,
"receiverCode": "1",
"receiverID": 1,
"receiverName": "Organization",
"reflectInformationInEntries": false,
"rS_TaxDocument_ID": 0,
"rS_TaxDocument_Number": "",
"rS_VAT_ID": 0,
"rS_VAT_Number": "",
"rS_Waybill_ID": 0,
"rS_Waybill_Number": "",
"status": "",
"supplier": "",
"supplierAddress": "",
"supplierAddressesID": 0,
"supplierID": 0,
"vat": 15.25,
"returnFromCustomerItems": [
  {
    "oA_ManagerialEntriesID": 3003,
    "oA_TransactionsID": 3004,
    "code": "1",
    "item": "item",
    "oA_ItemsID": 2,
    "quantity": 1,
    "unit": "Piece",
    "oA_StockUnitsID": 1,
    "price": 100,
    "amount": 100,
    "vatRate": 18,
    "vat": 15.25,
    "exciseRate": 0,
    "excise": 0,
    "quantityInRegisterUnit": 0,
    "itemType": 0,
  }
]
```

```

    "unitPrimeCost": 0,
    "primeCost": 0,
    "oA_EntriesID": 3018,
    "oA_AccountsID": 69,
    "registerParentAccount": "1610",
    "rsItemCode": "",
    "rsItemName": "",
    "rsItemUnit": "",
    "itemVATType": 0,
    "rsItemVATType": 0,
    "hasADoubleRegistrationUnit": false,
    "oA_StockUnitsIDRegisterUnit": 1,
    "registerUnit": "Piece",
    "costCentre": "",
    "oA_CostCentreID": 0,
    "sequenceNumber": 1
  }
]
}

```

Server response Code(400):

```

[
{
  "errorNumber": 51051,
  "description": "Record does not exist or read prohibited for user!",
  "comment": "OA_TransactionsID = 3005",
  "command": "",
  "errorType": "ErrorType_Error",
  "source": "OA_BusinessObject.dll",
  "class": "Transaction",
  "entryPoint": "GetProperties",
  "resourceDLL_ForTranslation": "ORIS_Accounting",
  "errorPointer": 0
}
]

```

ReturnFromCustomersList

POST/api/ReturnFromCustomersList

Gets Return From Customers. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

"**token**": Set [token] string from [LogIn] response value

"**topRecordsCount**": The number of records to accept. If you specify 0 you will get all records. Default=0

"**filter**": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1.

"**sort**": Use SQL ORDER BY string with field names.

"**pageNumber**": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get list of ReturnFromCustomersList.

Request body:

```

{

```

```
"token": "... paste here [LogIn] response value ...",
"topRecordsCount":1 ,
"filter": "[ViewOA_ReturnFromCustomerOperations].[OA_TransactionsID]=3004",
"sort": "[Buyer code],[Buyer name]",
"pageNumber": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 1,
      "PageRowCount": 1,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_TransactionsID": 3004,
      "Document date": "2023-05-29T18:00:02",
      "ReceiverID": 1,
      "Amount": 0.00000000,
      "SupplierID": null,
      "Supplier": "Organization",
      "Supplier address": "",
      "OA_TransactionTypesID": 26,
      "Comment": "",
      "Document number": "3",
      "BuyerID": 1,
      "Buyer code": "1",
      "Buyer name": "Organization",
      "OA_ParentAccountForBuyerID": 58,
      "Parent account for a buyer": "1420",
      "OA_IncomeAccountForEqualizeID": 211,
      "Income account for equalize": "6110 1",
      "OA_PayedVATAccountID": 133,
      "Payed VAT account": "3340",
      "OA_PayableExciseAccountsID": 120,
      "Payed excise account": "3120",
      "Receiver code": "1",
      "Receiver name": "Organization",
      "Receiver address": "",
      "Delay": false,
      "Documental confirm": false,
      "Has entries": false,
      "ReceiverAddressID": null,
      "SupplierAddressID": null,
      "VAT": 0.00000000,
      "Excise": 0.00000000,
      "RS_Waybill_ID": null,
      "RS Waybill number": null,
      "RS_VAT_ID": null,
      "RS VAT number": null,
      "OA_ReturnedGoodLossAccountID": 181,
      "Returned good loss account": "7220",
      "OA_UsersID": 1,
      "OA_DepartmentsID": 1,
      "User": "sa",
      "Department": "Administrators",
      "OA_CurrenciesID": 1,
      "Currency": "GEL",
      "Rate": 1.00000000,
      "Is rate fixed": false,
      "Cost centre": "General cost center",
      "OA_CostCentreID": 1,
      "Record related binary documents": null,
    }
  ]
}
```



```

"Owner VAT type": 0,
"DebitorCreditor VAT type": 0,
"RS_TaxDocument_ID": null,
"RS Tax document number": null,
"OA_IncomeAccountForZeroVATID": 212,
"Incomes account for zero VAT return": "6110 2",
"OA_IncomeAccountForUntaxedID": 213,
"Incomes account for untaxed return": "6110 3",
"Due date": null,
"OA_StatusID": null,
>Status": null,
"Numeric document number": 3,
"NUM_ROW": 1
}
]
}

```

OrderToSupplier

POST/api/OrderToSupplier

Adds new Order To Supplier. Returns ID value or Error description Json.

Schema

```

{
  "token": "string",
  "costCentreName": "string",
  "currency": "string",
  "documentComment": "string",
  "documentDate": "2023-05-03T13:32:28.865Z",
  "documentNumber": "string",
  "dueDate": "2023-05-03T13:32:28.865Z",
  "isRateFixed": true,
  "oA_CostCentreID": 0,
  "oA_CurrenciesID": 0,
  "oA_StatusID": 0,
  "rate": 0,
  "receiverAddress": "string",
  "receiverAddressesID": 0,
  "sellerCode": "string",
  "sellerID": 0,
  "status": "string",
  "supplierAddress": "string",
  "supplierAddressID": 0,
  "supplierCode": "string",
  "supplierID": 0,
  "orderToSupplierItems": [
    {
      "code": "string",
      "oA_ItemsID": 0,
      "quantity": 0,
      "unit": "string",
      "oA_StockUnitsID": 0,
      "price": 0,
      "amount": 0,
      "vatRate": 0,
      "vat": 0,
      "exciseRate": 0,
      "excise": 0,
      "quantityInRegisterUnit": 0,
      "supplierItemCode": "string",
      "supplierItemName": "string",
      "supplierItemUnit": "string",
      "supplierItemVATType": 0,
      "oA_AccountsID": 0,
    }
  ]
}

```

```

    "registerParentAccount": "string",
    "costCentre": "string",
    "oA_CostCentreID": 0
  }
]
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"costCentreName": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"currency": See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified. If omitted, gets first register currency from parameters.

"documentComment":

"documentDate": Sets operation date. If not specified - sets current date.

"documentNumber": Sets the transaction document number. If not specified, sets a unique number in the current year.

"dueDate":

"isRateFixed": Sets a fixed rate flag. If true, then equivalent values in accounting entries of this operation will not be recalculated when the exchange rate changes.

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreName] property specified. If omitted, gets cost centre from parameters.

"oA_CurrenciesID": See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currency] property specified. If omitted, gets first register currency from parameters.

"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified.

"rate": Sets the currency rate. If not specified, gets the rate by [DocumentDate]. See [CurrencyRatesList]

"receiverAddress": See [Owner] method (use [Address] value or [Address] from from [OwnerAddresses]).

"receiverAddressesID": See [Owner] method (use [OA_OwnerAddressesID] value from from [OwnerAddresses]).

"sellerCode": See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [SellerID] property specified.

"sellerID": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [SellerCode] property specified.

"status": See [StatusesList] method (use [Name] value). Can be omitted if [OA_StatusID] property specified.

"supplierAddress": See [GetDebitorCreditorByID] method (use [Address] value from [DebitorCreditorAddresses]).

"supplierAddressID": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorAddressesID] value from [DebitorCreditorAddresses]).

"supplierCode": See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [SupplierID] property specified.

"supplierID": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [SupplierCode] property specified.

"orderToSupplierItems": OrderToSupplierItemAdd

"code": See [ItemsList] method (use [Code] value). Can be omitted if [OA_ItemsID] property specified.

"oA_ItemsID": See [ItemsList] method (use [OA_ItemsID] value). Can be omitted if [Code] property specified.

"quantity":

"unit": See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified. If both are omitted, gets the Item register minimum unit.

"oA_StockUnitsID": See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified. If both are omitted, gets the OA_StockUnitsID for Item register minimum unit.

"price": If omitted, calculated as [Amount]/[Quantity]

"amount": If omitted, calculated as [Quantity]*[Price]

"vatRate": If not specified, depends on [vatType] values of [Item], [Seller] and [owner] (See [GetItemByID], [GetDebitorCreditorByID] and [Owner] methods)

"vat": If omitted, calculated as [Amount]*[VATRate]/100

"exciseRate": If not specified, gets from [exciseRate] value of [Item] object (See [GetItemByID] method)

"excise": If omitted, calculated as [Quantity]*[ExciseRate]

"quantityInRegisterUnit": Required if [hasDoubleRegisterUnit] is true for the [Item] object

"supplierItemCode":

"supplierItemName":

"supplierItemUnit":

"supplierItemVATType":

"oA_AccountsID": If not specified, gets from [oA_AccountsID] value of [Item] object (See [GetItemByID] method)

"registerParentAccount": If not specified, gets account by [OA_AccountsID] of [Item] object (See [GetItemByID] method)

"costCentre": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters.

Examples

1. Adds new Order To Supplier
Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "costCentreName": "General cost center",
  "currency": "GEL",
  "oA_CostCentreID": 1,
  "receiverAddressesID": 1,
  "sellerID": 1,
  "orderToSupplierItems": [
    {
      "code": "1",
      "oA_ItemsID": 1,
      "quantity": 5,
      "unit": "Piece",
      "oA_StockUnitsID": 0,
      "registerParentAccount": "1610",
      "costCentre": "General cost center",
      "oA_CostCentreID": 1
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 2011,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51190,
    "description": "Property not found!",
    "comment": "OA_TransactionsID=0 [3]\n[SellerID]=12",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "OrdersToSuppliers",
    "entryPoint": "set property",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/OrderToSupplier

Updates Order To Supplier. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "costCentreName": "string",
  "currency": "string",
  "documentComment": "string",
  "documentDate": "2023-05-04T08:01:40.961Z",
  "documentNumber": "string",
  "dueDate": "2023-05-04T08:01:40.961Z",
  "isRateFixed": true,
}
```

```

"oA_CostCentreID": 0,
"oA_CurrenciesID": 0,
"oA_StatusID": 0,
"rate": 0,
"receiverAddress": "string",
"receiverAddressesID": 0,
"sellerCode": "string",
"sellerID": 0,
"status": "string",
"supplierAddress": "string",
"supplierAddressID": 0,
"supplierCode": "string",
"supplierID": 0,
"clearItems": true,
"orderToSupplierItems": [
  {
    "oA_ManualEntriesID": 0,
    "deleteRow": true,
    "code": "string",
    "oA_ItemsID": 0,
    "quantity": 0,
    "unit": "string",
    "oA_StockUnitsID": 0,
    "price": 0,
    "amount": 0,
    "vatRate": 0,
    "vat": 0,
    "exciseRate": 0,
    "excise": 0,
    "quantityInRegisterUnit": 0,
    "supplierItemCode": "string",
    "supplierItemName": "string",
    "supplierItemUnit": "string",
    "supplierItemVATType": 0,
    "oA_AccountsID": 0,
    "registerParentAccount": "string",
    "costCentre": "string",
    "oA_CostCentreID": 0
  }
]
}

```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change.

"**token**": Set [token] string from [LogIn] response value

"**oA_TransactionsID**":

"**costCentreName**": See [CostCentresList] method (use [Name] value)

"**currency**": See [CurrenciesList] method (use [Currency] value).

"**documentComment**":

"**documentDate**": Sets operation date

"**documentNumber**":

"**dueDate**":

"**isRateFixed**": Sets a fixed rate flag. If true, then equivalent values in accounting entries of this operation will not be recalculated when the exchange rate changes.

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value)

"**oA_CurrenciesID**": See [CurrenciesList] method (use [OA_CurrenciesID] value)

"**oA_StatusID**": See [StatusesList] method (use [OA_StatusID] value).

"**rate**": Sets the currency rate.

"**receiverAddress**": See [Owner] method (use [Address] value or [Address] from [OwnerAddresses])

"**receiverAddressesID**": See [Owner] method (use [OA_OwnerAddressesID] value from [OwnerAddresses])

"**sellerCode**": See [DebitorCreditorsList] method (use [Code] value)

"**sellerID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value)

"**status**":

"**supplierAddress**": See [GetDebitorCreditorByID] method (use [Address] value from [DebitorCreditorAddresses])

"**supplierAddressID**": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorAddressesID] value from [DebitorCreditorAddresses])

"**supplierCode**": See [DebitorCreditorsList] method (use [Code] value)

"**supplierID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value).

"**clearItems**": Clears all items for the operation.

"**orderToSupplierItems**": Set OA_ManagerialEntriesID=0 to add a new Item specification. Specify the OA_ManagerialEntriesID value to update information about Item fields. To delete existing row, specify the value OA_ManagerialEntriesID and set "DeleteRow"=true.

"**oA_ManagerialEntriesID**": Specify an existing OA_ManagerialEntriesID to update or remove Item information, or 0 to add a new Item.

"**deleteRow**": If true, deletes Item information for specified OA_ManagerialEntriesID value.

"**code**": See [ItemsList] method (use [Code] value).

"**oA_ItemsID**": See [ItemsList] method (use [OA_ItemsID] value)

"**quantity**": 0,

"**unit**": See [StockUnitsList] method (use [Unit] value)

"**oA_StockUnitsID**": See [StockUnitsList] method (use [OA_StockUnitsID] value).

"**price**":

"**amount**":

"**vatRate**":

"**vat**":

"**exciseRate**":

"**excise**":

"**quantityInRegisterUnit**":

"**supplierItemCode**":

"**supplierItemName**":

"**supplierItemUnit**":

"**supplierItemVATType**":

"**oA_AccountsID**":

"**registerParentAccount**":

"**costCentre**": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters.

Examples

1. Update Order To Supplier

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 2011,
  "costCentreName": "New cost center",
  "oA_CostCentreID": 2
}
```

Server response Code(200):

```
{
  "id": 2011,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51190,
    "description": "Property not found!",
    "comment": "OA_TransactionsID=2011 [1]\n[OA_CostCentreID]=3",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "OrdersToSuppliers",
    "entryPoint": "set property",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/OrderToSupplier
Returns OK or Error description Json.

Schema

```
{  
  "token": "string",  
  "oA_TransactionsID": 0  
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value
"oA_TransactionsID":

Examples

1. Delete Order To Supplier

Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "oA_TransactionsID": 2011  
}
```

Server response Code(200):

```
{  
  "id": 2011,  
  "idFieldName": "OA_TransactionsID"  
}
```

Server response Code(400):

```
[  
  {  
    "errorNumber": 51051,  
    "description": "Record does not exist or read prohibited for user!",  
    "comment": "OA_TransactionsID = 222",  
    "command": "",  
    "errorType": "ErrorType_Error",  
    "source": "OA_BusinessObject.dll",  
    "class": "OrdersToSuppliers",  
    "entryPoint": "GetProperties",  
    "resourceDLL_ForTranslation": "ORIS_Accounting",  
    "errorPointer": 0  
  }  
]
```

GetOrderToSupplierById

POST/api/GetOrderToSupplierById

Gets Order To Supplier and its Items by OA_TransactionsID value.

Schema

```
{  
  "token": "string",  
  "oA_TransactionsID": 0  
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"OA_TransactionsID": See [TransactionsList] method (use [OA_TransactionsID] value)

Examples

1. Gets Order To Supplier and its Items by OA_TransactionsID =2014

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 2014
}
```

Server response Code(200):

```
{
  "oA_TransactionsID": 2014,
  "amount": 100,
  "costCentreName": "",
  "currency": "",
  "debitorCreditorVATType": 0,
  "documentComment": "",
  "documentDate": "2023-05-10T17:52:20",
  "documentNumber": 1,
  "dueDate": "2023-05-10T00:00:00",
  "excise": 0,
  "isDueDateNULL": true,
  "isRateFixed": false,
  "oA_CostCentreID": 1,
  "oA_CurrenciesID": 1,
  "oA_StatusID": 0,
  "oA_TransactionTypesID": 41,
  "ownerVATType": 0,
  "rate": 0,
  "receiver": "",
  "receiverAddress": "",
  "receiverAddressesID": 0,
  "sellerCode": "",
  "sellerID": 0,
  "sellerName": "",
  "status": "",
  "supplierAddress": "",
  "supplierAddressID": 0,
  "supplierCode": "",
  "supplierID": 0,
  "supplierName": "",
  "vat": 15.25,
  "orderToSupplierItems": [
    {
      "oA_ManualEntriesID": 2011,
      "oA_TransactionsID": 2014,
      "code": "1",
      "item": "item",
      "oA_ItemsID": 2,
      "quantity": 1,
      "unit": "Piece",
      "oA_StockUnitsID": 1,
      "price": 100,
      "amount": 100,
      "vatRate": 18,
      "vat": 15.25,
      "exciseRate": 0,
      "excise": 0,
      "quantityInRegisterUnit": 0,
      "supplierItemCode": "",
      "supplierItemName": "",
      "supplierItemUnit": "",
      "itemType": 0,
    }
  ]
}
```

```

    "itemVATType": 0,
    "supplierItemVATType": 0,
    "oA_AccountsID": 0,
    "registerParentAccount": "",
    "hasADoubleRegistrationUnit": false,
    "oA_StockUnitsIDRegisterUnit": 1,
    "registerUnit": "Piece",
    "costCentre": "",
    "oA_CostCentreID": 0,
    "sequenceNumber": 1
  }
]
}

```

Server response Code(400):

```

[
{
  "errorNumber": 51051,
  "description": "Record does not exist or read prohibited for user!",
  "comment": "OA_TransactionsID = 2015",
  "command": "",
  "errorType": "ErrorType_Error",
  "source": "OA_BusinessObject.dll",
  "class": "OrdersToSuppliers",
  "entryPoint": "GetProperties",
  "resourceDLL_ForTranslation": "ORIS_Accounting",
  "errorPointer": 0
}
]

```

OrderToSuppliersList

POST/api/OrderToSuppliersList

Gets Order To Suppliers. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names.

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get list debtor/creditors (organization or person) of Order To Supplier under 1

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "filter": "UPPER([ViewOA_OrdersToSuppliers].[Seller code]) LIKE N'1%'",
  "sort": "[OA_TransactionsID],[Numeric document number]",
  "pageNumber": 1
}

```



```
}
```

Server response Code(200):

```
"Info": [  
  {  
    "RowCount": 1,  
    "PageCount": 1,  
    "PageRowCount": 1,  
    "PageNumber": 1  
  }  
],  
"Result": [  
  {  
    "OA_TransactionsID": 2011,  
    "Document date": "2023-05-10T00:00:00",  
    "Document number": "3",  
    "SellerID": 1,  
    "Seller code": "1",  
    "Seller": "Organization",  
    "SupplierID": 1,  
    "Supplier code": "1",  
    "Supplier": "Organization",  
    "Supplier address": "",  
    "Receiver": "",  
    "Receiver address": "",  
    "OA_CostCentreID": 1,  
    "Cost centre": "General cost center",  
    "Comment": "",  
    "OA_TransactionTypesID": 40,  
    "OA_CurrenciesID": 1,  
    "Currency": "GEL",  
    "Rate": 1.00000000,  
    "Is rate fixed": false,  
    "Amount": 0.00000000,  
    "SupplierAddressID": null,  
    "ReceiverAddressID": null,  
    "VAT": 0.00000000,  
    "Excise": 0.00000000,  
    "OA_UsersID": 1,  
    "OA_DepartmentsID": 1,  
    "Record related binary documents": null,  
    "Owner VAT type": 0,  
    "DebitorCreditor VAT type": 0,  
    "Due date": null,  
    "Linked operations": null,  
    "OA_StatusID": null,  
    "Status": null,  
    "Numeric document number": 3,  
    "NUM_ROW": 1  
  }  
]  
}
```

OrderOfTransfer

POST/api/OrderOfTransfer

Adds new Order Of Transfer. Returns ID value or Error description Json.

Schema

```
{  
  "token": "string",  
  "costCentreFrom": "string",  
  "costCentreTo": "string",  
  "documentComment": "string",  
}
```

```

"documentDate": "2023-05-03T10:50:51.164Z",
"documentNumber": "string",
"oA_CostCentreID": 0,
"oA_SalePriceTypesID": 0,
"oA_StatusID": 0,
"salePriceType": "string",
"status": "string",
"toOA_CostCentreID": 0,
"orderOfTransferItems": [
  {
    "code": "string",
    "oA_ItemsID": 0,
    "quantity": 0,
    "unit": "string",
    "oA_StockUnitsID": 0,
    "price": 0,
    "amount": 0,
    "quantityInRegisterUnit": 0,
    "oA_AccountsID": 0,
    "registerParentAccount": "string",
    "costCentreFrom": "string",
    "oA_CostCentreID": 0,
    "costCentreTo": "string",
    "toOA_CostCentreID": 0
  }
]
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"costCentreFrom": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"costCentreTo": See [CostCentresList] method (use [Name] value). Can be omitted if [ToOA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"documentComment":

"documentDate": Sets operation date. If not specified - sets current date.

"documentNumber": Sets the transaction document number. If not specified, sets a unique number in the current year.

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreFrom] property specified. If omitted, gets cost centre from parameters.

"oA_SalePriceTypesID": See [SalePriceTypesList] method (use [oA_SalePriceTypesID] value). Can be omitted if [SalePriceType] property specified.

"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified.

"salePriceType": See [SalePriceTypesList] method (use [description] value). Can be omitted if [OA_SalePriceTypesID] property specified.

"status": See [StatusesList] method (use [Name] value). Can be omitted if [OA_StatusID] property specified.

"toOA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreTo] property specified. If omitted, gets cost centre from parameters.

"orderOfTransferItems": OrderOfTransferItemAdd

"code": See [ItemsList] method (use [Code] value). Can be omitted if [OA_ItemsID] property specified.

"oA_ItemsID": See [ItemsList] method (use [OA_ItemsID] value). Can be omitted if [Code] property specified.

"quantity":

"unit": See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified. If both are omitted, gets the Item register minimum unit.

"oA_StockUnitsID": See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified. If both are omitted, gets the OA_StockUnitsID for Item register minimum unit.

"price": If omitted, calculated as [Amount]/[Quantity]

"amount": If omitted, calculated as [Quantity]*[Price]

"quantityInRegisterUnit": Required if [hasDoubleRegisterUnit] is true for the [Item] object

"oA_AccountsID": If not specified, gets from [oA_AccountsID] value of [Item] object (See [GetItemByID] method)

"registerParentAccount": If not specified, gets account by [OA_AccountsID] of [Item] object (See [GetItemByID] method)

"costCentreFrom": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreFrom] property specified. If omitted, gets cost centre from parameters.
 "costCentreTo": See [CostCentresList] method (use [Name] value). Can be omitted if [ToOA_CostCentreID] property specified. If omitted, gets cost centre from parameters.
 "toOA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreTo] property specified. If omitted, gets cost centre from parameters.

Examples

1. Add new Order Of Transfer.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "costCentreFrom": "General cost center",
  "costCentreTo": "New cost center",
  "toOA_CostCentreID": 2,
  "orderOfTransferItems": [
    {
      "code": "1",
      "oA_ItemsID": 1,
      "quantity": 1,
      "unit": "Piece",
      "oA_StockUnitsID": 1,
      "oA_AccountsID": 1,
      "registerParentAccount": "1610",
      "costCentreFrom": "General cost center",
      "oA_CostCentreID": 1,
      "costCentreTo": "New cost center",
      "toOA_CostCentreID": 2
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 2013,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51053,
    "description": "OA_CostCentreID does not exist!",
    "comment": "[3] Record is not in table or user have no permissions to read this record.\n Entry number [1]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "OrdersOfTransfers",
    "entryPoint": "SetToOA_CostCentreID",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/OrderOfTransfer

Updates Order Of Transfer. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "costCentreFrom": "string",
  "costCentreTo": "string",
}
```

```

"documentComment": "string",
"documentDate": "2023-05-03T11:40:06.539Z",
"documentNumber": "string",
"oA_CostCentreID": 0,
"oA_SalePriceTypesID": 0,
"oA_StatusID": 0,
"salePriceType": "string",
"status": "string",
"toOA_CostCentreID": 0,
"clearItems": true,
"orderOfTransferItems": [
  {
    "oA_ManagerialEntriesID": 0,
    "deleteRow": true,
    "code": "string",
    "oA_ItemsID": 0,
    "quantity": 0,
    "unit": "string",
    "oA_StockUnitsID": 0,
    "price": 0,
    "amount": 0,
    "quantityInRegisterUnit": 0,
    "oA_AccountsID": 0,
    "registerParentAccount": "string",
    "costCentreFrom": "string",
    "oA_CostCentreID": 0,
    "costCentreTo": "string",
    "toOA_CostCentreID": 0
  }
]
}

```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID": See [CostCentresList] method (use [Name] value).

"costCentreFrom": See [CostCentresList] method (use [Name] value).

"costCentreTo": See [CostCentresList] method (use [Name] value).

"documentComment":

"documentDate": Sets operation date

"documentNumber":

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value).

"oA_SalePriceTypesID": See [SalePriceTypesList] method (use [oA_SalePriceTypesID] value).

"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value).

"salePriceType": See [SalePriceTypesList] method (use [description] value).

"status": See [StatusesList] method (use [Name] value).

"toOA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value).

"clearItems": Clears all items for the operation.

"orderOfTransferItems": Set OA_ManagerialEntriesID=0 to add a new Item specification. Specify the OA_ManagerialEntriesID value to update information about Item fields. To delete existing row, specify the value OA_ManagerialEntriesID and set

"DeleteRow"=true.

"oA_ManagerialEntriesID": Specify an existing OA_ManagerialEntriesID to update or remove Item information, or 0 to add a new Item.

"deleteRow": If true, deletes Item information for specified OA_ManagerialEntriesID value.

"code": See [ItemsList] method (use [Code] value).

"oA_ItemsID": See [ItemsList] method (use [OA_ItemsID] value).

"quantity":

"unit": See [StockUnitsList] method (use [Unit] value).

"oA_StockUnitsID": See [StockUnitsList] method (use [OA_StockUnitsID] value).

"price":

"amount":

"quantityInRegisterUnit":

"oA_AccountsID":

"registerParentAccount":

"costCentreFrom": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreFrom] property specified. If omitted, gets cost centre from parameters.

"costCentreTo": See [CostCentresList] method (use [Name] value). Can be omitted if [ToOA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"toOA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreTo] property specified. If omitted, gets cost centre from parameters.

Examples

1. Updates Order Of Transfer with OA_TransactionsID= 2013.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 2013,
  "costCentreFrom": "General cost center",
  "costCentreTo": "New cost center2",
  "oA_CostCentreID": 1,
  "toOA_CostCentreID": 1002,
  "orderOfTransferItems": [
    {
      "code": "1",
      "oA_ItemsID": 1,
      "quantity": 1,
      "unit": "Piece",
      "oA_StockUnitsID": 1,
      "oA_AccountsID": 1,
      "registerParentAccount": "1610",
      "costCentreFrom": "General cost center",
      "oA_CostCentreID": 1,
      "costCentreTo": "New cost center2",
      "toOA_CostCentreID": 1002
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 2013,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 214",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "OrdersOfTransfers",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/OrderOfTransfer

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

Examples

1. Delete Order Of Transfer with OA_TransactionsID= 2013
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 2013,
}
```

Server response Code(200):

```
{
  "id": 2013,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 213",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "OrdersOfTransfers",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

GetOrderOfTransferByID

POST/api/GetOrderOfTransferByID

Get Order Of Transfer and its Items by OA_TransactionsID value.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"OA_TransactionsID": See [TransactionsList] method (use [OA_TransactionsID] value)

Examples

1. Get Order Of Transfer and its Items by OA_TransactionsID=2013
Request body:

```
{
```

```
"token": "... paste here [LogIn] response value ...",
"oA_TransactionsID": 2013
}
```

Server response Code(200):

```
{
  "oA_TransactionsID": 2013,
  "amount": 0,
  "costCentreFrom": "General cost center",
  "costCentreTo": "General cost center",
  "documentComment": "",
  "documentDate": "2023-05-10T17:48:28",
  "documentNumber": 1,
  "oA_CostCentreID": 1,
  "oA_SalePriceTypesID": 0,
  "oA_StatusID": 0,
  "oA_TransactionTypesID": 42,
  "reflectInformationInEntries": false,
  "salePriceChangingLastReason": 0,
  "salePriceType": null,
  "status": "",
  "toOA_CostCentreID": null,
  "orderOfTransferItems": [
    {
      "oA_ManualEntriesID": 2010,
      "oA_TransactionsID": 2013,
      "code": "1",
      "item": "item",
      "oA_ItemsID": 2,
      "quantity": 2,
      "unit": "Piece",
      "oA_StockUnitsID": 1,
      "price": 0,
      "amount": 0,
      "quantityInRegisterUnit": 0,
      "itemType": 0,
      "errorPointer": 0,
      "oA_EntriesID": 0,
      "oA_AccountsID": 0,
      "registerParentAccount": "",
      "hasADoubleRegistrationUnit": false,
      "registerUnit": "Piece",
      "oA_StockUnitsIDRegisterUnit": 1,
      "costCentreFrom": "",
      "oA_CostCentreID": 0,
      "costCentreTo": "",
      "toOA_CostCentreID": 0,
      "sequenceNumber": 1
    }
  ]
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 2014",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "OrdersOfTransfers",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

]

OrderOfTransfersList

POST/api/OrderOfTransfersList

Gets Order Of Transfers. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [Login] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property: [TopRecordsCount]=1.

"sort": Use SQL ORDER BY string with field names.

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get Order Of Transfers Info and Result tables.

Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "filter": "[ViewOA_OrdersOfTransfers].[OA_TransactionsID]=4008",
  "sort": "[OA_TransactionTypesID],[OA_CostCentreID]"
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_TransactionsID": 4008,
      "Document date": "2023-05-24T12:09:04",
      "OA_TransactionTypesID": 42,
      "Comment": "",
      "Document number": "2",
      "OA_CostCentreID": 1,
      "Cost centre from": "General cost center",
      "Amount": 0.00000000,
      "OA_StatusID": 2,
      "Status": "Confirmed",
      "Numeric document number": 2
    }
  ]
}
```


OrderFromByuer

POST/api/OrderFromByuer

Adds new Order From Byuer. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "buyerCode": "string",
  "buyerID": 0,
  "costCentreName": "string",
  "currency": "string",
  "documentComment": "string",
  "documentDate": "2023-05-03T06:11:08.331Z",
  "documentNumber": "string",
  "dueDate": "2023-05-03T06:11:08.331Z",
  "isRateFixed": true,
  "oA_CostCentreID": 0,
  "oA_CurrenciesID": 0,
  "oA_SalePriceTypesID": 0,
  "oA_StatusID": 0,
  "rate": 0,
  "receiverAddress": "string",
  "receiverAddressesID": 0,
  "receiverCode": "string",
  "receiverID": 0,
  "salePriceType": "string",
  "status": "string",
  "supplierAddress": "string",
  "supplierAddressesID": 0,
  "supplierID": 0,
  "ordersFromByuerItems": [
    {
      "code": "string",
      "oA_ItemsID": 0,
      "quantity": 0,
      "unit": "string",
      "oA_StockUnitsID": 0,
      "price": 0,
      "amount": 0,
      "vatRate": 0,
      "vat": 0,
      "exciseRate": 0,
      "excise": 0,
      "quantityInRegisterUnit": 0,
      "rsItemCode": "string",
      "rsItemName": "string",
      "rsItemUnit": "string",
      "rsItemVATType": 0,
      "itemSalePrice": 0,
      "itemSalePriceOA_CurrenciesID": 0,
      "itemSalePriceCurrency": "string",
      "itemSalePriceOA_StockUnitsID": 0,
      "itemSalePriceUnit": "string",
      "itemSalePriceInSaleCurrencyAndUnit": 0,
      "oA_AccountsID": 0,
      "registerParentAccount": "string",
      "costCentre": "string",
      "oA_CostCentreID": 0
    }
  ]
}
```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"**token**": "string",
"**buyerCode**": See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [BuyerID] property specified.
"**buyerID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [BuyerCode] property specified.
"**costCentreName**": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.
"**currency**": See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified. If omitted, gets first register currency from parameters.
"**documentComment**":
"**documentDate**": Sets operation date. If not specified - sets current date.
"**documentNumber**": Sets the transaction document number. If not specified, sets a unique number in the current year.
"**dueDate**":
"**isRateFixed**": Sets a fixed rate flag. If true, then equivalent values in accounting entries of this operation will not be recalculated when the exchange rate changes.
"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentreName] property specified. If omitted, gets cost centre from parameters.
"**oA_CurrenciesID**": See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currency] property specified. If omitted, gets first register currency from parameters.
"**oA_SalePriceTypesID**": See [SalePriceTypesList] method (use [oA_SalePriceTypesID] value). Can be omitted if [SalePriceType] property specified.
"**oA_StatusID**": See [StatusesList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified.
"**rate**": See [StatusesList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified.
"**receiverAddress**": See [GetDebitorCreditorByID] method (use [Address] value from [DebitorCreditorAddresses]).
"**receiverAddressesID**": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorAddressesID] value from [DebitorCreditorAddresses]).
"**receiverCode**": See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [ReceiverID] property specified.
"**receiverID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [ReceiverCode] property specified.
"**salePriceType**": See [SalePriceTypesList] method (use [description] value). Can be omitted if [OA_SalePriceTypesID] property specified.
"**status**": See [StatusesList] method (use [Name] value). Can be omitted if [OA_StatusID] property specified.
"**supplierAddress**": See [Owner] method (use [Address] value or [Address] from [OwnerAddresses]).
"**supplierAddressesID**": See [Owner] method (use [OA_OwnerAddressesID] value from [OwnerAddresses]).
"**supplierID**":

"**ordersFromBuyerItems**": OrderFromBuyerItemAdd
"**code**": See [ItemsList] method (use [Code] value). Can be omitted if [OA_ItemsID] property specified.
"**oA_ItemsID**": See [ItemsList] method (use [OA_ItemsID] value). Can be omitted if [Code] property specified.
"**quantity**":
"**unit**": See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified. If both are omitted, gets the Item sale unit.
"**oA_StockUnitsID**": See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified. If both are omitted, gets the OA_StockUnitsID for Item sale minimum unit.
"**price**": If omitted, calculated as [Amount]/[Quantity]
"**amount**": If omitted, calculated as [Quantity]*[Price]
"**vatRate**": If not specified, depends on [vatType] values of [Item], [Buyer] and [owner] (See [GetItemByID], [GetDebitorCreditorByID] and [Owner] methods)
"**vat**": If omitted, calculated as [Amount]*[VATRate]/100
"**exciseRate**": If not specified, gets from [exciseRate] value of [Item] object (See [GetItemByID] method)
"**excise**": If omitted, calculated as [Quantity]*[ExciseRate]
"**quantityInRegisterUnit**": Required if [hasDoubleRegisterUnit] is true for the [Item] object
"**rsItemCode**":
"**rsItemName**":
"**rsItemUnit**":
"**rsItemVATType**":
"**itemSalePrice**":
"**itemSalePriceOA_CurrenciesID**":
"**itemSalePriceCurrency**":
"**itemSalePriceOA_StockUnitsID**":
"**itemSalePriceUnit**":
"**itemSalePriceInSaleCurrencyAndUnit**":
"**oA_AccountsID**": If not specified, gets from [OA_AccountsID] value of [Item] object (See [GetItemByID] method)
"**registerParentAccount**": If not specified, gets account by [OA_AccountsID] of [Item] object (See [GetItemByID] method)

"**costCentre**": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If omitted, gets cost centre from parameters.

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If omitted, gets cost centre from parameters.

Examples

1. Add new Order From Buyer.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "buyerCode": "1",
  "buyerID": 1,
  "oA_CostCentreID": 1,
  "oA_CurrenciesID": 1,
  "receiverAddress": "1",
  "receiverAddressesID": 1,
  "receiverCode": "",
  "receiverID": 1,
  "supplierAddressesID": 1,
  "supplierID": 1,
  "ordersFromBuyerItems": [
    {
      "code": "1",
      "oA_ItemsID": 1,
      "quantity": 1,
      "unit": "Piece",
      "price": 5,
      "itemSalePrice": 1,
      "itemSalePriceInSaleCurrencyAndUnit": 1,
      "oA_AccountsID": 1001,
      "registerParentAccount": "1610",
      "oA_CostCentreID": 1
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 2014,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51161,
    "description": "Unit does not belong to item units!",
    "comment": "[1] Unit does not belong to item units !\n Entry number [1]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "OrdersFromBuyers",
    "entryPoint": "SetUnit",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/OrderFromBuyer

Updates Order From Buyer. Returns OK or Error description Json.

Schema

```
{
```

```

"token": "string",
"oA_TransactionsID": 0,
"buyerCode": "string",
"buyerID": 0,
"costCentreName": "string",
"currency": "string",
"documentComment": "string",
"documentDate": "2023-05-03T08:01:24.946Z",
"documentNumber": "string",
"dueDate": "2023-05-03T08:01:24.946Z",
"isRateFixed": true,
"oA_CostCentreID": 0,
"oA_CurrenciesID": 0,
"oA_SalePriceTypesID": 0,
"oA_StatusID": 0,
"rate": 0,
"receiverAddress": "string",
"receiverAddressesID": 0,
"receiverCode": "string",
"receiverID": 0,
"salePriceType": "string",
"status": "string",
"supplierAddress": "string",
"supplierAddressesID": 0,
"supplierID": 0,
"clearItems": true,
"ordersFromByuerItems": [
  {
    "oA_ManagerialEntriesID": 0,
    "deleteRow": true,
    "code": "string",
    "oA_ItemsID": 0,
    "quantity": 0,
    "unit": "string",
    "oA_StockUnitsID": 0,
    "price": 0,
    "amount": 0,
    "vatRate": 0,
    "vat": 0,
    "exciseRate": 0,
    "excise": 0,
    "quantityInRegisterUnit": 0,
    "rsItemCode": "string",
    "rsItemName": "string",
    "rsItemUnit": "string",
    "rsItemVATType": 0,
    "itemSalePrice": 0,
    "itemSalePriceOA_CurrenciesID": 0,
    "itemSalePriceCurrency": "string",
    "itemSalePriceOA_StockUnitsID": 0,
    "itemSalePriceUnit": "string",
    "itemSalePriceInSaleCurrencyAndUnit": 0,
    "oA_AccountsID": 0,
    "registerParentAccount": "string",
    "costCentre": "string",
    "oA_CostCentreID": 0
  }
]
}

```

Field description

[Token] and [OA_TransactionsID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

"buyerCode": See [DebitorCreditorsList] method (use [Code] value).

"**buyerID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value).

"**costCentreName**": See [CostCentresList] method (use [Name] value).

"**currency**": See [CurrenciesList] method (use [Currency] value).

"**documentComment**":

"**documentDate**": Sets operation date.

"**documentNumber**":

"**dueDate**":

"**isRateFixed**": Sets a fixed rate flag. If true, then equivalent values in accounting entries of this operation will not be recalculated when the exchange rate changes.

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value).

"**oA_CurrenciesID**": See [CurrenciesList] method (use [OA_CurrenciesID] value).

"**oA_SalePriceTypesID**": See [SalePriceTypesList] method (use [oA_SalePriceTypesID] value).

"**oA_StatusID**": See [StatusesList] method (use [OA_StatusID] value).

"**rate**": Sets the currency rate.

"**receiverAddress**": See [GetDebitorCreditorByID] method (use [Address] value from [DebitorCreditorAddresses]).

"**receiverAddressesID**": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorAddressesID] value from [DebitorCreditorAddresses]).

"**receiverCode**": See [DebitorCreditorsList] method (use [Code] value).

"**receiverID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value).

"**salePriceType**": See [SalePriceTypesList] method (use [description] value).

"**status**": See [StatusesList] method (use [Name] value).

"**supplierAddress**": See [Owner] method (use [Address] value or [address] value from [OwnerAddresses]).

"**supplierAddressesID**": See [Owner] method (use [OA_OwnerAddressesID] value from [OwnerAddresses]).

"**supplierID**":

"**clearItems**": Clears all items for the operation.

"**ordersFromByuerItems**": Set OA_ManagerialEntriesID=0 to add a new Item specification. Specify the OA_ManagerialEntriesID value to update information about Item fields. To delete existing row, specify the value OA_ManagerialEntriesID and set "DeleteRow"=true.

"**oA_ManagerialEntriesID**": Specify an existing OA_ManagerialEntriesID to update or remove Item information, or 0 to add a new Item.

"**deleteRow**": If true, deletes Item information for specified OA_ManagerialEntriesID value

"**code**": See [ItemsList] method (use [Code] value).

"**oA_ItemsID**": See [ItemsList] method (use [OA_ItemsID] value).

"**quantity**":

"**unit**": See [StockUnitsList] method (use [Unit] value).

"**oA_StockUnitsID**": See [StockUnitsList] method (use [OA_StockUnitsID] value).

"**price**":

"**amount**":

"**vatRate**":

"**vat**":

"**exciseRate**":

"**excise**":

"**quantityInRegisterUnit**":

"**rsItemCode**":

"**rsItemName**":

"**rsItemUnit**":

"**rsItemVATType**":

"**itemSalePrice**":

"**itemSalePriceOA_CurrenciesID**":

"**itemSalePriceCurrency**":

"**itemSalePriceOA_StockUnitsID**":

"**itemSalePriceUnit**":

"**itemSalePriceInSaleCurrencyAndUnit**":

"**oA_AccountsID**":

"**registerParentAccount**":

"**costCentre**": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified. If ommite, gets cost centre from parameters.

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If ommite, gets cost centre from parameters

Examples

1. Updates Order From Byuer By OA_CostCentreID= 2
Request body:

```
{
  "token": "... paste here [Login] response value ...",
```

```

"oA_TransactionsID":2014,
"buyerCode": "1",
"buyerID": 1,
"oA_CostCentreID": 2,
"oA_CurrenciesID": 1,
"receiverAddress": "1",
"receiverAddressesID": 1,
"receiverCode": "",
"receiverID": 1,
"supplierAddressesID": 1,
"supplierID": 1,
"ordersFromByuerItems": [
  {
    "code": "1",
    "oA_ItemsID": 10,
    "quantity": 1,
    "price": 5,
    "itemSalePrice": 1,
    "itemSalePriceUnit": "",
    "itemSalePriceInSaleCurrencyAndUnit": 1,
    "oA_AccountsID": 1101,
    "registerParentAccount": "1610",
    "oA_CostCentreID": 2
  }
]
}

```

Server response Code(200):

```

{
  "id": 2014,
  "idFieldName": "OA_TransactionsID"
}

```

Server response Code(400):

```

[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 204",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "OrdersFromByuers",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

DELETE/api/OrderFromByuer

Returns OK or Error description Json.

Schema

```

{
  "token": "string",
  "oA_TransactionsID": 0
}

```

Field description

[Token] and [OA_TransactionsID] is necessary.

"**token**": Set [token] string from [LogIn] response value

"**oA_TransactionsID**":

Examples

1. Delete OrderFromByuer with OA_TransactionsID=2014
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 2014
}
```

Server response Code(200):

```
{
  "id": 2014,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 214",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "OrdersFromByuers",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

GetOrderFromByuerByID

POST/api/GetOrderFromByuerByID

Gets Order From Byuer and its Items by OA_TransactionsID value.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0
}
```

Field description

[Token] and [OA_TransactionsID] is necessary.

"token": Set [token] string from [LogIn] response value

"OA_TransactionsID": See [TransactionsList] method (use [OA_TransactionsID] value)

Examples

1. Get Order From Byuer and its Items by OA_TransactionsID =2014
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 2014
}
```

Server response Code(200):

```
{
  "oA_TransactionsID": 2014,
  "amount": 5,
  "buyerCode": "",
  "buyerID": 0,
}
```

```
"buyerName": "",
"costCentreName": "",
"currency": "",
"debitorCreditorVATType": 0,
"documentComment": "",
"documentDate": "2023-05-10T00:00:00",
"documentNumber": 27,
"dueDate": "2023-05-10T14:42:44",
"excise": 0,
"isDueDateNULL": true,
"isRateFixed": false,
"oA_CostCentreID": 1,
"oA_CurrenciesID": 1,
"oA_SalePriceTypesID": 0,
"oA_StatusID": 0,
"oA_TransactionTypesID": 16,
"ownerVATType": 0,
"rate": 1,
"receiverAddress": "",
"receiverAddressesID": 0,
"receiverCode": "",
"receiverID": 0,
"receiverName": "",
"reflectInformationInEntries": false,
"salePriceChangingLastReason": 0,
"salePriceType": "",
"status": "",
"supplier": "",
"supplierAddress": "",
"supplierAddressesID": 0,
"supplierID": 1,
"vat": 0.76,
"ordersFromByuerItems": [
  {
    "oA_ManualEntriesID": 2002,
    "oA_TransactionsID": 2014,
    "code": "1",
    "oA_ItemsID": 1,
    "quantity": 5,
    "unit": "Piece",
    "oA_StockUnitsID": 1,
    "price": 1,
    "amount": 5,
    "vatRate": 18,
    "vat": 0.76,
    "exciseRate": 0,
    "excise": 0,
    "quantityInRegisterUnit": 0,
    "itemType": 0,
    "rsItemCode": "",
    "rsItemName": "",
    "rsItemUnit": "",
    "errorPointer": 0,
    "itemVATType": 0,
    "rsItemVATType": 0,
    "itemSalePrice": 0,
    "itemSalePriceOA_CurrenciesID": 0,
    "itemSalePriceCurrency": "",
    "itemSalePriceOA_StockUnitsID": 0,
    "itemSalePriceUnit": "",
    "itemSalePriceInSaleCurrencyAndUnit": 0,
    "oA_AccountsID": 69,
    "registerParentAccount": "1610",
    "hasADoubleRegistrationUnit": false,
    "registerUnit": "Piece",
    "oA_StockUnitsIDRegisterUnit": 1,
    "costCentre": "",
```



```
"oA_CostCentreID": 0,
"sequenceNumber": 1
}
]
}
```

Server response Code(400):

```
[
{
  "errorNumber": 51051,
  "description": "Record does not exist or read prohibited for user!",
  "comment": "OA_TransactionsID = 2012",
  "command": "",
  "errorType": "ErrorType_Error",
  "source": "OA_BusinessObject.dll",
  "class": "OrdersFromByuers",
  "entryPoint": "GetProperties",
  "resourceDLL_ForTranslation": "ORIS_Accounting",
  "errorPointer": 0
}
]
```

OrdersFromByuersList

POST/api/OrdersFromByuersList

Gets Orders From Byuers. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1.

"sort": Use SQL ORDER BY string with field names.

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get Order From Byuers Info and Result tables.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "filter": "",
  "sort": "",
  "pageNumber": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 3,
      "PageCount": 3,

```

```

    "PageRowCount": 1,
    "PageNumber": 1
  }
],
"Result": [
  {
    "OA_TransactionsID": 4004,
    "Document date": "2023-05-23T16:51:22",
    "Amount": 45.00000000,
    "Supplier": "",
    "Supplier address": "",
    "OA_TransactionTypesID": 41,
    "Comment": "",
    "Document number": "2",
    "BuyerID": 1,
    "Buyer code": "1",
    "Buyer name": "Organization",
    "OA_CostCentreID": 2,
    "Cost centre": "New cost center",
    "Receiver address": "",
    "ReceiverAddressID": null,
    "SupplierAddressID": null,
    "VAT": 6.85000000,
    "Excise": 0.00000000,
    "Rate": 1.00000000,
    "Is rate fixed": false,
    "OA_CurrenciesID": 1,
    "Currency": "GEL",
    "Record related binary documents": null,
    "Owner VAT type": 0,
    "DebitorCreditor VAT type": 0,
    "ReceiverID": 1,
    "Receiver code": "1",
    "Receiver name": "Organization",
    "OA_SalePriceTypesID": null,
    "Price type": null,
    "OA_DebitorCreditorGroupID": null,
    "Due date": null,
    "Linked operations": null,
    "OA_StatusID": 4,
    "Status": "Rejected",
    "Numeric document number": 2,
    "NUM_ROW": 1
  }
]
}

```

Menegerial accounting – results

ItemsBalancesList

POST/api/ItemsBalancesList

Gets Items Balances. Returns [Info] and [Result] tables

Schema

```

{
  "token": "string",
  "currenciesIDs": "string",
  "costCentreIDs": "string",
  "costUnitIDs": "string",
  "date": "2023-05-08T08:39:51.900Z",
  "showZeroResults": true,
  "showStockValues": true,

```

```
"showFinanceValues": true,
"topRecordsCount": 0,
"filter": "string",
"sort": "string",
"showResultsInSaleUnit": true,
"showOnlyTotalResults": true,
"pageNumber": 0
}
```

Field description

"token": Set [token] string from [Login] response value

"currenciesIDs":

"costCentreIDs":

"costUnitIDs":

"date": Date of balance (including date).

"showZeroResults":

"showStockValues":

"showFinanceValues":

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": "string",

"sort": "string",

"showResultsInSaleUnit": true,

"showOnlyTotalResults": true,

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get Items List Balances

Request body:

```
{
  "token": "... paste here [Login] response value ...",
  "currenciesIDs": "1",
  "costCentreIDs": "1",
  "costUnitIDs": "1",
  "date": "2023-06-02T13:47:27.515Z",
  "showZeroResults": true,
  "showStockValues": true,
  "showFinanceValues": true,
  "topRecordsCount": 1,
  "filter": "UPPER([ViewOA_Items].[Code])=N'1' AND UPPER([ViewOA_Items].[Name])=N'ITEM2'",
  "sort": "[Code],[Name]",
  "showResultsInSaleUnit": true,
  "showOnlyTotalResults": true,
  "pageNumber": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 0,
      "PageCount": 0,
      "PageRowCount": 1,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_ItemsID": 1,
      "Code": "1",
      "Name": "item2",
      "Unit": "Piece",
      "OA_StockUnitsID": 1,
      "Item type": 0,
      "OA_AccountsID": 69,
      "Register parent account": "1610",
    }
  ]
}
```

```

"OA_ItemGroupID": null,
"Group": null,
"Sale price": 0.00000000,
"Close date": null,
"Written off": false,
"Closed by desintegration": null,
"Closed by integration": null,
"OA_CurrenciesID": null,
"Currency": null,
"OA_StockUnitsID_SaleUnit": 1,
"Sale unit": "Piece",
"Record related binary documents": null,
"VAT type": 0,
"Inventory number": "",
"OA_ManufacturersID": null,
"Manufacturer": null,
"Comment": "",
"Register parent account name": "Merchandise inventory",
"Date of expiration": null,
"Manufacturer number": "",
"Code contains weight": null,
"Has a double registration unit": null,
"Excise rate": 0.00000000,
"OA_StockUnitsID_ExciseUnit": null,
"Excise unit": null,
"Price increase percentage": 0.00000000,
"OA_AccountsIDExpenseAccount": null,
"Item write-off expense account": null,
"OA_DebitorCreditorID": null,
"Responsible person": null,
"OA_CostUnitID": null,
"Cost unit": null,
"HasResult": true,
"Stock": 106.0,
"Cost": 6071.33079022,
"Unit cost": 57.2767055681132
}
]
}

```

ItemsTurnoversList

POST/api/ItemsTurnoversList

Gets Items Turnovers. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "currenciesIDs": "string",
  "costCentreIDs": "string",
  "costUnitIDs": "string",
  "diapazonStartDate": "2023-05-08T08:49:53.093Z",
  "diapazonEndDate": "2023-05-08T08:49:53.093Z",
  "showZeroResults": true,
  "showStockValues": true,
  "showFinanceValues": true,
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"currenciesIDs":

"costCentreIDs":

"costUnitIDs":

"diapazonStartDate": Diapazon start date.

"diapazonEndDate": Diapazon end date (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01", "DiapazonEndDate":"2011-01-01"

"showZeroResults":

"showStockValues":

"showFinanceValues":

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": "string",

"sort": "string",

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get Items List Turnovers

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "diapazonStartDate": "2023-06-05T10:45:05.693Z",
  "diapazonEndDate": "2024-01-01T10:45:05.693Z",
  "showZeroResults": true,
  "showStockValues": true,
  "showFinanceValues": true,
  "topRecordsCount": 1,
  "filter": "UPPER([ViewOA_Items].[Name])=N'ITEM2'",
  "sort": "[Code],[Name],[Start (Quantity)]",
  "pageNumber": 0
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 0,
      "PageRowCount": 1,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_ItemsID": 1,
      "Code": "1",
      "Name": "item2",
      "Unit": "Piece",
      "OA_StockUnitsID": 1,
      "Item type": 0,
      "OA_AccountsID": 69,
      "Register parent account": "1610",
      "OA_ItemGroupID": null,
      "Group": null,
      "Sale price": 0.00000000,
      "Close date": null,
      "Written off": false,
      "Closed by desintegration": null,
      "Closed by integration": null,
      "OA_CurrenciesID": null,
      "Currency": null,
      "OA_StockUnitsID_SaleUnit": null,
      "Sale unit": null,
      "Record related binary documents": null,
      "VAT type": 0,
      "Inventory number": "",
      "OA_ManufacturersID": null,
    }
  ]
}
```

```
"Manufacturer": null,
"Comment": "",
"Register parent account name": "Merchandise inventory",
"Date of expiration": null,
"Manufacturer number": "",
"Code contains weight": null,
"Has a double registration unit": null,
"Excise rate": 0.00000000,
"OA_StockUnitsID_ExciseUnit": null,
"Excise unit": null,
"Price increase percentage": 0.00000000,
"OA_AccountsIDExpenseAccount": null,
"Item write-off expense account": null,
"OA_DebitorCreditorID": null,
"Responsible person": null,
"OA_CostUnitID": null,
"Cost unit": null,
"HasResult": true,
"Start (Quantity)": 111.0,
"Accept operations (Quantity)": null,
"Transfer operations - out (Quantity)": 0.0,
"Transfer operations - in (Quantity)": null,
"Supply operations (Quantity)": 0.0,
"Write off operations (Quantity)": 0.0,
"Return to supplier operations (Quantity)": 0.0,
"Return from customer operations (Quantity)": null,
"Prodused in manufacturing (Quantity)": null,
"Used in manufacturing (Quantity)": 0.0,
"Inventory - Excess (Quantity)": null,
"Inventory - Shortage (Quantity)": 0.0,
"Miscellaneous - plus (Quantity)": null,
"Miscellaneous - minus (Quantity)": 0.0,
"Balance (Quantity)": 111.0,
"Start (prime cost)": 6371.68181944,
"Accept operations (prime cost)": null,
"Transfer operations - out (prime cost)": 0.0,
"Transfer operations - in (prime cost)": null,
"Supply operations (prime cost)": 0.0,
"Write off operations (prime cost)": 0.0,
"Return to supplier operations (prime cost)": 0.0,
"Return from customer operations (prime cost)": null,
"Prodused in manufacturing (prime cost)": null,
"Used in manufacturing (prime cost)": 0.0,
"Inventory - Excess (prime cost)": null,
"Inventory - Shortage (prime cost)": 0.0,
"Miscellaneous - plus (prime cost)": null,
"Miscellaneous - minus (prime cost)": 0.0,
"Balance (prime cost)": 6371.68181944,
"Accept operations (amount)": null,
"Accept operations (amount without VAT)": null,
"Supply operations (amount)": null,
"Supply operations (amount without VAT)": null,
"Return to supplier operations (amount)": null,
"Return to supplier operations (amount without VAT)": null,
"Return from customer operations (amount)": null,
"Return from customer operations (amount without VAT)": null,
"Supply operations (Unit cost)": null,
"Supply operations (Unit sale price)": null,
"Inventory - Excess (amount)": null,
"Inventory - Shortage (amount without VAT)": null,
"Profit and loss": 0.0,
"Margin": 0.0
}
]
}
```

Salary

Employee

POST/api/Employee

Adds new employee. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "bankAccount": "string",
  "closeDate": "2023-05-01T11:59:39.076Z",
  "comment": "string",
  "costCentre": "string",
  "department": "string",
  "idCardNumber": "string",
  "identityNumber": "string",
  "isCloseDateNULL": true,
  "isValidFromNULL": true,
  "oA_CostCentreID": 0,
  "oA_DebitorCreditorBankAccountsID": 0,
  "oA_DebitorCreditorID": 0,
  "oA_DebitorCreditorPhonesID": 0,
  "oA_EmployeeDepartmentID": 0,
  "oA_EmployeePositionTitleID": 0,
  "oA_LegalIdentifiersID": 0,
  "phone": "string",
  "positionTitle": "string",
  "validFrom": "2023-05-01T11:59:39.076Z"
}
```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"bankAccount": See [GetDebitorCreditorByID] method (use [account] value from [DebitorCreditorBankAccounts]). Can be omitted if [OA_DebitorCreditorBankAccountsID] property specified

"closeDate":

"comment":

"costCentre": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified

"department": See [EmployeeDepartmentsList] method (use [Name] value). Can be omitted if [OA_EmployeeDepartmentID] property specified

"idCardNumber": See [GetDebitorCreditorByID] method (use [Number] value from [DebitorCreditorLegalIdentifiers]). Can be omitted if [OA_LegalIdentifiersID] property specified

"identityNumber": See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [OA_DebitorCreditorID] property specified

"isCloseDateNULL": Set True to clear [CloseDate] value

"isValidFromNULL": Set True to clear [ValidFrom] value

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified

"oA_DebitorCreditorBankAccountsID": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorBankAccountsID] value from [DebitorCreditorBankAccounts]). Can be omitted if [BankAccount] property specified

"oA_DebitorCreditorID": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [IdentityNumber] property specified

"oA_DebitorCreditorPhonesID": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorPhonesID] value from [DebitorCreditorPhones]). Can be omitted if [Phone] property specified

"oA_EmployeeDepartmentID": See [EmployeeDepartmentsList] method (use [OA_EmployeeDepartmentID] value). Can be omitted if [Department] property specified

"oA_EmployeePositionTitleID": See [GetEmployeeDepartmentByID] method (use [OA_EmployeePositionTitleID] value from [EmployeePosition]). Can be omitted if [PositionTitle] property specified

"oA_LegalIdentifiersID": See [GetDebitorCreditorByID] method (use [OA_LegalIdentifiersID] value from [DebitorCreditorLegalIdentifiers]). Can be omitted if [IDCardNumber] property specified

"**phone**": See [GetDebtorCreditorByID] method (use [phone] value from [DebtorCreditorPhones]). Can be omitted if [OA_DebtorCreditorPhonesID] property specified

"**positionTitle**": See [GetEmployeeDepartmentByID] method (use [Name] value from [EmployeePosition]). Can be omitted if [OA_EmployeePositionTitleID] property specified

"**validFrom**":

Examples

1. Add employee using "oA_DebtorCreditorID" – 2

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_DebtorCreditorID": 2
}
```

Server response Code(200):

```
{
  "id": 12,
  "idFieldName": "OA_EmployeeID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51148,
    "description": "Debtor/Creditor does not exist!",
    "comment": "OA_DebtorCreditorID=28",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Employee",
    "entryPoint": "set property",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

2. Add employee using "identityNumber" -12345678910

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "identityNumber": "12345678910"
}
```

PUT/api/Employee

Updates employee. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_EmployeeID": 0,
  "bankAccount": "string",
  "closeDate": "2023-05-01T12:29:12.841Z",
  "comment": "string",
  "costCentre": "string",
  "department": "string",
  "idCardNumber": "string",
  "identityNumber": "string",
  "isCloseDateNULL": true,
  "isValidFromNULL": true,
  "oA_CostCentreID": 0,
  "oA_DebtorCreditorBankAccountsID": 0,
  "oA_DebtorCreditorID": 0,
  "oA_DebtorCreditorPhonesID": 0,
}
```



```

"oA_EmployeeDepartmentID": 0,
"oA_EmployeePositionTitleID": 0,
"oA_LegalIdentifiersID": 0,
"phone": "string",
"positionTitle": "string",
"validFrom": "2023-05-01T12:29:12.842Z"
}

```

Field description

[Token] and [OA_EmployeeID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_EmployeeID":

"bankAccount": See [GetDebitorCreditorByID] method (use [account] value from [DebitorCreditorBankAccounts]). Can be omitted if [OA_DebitorCreditorBankAccountsID] property specified

"closeDate":

"comment":

"costCentre": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified

"department": See [EmployeeDepartmentsList] method (use [Name] value). Can be omitted if [OA_EmployeeDepartmentID] property specified

"idCardNumber": See [GetDebitorCreditorByID] method (use [Number] value from [DebitorCreditorLegalIdentifiers]). Can be omitted if [OA_LegalIdentifiersID] property specified

"identityNumber": See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [OA_DebitorCreditorID] property specified

"isCloseDateNULL": Set True to clear [CloseDate] value

"isValidFromNULL": Set True to clear [ValidFrom] value

"oA_CostCentreID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified

"oA_DebitorCreditorBankAccountsID": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorBankAccountsID] value from [DebitorCreditorBankAccounts]). Can be omitted if [BankAccount] property specified

"oA_DebitorCreditorID": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [IdentityNumber] property specified

"oA_DebitorCreditorPhonesID": See [GetDebitorCreditorByID] method (use [OA_DebitorCreditorPhonesID] value from [DebitorCreditorPhones]). Can be omitted if [Phone] property specified

"oA_EmployeeDepartmentID": See [EmployeeDepartmentsList] method (use [OA_EmployeeDepartmentID] value). Can be omitted if [Department] property specified

"oA_EmployeePositionTitleID": See [GetEmployeeDepartmentByID] method (use [OA_EmployeePositionTitleID] value from [EmployeePosition]). Can be omitted if [PositionTitle] property specified

"oA_LegalIdentifiersID": See [GetDebitorCreditorByID] method (use [OA_LegalIdentifiersID] value from [DebitorCreditorLegalIdentifiers]). Can be omitted if [IDCardNumber] property specified

"phone": See [GetDebitorCreditorByID] method (use [phone] value from [DebitorCreditorPhones]). Can be omitted if [OA_DebitorCreditorPhonesID] property specified

"positionTitle": See [GetEmployeeDepartmentByID] method (use [Name] value from [EmployeePosition]). Can be omitted if [OA_EmployeePositionTitleID] property specified

"validFrom":

Examples

1. Update bankAccount by account number and cost centre by "oA_CostCentreID" - 2 , using "oA_EmployeeID" - 12

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_EmployeeID": 12,
  "bankAccount": "1111",
  "oA_CostCentreID": 2
}

```

Server response Code(200):

```

{
  "id": 12,
  "idFieldName": "OA_EmployeeID"
}

```

Server response Code(400):

```
[
  {
    "errorNumber": 51150,
    "description": "Employee does not exist!",
    "comment": "[WHERE OA_EmployeeID = 12] Record is not in table or user have no permissions to read this record.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Employee",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/Employee

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_EmployeeID": 0
}
```

Field description

[Token] and [OA_EmployeeID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_EmployeeID": 0

Examples

1. Delete employee from list using "oA_EmployeeID" - 12

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_EmployeeID": 12
}
```

Server response Code(200):

```
{
  "id": 12,
  "idFieldName": "OA_EmployeeID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "OA_EmployeeID=12 [2] [Person]\nExist relation to another table.\n\nRelated table [OA_SalaryResults].\n\n[547] The DELETE statement conflicted with the REFERENCE constraint \"FK_OA_SalaryResults_OA_Employee\". The conflict occurred in database \"NDatabaseEN\", table \"dbo.OA_SalaryResults\", column 'OA_EmployeeID'.[3621] The statement has been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Employee",
    "entryPoint": "RealDelete",
    "resourceDLL_ForTranslation": "ORIS_Administrate",
    "errorPointer": 0
  }
]
```

EmployeesList

POST/api/EmployeesList

Gets Employee. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get first 2 employees from list

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 2,
      "PageCount": 0,
      "PageRowCount": 2,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_EmployeeID": 12,
      "OA_DebitorCreditorID": 2,
      "OA_DebitorCreditorBankAccountsID": 2,
      "Comment": "",
      "Identity number": "2",
      "Legal name": "Person",
      "Clock number": "",
      "Address": "",
      "Bank account": "1111",
      "OA_BanksID": 8,
      "Bank code": "TBCBGE22",
      "Bank name": "TBC Bank",
      "OA_CostCentreID": 2,
      "Cost centre": "New cost center",
      "OA_CostUnitID": 1,
      "Cost unit": "Total cost unit"
    },
    {
      "OA_EmployeeID": 13,
```

```

"OA_DebitorCreditorID": 1002,
"Comment": "",
"Identity number": "12345678910",
"Legal name": "someone",
"Clock number": "",
"Address": "",
"OA_CostCentreID": 1,
"Cost centre": "General cost center",
"OA_CostUnitID": 1,
"Cost unit": "Total cost unit"
}
]
}

```

2. Get a list of Employees only from General cost center and sort by Identity number decreasing
Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"filter": "([ViewOA_Employee].[Cost centre])=N'General cost center' ",
"sort": "[Identity number] DESC"
}

```

EmployeeDepartment

POST/api/EmployeeDepartment

Adds new Employee Department. Returns ID value or Error description Json.

Schema

```

{
"token": "string",
"comment": "string",
"name": "string",
"parentID": 0,
"employeePosition": [
{
"name": "string"
}
]
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"comment":

"name": Name of department

"parentID": Parent Employee Department ID to form tree structure. Specify 0 to specify the root Employee Department

"employeePosition":

"name": Position name for employee department

Examples

1. Add new department "Financial"
Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"name": "Financial"
}

```

Server response Code(200):

```

{
"id": 3,
"idFieldName": "OA_EmployeeDepartmentID"
}

```

```
}
```

Server response Code(400):

```
[  
  {  
    "errorNumber": 51109,  
    "description": "ORIS_Tools internal error!",  
    "comment": "OA_EmployeeDepartmentID=0 [Financial]\nA duplicate value cannot be  
inserted.\r\n[Name]\r\n\r\n[2601] Cannot insert duplicate key row in object 'dbo.OA_EmployeeDepartment' with  
unique index 'IdxOA_EmployeeDepartment1'. The duplicate key value is (Financial).[3621] The statement has been  
terminated.",  
    "command": "",  
    "errorType": "ErrorType_Error",  
    "source": "OA_BusinessObject.dll",  
    "class": "EmployeeDepartment",  
    "entryPoint": "Update",  
    "resourceDLL_ForTranslation": "ORIS_Administrate",  
    "errorPointer": 0  
  }  
]
```

2. Add child department "Accountants" under "Financial" and employee position "Chief Accountant"

Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "name": "Accountants",  
  "parentID": 3,  
  "employeePosition": [  
    {  
      "name": "Chief Accountant"  
    }  
  ]  
}
```

PUT/api/EmployeeDepartment

Updates Employee Department. Returns OK or Error description Json

Schema

```
{  
  "token": "string",  
  "oA_EmployeeDepartmentID": 0,  
  "comment": "string",  
  "name": "string",  
  "parentID": 0,  
  "clearEmployeePosition": true,  
  "employeePosition": [  
    {  
      "oA_EmployeePositionTitleID": 0,  
      "deleteRow": true,  
      "name": "string"  
    }  
  ]  
}
```

Field description

"token": Set [token] string from [LogIn] response value

"oA_EmployeeDepartmentID": See [EmployeeDepartmentsList] method (use [OA_EmployeeDepartmentID] value)

"comment":

"name":

"parentID": Parent Employee Department ID to form tree structure. Specify 0 to specify the root Employee Department

"clearEmployeePosition": Clears all related Employee Position for the Department

"employeePosition":

"oA_EmployeePositionTitleID": Specify an existing OA_EmployeePositionTitleID to update or remove related item information, or 0 to add a new related position

"deleteRow": If true, deletes related unit information for specified OA_EmployeePositionTitleID value

"name": Position name for employee department

Examples

1. Update employee position – add “Accountant” and “Cashier” under child department “Accountants” using “oA_EmployeeDepartmentID” - 6

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_EmployeeDepartmentID": 6,
  "clearEmployeePosition": false,
  "employeePosition": [
    {
      "name": "Accountant"
    },
    {
      "name": "Cashier"
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 6,
  "idFieldName": "OA_EmployeeDepartmentID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_EmployeeDepartmentID = 118",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "EmployeeDepartment",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/EmployeeDepartment

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_EmployeeDepartmentID": 0
}
```

Field description

[Token] and [OA_EmployeeDepartmentID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_EmployeeDepartmentID": 0

Examples

1. Delet Employee Department by "oA_EmployeeDepartmentID" - 6

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_EmployeeDepartmentID": 6
}
```

Server response Code(200):

```
{
  "id": 6,
  "idFieldName": "OA_EmployeeDepartmentID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "OA_EmployeeDepartmentID=6 [Accountants]\nExist relation to another table.\r\nRelated table [OA_Employee].\r\n\r\n[547] The DELETE statement conflicted with the REFERENCE constraint \"FK_OA_Employee_OA_EmployeePositionTitle\". The conflict occurred in database \"NDatabaseEN\", table \"dbo.OA_Employee\", column 'OA_EmployeePositionTitleID'.[3621] The statement has been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "EmployeeDepartment",
    "entryPoint": "RealDelete",
    "resourceDLL_ForTranslation": "ORIS_Administrate",
    "errorPointer": 0
  }
]
```

EmployeeDepartmentsList

POST/api/EmployeeDepartmentsList

Gets Employee Departments. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

[Token] and [OA_EmployeeDepartmentID] is necessary.

"**token**": Set [token] string from [LogIn] response value

"**topRecordsCount**": The number of records to accept. If you specify 0 you will get all records. Default=0

"**filter**": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"**sort**": Use SQL ORDER BY string with field names

"**pageNumber**": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get list of all Employee Departments sort by level

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "sort": [Level]
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 2,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_EmployeeDepartmentID": 3,
      "ParentID": 0,
      "Level": 0,
      "Name": "Financial",
      "Comment": ""
    },
    {
      "OA_EmployeeDepartmentID": 6,
      "ParentID": 3,
      "Level": 1,
      "Name": "Accountants",
      "Comment": ""
    }
  ]
}
```

EmployeePositionsList

POST/api/EmployeePositionsList

Gets Employee Positions. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property: [TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get list of employees, who has specified position title "Accountant"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[Position title] IN (N'Accountant')"
```

Server response Code(200):

```
{
  "Info": [
    {
```



```

    "RowCount": 1,
    "PageCount": 0,
    "PageRowCount": 0,
    "PageNumber": 0
  }
],
"Result": [
  {
    "OA_EmployeePositionTitleID": 18,
    "Position title": "Accountant",
    "OA_EmployeeDepartmentID": 6,
    "SequenceNumber": 2,
    "Department": "Accountants"
  }
]
}

```

GetDataTableEmployeeMonthAccrueds

POST/api/GetDataTableEmployeeMonthAccrueds

Gets Employee Month Accrueds. Returns [Info] and [Result] tables

Schema

```

{
  "token": "string",
  "oA_EmployeeID": 0,
  "year": 0,
  "month": 0,
  "oA_MacrosID": 0,
  "reportNumber": "string"
}

```

Field description

"token": Set [token] string from [LogIn] response value

"oA_EmployeeID":

"year":

"month":

"oA_MacrosID":

"reportNumber":

Examples

1. Get employee month accrued by "oA_EmployeeID" -12, year - 2023 and month - 5
Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_EmployeeID": 12 ,
  "year":2023,
  "month": 5
}

```

Server response Code(200):

```

{
  "Info": [
    {
      "RowCount": 1
    }
  ],
  "Result": [
    {
      "OA_SalaryAccruedsID": 137,
      "OA_EmployeeID": 12,
      "Year": 2023,

```

```
"Month": 5,
"A": 1000.00000000
}
]
}
```

GetDataTableEmployeeResults

POST/api/GetDataTableEmployeeResults

Gets Employee salary results. Returns [Info] and [Result] tables

Schema

```
{
  "token": "string",
  "oA_EmployeeID": 0,
  "year": 0,
  "month": 0,
  "oA_MacrosID": 0,
  "reportNumber": "string"
}
```

Field description

All fields is necessary.

"token": Set [token] string from [LogIn] response value

"oA_EmployeeID":

"year":

"month":

"oA_MacrosID":

"reportNumber":

Examples

1. Get employee salary results by "oA_EmployeeID" -12, year - 2023 and month - 8

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_EmployeeID": 12 ,
  "year":2023,
  "month": 8,
  "oA_MacrosID": 298,
  "reportNumber": "8"
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1
    }
  ],
  "Result": [
    {
      "OA_SalaryResultsID": 11,
      "OA_EmployeeID": 12,
      "OA_MacrosID": 298,
      "Year": 2023,
      "Month": 8,
      "ReportNumber": "8",
      "A": 1000.00000000,
      "B": 23.00000000,
      "C": 1000.00000000,
      "D": 0.00000000,
      "E": 0.00000000,
    }
  ]
}
```

```

"F": 0.00000000,
"G": 0.00000000,
"H": 1000.00000000,
"I": 1000.00000000,
"J": 200.00000000,
"K": 800.00000000,
"L": 3000.00000000,
"M": 600.00000000,
"N": 0.00000000,
"O": 0.00000000,
"P": 0.00000000,
"BankAccount": "1111",
"OA_BankID": 8,
"OA_EmployeePositionTitleID": 0,
"OA_EmployeeDepartmentID": 0,
"AE": 23,
"BA": 0.00000000,
"BB": 0.00000000,
"BF": 0.00000000,
"AG": 0.00000000,
"OA_CostCentreID": 2,
"AH": 0.00000000,
"AI": 0.00000000,
"BC": 1000.00000000,
"AJ": 0.00000000,
"AK": 0.00000000,
"AL": 0.00000000,
"AM": 0.00000000,
"AN": 0.00000000,
"AO": 0.00000000,
"OA_CostUnitID": 1
}
]
}

```

GetDataTableEmployeeYearAccrueds

POST/api/GetDataTableEmployeeYearAccrueds

Gets Employee Current Year Accrueds. Returns [Info] and [Result] tables

Schema

```

{
  "token": "string",
  "oA_EmployeeID": 0,
  "topRecordsCount": 0,
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"oA_EmployeeID":

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get employee current year accrued by "oA_EmployeeID" -12, first 2 month
Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_EmployeeID": 12,
  "topRecordsCount": 2
}

```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 0,
      "PageCount": 0,
      "PageRowCount": 2,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_SalaryAccruedsID": 133,
      "OA_EmployeeID": 12,
      "Year": 2023,
      "Month": 1,
      "A": 1000.00000000,
      "J": 22.00000000,
      "AA": 22
    },
    {
      "OA_SalaryAccruedsID": 134,
      "OA_EmployeeID": 12,
      "Year": 2023,
      "Month": 2,
      "A": 1000.00000000
    }
  ]
}
```

GetDataTableReportNumbers

POST/api/GetDataTableReportNumbers

Gets salary Report Numbers. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get all salary report by current year

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 0,
      "PageCount": 0,
      "PageRowCount": 0,

```

```

    "PageNumber": 0
  }
],
"Result": [
  {
    "Year": 2023,
    "Month": 8,
    "OA_MacrosID": 298,
    "ReportNumber": "8",
    "Comment": "Salary accrued report"
  },
  {
    "Year": 2023,
    "Month": 2,
    "OA_MacrosID": 298,
    "ReportNumber": "2",
    "Comment": "Salary accrued report"
  },
  {
    "Year": 2023,
    "Month": 1,
    "OA_MacrosID": 298,
    "ReportNumber": "1",
    "Comment": "Salary accrued report"
  }
]
}

```

GetDataTableReportResult

POST/api/GetDataTableReportResult

Gets salary Report Result. Returns [Info] and [Result] tables

Schema

```

{
  "token": "string",
  "year": 0,
  "month": 0,
  "oA_MacrosID": 0,
  "reportNumber": "string",
  "topRecordsCount": 0,
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"year":

"month":

"oA_MacrosID":

"reportNumber":

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Gets salary report result year - 2023, month - 1, "oA_MacrosID" - 298 – SalaryAccruedReport
Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "year": 2023,
  "month": 1,
  "oA_MacrosID": 298,
  "reportNumber": "1",

```

```
"topRecordsCount":1,
"pageNumber": 0
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 0,
      "PageCount": 0,
      "PageRowCount": 1,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_SalaryResultsID": 2,
      "OA_EmployeeID": 12,
      "OA_MacrosID": 298,
      "Year": 2023,
      "Month": 1,
      "A": 1000.00000000,
      "B": 22.00000000,
      "C": 1000.00000000,
      "D": 0.00000000,
      "E": 0.00000000,
      "F": 0.00000000,
      "G": 0.00000000,
      "H": 1000.00000000,
      "I": 1000.00000000,
      "J": 200.00000000,
      "K": 800.00000000,
      "L": 1000.00000000,
      "M": 200.00000000,
      "N": 0.00000000,
      "O": 0.00000000,
      "P": 0.00000000,
      "ReportNumber": "1",
      "OA_DebitorCreditorID": 2,
      "Employee": "Person",
      "Identity number": "2",
      "Macro": "Salary accrued report",
      "OA_BankID": 8,
      "Bank account": "1111",
      "Bank code": "TBCBGE22",
      "Bank name": "TBC Bank",
      "OA_EmployeePositionTitleID": 0,
      "Position title": null,
      "OA_EmployeeDepartmentID": 0,
      "Department": null,
      "AE": 22,
      "OA_AccountsID_AccruedSalary": null,
      "OA_AccountsID_AccruedBulletin": null,
      "AF": null,
      "BA": 0.00000000,
      "BB": 0.00000000,
      "BF": 0.00000000,
      "AG": 0.00000000,
      "OA_CostCentreID": 2,
      "Cost centre": "New cost center",
      "AH": 0.00000000,
      "AI": 0.00000000,
      "BC": 1000.00000000,
      "AO": 0.00000000,
      "BG": null,
      "AJ": 0.00000000,
      "AK": 0.00000000,
    }
  ]
}
```

```
"AL": 0.00000000,  
"AM": 0.00000000,  
"AN": 0.00000000,  
"OA_CostUnitID": 1,  
"Cost unit": "Total cost unit"  
}  
]  
}
```

GetEmployeeDepartmentByID

POST/api/GetEmployeeDepartmentByID

Gets employee department and its related positions list.

Schema

```
{  
  "token": "string",  
  "oA_EmployeeDepartmentID": 0  
}
```

Field description

"token": Set [token] string from [LogIn] response value

"oA_EmployeeDepartmentID":

Examples

1. Get list of employee departments using "oA_EmployeeDepartmentID" - 6

Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "oA_EmployeeDepartmentID":6  
}
```

Server response Code(200):

```
{  
  "oA_EmployeeDepartmentID": 6,  
  "comment": "",  
  "name": "Accountants",  
  "parentID": 3,  
  "parentName": "Financial",  
  "employeePosition": [  
    {  
      "oA_EmployeePositionTitleID": 13,  
      "name": "Chief Accountant"  
    },  
    {  
      "oA_EmployeePositionTitleID": 20,  
      "name": "Accountant"  
    },  
    {  
      "oA_EmployeePositionTitleID": 21,  
      "name": "Cashier"  
    }  
  ]  
}
```

GetDataTableSalaryResultArchives

POST/api/GetDataTableSalaryResultArchives

Gets Salary Results from Archive. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter":

"sort":

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get first two salary results from archives, filter by year- 2023, month -1 and sort by monthly accrued
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2,
  "filter": "[Year] IN (2023) AND [ViewOA_SalaryResults].[Month]=1",
  "sort": "[A]",
  "pageNumber": 0
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 0,
      "PageCount": 0,
      "PageRowCount": 2,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_SalaryResultsID": 5,
      "OA_EmployeeID": 15,
      "OA_MacrosID": 298,
      "Year": 2023,
      "Month": 1,
      "A": 600.00000000,
      "B": 22.00000000,
      "C": 600.00000000,
      "D": 0.00000000,
      "E": 0.00000000,
      "F": 0.00000000,
      "G": 0.00000000,
      "H": 600.00000000,
      "I": 600.00000000,
      "J": 120.00000000,
      "K": 480.00000000,
      "L": 600.00000000,
      "M": 120.00000000,
      "N": 0.00000000,
      "O": 0.00000000,
      "P": 0.00000000,
      "ReportNumber": "1",
      "OA_DebitorCreditorID": 1013,
      "Employee": "Person6",
      "Identity number": "8",
    }
  ]
}
```



```
"Macro": "Salary accrued report",
"OA_BankID": 1,
"Bank account": "2222",
"Bank code": "BNLNGE22",
"Bank name": "National Bank Of Georgia",
"OA_EmployeePositionTitleID": 13,
"Position title": "Chief Accountant",
"OA_EmployeeDepartmentID": 6,
"Department": "Accountants",
"AE": 22,
"BA": 0.00000000,
"BB": 0.00000000,
"BF": 0.00000000,
"AG": 0.00000000,
"OA_CostCentreID": 1,
"Cost centre": "General cost center",
"AH": 0.00000000,
"AI": 0.00000000,
"BC": 600.00000000,
"AO": 0.00000000,
"AJ": 0.00000000,
"AK": 0.00000000,
"AL": 0.00000000,
"AM": 0.00000000,
"AN": 0.00000000,
"OA_CostUnitID": 1,
"Cost unit": "Total cost unit"
},
{
"OA_SalaryResultsID": 4,
"OA_EmployeeID": 14,
"OA_MacrosID": 298,
"Year": 2023,
"Month": 1,
"A": 700.00000000,
"B": 22.00000000,
"C": 700.00000000,
"D": 0.00000000,
"E": 0.00000000,
"F": 0.00000000,
"G": 0.00000000,
"H": 700.00000000,
"I": 700.00000000,
"J": 140.00000000,
"K": 560.00000000,
"L": 700.00000000,
"M": 140.00000000,
"N": 0.00000000,
"O": 0.00000000,
"P": 0.00000000,
"ReportNumber": "1",
"OA_DebitorCreditorID": 1008,
"Employee": "Person5",
"Identity number": "7",
"Macro": "Salary accrued report",
"OA_BankID": 1,
"Bank account": "2222",
"Bank code": "BNLNGE22",
"Bank name": "National Bank Of Georgia",
"OA_EmployeePositionTitleID": 0,
"OA_EmployeeDepartmentID": 0,
"AE": 22,
"BA": 0.00000000,
"BB": 0.00000000,
"BF": 0.00000000,
"AG": 0.00000000,
"OA_CostCentreID": 1,
```

```

"Cost centre": "General cost center",
"AH": 0.00000000,
"AI": 0.00000000,
"BC": 700.00000000,
"AO": 0.00000000,
"AJ": 0.00000000,
"AK": 0.00000000,
"AL": 0.00000000,
"AM": 0.00000000,
"AN": 0.00000000,
"OA_CostUnitID": 1,
"Cost unit": "Total cost unit"
}
]
}

```

GetDataTableSalaryAccruedsArchives

POST/api/GetDataTableSalaryAccruedsArchives

Gets Salary Accrueds from Archive. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "filter": "string",
  "sort": "string",
  "topRecordsCount": 0,
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property: [TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get first two data from salary accrued archives

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2
}

```

Server response Code(200):

```

{
  "Info": [
    {
      "RowCount": 0,
      "PageCount": 0,
      "PageRowCount": 2,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_SalaryAccruedsID": 133,
      "OA_EmployeeID": 12,
      "OA_DebitorCreditorID": 2,
      "Identity number": "2",

```

```

"Employee": "Person",
"Cost centre": "New cost center",
"Year": 2023,
"Month": 1,
"A": 1000.00000000,
"J": 22.00000000,
"AA": 22
},
{
"OA_SalaryAccruedsID": 134,
"OA_EmployeeID": 12,
"OA_DebitorCreditorID": 2,
"Identity number": "2",
"Employee": "Person",
"Cost centre": "New cost center",
"Year": 2023,
"Month": 2,
"A": 1000.00000000,
"J": 20.00000000,
"AA": 20
}
]
}

```

2. Get all data from salary accrued archives
Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"topRecordsCount": 0
}

```

3. Get first all data from salary accrued archives by year – 2023
Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"topRecordsCount": 0,
"filter": "[Year]=2023"
}

```

Fixed assets

FixedAssets

POST/api/FixedAsset

Adds Fixed Asset. Returns ID value or Error description Json.

Schema

```

{
"token": "string",
"balanceCost": 0,
"balanceCostParentAccount": "string",
"balanceCostParentAccountOA_AccountsID": 0,
"balanceCostRevaluationAmount": 0,
"chargingMonthsQuantity": 0,
"closeDate": "2023-05-03T13:09:29.968Z",
"closedByDesintegration": true,
"closedByIntegration": true,
"code": "string",
"comment": "string",
"costCentre": "string",
"currencies": "string",
"dateFunctioningBegan": "2023-05-03T13:09:29.968Z",
"dateOfExpiration": "2023-05-03T13:09:29.968Z",

```

```

"department": "string",
"depreciationAccrueAmount": 0,
"depreciationAccrueMethodInt": 0,
"depreciationCost": 0,
"incomeDate": "2023-05-03T13:09:29.968Z",
"inventoryNumber": "string",
"itemGroup": "string",
"itemWriteOffExpenseAccount": "string",
"liquidationValue": 0,
"manufacturer": "string",
"manufacturerNumber": "string",
"name": "string",
"notValuable": true,
"oA_AccountsIDExpenseAccount": 0,
"oA_CostCentreID": 0,
"oA_CurrenciesID": 0,
"oA_DebitorCreditorID": 0,
"oA_EmployeeDepartmentID": 0,
"oA_ItemGroupID": 0,
"oA_ManufacturersID": 0,
"oA_StockUnitsID": 0,
"productionDate": "2023-05-03T13:09:29.968Z",
"productionQuantityChargingPeriod": 0,
"productionQuantityServiceLife": 0,
"quantity": 0,
"remainedMonthsQuantityServiceLife": 0,
"responsiblePersonIdentityNumber": "string",
"salePrice": 0,
"serviceLife": 0,
"technicalDatas": "string",
"totalWear": 0,
"totalWearByPurchaseDocument": 0,
"totalWearOA_AccountsID": 0,
"totalWearRevaluationAmount": 0,
"transmitted": "2023-05-03T13:09:29.968Z",
"unit": "string",
"vat": 0,
"vatType": 0,
"wearAccrueParentAccount": "string",
"writtenOff": true,
"yearDepreciationCoefficient": 0
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"**token**": Set [token] string from [Login] response value

"**balanceCost**":

"**balanceCostParentAccount**": See [AccountsList] method (use [Account] value). Can be omitted if [BalanceCostParentAccountOA_AccountsID] property specified.

"**balanceCostParentAccountOA_AccountsID**": Balance Cost parent account. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [BalanceCostParentAccount] property specified.

"**balanceCostRevaluationAmount**":

"**chargingMonthsQuantity**":

"**closeDate**":

"**closedByDesintegration**":

"**closedByIntegration**":

"**code**":

"**comment**":

"**costCentre**": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified.

"**currencies**":

"**dateFunctioningBegan**":

"**dateOfExpiration**":

"**department**": "See [EmployeeDepartmentsList] method (use [Name] value). Can be omitted if [OA_EmployeeDepartmentID] property specified.

"**depreciationAccrueAmount**":

"**depreciationAccrueMethodInt**": Depreciation accrue method. None = 0, Taxation = 1, Linear = 2, WithProduction =, SumOfNumbers = 4, Doubled = 5, Budgetary = 6

"**depreciationCost**":

"**incomeDate**": "2023-05-03T11:26:13.024Z",

"**inventoryNumber**":

"**itemGroup**": See [ItemGroupsList] method (use [Name] value). Can be omitted if [OA_ItemGroupID] property specified.

"**itemWriteOffExpenseAccount**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_AccountsIDExpenseAccount] property specified.

"**liquidationValue**":

"**manufacturer**": See [ManufacturersList] method (use [Name] value). Can be omitted if [OA_ManufacturersID] property specified.

"**manufacturerNumber**":

"**name**": Name of fixed asset.

"**notValuable**":

"**oA_AccountsIDExpenseAccount**": Will be used in managarial transaction. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [ItemWriteOffExpenseAccount] property specified.

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If ommite, gets cost centre from parameters.

"**oA_CurrenciesID**": Sets sale price currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currencies] property specified. "**oA_DebitorCreditorID**": Responsible Person. See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [ResponsiblePerson] property specified.

"**oA_EmployeeDepartmentID**": See [EmployeeDepartmentsList] method (use [OA_EmployeeDepartmentID] value). Can be omitted if [Department] property specified.

"**oA_ItemGroupID**": See [ItemGroupsList] method (use [OA_ItemGroupID] value). Can be omitted if [ItemGroup] property specified.

"**oA_ManufacturersID**": See [ManufacturersList] method (use [OA_ManufacturersID] value). Can be omitted if [Manufacturer] property specified.

"**oA_StockUnitsID**": Registe9r minimum unit. See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified.

"**productionDate**":

"**productionQuantityChargingPeriod**":

"**productionQuantityServiceLife**":

"**quantity**":

"**remainedMonthsQuantityServiceLife**":

"**responsiblePersonIdentityNumber**":

"**salePrice**":

"**serviceLife**":

"**technicalDatas**":

"**totalWear**":

"**totalWearByPurchaseDocument**":

"**totalWearOA_AccountstID**": Parent account for total wear. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [WearAccrueParentAccount] property specified

"**totalWearRevaluationAmount**":

"**transmetted**": "2023-05-03T11:26:13.024Z",

"**unit**": Register minimum unit. See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified.

"**vat**":

"**vatType**": Standard = 0, ZeroRate = 1, Untaxed = 2

"**wearAccrueParentAccount**": "Parent account for total wear. See [AccountsList] method (use [Account] value). Can be omitted if [TotalWearOA_AccountstID] property specified.

"**writtenOff**":

"**yearDepreciationCoefficient**":

Examples

1. Add new FixedAssets, name- FixedAsset, code -21, balanceCostParentAccount – "2110", wearAccrueParentAccount – "2210", balanceCost - 200, depreciationAccrueMethodInt – "Linear".
Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "balanceCost": 200,  
  "balanceCostParentAccount": "2110",  
  "code": "21",  
  "name": "FixedAsset",  
  "quantity": 1,  
  "wearAccrueParentAccount": "2210",
```

```
"depreciationAccrueMethodInt": 2,
"depreciationCost": 170,
"liquidationValue": 180,
"totalWear": 200,
"serviceLife": 2,
"totalWearByPurchaseDocument": 1,
"chargingMonthsQuantity": 1
}
```

Server response Code (200):

```
{
  "id": 1017,
  "idFieldName": "OA_ItemsID"
}
```

Server response Code (400):

```
{
  "errorNumber": 51190,
  "description": "Property not found!",
  "comment": "OA_ItemsID=0 [1032]\n[BalanceCostParentAccount]=2195",
  "command": "",
  "errorType": "ErrorType_Error",
  "source": "OA_BusinessObject.dll",
  "class": "FixedAssets",
  "entryPoint": "set property",
  "resourceDLL_ForTranslation": "ORIS_Accounting",
  "errorPointer": 0
},
{
  "errorNumber": 51190,
  "description": "Property not found!",
  "comment": "OA_ItemsID=0 [22]\n[WearAccrueParentAccount]=2295",
  "command": "",
  "errorType": "ErrorType_Error",
  "source": "OA_BusinessObject.dll",
  "class": "FixedAssets",
  "entryPoint": "set property",
  "resourceDLL_ForTranslation": "ORIS_Accounting",
  "errorPointer": 0
}
```

PUT/api/FixedAsset

Updates Fixed Asset. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_ItemsID": 0,
  "balanceCost": 0,
  "balanceCostParentAccount": "string",
  "balanceCostParentAccountOA_AccountsID": 0,
  "balanceCostRevaluationAmount": 0,
  "chargingMonthsQuantity": 0,
  "closeDate": "2023-05-18T07:46:17.327Z",
  "closedByDesintegration": true,
  "closedByIntegration": true, "code": "string",
  "comment": "string",
  "costCentre": "string",
  "currencies": "string",
  "dateFunctioningBegan": "2023-05-18T07:46:17.327Z",
  "dateOfExpiration": "2023-05-18T07:46:17.327Z",
  "department": "string",
  "depreciationAccrueAmount": 0,
  "depreciationAccrueMethodInt": 0,
}
```

```

"depreciationCost": 0,
"incomeDate": "2023-05-18T07:46:17.327Z",
"inventoryNumber": "string",
"itemGroup": "string",
"itemWriteOffExpenseAccount": "string",
"liquidationValue": 0,
"manufacturer": "string",
"manufacturerNumber": "string",
"name": "string",
"notValuable": true,
"oA_AccountsIDExpenseAccount": 0,
"oA_CostCentreID": 0,
"oA_CurrenciesID": 0,
"oA_DebitorCreditorID": 0,
"oA_EmployeeDepartmentID": 0,
"oA_ItemGroupID": 0,
"oA_ManufacturersID": 0,
"oA_StockUnitsID": 0,
"productionDate": "2023-05-18T07:46:17.327Z",
"productionQuantityChargingPeriod": 0,
"productionQuantityServiceLife": 0,
"quantity": 0,
"remainedMonthsQuantityServiceLife": 0,
"responsiblePersonIdentityNumber": "string",
"salePrice": 0,
"serviceLife": 0,
"technicalDatas": "string",
"totalWear": 0,
"totalWearByPurchaseDocument": 0,
"totalWearOA_AccountsID": 0,
"totalWearRevaluationAmount": 0,
"transmitted": "2023-05-18T07:46:17.327Z",
"unit": "string",
"vat": 0,
"vatType": 0,
"wearAccrueParentAccount": "string",
"writtenOff": true,
"yearDepreciationCoefficient": 0
}

```

Field description

[Token] and [OA_ItemsID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_ItemsID": See [FixedAssetsList] method (use [OA_ItemsID] value).

"balanceCost":

"balanceCostParentAccount": See [AccountsList] method (use [Account] value). Can be omitted if [BalanceCostParentAccountOA_AccountsID] property specified.

"balanceCostParentAccountOA_AccountsID": Balance Cost parent account. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [BalanceCostParentAccount] property specified.0,

"balanceCostRevaluationAmount":

"chargingMonthsQuantity":

"closeDate":

"closedByDesintegration":

"closedByIntegration":

"code":

"comment":

"costCentre": See [CostCentresList] method (use [Name] value). Can be omitted if [OA_CostCentreID] property specified."string",

"currencies": "Sets sale price currency. See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified.string",

"dateFunctioningBegan":

"dateOfExpiration":

"department": See [EmployeeDepartmentsList] method (use [Name] value). Can be omitted if [OA_EmployeeDepartmentID] property specified.

"depreciationAccrueAmount":

"**depreciationAccrueMethodInt**": Depreciation accrue method. None = 0, Taxation = 1, Linear = 2, WithProduction = 3, SumOfNumbers = 4, Doubled = 5, Budgetary = 6

"**depreciationCost**":

"**incomeDate**":

"**inventoryNumber**":

"**itemGroup**": See [ItemGroupsList] method (use [Name] value). Can be omitted if [OA_ItemGroupID] property specified.

"**itemWriteOffExpenseAccount**": See [AccountsList] method (use [Account] value). Can be omitted if [OA_AccountsIDExpenseAccount] property specified."string",

"**liquidationValue**":

"**manufacturer**": See [ManufacturersList] method (use [Name] value). Can be omitted if [OA_ManufacturersID] property specified.

"**manufacturerNumber**":

"**name**": Name of fixed asset.

"**notValuable**":

"**oA_AccountsIDExpenseAccount**": Will be used in managarial transaction. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [ItemWriteOffExpenseAccount] property specified.

"**oA_CostCentreID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [CostCentre] property specified. If ommite, gets cost centre from parameters

"**oA_CurrenciesID**": Sets sale price currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currencies] property specified.

"**oA_DebitorCreditorID**": Responsible Person. See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [ResponsiblePerson] property specified.

"**oA_EmployeeDepartmentID**": See [EmployeeDepartmentsList] method (use [OA_EmployeeDepartmentID] value). Can be omitted if [Department] property specified.

"**oA_ItemGroupID**": See [ItemGroupsList] method (use [OA_ItemGroupID] value). Can be omitted if [ItemGroup] property specified.

"**oA_ManufacturersID**": See [ManufacturersList] method (use [OA_ManufacturersID] value). Can be omitted if [Manufacturer] property specified.

"**oA_StockUnitsID**": Register minimum unit. See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified.

"**productionDate**":

"**productionQuantityChargingPeriod**":

"**productionQuantityServiceLife**":

"**quantity**":

"**remainedMonthsQuantityServiceLife**":

"**responsiblePersonIdentityNumber**":

"**salePrice**":

"**serviceLife**":

"**technicalDatas**":

"**totalWear**":

"**totalWearByPurchaseDocument**": 0,

"**totalWearOA_AccountstID**":Parent account for total wear. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [WearAccrueParentAccount] property specified

"**totalWearRevaluationAmount**":

"**transmetted**":

"**unit**": Register minimum unit. See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified.

"**vat**":

"**vatType**": Standard = 0, ZeroRate = 1, Untaxed = 2

"**wearAccrueParentAccount**": Parent account for total wear. See [AccountsList] method (use [Account] value). Can be omitted if [TotalWearOA_AccountstID] property specified

"**writtenOff**":

"**yearDepreciationCoefficient**":

Examples

1. Update FixedAssets. Name - FixedAsset, code -21, balanceCostParentAccount – "2150", wearAccrueParentAccount – "2250", balanceCost- 150,using "oA_ItemsID" - 2004.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ItemsID": 2004,
  "balanceCost": 150,
  "balanceCostParentAccount": "2150",
  "code": "21",
  "name": "FixedAsset",
```



```
"quantity": 1,
"wearAccrueParentAccount": "2250",
"depreciationAccrueMethodInt": 2,
"depreciationCost": 170,
"liquidationValue": 180,
"totalWear": 200,
"serviceLife": 2,
"totalWearByPurchaseDocument": 1,
"chargingMonthsQuantity": 1
}
```

Server response Code (200):

```
{
  "id": 2004,
  "idFieldName": "OA_ItemsID"
}
```

Server response Code (400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_ItemsID = 2004",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "FixedAssets",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting", "errorPointer": 0
  }
]
```

DELETE/api/FixedAsset

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_ItemsID": 0
}
```

Field description

[Token] and [OA_ItemsID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_ItemsID":

Examples

1. Delete FixedAssets from list using "oA_ItemsID" - 2002.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ItemsID": 2002
}
```

Server response Code(200):

```
{
  "id": 2002,
  "idFieldName": "OA_ItemsID"
}
```

Server response Code(400):

```
[
  {
```

```

"errorNumber": 51051,
"description": "Record does not exist or read prohibited for user!",
"comment": "OA_ItemsID = 2005",
"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "FixedAssets",
"entryPoint": "GetProperties",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

FixedAssetsList

POST/api/FixedAssetsList

Gets Fixed Assets. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get fixed assets list from database in specified filter - fixed assets name (FixedAssets).

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 6,
  "filter": "UPPER([ViewOA_FixedAssets].[Name]) LIKE N'%FIXEDASSET'",
  "pageNumber": 1
}

```

Server response Code (200):

```

{
  "Info": [
    {
      "RowCount": 27,
      "PageCount": 5,
      "PageRowCount": 6,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_ItemsID": 1006,
      "Code": "21",
      "Name": "FixedAsset",
      "Item type": 1,
      "Depreciation accrue method": 0,

```

```
"Inventory number": "",
"Balance cost": 200.00000000,
"Quantity": 0.00000000,
"Liquidation value": 0.00000000,
"Service life": 0.00000000,
"Production quantity service life": 0.00000000,
"Production quantity charging period": 0.00000000,
"Remained months quantity service life": 0.00000000,
"Year depreciation coefficient": 0.00000000,
"Charging months quantity": 0.00000000,
"Depreciation cost": 0.00000000,
"Depreciation accrue amount": 0.00000000,
"Total wear": 0.00000000,
"Balance cost revaluation amount": 0.00000000,
"VAT": 0.00000000,
"Balance cost parent account": "2110",
"Comment": "",
"OA_AccountsID": 82,
"Manufacturer number": "",
"Written off": false,
"Total wear by purchase document": 0.00000000,
"Sale price": 0.00000000,
"Not valuable": false,
"Total wear revaluation amount": 0.00000000,
"Closed by desintegration": false,
"Closed by integration": false,
"VAT type": 0,
"NUM_ROW": 1
},
{
"OA_ItemsID": 1007,
"Code": "21",
"Name": "FixedAsset",
"Item type": 1,
"Depreciation accrue method": 0,
"Inventory number": "",
"Balance cost": 200.00000000,
"Quantity": 0.00000000,
"Liquidation value": 0.00000000,
"Service life": 0.00000000,
"Production quantity service life": 0.00000000,
"Production quantity charging period": 0.00000000,
"Remained months quantity service life": 0.00000000,
"Year depreciation coefficient": 20.00000000,
"Charging months quantity": 1.00000000,
"Depreciation cost": 0.00000000,
"Depreciation accrue amount": 0.00000000,
"Total wear": 0.00000000,
"Balance cost revaluation amount": 0.00000000,
"VAT": 0.00000000,
"Balance cost parent account": "2110",
"Comment": "",
"OA_AccountsID": 82,
"Manufacturer number": "",
"Written off": false,
"Total wear by purchase document": 0.00000000,
"Sale price": 0.00000000,
"TotalWearOA_AccountsID": 91,
"Wear accrue parent account": "2210",
"Not valuable": false,
"Total wear revaluation amount": 0.00000000,
"Closed by desintegration": false,
"Closed by integration": false,
"VAT type": 0,
"NUM_ROW": 2
},
{
```

"OA_ItemsID": 1008,
"Code": "21",
"Name": "FixedAsset",
"Item type": 1,
"Depreciation accrue method": 0,
"Inventory number": "",
"Balance cost": 200.00000000,
"Quantity": 0.00000000,
"Liquidation value": 0.00000000,
"Service life": 0.00000000,
"Production quantity service life": 0.00000000,
"Production quantity charging period": 0.00000000,
"Remained months quantity service life": 0.00000000,
"Year depreciation coefficient": 0.00000000,
"Charging months quantity": 0.00000000,
"Depreciation cost": 0.00000000,
"Depreciation accrue amount": 0.00000000,
"Total wear": 0.00000000,
"Balance cost revaluation amount": 0.00000000,
"VAT": 0.00000000,
"Balance cost parent account": "2110",
"Comment": "",
"OA_AccountsID": 82,
"Manufacturer number": "",
"Written off": false,
"Total wear by purchase document": 0.00000000,
"Sale price": 0.00000000,
"TotalWearOA_AccountsID": 91,
"Wear accrue parent account": "2210",
"Not valuable": false,
"Total wear revaluation amount": 0.00000000,
"Closed by desintegration": false,
"Closed by integration": false,
"VAT type": 0,
"NUM_ROW": 3

},
{

"OA_ItemsID": 1009,
"Code": "21",
"Name": "FixedAsset",
"Item type": 1,
"Depreciation accrue method": 0,
"Inventory number": "",
"Balance cost": 200.00000000,
"Quantity": 0.00000000,
"Liquidation value": 0.00000000,
"Service life": 0.00000000,
"Production quantity service life": 0.00000000,
"Production quantity charging period": 0.00000000,
"Remained months quantity service life": 0.00000000,
"Year depreciation coefficient": 0.00000000,
"Charging months quantity": 0.00000000,
"Depreciation cost": 0.00000000,
"Depreciation accrue amount": 0.00000000,
"Total wear": 0.00000000,
"Balance cost revaluation amount": 0.00000000,
"VAT": 0.00000000,
"Balance cost parent account": "2110",
"Comment": "",
"OA_AccountsID": 82,
"Manufacturer number": "",
"Written off": false,
"Total wear by purchase document": 0.00000000,
"Sale price": 0.00000000,
"TotalWearOA_AccountsID": 100,
"Wear accrue parent account": "2310",
"Not valuable": false,

```
"Total wear revaluation amount": 0.00000000,
"Closed by desintegration": false,
"Closed by integration": false,
"VAT type": 0,
"NUM_ROW": 4
},
{
  "OA_ItemsID": 1010,
  "Code": "21",
  "Name": "FixedAsset",
  "Item type": 1,
  "Depreciation accrue method": 0,
  "Inventory number": "",
  "Production date": "2023-05-10T12:32:49",
  "Balance cost": 200.00000000,
  "Quantity": 0.00000000,
  "Liquidation value": 0.00000000,
  "Service life": 0.00000000,
  "Production quantity service life": 0.00000000,
  "Production quantity charging period": 0.00000000,
  "Remained months quantity service life": 0.00000000,
  "Year depreciation coefficient": 0.00000000,
  "Charging months quantity": 0.00000000,
  "Depreciation cost": 0.00000000,
  "Depreciation accrue amount": 0.00000000,
  "Total wear": 0.00000000,
  "Balance cost revaluation amount": 0.00000000,
  "VAT": 0.00000000,
  "Balance cost parent account": "2110",
  "Comment": "",
  "Unit": "ცალი",
  "OA_AccountsID": 82,
  "OA_StockUnitsID": 4,
  "Manufacturer number": "",
  "Written off": false,
  "Total wear by purchase document": 0.00000000,
  "Sale price": 0.00000000,
  "TotalWearOA_AccountsID": 95,
  "Wear accrue parent account": "2250",
  "Not valuable": false,
  "OA_CostCentreID": 1,
  "Cost centre": "General cost center",
  "Total wear revaluation amount": 0.00000000,
  "Closed by desintegration": false,
  "Closed by integration": false,
  "VAT type": 0,
  "OA_CostUnitID": 1,
  "Cost unit": "Total cost unit",
  "NUM_ROW": 5
},
{
  "OA_ItemsID": 1011,
  "Code": "22",
  "Name": "FixedAsset",
  "Item type": 1,
  "Depreciation accrue method": 0,
  "Inventory number": "",
  "Balance cost": 200.00000000,
  "Quantity": 0.00000000,
  "Liquidation value": 0.00000000,
  "Service life": 0.00000000,
  "Production quantity service life": 0.00000000,
  "Production quantity charging period": 0.00000000,
  "Remained months quantity service life": 0.00000000,
  "Year depreciation coefficient": 0.00000000,
  "Charging months quantity": 0.00000000,
  "Depreciation cost": 0.00000000,
```

```

    "Depreciation accrue amount": 0.00000000,
    "Total wear": 0.00000000,
    "Balance cost revaluation amount": 0.00000000,
    "VAT": 0.00000000,
    "Balance cost parent account": "2110",
    "Comment": "",
    "OA_AccountsID": 82,
    "Manufacturer number": "",
    "Written off": false,
    "Total wear by purchase document": 0.00000000,
    "Sale price": 0.00000000,
    "TotalWearOA_AccountsID": 91,
    "Wear accrue parent account": "2210",
    "Not valuable": false,
    "Total wear revaluation amount": 0.00000000,
    "Closed by desintegration": false,
    "Closed by integration": false,
    "VAT type": 0,
    "NUM_ROW": 6
  }
}
}

```

2. Get first FixedAssets from list.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "pageNumber": 1
}

```

FixedAssetsStatementsList

Post/api/FixedAssetsStatementsList

Gets Fixed Asset Statement List. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "currenciesIDs": "string",
  "costCentreIDs": "string",
  "costUnitIDs": "string",
  "diapazonStartDate": "2023-05-03T13:34:42.836Z",
  "diapazonEndDate": "2023-05-03T13:34:42.836Z",
  "showZeroResults": true,
  "showStockValues": true,
  "includeInternalTurns": true,
  "itemID": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"currenciesIDs":

"costCentreIDs":

"costUnitIDs":

"diapazonStartDate": Diapazon start date

"diapazonEndDate": Diapazon end date (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01", "DiapazonEndDate":"2011-01-01"

"showZeroResults":

"showStockValues":

"includeInternalTurns":

"itemID":

Examples

1. Get Fixed Assets Statements List from database in specified diapason, using "itemID" - 1006
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "diapazonStartDate": "2020-05-04",
  "diapazonEndDate": "2023-05-10",
  "itemID": 1006
}
```

Server response Code (200):

```
{
  "Info": [
    {
      "RowCount": 3,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "Correspondent account": "Start",
      "Delay": false,
      "Documental confirm": false,
      "Main entry": false,
      "Debit balance": 0.0,
      "Credit balance": 0.0,
      "Debit equivalent balance": 0.0,
      "Credit equivalent balance": 0.0,
      "Debit equivalent2 balance": 0.0,
      "Credit equivalent2 balance": 0.0,
      "Stock balance": 0.0,
      "SortType": 0
    },
    {
      "OA_EntriesID": 4041,
      "OA_AccountsID": 4217,
      "ParentID": 41,
      "CorrAccountID": 198,
      "Correspondent account": "7455",
      "Category": 0,
      "Correspondent account name": "Depreciation and amortization",
      "OA_CurrenciesID": 1,
      "OA_TransactionsID": 5029,
      "OA_CostCentreID": 2,
      "OA_CostUnitID": 1,
      "Delay": false,
      "Documental confirm": false,
      "Transaction date": "2023-05-09T16:51:14",
      "Document number": "",
      "Debit amount": 0.0,
      "Credit amount": 0.4,
      "Debit equivalent": 0.0,
      "Credit equivalent": 0.4,
      "Debit equivalent2": 0.0,
      "Credit equivalent2": 0.0,
      "Debit stock": 0.0,
      "Credit stock": 0.0,
      "Account": "2210 1006",
      "TransactionTypesID": 15,
      "Transaction type": "Fixad assets depreciation accrue",
      "Main entry": false,
      "Is start balance": false,
      "Transaction comment": "ცვეთის დარიცხვა",
      "Debit balance": 0.0,

```

```

"Credit balance": 0.4,
"Debit equivalent balance": 0.0,
"Credit equivalent balance": 0.4,
"Debit equivalent2 balance": 0.0,
"Credit equivalent2 balance": 0.0,
"Stock balance": 0.0,
"SortType": 1,
"Cost centre": "New cost center",
"OA_UsersID": 1,
"User": "sa",
"SequenceNumber": 2,
"Currency": "GEL",
"Cost unit": "Total cost unit"
},
{
"Correspondent account": "Final",
"Delay": false,
"Documental confirm": false,
"Debit amount": 0.0,
"Credit amount": 0.4,
"Debit equivalent": 0.0,
"Credit equivalent": 0.4,
"Debit equivalent2": 0.0,
"Credit equivalent2": 0.0,
"Main entry": false,
"Credit balance": 0.4,
"Credit equivalent balance": 0.4,
"Credit equivalent2 balance": 0.0,
"Stock balance": 0.0,
"SortType": 2
}
}

```

1. Get Fixed Assets Statements Detailed List from database using "itemID" -1007, "costCentreIDs" 1 - General cost center, "costUnitIDs" 1 - Total cost unit.

Request body

```

{
"token": "... paste here [Login] response value ...",
"costCentreIDs": "1",
"costUnitIDs": "1",
"diapazonStartDate": "01.05.2020",
"diapazonEndDate": "19.05.2023",
"showZeroResults": true,
"showStockValues": true,
"includeInternalTurns": true,
"itemID": 1007
}

```

FixedAssetsStatementsDetailedList

POST/api/FixedAssetsStatementsDetailedList

Gets Fixed Asset Statement Detailed List. Returns [Info] and [Result] tables.

Schema

```

{
"token": "string",
"currenciesIDs": "string",
"costCentreIDs": "string",
"costUnitIDs": "string",
"diapazonStartDate": "2023-05-03T13:11:12.889Z",
"diapazonEndDate": "2023-05-03T13:11:12.889Z",
"showZeroResults": true,
"showStockValues": true,
"includeInternalTurns": true,
"itemID": 0
}

```



```
}
```

Field description

"token": Set [token] string from [LogIn] response value

"currenciesIDs":

"costCentreIDs":

"costUnitIDs":

"diapazonStartDate": Diapazon start date

"diapazonEndDate": Diapazon end date (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01", "DiapazonEndDate":"2011-01-01"

"showZeroResults":

"showStockValues":

"includeInternalTurns":

"itemID":

Examples

1. Get Fixed Assets Statements DetailedList from database in specified diapason using "itemID" - 1008

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "diapazonStartDate": "2020-05-04",
  "diapazonEndDate": "2023-05-10",
  "itemID": 1008
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 3,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "Correspondent account": "Start",
      "Delay": false,
      "Documental confirm": false,
      "Main entry": false,
      "Debit balance": 0.0,
      "Credit balance": 0.0,
      "Debit equivalent balance": 0.0,
      "Credit equivalent balance": 0.0,
      "Debit equivalent2 balance": 0.0,
      "Credit equivalent2 balance": 0.0,
      "Stock balance": 0.0,
      "SortType": 0
    },
    {
      "OA_EntriesID": 4045,
      "OA_AccountsID": 1229,
      "CorrAccountID": 198,
      "Correspondent account": "7455",
      "Category": 0,
      "Correspondent account name": "Depreciation and amortization",
      "OA_CurrenciesID": 1,
      "OA_TransactionsID": 5029,
      "OA_CostCentreID": 2,
      "OA_CostUnitID": 1,
      "Delay": false,
      "Documental confirm": false,
      "Transaction date": "2023-05-09T16:51:14",
      "Debit amount": 0.0,
    }
  ]
}
```

```

    "Credit amount": 0.02,
    "Debit equivalent": 0.0,
    "Credit equivalent": 0.02,
    "Debit equivalent2": 0.0,
    "Credit equivalent2": 0.0,
    "Debit stock": 0.0,
    "Credit stock": 0.0,
    "Account": "2210 1008",
    "TransactionTypesID": 15,
    "Transaction type": "Fixad assets depreciation accrue",
    "Main entry": false,
    "Is start balance": false,
    "Transaction comment": "ცვეთის დარიცხვა",
    "Debit balance": 0.0,
    "Credit balance": 0.02,
    "Debit equivalent balance": 0.0,
    "Credit equivalent balance": 0.02,
    "Debit equivalent2 balance": 0.0,
    "Credit equivalent2 balance": 0.0,
    "Stock balance": 0.0,
    "SortType": 1,
    "Is reverse entry": false,
    "Document number": "",
    "Cost centre": "New cost center",
    "OA_UsersID": 1,
    "User": "sa",
    "SequenceNumber": 6,
    "Currency": "GEL",
    "Cost unit": "Total cost unit"
  },
  {
    "Correspondent account": "Final",
    "Delay": false,
    "Documental confirm": false,
    "Debit amount": 0.0,
    "Credit amount": 0.02,
    "Debit equivalent": 0.0,
    "Credit equivalent": 0.02,
    "Debit equivalent2": 0.0,
    "Credit equivalent2": 0.0,
    "Main entry": false,
    "Credit balance": 0.02,
    "Credit equivalent balance": 0.02,
    "Credit equivalent2 balance": 0.0,
    "Stock balance": 0.0,
    "SortType": 2
  }
]
}

```

2. Get FixedAssetsStatementsDetailedList from database "itemID" -1010, "costCentreIDs" 1 - General cost center, "costUnitIDs" 1 -Total cost unit.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "costCentreIDs": "1",
  "costUnitIDs": "1",
  "diapazonStartDate": "04.05.2020",
  "diapazonEndDate": "11.05.2023",
  "showZeroResults": true,
  "showStockValues": true,
  "includeInternalTurns": true,
  "itemID": 1010
}

```

Manufacturing

ManufacturingDesign

POST/api/ManufacturingDesign

Adds new Manufacturing Design. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "oA_ManufacturingDesignID": 0,
  "code": "string",
  "comment": "string",
  "name": "string",
  "oA_ItemsID": 0,
  "oA_StockUnitsID": 0,
  "quantity": 0,
  "unit": "string",
  "manufacturingDesignBOMs": [
    {
      "oA_ManufacturingDesignBOMID": 0,
      "oA_ManufacturingDesignID": 0,
      "oA_ItemsID": 0,
      "oA_StockUnitsID": 0,
      "code": "string",
      "unit": "string",
      "sequenceNumber": 0,
      "itemType": 0,
      "materialName": "string",
      "quantity": 0,
      "oA_CostDriverID": 0,
      "costDriver": "string"
    }
  ]
}
```

Field description

"token": Set [token] string from [LogIn] response valu

"oA_ManufacturingDesignID":

"code":

"comment":

"name":

"oA_ItemsID":

"oA_StockUnitsID":

"quantity":

"unit":

"manufacturingDesignBOMs":

"oA_ManufacturingDesignBOMID":

"oA_ManufacturingDesignID":

"oA_ItemsID":

"oA_StockUnitsID":

"code":

"unit":

"sequenceNumber":

"itemType":

"materialName":

"quantity":

"oA_CostDriverID":

"costDriver":

Examples

1. Add new ManufacturingDesign "code" -1032, name-cake, code -1033, materialName-egg.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ManufacturingDesignID": 13,
  "code": "1032",
  "name": "cake",
  "oA_ItemsID": 1032,
  "oA_StockUnitsID": 1,
  "manufacturingDesignBOMs": [
    {
      "oA_ManufacturingDesignBOMID": 112,
      "oA_ManufacturingDesignID": 13,
      "oA_ItemsID": 1033,
      "code": "1033",
      "materialName": "egg"
    }
  ]
}
```

Server response Code (200):

```
{
  "id": 27,
  "idFieldName": "OA_ManufacturingDesignID"
}
```

Server response Code (400):

```
[
  {
    "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
    "title": "One or more validation errors occurred.",
    "status": 400,
    "traceId": "00-f78f003dc3471faacdde796ed54d18b6-3ca64b7d0395bdcd-00",
    "errors": {
      "manufacturingDesign": [
        "The manufacturingDesign field is required." ],
      "$. manufacturingDesignBOMs[0]": [
        "'1' is an invalid start of a property name. Expected a '\\\"'. Path: $. manufacturingDesignBOMs[0] | LineNumber: 12 | BytePositionInLine: 25."
      ]
    }
  }
]
```

2. Add new ManufacturingDesign code-3011, name-cake2, materialName- floure. code-1035.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ManufacturingDesignID": 0,
  "code": "3011",
  "name": "cake2",
  "oA_ItemsID": 4003,
  "oA_StockUnitsID": 1,
  "unit": "piece",
  "manufacturingDesignBOMs": [
    {
      "oA_ManufacturingDesignBOMID": 2118,
      "oA_ManufacturingDesignID": 2013,
      "oA_ItemsID": 1035,
      "code": "1035",
      "unit": "piece",
      "sequenceNumber": 1,
      "itemType": 0,
      "materialName": "floure"
    }
  ]
}
```

```
}
```

PUT/api/ManufacturingDesign

Updates Manufacturing Design.Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_ManufacturingDesignID": 0,
  "code": "string",
  "comment": "string",
  "name": "string",
  "oA_ItemsID": 0,
  "oA_StockUnitsID": 0,
  "quantity": 0,
  "unit": "string",
  "manufacturingDesignBOMs": [
    {
      "oA_ManufacturingDesignBOMID": 0,
      "oA_ManufacturingDesignID": 0,
      "oA_ItemsID": 0,
      "oA_StockUnitsID": 0,
      "code": "string",
      "unit": "string",
      "sequenceNumber": 0,
      "itemType": 0,
      "materialName": "string",
      "quantity": 0,
      "oA_CostDriverID": 0,
      "costDriver": "string"
    }
  ]
}
```

Field description

"token": Set [token] string from [LogIn] response value

"oA_ManufacturingDesignID":

"code":

"comment":

"name":

"oA_ItemsID":

"oA_StockUnitsID":

"quantity":

"unit":

"manufacturingDesignBOMs":

"oA_ManufacturingDesignBOMID":

"oA_ManufacturingDesignID":

"oA_ItemsID":

"oA_StockUnitsID":

"code":

"unit":

"sequenceNumber":

"itemType":

"materialName":

"quantity":

"oA_CostDriverID":

"costDriver":

Examples

1. Update Manufacturing Design. oA_ManufacturingDesignID-27, Code-3010, name-sweetcake, OA_ItemsID-3010.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ManufacturingDesignID": 27,
  "code": "3010",
  "name": "sweetcake",
  "oA_ItemsID": 3010,
  "oA_StockUnitsID": 1,
  "manufacturingDesignBOMs": [
    {
      "oA_ManufacturingDesignBOMID": 1116,
      "oA_ManufacturingDesignID": 27,
      "oA_ItemsID": 1035,
      "code": "1035",
      "materialName": "floure"
    }
  ]
}
```

Server response Code (200):

```
{
  "id": 27,
  "idFieldName": "OA_ManufacturingDesignID"
}
```

Server response Code (400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_ManufacturingDesignID = 13",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "ManufacturingDesign",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

2. Update Manufacturing Design. oA_ManufacturingDesignID-1014, Code-300, name"-sweetcake, OA_ItemsID-3010, oA_ManufacturingDesignID-1014, oA_ItemsID-1033, Code-1033, materialName-egg.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ManufacturingDesignID": 1014,
  "code": "3009",
  "name": "sweetcake",
  "oA_ItemsID": 3010,
  "oA_StockUnitsID": 1,
  "manufacturingDesignBOMs": [
    {
      "oA_ManufacturingDesignBOMID": 1121,
      "oA_ManufacturingDesignID": 1014,
      "oA_ItemsID": 1033,
      "code": "1033",
      "materialName": "egg"
    }
  ]
}
```

DELETE/api/ManufacturingDesign

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_ManufacturingDesignID": 0,
  "code": "string",
  "comment": "string",
  "name": "string",
  "oA_ItemsID": 0,
  "oA_StockUnitsID": 0,
  "quantity": 0,
  "unit": "string",
  "manufacturingDesignBOMs": [
    {
      "oA_ManufacturingDesignBOMID": 0,
      "oA_ManufacturingDesignID": 0,
      "oA_ItemsID": 0,
      "oA_StockUnitsID": 0,
      "code": "string",
      "unit": "string",
      "sequenceNumber": 0,
      "itemType": 0,
      "materialName": "string",
      "quantity": 0,
      "oA_CostDriverID": 0,
      "costDriver": "string"
    }
  ]
}
```

Field description

"token": Set [token] string from [LogIn] response value

"oA_ManufacturingDesignID":

"code":

"comment":

"name":

"oA_ItemsID":

"oA_StockUnitsID":

"quantity":

"unit":

"manufacturingDesignBOMs":

"oA_ManufacturingDesignBOMID":

"oA_ManufacturingDesignID":

"oA_ItemsID":

"oA_StockUnitsID":

"code":

"unit":

"sequenceNumber":

"itemType":

"materialName":

"quantity":

"oA_CostDriverID":

"costDriver":

Examples

1. Delete "oA_ManufacturingDesignID-1013, oA_ManufacturingDesignBOMID-1119.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ManufacturingDesignID": 1013,
  "manufacturingDesignBOMs": [
    {
      "oA_ManufacturingDesignBOMID": 1119,
```

```
"oA_ManufacturingDesignID": 1013
}
]
}
```

Server response Code (200):

```
{
  "id": 1013,
  "idFieldName": "OA_ManufacturingDesignID"
}
```

Server response Code (400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_ManufacturingDesignID = 1013",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "ManufacturingDesign",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

GetManufacturingDesignByID

POST/api/GetManufacturingDesignByID

Gets Manufacturing Design (products) and its BillsOf Materials by OA_ManufacturingDesignID value.

Schema

```
{
  "token": "string",
  "oA_ManufacturingDesignID": 0,
  "code": "string",
  "comment": "string",
  "name": "string",
  "oA_ItemsID": 0,
  "oA_StockUnitsID": 0,
  "quantity": 0,
  "unit": "string",
  "manufacturingDesignBOMs": [
    {
      "oA_ManufacturingDesignBOMID": 0,
      "oA_ManufacturingDesignID": 0,
      "oA_ItemsID": 0,
      "oA_StockUnitsID": 0,
      "code": "string",
      "unit": "string",
      "sequenceNumber": 0,
      "itemType": 0,
      "materialName": "string",
      "quantity": 0,
      "oA_CostDriverID": 0,
      "costDriver": "string"
    }
  ]
}
```

Field description


```
"token": " Set [token] string from [LogIn] response value
"oA_ManufacturingDesignID":
"code":
"comment":
"name":
"oA_ItemsID":
"oA_StockUnitsID":
"quantity":
"unit":
```

```
"manufacturingDesignBOMs":
"oA_ManufacturingDesignBOMID":
"oA_ManufacturingDesignID":
"oA_ItemsID":
"oA_StockUnitsID":
"code":
"unit":
"sequenceNumber":
"itemType":
"materialName":
"quantity":
"oA_CostDriverID":
"costDriver":
```

examples

1. Get ManufacturingDesign.ID - "2013".

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ManufacturingDesignID": 2013
}
```

Server response Code (200):

```
{
  "oA_ManufacturingDesignID": 2013,
  "code": "3011",
  "comment": "",
  "name": "cake2",
  "oA_ItemsID": 4003,
  "oA_StockUnitsID": 1,
  "quantity": 1,
  "unit": "Piece",
  "manufacturingDesignBOMs": [
    {
      "oA_ManufacturingDesignBOMID": 2118,
      "oA_ManufacturingDesignID": 2013,
      "oA_ItemsID": 1035,
      "oA_StockUnitsID": 1,
      "code": "1035",
      "unit": "Piece",
      "sequenceNumber": 1,
      "itemType": 0,
      "materialName": "Flour",
      "quantity": 1,
      "oA_CostDriverID": 0,
      "costDriver": ""
    }
  ]
}
```

Server response Code (400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_ManufacturingDesignID = 2000",
  }
]
```

```

"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "ManufacturingDesign",
"entryPoint": "GetProperties",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

ManufacturingDesignsList

POST/api/ManufacturingDesignsList

Gets Manufacturing Designs.Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1.

"sort": Use SQL ORDER BY string with field names.

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get ManufacturingDesignsList from database in specified filter.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2,
  "filter": "UPPER([ViewOA_ManufacturingDesign].[Code]) LIKE N'%1032'",
  "pageNumber": 1
}

```

Server response Code (200):

```

{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 1,
      "PageRowCount": 2,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_ManufacturingDesignID": 27,
      "OA_ItemsID": 1032,
      "OA_StockUnitsID": 1,
      "Code": "1032",
      "Product name": "CAKE",
      "Quantity": 1.00000000,
      "Unit": "Piece",
      "Comment": ""
    }
  ]
}

```

```
"Item type": 0,
"NUM_ROW": 1
}
]
}
```

2. Get a list of ManufacturingDesigns (Code,Product name).
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "sort": "[Code],[Product name]",
  "pageNumber": 1
}
```

CostDriver

POST/api/CostDriver

Adds new Cost Driver.Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "oA_CostDriverID": 0,
  "code": "string",
  "description": "string",
  "name": "string",
  "oA_ItemsID": 0,
  "after_update_fill_ID_field_values_in_BOM_datatable": true
}
```

Field description

"token": Set [token] string from [LogIn] response value
"oA_CostDriverID":
"code":
"description":
"name":
"oA_ItemsID":
"after_update_fill_ID_field_values_in_BOM_datatable":

Examples

1. Add new CostDriver, code-2006, name-CostDriver., description-cost driver N1007.
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_CostDriverID": 1007,
  "code": "2006",
  "description": "cost driver N1007",
  "name": "CostDriver",
  "oA_ItemsID": 3003,
  "after_update_fill_ID_field_values_in_BOM_datatable": true
}
```

Server response Code (200):

```
{
  "id": 1017,
  "idFieldName": "OA_CostDriverID"
}
```

Server response Code (400):

```
{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
}
```

```

"title": "One or more validation errors occurred.",
"status": 400,
"tracelId": "00-83e073ad8b927eeced691b3f52e17cb5-97e65f3a4b8f8872-00",
"errors": {
  "costDriver": [
    "The costDriver field is required."
  ],
  "$.oA_CostDriverID": [
    "Invalid leading zero before '7'. Path: $.oA_CostDriverID | LineNumber: 2 | BytePositionInLine: 24."
  ]
}
}

```

2. Add new CostDriver, code: 1032, name: CostDriver.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_CostDriverID": 1007,
  "code": "2006",
  "name": "CostDriver",
  "oA_ItemsID": 3003
}

```

PUT/api/CostDriver

Updates Cost Driver. Returns OK or Error description Json.

Schema

```

{
  "token": "string",
  "oA_CostDriverID": 0,
  "code": "string",
  "description": "string",
  "name": "string",
  "oA_ItemsID": 0,
  "after_update_fill_ID_field_values_in_BOM_datatable": true
}

```

Field description

```

"token": Set [token] string from [LogIn] response value
"oA_CostDriverID":
"code":
"description":
"name":
"oA_ItemsID":
"after_update_fill_ID_field_values_in_BOM_datatable":

```

Examples

1. Add new CostDriver, code-2008, name-CostDriver, description-cost driver N.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_CostDriverID": 1008,
  "code": "2008",
  "description": "cost driver N",
  "name": "CostDriver",
  "oA_ItemsID": 3003,
  "after_update_fill_ID_field_values_in_BOM_datatable": true
}

```

Server response Code (200):

```

{
  "id": 1008,

```

```
"idFieldName": "OA_CostDriverID"
}
```

Server response Code (400):

```
{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
  "title": "One or more validation errors occurred.",
  "status": 400,
  "traceId": "00-10a74e711fb808624c4ada782ddd303a-458b58831eee4212-00",
  "errors": {
    "costDriver": [
      "The costDriver field is required."
    ],
    "$.oA_CostDriverID": [
      "Invalid leading zero before '0'. Path: $.oA_CostDriverID | LineNumber: 2 | BytePositionInLine: 21."
    ]
  }
}
```

DELETE/api/CostDriver

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_CostDriverID": 0,
  "code": "string",
  "description": "string",
  "name": "string",
  "oA_ItemsID": 0,
  "after_update_fill_ID_field_values_in_BOM_datatable": true
}
```

Field description

```
"token": Set [token] string from [LogIn] response value
"oA_CostDriverID":
"code":
"description":
"name":
"oA_ItemsID":
"after_update_fill_ID_field_values_in_BOM_datatable":
```

Examples

1. Delete CostDriver. "oA_CostDriverID" -1006, "oA_ItemsID" -1033.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_CostDriverID":1006,
  "oA_ItemsID": 1033
}
```

Server response Code (200):

```
{
  "id": 1006,
  "idFieldName": "OA_CostDriverID"
}
```

Server response Code (400):

```
[
  {
    "errorNumber": 51051,
```

```

"description": "Record does not exist or read prohibited for user!",
"comment": "OA_CostDriverID = 1033",
"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "CostDriver",
"entryPoint": "GetProperties",
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]

```

CostDriversList

POST/api/CostDriversList

Gets Cost Drivers. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [Login] response value.

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0.

filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names.

"pageNumber": Use SQL ORDER BY string with field names.

Examples

1. Get CostDriversList from database in specified filter (CODE).

Request body:

```

{
  "token": "... paste here [Login] response value ...",
  "topReco1Q2rdsCount": 5,
  "filter": "UPPER([ViewOA_CostDriver].[Code]) LIKE N'%1032'",
  "pageNumber": 1
}

```

Server response Code (200):

```

{
  "Info": [
    {
      "RowCount": 4,
      "PageCount": 1,
      "PageRowCount": 5,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_Cos` tDriverID": 2,
      "OA_ItemsID": 1032,
      "Description": "",
      "Code": "1032",
      "Material name": "CAKE",
      "Item type": 0,

```

```

"NUM_ROW": 1
},
{
"OA_CostDriverID": 3,
"OA_ItemsID": 1032,
"Description": "",
"Code": "1032",
"Material name": "CAKE",
"Item type": 0,
"NUM_ROW": 2
}
{
"OA_CostDriverID": 4,
"OA_ItemsID": 1032,
"Description": "",
"Code": "1032",
"Material name": "CAKE",
"Item type": 0,
"NUM_ROW": 3
},
{
"OA_CostDriverID": 5,
"OA_ItemsID": 1032,
"Description": "",
"Code": "1032",
"Material name": "CAKE",
"Item type": 0,
"NUM_ROW": 4
}
]
}

```

Server response Code (400):

```

]
{
"type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
"title": "One or more validation errors occurred.",
"status": 400,
"traceId": "00-fb55b1ac020277efa5e84c5b9b0c957b-9a82b273a0e31ece-00",
"errors": {
"LP": [
"The LP field is required."],
"$topRecordsCount": [
", ' is an invalid start of a value. Path: $.topRecordsCount | LineNumber: 2 | BytePositionInLine: 21."
]
}
}
}
]

```

2. Get CostDriversList from database in sort (Code).

Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"topRecordsCount": 5,
"sort": "[Code]",
"pageNumber": 2
}

```

ManufacturingPlanning

POST/api/ManufacturingPlanning

Adds new Manufacturing Planning. Returns ID value or Error description Json.

Schema

```

{
  "token": "string",
  "oA_ManufacturingPlanningID": 0,
  "comment": "string",
  "oA_ItemsID": 0,
  "oA_ManufacturingPlanningOrderID": 0,
  "oA_StockUnitsID": 0,
  "quantity": 0,
  "unit": "string",
  "manufacturingPlanningBOMTable": [
    {
      "oA_ManufacturingPlanningBOMID": 0,
      "oA_ManufacturingPlanningID": 0,
      "oA_ItemsID": 0,
      "oA_StockUnitsID": 0,
      "oA_CostDriverID": 0,
      "quantity": 0,
      "unit": "string",
      "quantityForOneProduct": 0,
      "stockRegisterRate": 0,
      "quantityInRegisterUnit": 0,
      "oA_StockUnitsIDItem": 0,
      "registerUnit": "string",
      "sequenceNumber": 0,
      "code": "string",
      "materialName": "string",
      "itemType": 0,
      "costDriver": "string",
      "purchasePrice": 0,
      "purchaseAmount": 0,
      "purchaseEquivalent": 0,
      "purchaseEquivalent2": 0,
      "oA_CurrenciesID": 0,
      "purchaseCurrency": "string",
      "vatRate": 0,
      "vat": 0,
      "purchaseAmountWithoutVAT": 0,
      "purchaseEquivalentWithoutVAT": 0,
      "purchaseEquivalent2WithoutVAT": 0,
      "priceLastUpdateDate": "2023-05-15T13:02:31.908Z",
      "purchaseRate": 0,
      "purchaseRate2": 0,
      "purchaseRateIsFixed": true
    }
  ]
}

```

Field description

"token": Set [token] string from [LogIn] response value

"oA_ManufacturingPlanningID":

"comment":

"oA_ItemsID":

"oA_ManufacturingPlanningOrderID":

"oA_StockUnitsID":

"quantity":

"unit":

"manufacturingPlanningBOMTable":

"oA_ManufacturingPlanningBOMID":

"oA_ManufacturingPlanningID":

"oA_ItemsID": 0,

"oA_StockUnitsID":

"oA_CostDriverID":

"quantity":

"unit":

"quantityForOneProduct":


```

"stockRegisterRate":
"quantityInRegisterUnit":
"oA_StockUnitsIDItem":
"registerUnit":
"sequenceNumber":
"code":
"materialName":
"itemType":
"costDriver":
"purchasePrice":
"purchaseAmount":
"purchaseEquivalent":
"purchaseEquivalent2":
"oA_CurrenciesID":
"purchaseCurrency":
"vatRate":
"vat":
"purchaseAmountWithoutVAT":
"purchaseEquivalentWithoutVAT":
"purchaseEquivalent2WithoutVAT":
"priceLastUpdateDate":
"purchaseRate":
"purchaseRate2":
"purchaseRateIsFixed":

```

Examples

1. ADD ManufacturingPlanning.using "oA_ManufacturingPlanningID" -2, "oA_ItemsID" - 4007.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_ManufacturingPlanningID": 2,
  "oA_ItemsID": 4007,
  "oA_StockUnitsID": 1,
  "quantity": 1,
  "unit": "piece",
  "manufacturingPlanningBOMTable": [
    {
      "oA_ManufacturingPlanningBOMID": 2,
      "oA_ManufacturingPlanningID": 2,
      "oA_ItemsID": 4007,
      "oA_StockUnitsID": 1,
      "quantity": 2,
      "unit": "PIECE"
    }
  ]
}

```

Server response Code (200):

```

{
  "id": 3,
  "idFieldName": "OA_ManufacturingPlanningID"
}

```

Server response Code (400):

```

[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "OA_ManufacturingPlanningID=8\nExist relation to another table.\n\nRelated table [OA_ManufacturingPlanningBOM].\n\n[547] The INSERT statement conflicted with the FOREIGN KEY constraint \"FK_OA_ManufacturingPlanningBOM_OA_Items\". The conflict occurred in database \"NDatabaseEN\", table \"dbo.OA_Items\", column 'OA_ItemsID'.[3621] The statement has been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "ManufacturingPlanning",
  }
]

```

```

    "entryPoint": "Update",
    "resourceDLL_ForTranslation": "ORIS_Administrate",
    "errorPointer": 0
  }
]

```

2. Add "oA_ManufacturingPlanningID" - 6, "oA_ItemsID" - 4007, quantityInRegisterUnit - 2, "oA_StockUnitsIDItem" -1, register Unit - piece, sequenceNumber -1, code -1044, materialName - khachapuri, purchasePrice - 120, purchaseAmount -240.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_ManufacturingPlanningID": 6,
  "oA_ItemsID": 4007,
  "quantity": 1,
  "unit": "piece",
  "manufacturingPlanningBOMTable": [
    {
      "oA_ManufacturingPlanningBOMID": 7,
      "oA_ManufacturingPlanningID": 6,
      "oA_ItemsID": 4007,
      "oA_StockUnitsID": 1,
      "quantity": 2,
      "unit": "piece",
      "quantityInRegisterUnit": 2,
      "oA_StockUnitsIDItem": 1,
      "registerUnit": "piece",
      "sequenceNumber": 1,
      "code": "1044",
      "materialName": "khachapuri",
      "purchasePrice": 120,
      "purchaseAmount": 240,
      "oA_CurrenciesID": 1,
      "purchaseCurrency": "GEL",
      "vatRate": 18,
      "vat": 36,
      "purchaseAmountWithoutVAT": 203,
      "priceLastUpdateDate": "2023-05-31T08:04:34.576Z"
    }
  ]
}

```

PUT/api/ManufacturingPlanning

Updates Manufacturing Planning. Returns OK or Error description Json.

Schema

```

{
  "token": "string",
  "oA_ManufacturingPlanningID": 0,
  "comment": "string",
  "oA_ItemsID": 0,
  "oA_ManufacturingPlanningOrderID": 0,
  "oA_StockUnitsID": 0,
  "quantity": 0,
  "unit": "string",
  "manufacturingPlanningBOMTable": [
    {
      "oA_ManufacturingPlanningBOMID": 0,
      "oA_ManufacturingPlanningID": 0,
      "oA_ItemsID": 0,
      "oA_StockUnitsID": 0,
      "oA_CostDriverID": 0,
      "quantity": 0,
      "unit": "string",
    }
  ]
}

```

```

"quantityForOneProduct": 0,
"stockRegisterRate": 0,
"quantityInRegisterUnit": 0,
"oA_StockUnitsIDItem": 0,
"registerUnit": "string",
"sequenceNumber": 0,
"code": "string",
"materialName": "string",
"itemType": 0,
"costDriver": "string",
"purchasePrice": 0,
"purchaseAmount": 0,
"purchaseEquivalent": 0,
"purchaseEquivalent2": 0,
"oA_CurrenciesID": 0,
"purchaseCurrency": "string",
"vatRate": 0,
"vat": 0,
"purchaseAmountWithoutVAT": 0,
"purchaseEquivalentWithoutVAT": 0,
"purchaseEquivalent2WithoutVAT": 0,
"priceLastUpdateDate": "2023-05-24T07:14:43.940Z",
"purchaseRate": 0,
"purchaseRate2": 0,
"purchaseRateIsFixed": true
}
]
}

```

Field description

"token": Set [token] string from [LogIn] response value

"oA_ManufacturingPlanningID":

"comment":

"oA_ItemsID":

"oA_ManufacturingPlanningOrderID":

"oA_StockUnitsID":

"quantity":

"unit":

"manufacturingPlanningBOMTable":

"oA_ManufacturingPlanningBOMID":

"oA_ManufacturingPlanningID":

"oA_ItemsID": 0,

"oA_StockUnitsID":

"oA_CostDriverID":

"quantity":

"unit":

"quantityForOneProduct":

"stockRegisterRate":

"quantityInRegisterUnit":

"oA_StockUnitsIDItem":

"registerUnit":

"sequenceNumber":

"code":

"materialName":

"itemType":

"costDriver":

"purchasePrice":

"purchaseAmount":

"purchaseEquivalent":

"purchaseEquivalent2":

"oA_CurrenciesID":

"purchaseCurrency":

"vatRate":

"vat":

"purchaseAmountWithoutVAT":

"purchaseEquivalentWithoutVAT":
"purchaseEquivalent2WithoutVAT":
"priceLastUpdateDate":
"purchaseRate":
"purchaseRate2":
"purchaseRateIsFixed":

Examples

1. ADD item ID-3010, ManufacturingPlanningID -6, ItemsID-1035, code-1044, materialName-khachapuri, purchasePrice-120, purchaseAmount-240, vatRate- 18, vat-36, purchaseAmountWithoutVAT-203.

Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "oA_ManufacturingPlanningID": 6,  
  "oA_ItemsID": 3010,  
  "quantity": 1,  
  "unit": "piece",  
  "manufacturingPlanningBOMTable": [  
    "oA_ManufacturingPlanningBOMID": 7,  
    "oA_ManufacturingPlanningID": 6,  
    "oA_ItemsID": 1035,  
    "oA_StockUnitsID": 1,  
    "quantity": 2,  
    "unit": "piece",  
    "quantityInRegisterUnit": 2,  
    "oA_StockUnitsIDItem": 1,  
    "registerUnit": "piece",  
    "sequenceNumber": 1,  
    "code": "1044",  
    "materialName": "khachapuri",  
    "purchasePrice": 120,  
    "purchaseAmount": 240,  
    "oA_CurrenciesID": 1,  
    "purchaseCurrency": "GEL",  
    "vatRate": 18,  
    "vat": 36,  
    "purchaseAmountWithoutVAT": 203,  
    "priceLastUpdateDate": "2023-05-31T08:04:34.576Z"  
  ]  
}
```

Server response Code (200):

```
{  
  "id": 6,  
  "idFieldName": "OA_ManufacturingPlanningID"  
}
```

Server response Code (400):

```
[  
  {  
    "errorNumber": 51051,  
    "description": "Record does not exist or read prohibited for user!",  
    "comment": "OA_ManufacturingPlanningID = 66",  
    "command": "",  
    "errorType": "ErrorType_Error",  
    "source": "OA_BusinessObject.dll",  
    "class": "ManufacturingPlanning",  
    "entryPoint": "GetProperties",  
    "resourceDLL_ForTranslation": "ORIS_Accounting",  
    "errorPointer": 0  
  }  
]
```

DELETE/api/ManufacturingPlanning

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_ManufacturingPlanningID": 0,
  "comment": "string",
  "oA_ItemsID": 0,
  "oA_ManufacturingPlanningOrderID": 0,
  "oA_StockUnitsID": 0,
  "quantity": 0,
  "unit": "string",
  "manufacturingPlanningBOMTable": [
    {
      "oA_ManufacturingPlanningBOMID": 0,
      "oA_ManufacturingPlanningID": 0,
      "oA_ItemsID": 0,
      "oA_StockUnitsID": 0,
      "oA_CostDriverID": 0,
      "quantity": 0,
      "unit": "string",
      "quantityForOneProduct": 0,
      "stockRegisterRate": 0,
      "quantityInRegisterUnit": 0,
      "oA_StockUnitsIDItem": 0,
      "registerUnit": "string",
      "sequenceNumber": 0,
      "code": "string",
      "materialName": "string",
      "itemType": 0,
      "costDriver": "string",
      "purchasePrice": 0,
      "purchaseAmount": 0,
      "purchaseEquivalent": 0,
      "purchaseEquivalent2": 0,
      "oA_CurrenciesID": 0,
      "purchaseCurrency": "string",
      "vatRate": 0,
      "vat": 0,
      "purchaseAmountWithoutVAT": 0,
      "purchaseEquivalentWithoutVAT": 0,
      "purchaseEquivalent2WithoutVAT": 0,
      "priceLastUpdateDate": "2023-05-24T07:14:43.940Z",
      "purchaseRate": 0,
      "purchaseRate2": 0,
      "purchaseRateIsFixed": true
    }
  ]
}
```

Field description

"token": Set [token] string from [LogIn] response value

"oA_ManufacturingPlanningID":

"comment":

"oA_ItemsID":

"oA_ManufacturingPlanningOrderID":

"oA_StockUnitsID":

"quantity":

"unit":

"manufacturingPlanningBOMTable": [

"oA_ManufacturingPlanningBOMID":

"oA_ManufacturingPlanningID":

"oA_ItemsID": 0,

```

"oA_StockUnitsID":
"oA_CostDriverID":
"quantity":
"unit":
"quantityForOneProduct":
"stockRegisterRate":
"quantityInRegisterUnit":
"oA_StockUnitsIDItem":
"registerUnit":
"sequenceNumber":
"code":
"materialName":
"itemType":
"costDriver":
"purchasePrice":
"purchaseAmount":
"purchaseEquivalent":
"purchaseEquivalent2":
"oA_CurrenciesID":
"purchaseCurrency":
"vatRate":
"vat":
"purchaseAmountWithoutVAT":
"purchaseEquivalentWithoutVAT":
"purchaseEquivalent2WithoutVAT":
"priceLastUpdateDate":
"purchaseRate":
"purchaseRate2":
"purchaseRatelsFixed":

```

Examples

1. Delete ManufacturingPlanning ID - 6
Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_ManufacturingPlanningID": 6,
  "oA_ItemsID": 3010
}

```

Server response Code (200):

```

{
  "id": 6,
  "idFieldName": "OA_ManufacturingPlanningID"
}

```

Server response Code (400):

```

[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_ManufacturingPlanningID = 6",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "ManufacturingPlanning",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

2. Delete ManufacturingPlanning ID -12.
Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_ManufacturingPlanningID": 12
  "oA_ItemsID": 4007
}

```

```
}
```

GetManufacturingPlanningByID

POST/api/GetManufacturingPlanningByID

Gets Manufacturing Planning (products) and its Bills Of Materials by OA_ManufacturingPlanningID value.

Schema

```
{
  "token": "string",
  "oA_ManufacturingPlanningID": 0,
  "code": "string",
  "comment": "string",
  "name": "string",
  "oA_DebitorCreditorID": 0,
  "oA_ItemsID": 0,
  "oA_ManufacturingPlanningOrderID": 0,
  "oA_StockUnitsID": 0,
  "oA_StockUnitsIDItem": 0,
  "orderDate": "2023-05-15T08:09:37.489Z",
  "orderer": "string",
  "ordererCode": "string",
  "ordererVATType": 0,
  "orderNumber": 0,
  "ownerVATType": 0,
  "quantity": 0,
  "quantityInRegisterUnit": 0,
  "registerUnit": "string",
  "stockRegisterRate": 0,
  "unit": "string",
  "manufacturingPlanningBOMTable": [
    {
      "oA_ManufacturingPlanningBOMID": 0,
      "oA_ManufacturingPlanningID": 0,
      "oA_ItemsID": 0,
      "oA_StockUnitsID": 0,
      "oA_CostDriverID": 0,
      "quantity": 0,
      "unit": "string",
      "quantityForOneProduct": 0,
      "stockRegisterRate": 0,
      "quantityInRegisterUnit": 0,
      "oA_StockUnitsIDItem": 0,
      "registerUnit": "string",
      "sequenceNumber": 0,
      "code": "string",
      "materialName": "string",
      "itemType": 0,
      "costDriver": "string",
      "purchasePrice": 0,
      "purchaseAmount": 0,
      "purchaseEquivalent": 0,
      "purchaseEquivalent2": 0,
      "oA_CurrenciesID": 0,
      "purchaseCurrency": "string",
      "vatRate": 0,
      "vat": 0,
      "purchaseAmountWithoutVAT": 0,
      "purchaseEquivalentWithoutVAT": 0,
      "purchaseEquivalent2WithoutVAT": 0,
      "priceLastUpdateDate": "2023-05-15T08:09:37.489Z",
      "purchaseRate": 0,
      "purchaseRate2": 0,
    }
  ]
}
```

```
    "purchaseRatelsFixed": true
  }
}
```

Field description

"token": Set [token] string from [LogIn] response value

"oA_ManufacturingPlanningID":

"code":

"comment":

"name":

"oA_DebitorCreditorID":

"oA_ItemsID":

"oA_ManufacturingPlanningOrderID":

"oA_StockUnitsID":

"oA_StockUnitsIDItem":

"orderDate":

"orderer":

"ordererCode":

"ordererVATType":

"orderNumber":

"ownerVATType":

"quantity":

"quantityInRegisterUnit":

"registerUnit":

"stockRegisterRate":

"unit":

"manufacturingPlanningBOMTable":

"oA_ManufacturingPlanningBOMID":

"oA_ManufacturingPlanningID":

"oA_ItemsID":

"oA_StockUnitsID":

"oA_CostDriverID":

"quantity":

"unit":

"quantityForOneProduct":

"stockRegisterRate":

"quantityInRegisterUnit":

"oA_StockUnitsIDItem":

"registerUnit":

"sequenceNumber":

"code":

"materialName":

"itemType":

"costDriver":

"purchasePrice":

"purchaseAmount":

"purchaseEquivalent":

"purchaseEquivalent2":

"oA_CurrenciesID":

"purchaseCurrency":

"vatRate":

"vat":

"purchaseAmountWithoutVAT":

"purchaseEquivalentWithoutVAT":

"purchaseEquivalent2WithoutVAT":

"priceLastUpdateDate":

"purchaseRate":

"purchaseRate2":

"purchaseRatelsFixed":

Examples

1. Get ManufacturingPlanning, ID -4, ItemsID - 4007.
Request body:


```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ManufacturingPlanningID": 4,
  "oA_ItemsID": 4007
}
```

Server response Code (200):

```
{
  "oA_ManufacturingPlanningID": 4,
  "code": "40080",
  "comment": "coment",
  "name": "khachapuri",
  "oA_DebitorCreditorID": 1021,
  "oA_ItemsID": 4007,
  "oA_ManufacturingPlanningOrderID": 4,
  "oA_StockUnitsID": 0,
  "oA_StockUnitsIDItem": 0,
  "orderDate": "2023-05-31T15:37:41",
  "orderer": "Organization",
  "ordererCode": "1234567",
  "ordererVATType": 1,
  "orderNumber": 4,
  "ownerVATType": 0,
  "quantity": 1,
  "quantityInRegisterUnit": 1,
  "registerUnit": "",
  "stockRegisterRate": 1,
  "unit": "",
  "manufacturingPlanningBOMTable": [
    {
      "oA_ManufacturingPlanningBOMID": 5,
      "oA_ManufacturingPlanningID": 4,
      "oA_ItemsID": 4007,
      "oA_StockUnitsID": 1,
      "oA_CostDriverID": 0,
      "quantity": 2,
      "unit": "Piece",
      "quantityForOneProduct": 2,
      "stockRegisterRate": 1,
      "quantityInRegisterUnit": 2,
      "oA_StockUnitsIDItem": 1,
      "registerUnit": "Piece",
      "sequenceNumber": 1,
      "code": "40080",
      "materialName": "khachapuri",
      "itemType": 0,
      "costDriver": "",
      "purchasePrice": 0,
      "purchaseAmount": 0,
      "purchaseEquivalent": 0,
      "purchaseEquivalent2": 0,
      "oA_CurrenciesID": 1,
      "purchaseCurrency": "GEL",
      "vatRate": 18,
      "vat": 0,
      "purchaseAmountWithoutVAT": 0,
      "purchaseEquivalentWithoutVAT": 0,
      "purchaseEquivalent2WithoutVAT": 0,
      "priceLastUpdateDate": "2023-05-31T11:40:11",
      "purchaseRate": 1,
      "purchaseRate2": 0.38581735,
      "purchaseRateIsFixed": false
    }
  ]
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_ManufacturingPlanningID = 45",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "ManufacturingPlanning",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

ManufacturingPlanningsList

POST/api/ManufacturingPlanningsList

Gets. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property: [TopRecordsCount]=1.

"sort": Use SQL ORDER BY string with field names.

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page.

Examples

1. Get first 6 ManufacturingPlanning from database.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 6,
  "pageNumber": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 6,
      "PageCount": 1,
      "PageRowCount": 6,
      "PageNumber": 1
    }
  ],
  "Result": [
    {
      "OA_ManufacturingPlanningID": 4,
      "OA_ManufacturingPlanningOrderID": 4,
      "Planning order date": "2023-05-31T15:37:41",
    }
  ]
}
```

```
"Planning order document number": 4,
"OA_ItemsID": 4007,
"Code": "40080",
"Product name": "khachapuri",
"Quantity": 1.00000000,
"Stock register rate": 1.00000000,
"Quantity in register unit": 1.00000000,
"OA_StockUnitsIDItem": 1,
"Register unit": "Piece",
"Comment": "coment",
"Item type": 0,
"OA_DebitorCreditorID": 1021,
"Orderer code": "1234567",
"Orderer": "Organization",
"Orderer VAT type": 1,
"NUM_ROW": 1
},
{
  "OA_ManufacturingPlanningID": 5,
  "Quantity": 1.00000000,
  "Stock register rate": 1.00000000,
  "Quantity in register unit": 1.00000000,
  "Comment": "",
  "NUM_ROW": 2
},
{
  "OA_ManufacturingPlanningID": 7,
  "Quantity": 1.00000000,
  "Stock register rate": 1.00000000,
  "Quantity in register unit": 1.00000000,
  "Comment": "",
  "NUM_ROW": 3
},
{
  "OA_ManufacturingPlanningID": 9,
  "Quantity": 0.00000000,
  "Stock register rate": 1.00000000,
  "Quantity in register unit": 0.00000000,
  "Comment": "string",
  "NUM_ROW": 4
},
{
  "OA_ManufacturingPlanningID": 10,
  "Quantity": 0.00000000,
  "Stock register rate": 1.00000000,
  "Quantity in register unit": 0.00000000,
  "Comment": "string",
  "NUM_ROW": 5
},
{
  "OA_ManufacturingPlanningID": 11,
  "OA_ItemsID": 4007,
  "OA_StockUnitsID": 1,
  "Code": "40080",
  "Product name": "khachapuri",
  "Quantity": 1.00000000,
  "Unit": "Piece",
  "Stock register rate": 1.00000000,
  "Quantity in register unit": 1.00000000,
  "OA_StockUnitsIDItem": 1,
  "Register unit": "Piece",
  "Comment": "",
  "Item type": 0,
  "NUM_ROW": 6
}
]
}
```

2. Get a list of ManufacturingPlanning like (Code).

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "filter": "UPPER([ViewOA_ManufacturingPlanning].[Code]) LIKE N'%40080'"
}
```

3. Get a list of ManufacturingPlanning, sort it by Product name.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 6,
  "sort": "[Product name]"
}
```

ManufacturingExecution

POST/api/ManufacturingExecution

Adds new Manufacturing Execution. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "comment": "string",
  "costCentreName": "string",
  "date": "2023-05-15T10:32:57.013Z",
  "oA_CostCentreID": 0,
  "oA_ManufacturingDesignID": 0,
  "oA_ManufacturingPlanningID": 0,
  "oA_ItemsID": 0,
  "oA_ManufacturingExecutionOrderID": 0,
  "oA_ManufacturingPlanningOrderID": 0,
  "oA_StockUnitsID": 0,
  "oA_TransactionTypesID": 0,
  "producedQuantity": 0,
  "unit": "string",
  "manufacturingExecutionBOMTable": [
    {
      "oA_ManagerialEntriesID": 0,
      "oA_TransactionsID": 0,
      "oA_ItemsID": 0,
      "oA_StockUnitsID": 0,
      "oA_CostCentreID": 0,
      "code": "string",
      "unit": "string",
      "costCentreName": "string",
      "sequenceNumber": 0,
      "itemType": 0,
      "materialName": "string",
      "usedQuantity": 0,
      "quantityForOneProduct": 0,
      "oA_CostDriverID": 0,
      "costDriver": "string",
      "materialOA_AccountsID": 0,
      "materialParentAccount": "string",
      "oA_EntriesID": 0
    }
  ]
}
```

Field description

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

"comment":

"costCentreName":

"date":

"oA_CostCentreID":

"oA_ManufacturingDesignID":

"oA_ManufacturingPlanningID":

"oA_ItemsID":

"oA_ManufacturingExecutionOrderID":

"oA_ManufacturingPlanningOrderID":

"oA_StockUnitsID":

"oA_TransactionTypesID":

"producedQuantity":

"unit":

"manufacturingExecutionBOMTable":

"oA_ManagerialEntriesID":

"oA_TransactionsID":

"oA_ItemsID":

"oA_StockUnitsID":

"oA_CostCentreID":

"code":

"unit":

"costCentreName":

"sequenceNumber":

"itemType":

"materialName":

"usedQuantity":

"quantityForOneProduct":

"oA_CostDriverID":

"costDriver":

"materialOA_AccountsID":

"materialParentAccount":

"oA_EntriesID":

Examples

1. Add new ManufacturingExecution, "oA_TransactionsID" -2004, "oA_ItemsID" -1032,1033.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ItemsID": 1032,
  "oA_TransactionTypesID": 22,
  "manufacturingExecutionBOMTable": [
    {
      "oA_ManagerialEntriesID": 2003,
      "oA_TransactionsID": 2004,
      "oA_ItemsID": 1033
    }
  ]
}
```

Server response Code (200):

```
{
  "id": 5014,
  "idFieldName": "OA_TransactionsID"
}
```

Server response Code (400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!"
  }
]
```

```

"comment": "OA_TransactionsID=5023\nExist relation to another table\n\nRelated table
[OA_ManagerialEntries].\n\n\n[547] The INSERT statement conflicted with the FOREIGN KEY constraint
\FK_OA_ManagerialEntries_OA_Items\. The conflict occurred in database \NDatabaseEN\", table
\dbo.OA_Items\", column 'OA_ItemsID'.[3621] The statement has been terminated.",
"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "ManufacturingExecution",
"entryPoint": "Update",
"resourceDLL_ForTranslation": "ORIS_Administrate",
"errorPointer": 0
}
]

```

PUT/api/ManufacturingExecution

Updates Manufacturing Execution. Returns OK or Error description Json.

Schema

```

{
  "token": "string",
  "oA_TransactionsID": 0,
  "comment": "string",
  "costCentreName": "string",
  "date": "2023-06-09T11:49:06.575Z",
  "oA_CostCentreID": 0,
  "oA_ManufacturingDesignID": 0,
  "oA_ManufacturingPlanningID": 0,
  "oA_ItemsID": 0,
  "oA_ManufacturingExecutionOrderID": 0,
  "oA_ManufacturingPlanningOrderID": 0,
  "oA_StockUnitsID": 0,
  "oA_TransactionTypesID": 0,
  "producedQuantity": 0,
  "unit": "string",
  "manufacturingExecutionBOMTable": [
    {
      "oA_ManagerialEntriesID": 0,
      "oA_TransactionsID": 0,
      "oA_ItemsID": 0,
      "oA_StockUnitsID": 0,
      "oA_CostCentreID": 0,
      "code": "string",
      "unit": "string",
      "costCentreName": "string",
      "sequenceNumber": 0,
      "itemType": 0,
      "materialName": "string",
      "usedQuantity": 0,
      "quantityForOneProduct": 0,
      "oA_CostDriverID": 0,
      "costDriver": "string",
      "materialOA_AccountsID": 0,
      "materialParentAccount": "string",
      "oA_EntriesID": 0
    }
  ]
}

```

Field description

"**token**": Set [token] string from [LogIn] response value

"**oA_TransactionsID**":

"**comment**":

"**costCentreName**":

```

"date":
"oA_CostCentreID":
"oA_ManufacturingDesignID":
"oA_ManufacturingPlanningID":
"oA_ItemsID":
"oA_ManufacturingExecutionOrderID":
"oA_ManufacturingPlanningOrderID":
"oA_StockUnitsID":
"oA_TransactionTypesID":
"producedQuantity":
"unit":

"manufacturingExecutionBOMTable":
"oA_ManagerialEntriesID":
"oA_TransactionsID":
"oA_ItemsID":
"oA_StockUnitsID":
"oA_CostCentreID":
"code":
"unit":
"costCentreName":
"sequenceNumber":
"itemType":
"materialName":
"usedQuantity":
"quantityForOneProduct":
"oA_CostDriverID":
"costDriver":
"materialOA_AccountsID":
"materialParentAccount":
"oA_EntriesID":

```

Examples

1. Update manufacturingExecution , "oA_TransactionsID" - 5037, "oA_ItemsID" - 3010.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 5037,
  "oA_ItemsID": 3010,
  "manufacturingExecutionBOMTable": [
    {
      "oA_ManagerialEntriesID": 2003,
      "oA_TransactionsID": 2004,
      "oA_ItemsID": 1033
    }
  ]
}

```

Server response Code (200):

```

{
  "id": 5037,
  "idFieldName": "OA_TransactionsID"
}

```

Server response Code (400):

```

[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "OA_TransactionsID=5037\nExist relation to another table,\r\nRelated table [OA_ManagerialEntries].\r\n\r\n[547] The INSERT statement conflicted with the FOREIGN KEY constraint \r\n\"FK_OA_ManagerialEntries_OA_Items\". The conflict occurred in database \"NDatabaseEN\", table \"dbo.OA_Items\", column 'OA_ItemsID'.[3621] The statement has been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
  }
]

```

```
"class": "ManufacturingExecution",
"entryPoint": "Update",
"resourceDLL_ForTranslation": "ORIS_Administrate",
"errorPointer": 0
}
]
```

2. Update manufacturingExecution , "oA_TransactionsID" - 5037, "oA_ItemsID" - 3010, "oA_TransactionsID" - 2004, "oA_ItemsID" -1034.

Request body:

```
{
"token": "... paste here [LogIn] response value ...",
"oA_TransactionsID": 5037,
"oA_ItemsID": 3010,
"manufacturingExecutionBOMTable": [
{
"oA_ManagerialEntriesID": 2003,
"oA_TransactionsID": 2004,
"oA_ItemsID": 1034
}
]
}
```

DELETE/api/ManufacturingExecution

Returns OK or Error description Json.

Schema

```
{
"token": "string",
"oA_TransactionsID": 0,
"comment": "string",
"costCentreName": "string",
"date": "2023-06-09T11:54:34.058Z",
"oA_CostCentreID": 0,
"oA_ManufacturingDesignID": 0,
"oA_ManufacturingPlanningID": 0,
"oA_ItemsID": 0,
"oA_ManufacturingExecutionOrderID": 0,
"oA_ManufacturingPlanningOrderID": 0,
"oA_StockUnitsID": 0,
"oA_TransactionTypesID": 0,
"producedQuantity": 0,
"unit": "string",
"manufacturingExecutionBOMTable": [
{
"oA_ManagerialEntriesID": 0,
"oA_TransactionsID": 0,
"oA_ItemsID": 0,
"oA_StockUnitsID": 0,
"oA_CostCentreID": 0,
"code": "string",
"unit": "string",
"costCentreName": "string",
"sequenceNumber": 0,
"itemType": 0,
"materialName": "string",
"usedQuantity": 0,
"quantityForOneProduct": 0,
"oA_CostDriverID": 0,
"costDriver": "string",
"materialOA_AccountsID": 0,
"materialParentAccount": "string",
"oA_EntriesID": 0
}
]
}
```



```
}
```

Field description

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

"comment":

"costCentreName":

"date":

"oA_CostCentreID":

"oA_ManufacturingDesignID":

"oA_ManufacturingPlanningID":

"oA_ItemsID":

"oA_ManufacturingExecutionOrderID":

"oA_ManufacturingPlanningOrderID":

"oA_StockUnitsID":

"oA_TransactionTypesID":

"producedQuantity":

"unit":

"manufacturingExecutionBOMTable":

"oA_ManualEntriesID":

"oA_TransactionsID":

"oA_ItemsID":

"oA_StockUnitsID":

"oA_CostCentreID":

"code":

"unit":

"costCentreName":

"sequenceNumber":

"itemType":

"materialName":

"usedQuantity":

"quantityForOneProduct":

"oA_CostDriverID":

"costDriver":

"materialOA_AccountsID":

"materialParentAccount":

"oA_EntriesID":

Examples

1. Delete "oA_TransactionsID" -5037.

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 5037
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 50371",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "ManufacturingExecution",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

GetManufacturingExecutionByID

POST/api/GetManufacturingExecutionByID

Gets Manufacturing Execution (products) and its Bills Of Materials by OA_TransactionsID value.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "code": "string",
  "comment": "string",
  "costCentreName": "string",
  "date": "2023-06-09T12:07:02.936Z",
  "executionOrderDate": "2023-06-09T12:07:02.936Z",
  "executionOrderDocumentNumber": 0,
  "hasEntries": true,
  "name": "string",
  "oA_CostCentreID": 0,
  "oA_ItemsID": 0,
  "oA_ManufacturingExecutionOrderID": 0,
  "oA_ManufacturingPlanningOrderID": 0,
  "oA_StockUnitsID": 0,
  "oA_TransactionTypesID": 0,
  "producedQuantity": 0,
  "unit": "string",
  "workInProgress_Account": "string",
  "workInProgressOA_AccountsID": 0,
  "manufacturingExecutionBOMTable": [
    {
      "oA_ManualEntriesID": 0,
      "oA_TransactionsID": 0,
      "oA_ItemsID": 0,
      "oA_StockUnitsID": 0,
      "oA_CostCentreID": 0,
      "code": "string",
      "unit": "string",
      "costCentreName": "string",
      "sequenceNumber": 0,
      "itemType": 0,
      "materialName": "string",
      "usedQuantity": 0,
      "quantityForOneProduct": 0,
      "oA_CostDriverID": 0,
      "costDriver": "string",
      "materialOA_AccountsID": 0,
      "materialParentAccount": "string",
      "oA_EntriesID": 0
    }
  ]
}
```

Field description

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

"code":

"comment":

"costCentreName":

"date":

"executionOrderDate":

"executionOrderDocumentNumber":

"hasEntries":

"name":

"oA_CostCentreID":

"oA_ItemsID":

"oA_ManufacturingExecutionOrderID":
"oA_ManufacturingPlanningOrderID":
"oA_StockUnitsID":
"oA_TransactionTypesID":
"producedQuantity":
"unit":
"workInProgress_Account":
"workInProgressOA_AccountsID":

"manufacturingExecutionBOMTable":
"oA_ManualEntriesID":
"oA_TransactionsID":
"oA_ItemsID":
"oA_StockUnitsID":
"oA_CostCentreID":
"code":
"unit":
"costCentreName":
"sequenceNumber":
"itemType":
"materialName":
"usedQuantity":
"quantityForOneProduct":
"oA_CostDriverID":
"costDriver":
"materialOA_AccountsID":
"materialParentAccount":
"oA_EntriesID":

Examples

1. Get ManufacturingExecutionByID – "oA_TransactionsID" - 5006.

Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "oA_TransactionsID": 5006  
}
```

Server response Code(200):

```
{  
  "oA_TransactionsID": 5006,  
  "code": "1033",  
  "comment": "",  
  "costCentreName": "General cost center",  
  "date": "2023-05-29T14:31:38",  
  "executionOrderDate": "2023-05-30T08:29:55.707",  
  "executionOrderDocumentNumber": 11,  
  "hasEntries": false,  
  "name": "EGG",  
  "oA_CostCentreID": 1,  
  "oA_ItemsID": 1033,  
  "oA_ManufacturingExecutionOrderID": 1,  
  "oA_ManufacturingPlanningOrderID": 0,  
  "oA_StockUnitsID": 1,  
  "oA_TransactionTypesID": 0,  
  "producedQuantity": 0,  
  "unit": "Piece",  
  "workInProgress_Account": "1200",  
  "workInProgressOA_AccountsID": 11,  
  "manufacturingExecutionBOMTable": [  
    {  
      "oA_ManualEntriesID": 5005,  
      "oA_TransactionsID": 5006,  
      "oA_ItemsID": 1033,  
      "oA_StockUnitsID": 1,  
      "oA_CostCentreID": 1,  
      "code": "1033",
```

```

    "unit": "Piece",
    "costCentreName": "General cost center",
    "sequenceNumber": 1,
    "itemType": 0,
    "materialName": "EGG",
    "usedQuantity": 0,
    "quantityForOneProduct": 0,
    "oA_CostDriverID": 0,
    "costDriver": "",
    "materialOA_AccountsID": 70,
    "materialParentAccount": "1620",
    "oA_EntriesID": 0
  }
]
}

```

Server response Code (400):

```

[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_TransactionsID = 50066",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "ManufacturingExecution",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

ManufacturingExecutionsList

POST/api/ManufacturingExecutionsList

Gets Manufacturing Executions. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": " Use SQL WHERE string by field names. Get field names by launch this method with property: [TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names.

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get first ManufacturingExecutio from database.

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1
}

```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 0,
      "PageRowCount": 1,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_TransactionsID": 2004,
      "OA_ItemsID": 1032,
      "Code": "1032",
      "Product name": "CAKE",
      "OA_StockUnitsID": 1,
      "Unit": "Piece",
      "OA_CostCentreID": 1,
      "Cost centre name": "General cost center",
      "Comment": "",
      "Has entries": true,
      "Produced quantity": 1.00000000,
      "OA_ManufacturingExecutionOrderID": 1,
      "Execution order date": "2023-05-30T08:29:55.707",
      "Execution order document number": 11,
      "OA_TransactionTypesID": 22,
      "Delay": false,
      "Documental confirm": false,
      "OA_UsersID": 1,
      "User": "sa",
      "OA_DepartmentsID": 1,
      "Department": "Administrators",
      "WorkInProcessOA_AccountsID": 11,
      "Work in process account": "1200",
      "Execution order comment": "",
      "ProductOA_AccountsID": 73,
      "Product parent account": "1640",
      "Redistribute the total cost in proportion to the number of products produced": false,
      "Date": "2023-05-17T15:07:18",
      "Record related binary documents": null,
      "Item type": 0,
      "OA_ManufacturingPlanningOrderID": null,
      "Planning order date": null,
      "Planning order document number": null,
      "OA_DebitorCreditorID": null,
      "Orderer code": null,
      "Orderer": null,
      "OA_EntriesIDProduct": 4015,
      "OA_StatusID": null,
      "Status": null
    }
  ]
}
```

1. Get from database in specified filter, ManufacturingExecution-(Code).
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2,
  "filter": "UPPER([ViewOA_ManufacturingExecution].[Code]) LIKE N%1032%"
}
```

2. Get ManufacturingExecution from database in specified sort (Date)
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
```

```
"topRecordsCount": 1,  
"sort": "[Date]",  
"pageNumber": 1  
}
```

Related tables

DebitorCreditor

POST/api/DebitorCreditor

Adds debtor/creditor (organization or person). Returns ID value or Error description Json.

Schema

```
{  
  "token": "string",  
  "address": "string",  
  "birthDay": "2023-05-15T12:14:53.061Z",  
  "cityName": "string",  
  "clockNumber": "string",  
  "code": "string",  
  "comment": "string",  
  "countryName": "string",  
  "debitorCreditorGroup": "string",  
  "firstPerson": "string",  
  "isForeigner": true,  
  "isPerson": true,  
  "name": "string",  
  "oA_CityID": 0,  
  "oA_CountryID": 0,  
  "oA_DebitorCreditorGroupID": 0,  
  "oA_SalePriceTypesID": 0,  
  "salePriceType": "string",  
  "secondPerson": "string",  
  "swift_Code": "string",  
  "treasuryCode": "string",  
  "vatType": 0,  
  "zip": "string",  
  "debitorCreditorAddresses": [  
    {  
      "address": "string",  
      "comment": "string",  
      "city": "string",  
      "zip": "string",  
      "oA_CityID": 0,  
      "oA_CountryID": 0,  
      "country": "string"  
    }  
  ],  
  "debitorCreditorBankAccounts": [  
    {  
      "oA_BanksID": 0,  
      "account": "string",  
      "bankCode": "string",  
      "comment": "string"  
    }  
  ],  
  "debitorCreditorLegalIdentifiers": [  
    {  
      "number": "string",  
      "date": "2023-05-15T12:14:53.061Z",  
    }  
  ]  
}
```

```

    "issuedBy": "string",
    "comment": "string"
  }
],
"debtorCreditorPhones": [
  {
    "phone": "string",
    "comment": "string"
  }
]
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"address":

"birthDay":

"cityName": See [CitiesList] method (use [Name] value). Can be omitted if [OA_CityID] property specified

"clockNumber": Specify if [IsPerson]=true

"code": Specify identification code of organization if [IsPerson]=false or identity number of prson if [IsPerson]=true

"comment":

"countryName": See [CountriesList] method (use [Name] value). Can be omitted if [OA_CountryID] property specified

"debtorCreditorGroup": See [DebtorCreditorsList] method (use [Name] value). Can be omitted if [OA_DebtorCreditorGroupID] property specified

"firstPerson": Specify for organization if [IsPerson]=false

"isForeigner":

"isPerson": Specify true if debtor/creditor is "person", otherwise it will be "organization"

"name":

"oA_CityID": See [CitiesList] method (use [OA_CityID] value). Can be omitted if [CityName] property specified

"oA_CountryID": See [CountriesList] method (use [OA_CountryID] value). Can be omitted if [CountryName] property specified

"oA_DebtorCreditorGroupID": See [DebtorCreditorsList] method (use [OA_DebtorCreditorGroupID] value). Can be omitted if [DebtorCreditorGroup] property specified

"oA_SalePriceTypesID": See [SalePriceTypesList] method (use [OA_SalePriceTypesID] value). Can be omitted if [SalePriceType] property specified

"salePriceType": See [DebtorCreditorGroup] method (use [Description] value). Can be omitted if [OA_SalePriceTypesID] property specified

"secondPerson": Specify for organization if [IsPerson]=false

"swift_Code": Specify for organization if [IsPerson]=false

"treasuryCode": Specify for organization if [IsPerson]=false

"vatType": Undefined = -1, Standard = 0, ZeroRate = 1, Untaxed = 2

"zip":

"debtorCreditorAddresses":

"address":

"comment":

"city": See [CitiesList] method (use [Name] value). Can be omitted if [OA_CityID] property specified

"zip":

"oA_CityID": See [CitiesList] method (use [OA_CityID] value). Can be omitted if [City] property specified

"oA_CountryID": See [CountriesList] method (use [OA_CountryID] value). Can be omitted if [Country] property specified

"country": See [CountriesList] method (use [Name] value). Can be omitted if [OA_CountryID] property specified

"debtorCreditorBankAccounts":

"oA_BanksID": See [BanksList] method (use [OA_BanksID] value). Can be omitted if [BankCode] property specified

"account": See [AccountsList] method (use [Account] value)

"bankCode": See [BanksList] method (use [Code] value). Can be omitted if [OA_BanksID] property specified

"comment":

"debtorCreditorLegalIdentifiers":

"number": ",

"date":

"issuedBy":

"comment":

"debitorCreditorPhones":

"phone":

"comment":

Examples

1. Add "New Organization"- code : 12345

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "code": "12345",
  "name": "New organization"
}
```

Server response Code(200):

```
{
  "id": 1018,
  "idFieldName": "OA_DebitorCreditorID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51128,
    "description": "DebitorCreditor code already exists!",
    "comment": "Code [12345]\nName [New organization]\nOA_DebitorCreditorID [1018]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "DebitorCreditor",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

2. Add new Person, code "01010101010"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "code": "01010101010",
  "comment": "Add new person",
  "isForeigner": true,
  "isPerson": true,
  "name": "Person"
}
```

PUT/api/DebitorCreditor

Updates debtor/creditor (organization or person). Returns OK or Error description
Json.

Schema

```
{
  "token": "string",
  "oA_DebitorCreditorID": 0,
  "address": "string",
  "birthDay": "2023-05-16T07:41:57.367Z",
  "cityName": "string",
  "clockNumber": "string",
  "code": "string",
  "comment": "string",
  "countryName": "string",
  "debitorCreditorGroup": "string",
}
```



```

"firstPerson": "string",
"isForeigner": true,
"isPerson": true,
"name": "string",
"oA_CityID": 0,
"oA_CountryID": 0,
"oA_DebitorCreditorGroupID": 0,
"oA_SalePriceTypesID": 0,
"salePriceType": "string",
"secondPerson": "string",
"swift_Code": "string",
"treasuryCode": "string",
"vatType": 0,
"zip": "string",
"clearDebitorCreditorAddresses": true,
"clearDebitorCreditorBankAccounts": true,
"clearDebitorCreditorLegalIdentifiers": true,
"clearDebitorCreditorPhones": true,
"debitorCreditorAddresses": [
  {
    "oA_DebitorCreditorAddressesID": 0,
    "deleteRow": true,
    "address": "string",
    "comment": "string",
    "city": "string",
    "zip": "string",
    "oA_CityID": 0,
    "oA_CountryID": 0,
    "country": "string"
  }
],
"debitorCreditorBankAccounts": [
  {
    "oA_DebitorCreditorBankAccountsID": 0,
    "deleteRow": true,
    "oA_BanksID": 0,
    "account": "string",
    "bankCode": "string",
    "bankName": "string",
    "comment": "string"
  }
],
"debitorCreditorLegalIdentifiers": [
  {
    "oA_LegalIdentifiersID": 0,
    "deleteRow": true,
    "number": "string",
    "date": "2023-05-16T07:41:57.367Z",
    "issuedBy": "string",
    "comment": "string"
  }
],
"debitorCreditorPhones": [
  {
    "oA_DebitorCreditorPhonesID": 0,
    "deleteRow": true,
    "phone": "string",
    "comment": "string"
  }
]
}

```

Field description

[Token] and [OA_DebitorCreditorID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"**oA_DebitorCreditorID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value)

"**address**":

"**birthDay**":

"**cityName**": See [CitiesList] method (use [Name] value). Can be omitted if [OA_CityID] property specified

"**clockNumber**": Specify if [IsPerson]=true

"**code**": Specify identification code of organization if [IsPerson]=false or identity number of prson if [IsPerson]=true

"**comment**": "string",

"**countryName**": See [CountriesList] method (use [Name] value). Can be omitted if [OA_CountryID] property specified

"**debitorCreditorGroup**": See [DebitorCreditorsList] method (use [Name] value). Can be omitted if [OA_DebitorCreditorGroupID] property specified

"**firstPerson**": Specify for organization if [IsPerson]=false

"**isForeigner**":

"**isPerson**": Specify true if debtor/creditor is "person", otherwise it will be "organization"

"**name**":

"**oA_CityID**": See [CitiesList] method (use [OA_CityID] value). Can be omitted if [CityName] property specified

"**oA_CountryID**": See [CountriesList] method (use [OA_CountryID] value). Can be omitted if [CountryName] property specified

"**oA_DebitorCreditorGroupID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorGroupID] value). Can be omitted if [DebitorCreditorGroup] property specified

"**oA_SalePriceTypesID**": See [DebitorCreditorGroup] method (use [OA_SalePriceTypesID] value). Can be omitted if [SalePriceType] property specified

"**salePriceType**": See [DebitorCreditorGroup] method (use [Description] value). Can be omitted if [OA_SalePriceTypesID] property specified

"**secondPerson**": Specify for organization if [IsPerson]=false

"**swifT_Code**": Specify for organization if [IsPerson]=false

"**treasuryCode**": Specify for organization if [IsPerson]=false

"**vatType**": Undefined = -1, Standard = 0, ZeroRate = 1, Untaxed = 2

"**zip**":

"**clearDebitorCreditorAddresses**": Clears all addresses for the debtor/creditor

"**clearDebitorCreditorBankAccounts**": Clears all bank accounts for the debtor/creditor

"**clearDebitorCreditorLegalIdentifiers**": Clears all legal identifiers for the debtor/creditor

"**clearDebitorCreditorPhones**": Clears all phones for the debtor/creditor

"**debitorCreditorAddresses**": Set OA_DebitorCreditorAddressesID=0 to add a new address specification. Specify the OA_DebitorCreditorAddressesID value to update information about address fields. To delete existing address, specify the value OA_DebitorCreditorAddressesID and set "DeleteRow"=true

"**oA_DebitorCreditorAddressesID**": Specify an existing OA_DebitorCreditorAddressesID to update or remove address information, or 0 to add a new address

"**deleteRow**": If true, deletes address information for specified OA_DebitorCreditorAddressesID value

"**address**":

"**comment**":

"**city**": See [CitiesList] method (use [Name] value). Can be omitted if [OA_CityID] property specified

"**zip**":

"**oA_CityID**": See [CitiesList] method (use [OA_CityID] value). Can be omitted if [City] property specified

"**oA_CountryID**": See [CountriesList] method (use [OA_CountryID] value). Can be omitted if [Country] property specified

"**country**": See [CountriesList] method (use [Name] value). Can be omitted if [OA_CountryID] property specified

"**debitorCreditorBankAccounts**": Set OA_DebitorCreditorBankAccountsID=0 to add a new bank account specification. Specify the OA_DebitorCreditorBankAccountsID value to update information about bank account fields. To delete existing bank account, specify the value OA_DebitorCreditorBankAccountsID and set "DeleteRow"=true

"**oA_DebitorCreditorBankAccountsID**": Specify an existing OA_DebitorCreditorBankAccountsID to update or remove bank account information, or 0 to add a new bank account

"**deleteRow**": If true, deletes bank account information for specified OA_DebitorCreditorBankAccountsID value

"**oA_BanksID**": See [BanksList] method (use [OA_BanksID] value). Can be omitted if [BankCode] property specified

"**account**": See [AccountsList] method (use [Account] value)

"**bankCode**": See [BanksList] method (use [Code] value). Can be omitted if [OA_BanksID] property specified

"**bankName**":

"**comment**":

"**debitorCreditorLegalIdentifiers**": Set OA_LegalIdentifiersID=0 to add a new legal identifier specification. Specify the OA_LegalIdentifiersID value to update information about legal identifier fields. To delete existing legal identifier, specify the value OA_LegalIdentifiersID and set "DeleteRow"=true

"**oA_LegalIdentifiersID**": Specify an existing OA_LegalIdentifiersID to update or remove legal identifier information, or 0 to add a new legal identifier

"deleteRow": If true, deletes legal identifier information for specified OA_LegalIdentifiersID value

"number":

"date":

"issuedBy":

"comment":

"debtorCreditorPhones": Set OA_DebtorCreditorPhonesID=0 to add a new phones specification. Specify the OA_DebtorCreditorPhonesID value to update information about phones fields. To delete existing legal identifier, specify the value OA_DebtorCreditorPhonesID and set "DeleteRow"=true

"oA_DebtorCreditorPhonesID": Specify an existing OA_DebtorCreditorPhonesID to update or remove phone information, or 0 to add a new phone

"deleteRow": If true, deletes phone information for specified OA_DebtorCreditorPhonesID value

"phone":

"comment":

Examples

1. Update the details of "New Organization" using "oA_DebtorCreditorID" – 1018.
Add full address, Bank account - GE88BG0000000120121601 Bank of Georgia/BAGAGE22 and Phone number: +995 32 260 43 43

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_DebtorCreditorID": 1018,
  "address": "Al. Kazbegi Ave.",
  "cityName": "Tbilisi",
  "countryName": "Georgia",
  "zip": "0160",
  "debtorCreditorBankAccounts": [
    {
      "account": " GE88BG0000000120121601",
      "bankCode": "BAGAGE22",
      "bankName": "Bank of Georgia"
    }
  ],
  "debtorCreditorPhones": [
    {
      "phone": "+995 32 260 43 43"
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 1018,
  "idFieldName": "OA_DebtorCreditorID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_DebtorCreditorID = 1018",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "DebtorCreditor",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

2. Update the details of Person using "oA_DebtorCreditorID" – 1019.
Add full address, Bank account - GE88BG0000000120121601 Bank of Georgia/BAGAGE22
Phone number – +995 32 260 43 43

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_DebitorCreditorID": 1019,
  "address": "Al. Kazbegi Ave.",
  "cityName": "Tbilisi",
  "countryName": "Georgia",
  "zip": "0160",
  "debitorCreditorBankAccounts": [
    {
      "account": " GE88BG0000000120121601",
      "bankCode": "BAGAGE22",
      "bankName": "Bank of Georgia"
    }
  ],
  "debitorCreditorPhones": [
    {
      "phone": "+995 32 260 43 43"
    }
  ]
}
```

DELETE/api/DebitorCreditor

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_DebitorCreditorID": 0
}
```

Field description

[Token] and [OA_DebitorCreditorID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_DebitorCreditorID":

Examples

1. Delete "New Organization" From DebitorCreditor list, using "oA_DebitorCreditorID" – 1018

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_DebitorCreditorID": 1018
}
```

Server response Code(200):

```
{
  "id": 1018,
  "idFieldName": "OA_DebitorCreditorID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51251,
    "description": "Can't delete Debitor/Creditor. A link(s) has been found!",
    "comment": "ID [1018] Code [12345] Name [New organization]\n\nAccount [31 10 1018]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "DebitorCreditor",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
  }
]
```

```
"errorPointer": 0
}
]
```

GetDebtorCreditorByID

POST/api/GetDebtorCreditorByID

Gets Debtor/Creditor and its Addresses,Bank Accounts,Legal Identifiers and Phones by OA_DebtorCreditorID value.

Schema

```
{
  "token": "string",
  "oA_DebtorCreditorID": 0
}
```

Field description

[Token] and [OA_DebtorCreditorID] is necessary.

"token": Set [token] string from [LogIn] response value
"oA_DebtorCreditorID":

Examples

1. Get information about "New Organization" From DebtorCreditor list, using "oA_DebtorCreditorID" – 1018

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_DebtorCreditorID": 1018
}
```

Server response Code(200):

```
{
  "oA_DebtorCreditorID": 1018,
  "address": "Al. Kazbegi Ave.",
  "birthDay": "2023-05-17T16:30:06.0087686+04:00",
  "cityName": "Tbilisi",
  "clockNumber": "",
  "code": "12345",
  "comment": "",
  "countryName": "Georgia",
  "debtorCreditorGroup": "",
  "firstPerson": "",
  "isBithDayNull": true,
  "isForeigner": false,
  "isPerson": false,
  "name": "New organization",
  "oA_CityID": 1,
  "oA_CountryID": 82,
  "oA_DebtorCreditorGroupID": 0,
  "oA_SalePriceTypesID": 0,
  "salePriceType": "",
  "secondPerson": "",
  "swift_Code": "",
  "treasuryCode": "",
  "vatType": 0,
  "zip": "0160",
  "debtorCreditorAddresses": [],
  "debtorCreditorBankAccounts": [
    {
      "oA_DebtorCreditorBankAccountsID": 19,
      "oA_BanksID": 14,
```

```

    "account": " GE88BG000000120121601",
    "bankCode": "BAGAGE22",
    "bankName": "Bank of Georgia",
    "comment": ""
  }
],
"debitorCreditorLegalIdentifiers": [],
"debitorCreditorPhones": [
  {
    "oA_DebitorCreditorPhonesID": 1,
    "phone": "+995 32 260 43 43",
    "comment": ""
  }
]
}

```

Server response Code(400):

```

[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_DebitorCreditorID = 1018",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "DebitorCreditor",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

DebitorCreditorsList

POST/api/DebitorCreditorsList

Gets Debitor/Creditors. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get DebitorCreditor list, filter by name "New organization" and "Person", sort by code

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[Name] IN (N'New organization',N'Person')",
  "sort": "[Code]"
}

```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 2,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_DebitorCreditorID": 1019,
      "Code": "01010101010",
      "Name": "Person",
      "Is person": true,
      "City": "Tbilisi",
      "Address": "Al. Kazbegi Ave.",
      "Zip": "0160",
      "OA_CityID": 1,
      "SWIFT code": "",
      "First person": "",
      "Second person": "",
      "Treasury code": "",
      "Comment": "Add new person",
      "Clock number": "",
      "Is foreigner": true,
      "OA_CountryID": 82,
      "Country": "Georgia",
      "VAT type": 1
    },
    {
      "OA_DebitorCreditorID": 1018,
      "Code": "12345",
      "Name": "New organization",
      "Is person": false,
      "City": "Tbilisi",
      "Address": "Al. Kazbegi Ave.",
      "Zip": "0160",
      "OA_CityID": 1,
      "SWIFT code": "",
      "First person": "",
      "Second person": "",
      "Treasury code": "",
      "Comment": "",
      "Clock number": "",
      "Is foreigner": false,
      "OA_CountryID": 82,
      "Country": "Georgia",
      "VAT type": 0
    }
  ]
}
```

2. Get first two DebitorCreditor from list

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2
}
```

Organization

POST/api/Organization

Adds new Organization. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "address": "string",
  "cityName": "string",
  "code": "string",
  "comment": "string",
  "countryName": "string",
  "debitorCreditorGroup": "string",
  "firstPerson": "string",
  "isForeigner": true,
  "name": "string",
  "oA_CityID": 0,
  "oA_CountryID": 0,
  "oA_DebitorCreditorGroupID": 0,
  "oA_SalePriceTypesID": 0,
  "salePriceType": "string",
  "secondPerson": "string",
  "swifT_Code": "string",
  "treasuryCode": "string",
  "vatType": 0,
  "zip": "string",
  "organizationAddresses": [
    {
      "address": "string",
      "comment": "string",
      "sequenceNumber": 0,
      "city": "string",
      "zip": "string",
      "oA_CityID": 0,
      "oA_CountryID": 0,
      "country": "string"
    }
  ],
  "organizationBankAccounts": [
    {
      "oA_BanksID": 0,
      "account": "string",
      "bankCode": "string",
      "comment": "string"
    }
  ],
  "organizationPhones": [
    {
      "phone": "string",
      "comment": "string"
    }
  ]
}
```

Field description

"token": Set [token] string from [LogIn] response value

"address":

"cityName": See [CitiesList] method (use [Name] value). Can be omitted if [OA_CityID] property specified

"code": Specify identification code of organization

"comment":

"countryName": See [CountriesList] method (use [Name] value). Can be omitted if [OA_CountryID] property specified

"debitorCreditorGroup": See [DebitorCreditorsList] method (use [Name] value). Can be omitted if [OA_DebitorCreditorGroupID] property specified

"firstPerson":

"isForeigner":

"name":

"oA_CityID": See [CitiesList] method (use [OA_CityID] value). Can be omitted if [CityName] property specified
"oA_CountryID": See [CountriesList] method (use [OA_CountryID] value). Can be omitted if [CountryName] property specified
"oA_DebitorCreditorGroupID": See [DebitorCreditorsList] method (use [OA_DebitorCreditorGroupID] value). Can be omitted if [DebitorCreditorGroup] property specified
"oA_SalePriceTypesID": See [SalePriceTypesList] method (use [OA_SalePriceTypesID] value). Can be omitted if [SalePriceType] property specified
"salePriceType": See [SalePriceTypesList] method (use [Description] value). Can be omitted if [OA_SalePriceTypesID] property specified
"secondPerson":
"swifT_Code":
"treasuryCode":
"vatType": Undefined = -1, Standard = 0, ZeroRate = 1, Untaxed = 2
"zip":

"organizationAddresses":

"address":

"comment":

"sequenceNumber":

"city": See [CitiesList] method (use [Name] value). Can be omitted if [OA_CityID] property specified

"zip": "string",

"oA_CityID": See [CitiesList] method (use [OA_CityID] value). Can be omitted if [City] property specified

"oA_CountryID": See [CountriesList] method (use [OA_CountryID] value). Can be omitted if [Country] property specified

"country": See [CountriesList] method (use [Name] value). Can be omitted if [OA_CountryID] property specified

"organizationBankAccounts":

"oA_BanksID": See [BanksList] method (use [OA_BanksID] value). Can be omitted if [BankCode] property specified

"account": See [AccountsList] method (use [Account] value)

"bankCode": See [BanksList] method (use [Code] value). Can be omitted if [OA_BanksID] property specified

"comment":

"organizationPhones":

"phone":

"comment":

Examples

1. Add "Organization"- code : 1234567 , full address, Bank account - GE88BG000000120121601 Bank of Georgia/BAGAGE22 and Phone number - +995 32 260 43 43

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "address": "Al. Kazbegi Ave.",
  "cityName": "Tbilisi",
  "code": "1234567",
  "countryName": "Georgia",
  "name": "Organization",
  "organizationBankAccounts": [
    {
      "account": "GE88BG000000120121601",
      "bankCode": "BAGAGE22",
      "comment": "Bank of Georgia"
    }
  ],
  "organizationPhones": [
    {
      "phone": "+995 32 260 43 43"
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 1021,
  "idFieldName": "OA_DebitorCreditorID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "OA_DebitorCreditorID=1022 [1234567]\nA duplicate value cannot be inserted.\r\n[Code] = [1234567]\r\n\r\n[2601] Cannot insert duplicate key row in object 'dbo.OA_DebitorCreditor' with unique index 'IndxOA_DebitorCreditor1'. The duplicate key value is (1234567).[3621] The statement has been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Organization",
    "entryPoint": "Update",
    "resourceDLL_ForTranslation": "ORIS_Administrate",
    "errorPointer": 0
  }
]
```

PUT/api/Organization

Updates Organization. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_DebitorCreditorID": 0,
  "address": "string",
  "cityName": "string",
  "code": "string",
  "comment": "string",
  "countryName": "string",
  "debitorCreditorGroup": "string",
  "firstPerson": "string",
  "isForeigner": true,
  "name": "string",
  "oA_CityID": 0,
  "oA_CountryID": 0,
  "oA_DebitorCreditorGroupID": 0,
  "oA_SalePriceTypesID": 0,
  "salePriceType": "string",
  "secondPerson": "string",
  "swifT_Code": "string",
  "treasuryCode": "string",
  "vatType": 0,
  "zip": "string",
  "clearOrganizationAddresses": true,
  "clearOrganizationBankAccounts": true,
  "clearDebitorCreditorPhones": true,
  "organizationAddresses": [
    {
      "oA_DebitorCreditorAddressesID": 0,
      "deleteRow": true,
      "address": "string",
      "comment": "string",
      "city": "string",
      "zip": "string",
      "oA_CityID": 0,
      "oA_CountryID": 0,
      "country": "string"
    }
  ],
  "organizationBankAccounts": [
    {
      "oA_DebitorCreditorBankAccountsID": 0,
```

```

    "deleteRow": true,
    "oA_BanksID": 0,
    "account": "string",
    "bankCode": "string",
    "bankName": "string",
    "comment": "string"
  }
],
"organizationPhones": [
  {
    "oA_DebitorCreditorPhonesID": 0,
    "deleteRow": true,
    "phone": "string",
    "comment": "string"
  }
]
}

```

Field description

[Token] and [OA_DebitorCreditorID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_DebitorCreditorID":

"address":

"cityName": See [CitiesList] method (use [Name] value). Can be omitted if [OA_CityID] property specified

"code": Specify identification code of organization

"comment": "string",

"countryName": See [CountriesList] method (use [Name] value). Can be omitted if [OA_CountryID] property specified

"debitorCreditorGroup": See [DebitorCreditorsList] method (use [Name] value). Can be omitted if [OA_DebitorCreditorGroupID] property specified

"firstPerson": "string",

"isForeigner": true,

"name": "string",

"oA_CityID": See [CitiesList] method (use [OA_CityID] value). Can be omitted if [CityName] property specified

"oA_CountryID": See [CountriesList] method (use [OA_CountryID] value). Can be omitted if [CountryName] property specified

"oA_DebitorCreditorGroupID": See [DebitorCreditorsList] method (use [OA_DebitorCreditorGroupID] value). Can be omitted if [DebitorCreditorGroup] property specified

"oA_SalePriceTypesID": See [SalePriceTypesList] method (use [OA_SalePriceTypesID] value). Can be omitted if [SalePriceType] property specified

"salePriceType": See [SalePriceTypesList] method (use [Description] value). Can be omitted if [OA_SalePriceTypesID] property specified

"secondPerson":

"swifT_Code":

"treasuryCode":

"vatType": Undefined = -1, Standard = 0, ZeroRate = 1, Untaxed = 2

"zip":

"clearOrganizationAddresses": Clears all addresses for the Organization

"clearOrganizationBankAccounts": Clears all bank accounts for the Organization

"clearDebitorCreditorPhones": Clears all phones for the debitor/creditor

"organizationAddresses": Set OA_DebitorCreditorAddressesID=0 to add a new address specification. Specify the OA_DebitorCreditorAddressesID value to update information about address fields. To delete existing address, specify the value OA_DebitorCreditorAddressesID and set "DeleteRow"=true

"oA_DebitorCreditorAddressesID": Specify an existing OA_DebitorCreditorAddressesID to update or remove address information, or 0 to add a new address

"deleteRow": If true, deletes address information for specified OA_DebitorCreditorAddressesID value

"address":

"comment":

"city": See [CitiesList] method (use [Name] value). Can be omitted if [OA_CityID] property specified

"zip":

"oA_CityID": See [CitiesList] method (use [OA_CityID] value). Can be omitted if [City] property specified

"oA_CountryID": See [CountriesList] method (use [OA_CountryID] value). Can be omitted if [Country] property specified

"country": See [CountriesList] method (use [Name] value). Can be omitted if [OA_CountryID] property specified

"**organizationBankAccounts**": Set OA_DebitorCreditorBankAccountsID=0 to add a new bank account specification. Specify the OA_DebitorCreditorBankAccountsID value to update information about bank account fields. To delete existing bank account, specify the value OA_DebitorCreditorBankAccountsID and set "DeleteRow"=true
"oA_DebitorCreditorBankAccountsID": Specify an existing OA_DebitorCreditorBankAccountsID to update or remove bank account information, or 0 to add a new bank account
"deleteRow": If true, deletes bank account information for specified OA_DebitorCreditorBankAccountsID value
"oA_BanksID": See [BanksList] method (use [OA_BanksID] value). Can be omitted if [BankCode] property specified
"account": See [AccountsList] method (use [Account] value)
"bankCode": See [BanksList] method (use [Code] value). Can be omitted if [OA_BanksID] property specified
"bankName":
"comment":

"**organizationPhones**": Set OA_DebitorCreditorPhonesID=0 to add a new phones specification. Specify the OA_DebitorCreditorPhonesID value to update information about phones fields. To delete existing legal identifier, specify the value OA_DebitorCreditorPhonesID and set "DeleteRow"=true
"oA_DebitorCreditorPhonesID": Specify an existing OA_DebitorCreditorPhonesID to update or remove phone information, or 0 to add a new phone
"deleteRow": If true, deletes phone information for specified OA_DebitorCreditorPhonesID value
"phone": "string",
"comment": "string"

Examples

1. Update the details of Organization using "oA_DebitorCreditorID" – 1021.
 Add Debitor/creditor Group using "oA_DebitorCreditorGroupID" – 1,
 change VAT tipe to 'Zero Rate'- 1 and clear Phone number: +995 32 260 43 43
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_DebitorCreditorID": 1021,
  "oA_DebitorCreditorGroupID": 1,
  "vatType": 1,
  "clearDebitorCreditorPhones": true
}
```

Server response Code(200):

```
{
  "id": 1021,
  "idFieldName": "OA_DebitorCreditorID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_DebitorCreditorID = 1021",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Organization",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/Organization

Returns OK or Error desription Json.

Schema

```
{
  "token": "string",
```

```
"oA_DebitorCreditorID": 0
}
```

Field description

[Token] and [OA_DebitorCreditorID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_DebitorCreditorID":

Examples

1. Delete Organization using "oA_DebitorCreditorID" – 1021

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_DebitorCreditorID": 1021
}
```

Server response Code(200):

```
{
  "id": 1021,
  "idFieldName": "OA_DebitorCreditorID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51251,
    "description": "Can't delete Debitor/Creditor. A link(s) has been found!",
    "comment": "ID [1021] Code [1234567] Name [Organization]\n\nAccount [1410 1021]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Organization",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

GetOrganizationByID

POST/api/GetOrganizationByID

Gets Organization and its Addresses,Bank Accounts and Phones by OA_DebitorCreditorID value.

Schema

```
{
  "token": "string",
  "oA_DebitorCreditorID": 0
}
```

Field description

[Token] and [OA_DebitorCreditorID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_DebitorCreditorID":

Examples

1. Get information about Organization by "oA_DebitorCreditorID" -1021

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",

```

```
"oA_DebitorCreditorID": 1021
}
```

Server response Code(200):

```
{
  "oA_DebitorCreditorID": 0,
  "address": "Al. Kazbegi Ave.",
  "cityName": "Tbilisi",
  "code": "1234567",
  "comment": "",
  "countryName": "Georgia",
  "debitorCreditorGroup": "Debitor/creditor Group",
  "firstPerson": "",
  "isForeigner": false,
  "name": "Organization",
  "oA_CityID": 1,
  "oA_CountryID": 82,
  "oA_DebitorCreditorGroupID": 1,
  "oA_SalePriceTypesID": 0,
  "salePriceType": "",
  "secondPerson": "",
  "swift_Code": "",
  "treasuryCode": "",
  "vatType": 1,
  "zip": "",
  "organizationAddresses": [],
  "organizationBankAccounts": [
    {
      "oA_DebitorCreditorBankAccountsID": null,
      "oA_BanksID": 14,
      "account": "GE88BG000000120121601",
      "bankCode": "BAGAGE22",
      "bankName": null,
      "comment": "Bank of Georgia"
    }
  ],
  "organizationPhones": []
}
```

OrganizationsList

POST/api/OrganizationsList

Gets Organizations. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get first 2 organizations from list sort by "oA_DebitorCreditorID" in descending order

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2,
  "sort": "[OA_DebitorCreditorID] DESC"
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 2,
      "PageCount": 0,
      "PageRowCount": 2,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_DebitorCreditorID": 1021,
      "Identification code": "1234567",
      "Name": "Organization",
      "IsPerson": false,
      "City": "Tbilisi",
      "Address": "Al. Kazbegi Ave.",
      "Zip": "",
      "OA_CityID": 1,
      "SWIFT code": "",
      "First person": "",
      "Second person": "",
      "Treasury code": "",
      "Comment": "",
      "Is foreigner": false,
      "OA_CountryID": 82,
      "Country": "Georgia",
      "VAT type": 1,
      "OA_DebitorCreditorGroupID": 1,
      "Group": "Debitor/creditor Group"
    },
    {
      "OA_DebitorCreditorID": 1018,
      "Identification code": "12345",
      "Name": "New organization",
      "IsPerson": false,
      "City": "Tbilisi",
      "Address": "Al. Kazbegi Ave.",
      "Zip": "0160",
      "OA_CityID": 1,
      "SWIFT code": "",
      "First person": "",
      "Second person": "",
      "Treasury code": "",
      "Comment": "",
      "Is foreigner": false,
      "OA_CountryID": 82,
      "Country": "Georgia",
      "VAT type": 0
    }
  ]
}
```

2. Get a list of organizations with "Debitor/creditor Group"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[Group] IN (N'Debitor/creditor Group)"
}
```

```
}
```

Person

POST/api/Person

Adds new Person. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "address": "string",
  "birthDay": "2023-05-19T10:10:21.472Z",
  "cityName": "string",
  "clockNumber": "string",
  "code": "string",
  "comment": "string",
  "countryName": "string",
  "debitorCreditorGroup": "string",
  "isBithDayNull": true,
  "isForeigner": true,
  "name": "string",
  "oA_CityID": 0,
  "oA_CountryID": 0,
  "oA_DebitorCreditorGroupID": 0,
  "oA_SalePriceTypesID": 0,
  "salePriceType": "string",
  "vatType": 0,
  "zip": "string",
  "personAddresses": [
    {
      "address": "string",
      "comment": "string",
      "city": "string",
      "zip": "string",
      "oA_CityID": 0,
      "oA_CountryID": 0,
      "country": "string"
    }
  ],
  "personBankAccounts": [
    {
      "oA_BanksID": 0,
      "account": "string",
      "bankCode": "string",
      "comment": "string"
    }
  ],
  "personLegalIdentifiers": [
    {
      "number": "string",
      "date": "2023-05-19T10:10:21.472Z",
      "issuedBy": "string",
      "comment": "string"
    }
  ],
  "personPhones": [
    {
      "phone": "string",
      "comment": "string"
    }
  ]
}
```


Field description

[Token] is necessary. Specify only those fields that you want to fill.

"**token**": Set [token] string from [LogIn] response value

"**address**":

"**birthDay**":

"**cityName**": See [CitiesList] method (use [Name] value). Can be omitted if [OA_CityID] property specified

"**clockNumber**":

"**code**":

"**comment**":

"**countryName**": See [CountriesList] method (use [Name] value). Can be omitted if [OA_CountryID] property specified

"**debitorCreditorGroup**": See [DebitorCreditorsList] method (use [Name] value). Can be omitted if [OA_DebitorCreditorGroupID] property specified

"**isBithDayNull**": Set True for clear [BithDay] value

"**isForeigner**":

"**name**":

"**oA_CityID**": See [CitiesList] method (use [OA_CityID] value). Can be omitted if [CityName] property specified

"**oA_CountryID**": See [CountriesList] method (use [OA_CountryID] value). Can be omitted if [CountryName] property specified

"**oA_DebitorCreditorGroupID**": See [DebitorCreditorsList] method (use [OA_DebitorCreditorGroupID] value). Can be omitted if [DebitorCreditorGroup] property specified

"**oA_SalePriceTypesID**": See [SalePriceTypesList] method (use [OA_SalePriceTypesID] value). Can be omitted if [SalePriceType] property specified

"**salePriceType**": See [SalePriceTypesList] method (use [Description] value). Can be omitted if [OA_SalePriceTypesID] property specified

"**vatType**": Undefined = -1, Standard = 0, ZeroRate = 1, Untaxed = 2

"**zip**": "string",

"**personAddresses**":

"**address**":

"**comment**":

"**city**": See [CitiesList] method (use [Name] value). Can be omitted if [OA_CityID] property specified

"**zip**":

"**oA_CityID**": See [CitiesList] method (use [OA_CityID] value). Can be omitted if [City] property specified

"**oA_CountryID**": See [CountriesList] method (use [OA_CountryID] value). Can be omitted if [Country] property specified

"**country**": See [CountriesList] method (use [Name] value). Can be omitted if [OA_CountryID] property specified

"**personBankAccounts**":

"**oA_BanksID**": See [BanksList] method (use [OA_BanksID] value). Can be omitted if [BankCode] property specified

"**account**": See [AccountsList] method (use [Account] value)

"**bankCode**": See [BanksList] method (use [Code] value). Can be omitted if [OA_BanksID] property specified

"**comment**":

"**personLegalIdentifiers**":

"**number**":

"**date**":

"**issuedBy**":

"**comment**":

"**personPhones**":

"**phone**":

"**comment**":

Examples

1. Add Person 01011022033 – birth date "2005-05-19", address using "oA_CityID" – 1 'Tbilisi', "oA_CountryID" – 82 'Georgia',
Person Addresses: 0160, Al. Kazbegi Ave.12,Tbilisi,Georgia.
Bank account: GE88BG0000000120121601, BAGAGE22 , using "oA_BanksID" - 14 'Bank of Georgia'
Phone number : +995 32 260 43 43
Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "address": "Al. Kazbegi Ave.",  
  "birthDay": "2005-05-19",
```

```
"code": "01011022033",
"name": "Person",
"oA_CityID": 1,
"oA_CountryID": 82,
"personAddresses": [
  {
    "address": "Al. Kazbegi Ave.12",
    "city": "Tbilisi",
    "zip": "0160",
    "country": "Georgia"
  }
],
"personBankAccounts": [
  {
    "oA_BanksID": 14,
    "account": "GE88BG0000000120121601",
    "bankCode": "BAGAGE22"
  }
],
"personPhones": [
  {
    "phone": "+995 32 260 43 43"
  }
]
}
```

Server response Code(200):

```
{
  "id": 1024,
  "idFieldName": "OA_DebitorCreditorID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51128,
    "description": "DebitorCreditor code already exists!",
    "comment": "Code [01011022033]\nName [someone]\noA_DebitorCreditorID [1002]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Person",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/Person

Updates Person. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_DebitorCreditorID": 0,
  "address": "string",
  "birthDay": "2023-05-19T11:59:00.647Z",
  "cityName": "string",
  "clockNumber": "string",
  "code": "string",
  "comment": "string",
  "countryName": "string",
  "debitorCreditorGroup": "string",
  "isBithDayNull": true,
}
```

```

"isForeigner": true,
"name": "string",
"oA_CityID": 0,
"oA_CountryID": 0,
"oA_DebitorCreditorGroupID": 0,
"oA_SalePriceTypesID": 0,
"salePriceType": "string",
"vatType": 0,
"zip": "string",
"clearPersonAddresses": true,
"clearPersonBankAccounts": true,
"clearPersonLegalIdentifiers": true,
"clearPersonPhones": true,
"personAddresses": [
  {
    "oA_DebitorCreditorAddressesID": 0,
    "deleteRow": true,
    "address": "string",
    "comment": "string",
    "city": "string",
    "zip": "string",
    "oA_CityID": 0,
    "oA_CountryID": 0,
    "country": "string"
  }
],
"personBankAccounts": [
  {
    "oA_DebitorCreditorBankAccountsID": 0,
    "deleteRow": true,
    "oA_BanksID": 0,
    "account": "string",
    "bankCode": "string",
    "bankName": "string",
    "comment": "string"
  }
],
"personLegalIdentifiers": [
  {
    "oA_LegalIdentifiersID": 0,
    "deleteRow": true,
    "number": "string",
    "date": "2023-05-19T11:59:00.647Z",
    "issuedBy": "string",
    "comment": "string"
  }
],
"personPhones": [
  {
    "oA_DebitorCreditorPhonesID": 0,
    "deleteRow": true,
    "phone": "string",
    "comment": "string"
  }
]
}

```

Field description

[Token] and [OA_DebitorCreditorID] is necessary. Specify only those fields that you want to change.

"**token**": Set [token] string from [Login] response value

"**oA_DebitorCreditorID**":

"**address**":

"**birthDay**":

"**cityName**": See [CitiesList] method (use [Name] value). Can be omitted if [OA_CityID] property specified

"**clockNumber**":

"code":
"comment":
"countryName": See [CountriesList] method (use [Name] value). Can be omitted if [OA_CountryID] property specified
"debtorCreditorGroup": See [DebtorCreditorsList] method (use [Name] value). Can be omitted if [OA_DebtorCreditorGroupID] property specified
"isBithDayNull": Set True for clear [BithDay] value
"isForeigner":
"name":
"oA_CityID": See [CitiesList] method (use [OA_CityID] value). Can be omitted if [CityName] property specified
"oA_CountryID": See [CountriesList] method (use [OA_CountryID] value). Can be omitted if [CountryName] property specified
"oA_DebtorCreditorGroupID": See [DebtorCreditorsList] method (use [OA_DebtorCreditorGroupID] value). Can be omitted if [DebtorCreditorGroup] property specified
"oA_SalePriceTypesID": See [SalePriceTypesList] method (use [OA_SalePriceTypesID] value). Can be omitted if [SalePriceType] property specified
"salePriceType": See [SalePriceTypesList] method (use [Description] value). Can be omitted if [OA_SalePriceTypesID] property specified
"vatType": Undefined = -1, Standard = 0, ZeroRate = 1, Untaxed = 2
"zip":
"clearPersonAddresses": Clears all addresses for the debtor/creditor
"clearPersonBankAccounts": Clears all bank accounts for the debtor/creditor
"clearPersonLegalIdentifiers": Clears all legal identifiers for the debtor/creditor
"clearPersonPhones": Clears all phones for the debtor/creditor

"personAddresses": Set OA_DebtorCreditorAddressesID=0 to add a new address specification. Specify the OA_DebtorCreditorAddressesID value to update information about address fields. To delete existing address, specify the value OA_DebtorCreditorAddressesID and set "DeleteRow"=true
"oA_DebtorCreditorAddressesID": Specify an existing OA_DebtorCreditorAddressesID to update or remove address information, or 0 to add a new address
"deleteRow": true,
"address": If true, deletes address information for specified OA_DebtorCreditorAddressesID value
"comment":
"city": See [CitiesList] method (use [Name] value). Can be omitted if [OA_CityID] property specified.
"zip":
"oA_CityID": See [CitiesList] method (use [OA_CityID] value). Can be omitted if [City] property specified
"oA_CountryID": See [CountriesList] method (use [OA_CountryID] value). Can be omitted if [Country] property specified
"country": See [CountriesList] method (use [Name] value). Can be omitted if [OA_CountryID] property specified

"personBankAccounts": Set OA_DebtorCreditorBankAccountsID=0 to add a new bank account specification. Specify the OA_DebtorCreditorBankAccountsID value to update information about bank account fields. To delete existing bank account, specify the value OA_DebtorCreditorBankAccountsID and set "DeleteRow"=true
"oA_DebtorCreditorBankAccountsID": Specify an existing OA_DebtorCreditorBankAccountsID to update or remove bank account information, or 0 to add a new bank account
"deleteRow": If true, deletes bank account information for specified OA_DebtorCreditorBankAccountsID value
"oA_BanksID": See [BanksList] method (use [OA_BanksID] value). Can be omitted if [BankCode] property specified
"account": See [AccountsList] method (use [Account] value)
"bankCode": See [BanksList] method (use [Code] value). Can be omitted if [OA_BanksID] property specified
"bankName":
"comment":

"personLegalIdentifiers": Set OA_LegalIdentifiersID=0 to add a new legal identifier specification. Specify the OA_LegalIdentifiersID value to update information about legal identifier fields. To delete existing legal identifier, specify the value OA_LegalIdentifiersID and set "DeleteRow"=true
"oA_LegalIdentifiersID": Specify an existing OA_LegalIdentifiersID to update or remove legal identifier information, or 0 to add a new legal identifier
"deleteRow": If true, deletes legal identifier information for specified OA_LegalIdentifiersID value
"number":
"date":
"issuedBy":
"comment":

"personPhones": Set OA_DebtorCreditorPhonesID=0 to add a new phones specification. Specify the OA_DebtorCreditorPhonesID value to update information about phones fields. To delete existing legal identifier, specify the value OA_DebtorCreditorPhonesID and set "DeleteRow"=true

"oA_DebitorCreditorPhonesID": Specify an existing OA_DebitorCreditorPhonesID to update or remove phone information, or 0 to add a new phone

"deleteRow": If true, deletes phone information for specified OA_DebitorCreditorPhonesID value

"phone":

"comment":

Examples

1. Update the details of Person using "oA_DebitorCreditorID" – 1024.
Change second Address: Al. Kazbegi Ave.14, using oA_CityID :1 - Tbilisi, "oA_CountryID" - 82 Georgia.
Add Phone number : +995 32 260 43 41
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_DebitorCreditorID": 1024,
  "clearPersonBankAccounts": false,
  "personAddresses": [
    {
      "oA_DebitorCreditorAddressesID":10,
      "address": "Al. Kazbegi Ave.14",
      "oA_CityID":1,
      "oA_CountryID":82
    }
  ],
  "personPhones": [
    {
      "A_DebitorCreditorPhonesID":0,
      "phone": "+995 32 260 43 41"
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 1024,
  "idFieldName": "OA_DebitorCreditorID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "OA_DebitorCreditorID=1024 [01011022033]\nExist relation to another table.\n\nRelated table [OA_DebitorCreditorAddresses].\n\n\n[547] The UPDATE statement conflicted with the FOREIGN KEY constraint \"FK_OA_DebitorCreditorAddresses_OA_Country\". The conflict occurred in database \"NDatabaseEN\", table \"dbo.OA_Country\", column 'OA_CountryID'.[3621] The statement has been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Person",
    "entryPoint": "Update",
    "resourceDLL_ForTranslation": "ORIS_Administrate",
    "errorPointer": 0
  }
]
```

DELETE/api/Person

Returns OK or Error description Json.

Schema

```
{
  "token": "string",

```

```
"oA_DebitorCreditorID": 0
}
```

Field description

[Token] and [OA_DebitorCreditorID] is necessary

"token": Set [token] string from [LogIn] response value

"oA_DebitorCreditorID":

Examples

1. Delete Person using "oA_DebitorCreditorID " – 1024

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_DebitorCreditorID": 1024
}
```

Server response Code(200):

```
{
  "id": 1024,
  "idFieldName": "OA_DebitorCreditorID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51251,
    "description": "Can't delete Debitor/Creditor. A link(s) has been found!",
    "comment": "ID [1024] Code [01011022033] Name [Person]\n\nAccount [3130 1024]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Person",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

GetPersonByID

POST/api/GetPersonByID

Gets Debitor/Creditor and its Addresses,Bank Accounts,Legal Identifiers and Phones by OA_DebitorCreditorID value.

Schema

```
{
  "token": "string",
  "oA_DebitorCreditorID": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"oA_DebitorCreditorID":

Examples

1. Get information about Perdon using "oA_DebitorCreditorID" - 1024

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_DebitorCreditorID": 1024
}
```

Server response Code(200):

```
{
  "oA_DebitorCreditorID": 1024,
  "address": "Al. Kazbegi Ave.",
  "birthDay": "2023-05-25T11:59:28.1187958+04:00",
  "cityName": "Tbilisi",
  "clockNumber": "",
  "code": "01011022033",
  "comment": "",
  "countryName": "Georgia",
  "debitorCreditorGroup": "",
  "isBithDayNull": true,
  "isForeigner": false,
  "name": "Person",
  "oA_CityID": 1,
  "oA_CountryID": 82,
  "oA_DebitorCreditorGroupID": 0,
  "oA_SalePriceTypesID": 0,
  "salePriceType": "",
  "vatType": 0,
  "zip": "",
  "personAddresses": [
    {
      "oA_DebitorCreditorAddressesID": 10,
      "oA_DebitorCreditorID": 1024,
      "address": "Al. Kazbegi Ave.14",
      "comment": "",
      "sequenceNumber": 1,
      "city": "Tbilisi",
      "zip": "0160",
      "oA_CityID": 1,
      "oA_CountryID": 82,
      "country": "Georgia"
    }
  ],
  "personBankAccounts": [
    {
      "oA_DebitorCreditorBankAccountsID": 1021,
      "oA_DebitorCreditorID": 1024,
      "oA_BanksID": 14,
      "account": "GE88BG0000000120121601",
      "bankCode": "BAGAGE22",
      "bankName": "Bank of Georgia",
      "code": "01011022033",
      "comment": "",
      "sequenceNumber": 1
    }
  ],
  "personLegalIdentifiers": [],
  "personPhones": [
    {
      "oA_DebitorCreditorPhonesID": 1007,
      "oA_DebitorCreditorID": 1024,
      "phone": "+995 32 260 43 43",
      "comment": "",
      "sequenceNumber": 1
    },
    {
      "oA_DebitorCreditorPhonesID": 1008,
      "oA_DebitorCreditorID": 1024,
      "phone": "+995 32 260 43 41",
      "comment": "",
      "sequenceNumber": 2
    }
  ]
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_DebitorCreditorID = 1024",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Person",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PersonsList

POST/api/PersonsList

Gets Persons. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get first 2 Person from database

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 2,
      "PageCount": 0,
      "PageRowCount": 2,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_DebitorCreditorID": 2,
      "Identity number": "2",
      "Name": "Person333",
    }
  ]
}
```



```

    "IsPerson": true,
    "Clock number": "",
    "Address": "",
    "Zip": "",
    "Comment": "",
    "Is foreigner": false,
    "VAT type": 0
  },
  {
    "OA_DebitorCreditorID": 1002,
    "Identity number": "12345678910",
    "Name": "someone",
    "IsPerson": true,
    "Clock number": "",
    "Address": "",
    "Zip": "",
    "Comment": "",
    "Is foreigner": false,
    "VAT type": 0
  }
]

```

2. Get the 2nd page from the entire list of Persons, 1 records per page

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "pageNumber": 2
}

```

3. Get all field names for further use in the filter

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1
}

```

4. Get a list of Person with Tbilisi in city name and sort by Code

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[City] IN (N'Tbilisi)"
  "sort": "[Code]"
}

```

Item

POST/api/Item

Adds item (commodity, fixed asset or service). Returns ID value or Error description Json.

Schema

```

{
  "token": "string",
  "closeDate": "2023-05-24T08:19:55.147Z",
  "code": "string",
  "codeContainsWeight": true,
  "comment": "string",
  "currencies": "string",
  "dateOfExpiration": "2023-05-24T08:19:55.147Z",
  "exciseRate": 0,
  "exciseUnit": "string",
  "hasDoubleRegisterUnit": true,
  "inventoryNumber": "string",

```

```

"isCloseDateNULL": true,
"isDateOfExpirationNull": true,
"itemGroup": "string",
"itemType": 0,
"itemWriteOffExpenseAccount": "string",
"manufacturer": "string",
"manufacturerNumber": "string",
"name": "string",
"oA_AccountsID": 0,
"oA_AccountsIDExpenseAccount": 0,
"oA_CurrenciesID": 0,
"oA_DebitorCreditorID": 0,
"oA_ItemGroupID": 0,
"oA_ManufacturersID": 0,
"oA_StockUnitsID": 0,
"oA_StockUnitsID_ExciseUnit": 0,
"oA_StockUnitsID_SaleUnit": 0,
"priceIncreasePercentage": 0,
"registerParentAccount": "string",
"responsiblePerson": "string",
"salePrice": 0,
"saleUnit": "string",
"unit": "string",
"vatType": 0,
"writtenOff": true,
"itemUnits": [
  {
    "measure": 0,
    "relatedOA_StockUnitsID": 0,
    "relatedUnit": "string",
    "relatedUnitQuantity": 0
  }
]
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [Login] response value

"closeDate":

"code":

"codeContainsWeight": Sets code contains the weight. Code positions 1-7. Weight positions (whole part positions 8-9 and fraction part positions 10-12) will be used to managerial transaction . Default=False

"comment":

"currencies": Sets sale price currency. See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified

"dateOfExpiration":

"exciseRate":

"exciseUnit": See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID_ExciseUnit] property specified

"hasDoubleRegisterUnit": If true - when performing operations with this item, you must specify the quantity in both units: in the transaction unit [SaleUnit] and in the register minimum unit [Unit]

"inventoryNumber":

"isCloseDateNULL": Set True to clear [CloseDate] value

"isDateOfExpirationNull": Set True to clear [DateOfExpiration] value

"itemGroup": See [ItemGroupsList] method (use [Name] value). Can be omitted if [OA_ItemGroupID] property specified

"itemType": Commodity = 0, FixedAsset = 1, Service = 2

"itemWriteOffExpenseAccount": See [AccountsList] method (use [Account] value). Can be omitted if [OA_AccountsIDExpenseAccount] property specified

"manufacturer": See [ManufacturersList] method (use [Name] value). Can be omitted if [OA_ManufacturersID] property specified

"manufacturerNumber":

"name": Name of commodity if [ItemType]=0, fixed asset if [ItemType]=1 or service if [ItemType]=1

"oA_AccountsID": Register parent account. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [RegisterParentAccount] property specified

"**oA_AccountsIDExpenseAccount**": Will be used in managerial transaction. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [ItemWriteOffExpenseAccount] property specified

"**oA_CurrenciesID**": Sets sale price currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currencies] property specified

"**oA_DebitorCreditorID**": Responsible Person. See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [ResponsiblePerson] property specified

"**oA_ItemGroupID**": See [ItemGroupsList] method (use [OA_ItemGroupID] value). Can be omitted if [ItemGroup] property specified

"**oA_ManufacturersID**": See [ManufacturersList] method (use [OA_ManufacturersID] value). Can be omitted if [Manufacturer] property specified

"**oA_StockUnitsID**": Register minimum unit. See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified

"**oA_StockUnitsID_ExciseUnit**": See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [ExciseUnit] property specified

"**oA_StockUnitsID_SaleUnit**": Sale Unit. See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [SaleUnit] property specified

"**priceIncreasePercentage**": Will be used when forming sale price

"**registerParentAccount**": Register parent account. See [AccountsList] method (use [Account] value). Can be omitted if [OA_AccountsID] property specified

"**responsiblePerson**": Responsible Person. See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [OA_DebitorCreditorID] property specified

"**salePrice**":

"**saleUnit**": See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID_SaleUnit] property specified

"**unit**": Register minimum unit. See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified

"**vatType**": Standard = 0, ZeroRate = 1, Untaxed = 2

"**writtenOff**":

"**itemUnits**":

"**measure**": Quantity of goods in the corresponding unit in relation to the minimum register unit of goods. Example: If the product is "minimum register unit"="piece" and "related unit"="box" and there are 10 items in one box, then specify "Measure"=10

"**relatedOA_StockUnitsID**": See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [RelatedUnit] property specified

"**relatedUnit**": See [StockUnitsList] method (use [Unit] value). Can be omitted if [RelatedOA_StockUnitsID] property specified

"**relatedUnitQuantity**": Quantity of goods in the corresponding unit. Example: If the product is "minimum register unit"="piece" and "related unit"="box" and there are 12 items in 3 box, then specify "Measure"=12 and "RelatedUnitQuantity"=3

Examples

1. Add new item: type – commodity. Code - "1000001", name – "Commodity", ParentAccount – "1610", unit – "Piece" and related unit – "Kg" '1 piece = 10 kg'

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "code": "1000001",
  "itemType": 1,
  "name": "Commodity",
  "registerParentAccount": "1610",
  "unit": "Piece",
  "itemUnits": [
    {
      "measure": 1,
      "relatedUnit": "Kg",
      "relatedUnitQuantity": 10
    }
  ]
}
```

Server response Code(200):

```
{
  "id": 3005,
  "idFieldName": "OA_ItemsID"
}
```

Server response Code(400):

```
[
{
  "errorNumber": 51120,
  "description": "Unit does not exist!",
  "comment": "OA_ItemsID=0 [1000001] [Commodity]\nUnit=Kg",
  "command": "",
  "errorType": "ErrorType_Error",
  "source": "OA_BusinessObject.dll",
  "class": "Item",
  "entryPoint": "CheckAccountProperties",
  "resourceDLL_ForTranslation": "ORIS_Accounting",
  "errorPointer": 0
}
]
```

2. Add new item: type – Fixed asset. Code - “2000002”, name – “Fixed Asset”, ParentAccount – “2120”, unit – “Piece”

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "code": "2000002",
  "itemType": 1,
  "name": "Fixed Asset",
  "registerParentAccount": "2120",
  "unit": "Piece"
}
```

3. Add new item: type – Service. Code - “3000003”, name – “Service”, ParentAccount – “7425”, unit – “Service”

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "code": "3000003",
  "itemType": 2,
  "name": " Service ",
  "registerParentAccount": "7425",
  "unit": " Service "
}
```

PUT/api/Item

Updates item (commodity, fixed asset or service). Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_ItemsID": 0,
  "closeDate": "2023-05-25T10:44:43.562Z",
  "code": "string",
  "codeContainsWeight": true,
  "comment": "string",
  "currencies": "string",
  "dateOfExpiration": "2023-05-25T10:44:43.562Z",
  "exciseRate": 0,
  "exciseUnit": "string",
  "hasDoubleRegisterUnit": true,
  "inventoryNumber": "string",
  "isCloseDateNULL": true,
  "isDateOfExpirationNull": true,
  "itemGroup": "string",
  "itemType": 0,
  "itemWriteOffExpenseAccount": "string",
}
```

```

"manufacturer": "string",
"manufacturerNumber": "string",
"name": "string",
"oA_AccountsID": 0,
"oA_AccountsIDExpenseAccount": 0,
"oA_CurrenciesID": 0,
"oA_DebitorCreditorID": 0,
"oA_ItemGroupID": 0,
"oA_ManufacturersID": 0,
"oA_StockUnitsID": 0,
"oA_StockUnitsID_ExciseUnit": 0,
"oA_StockUnitsID_SaleUnit": 0,
"priceIncreasePercentage": 0,
"registerParentAccount": "string",
"responsiblePerson": "string",
"salePrice": 0,
"saleUnit": "string",
"unit": "string",
"vatType": 0,
"writtenOff": true,
"clearItemUnits": true,
"itemUnits": [
  {
    "oA_ItemStockUnitsID": 0,
    "deleteRow": true,
    "measure": 0,
    "relatedOA_StockUnitsID": 0,
    "relatedUnit": "string",
    "relatedUnitQuantity": 0
  }
]
}

```

Field description

[Token] and [OA_ItemsID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_ItemsID": See [ItemsList] method (use [OA_ItemsID] value)

"closeDate":

"code":

"codeContainsWeight": Sets code contains the weight. Code positions 1-7. Weight positions (whole part positions 8-9 and fraction part positions 10-12) will be used to managerial transaction . Default=False

"comment":

"currencies": Sets sale price currency. See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified

"dateOfExpiration":

"exciseRate":

"exciseUnit": See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID_ExciseUnit] property specified

"hasDoubleRegisterUnit": If true - when performing operations with this item, you must specify the quantity in both units: in the transaction unit [SaleUnit] and in the register minimum unit [Unit]

"inventoryNumber":

"isCloseDateNULL": Set True to clear [CloseDate] value

"isDateOfExpirationNull": Set True to clear [DateOfExpiration] value

"itemGroup": See [ItemGroupsList] method (use [Name] value). Can be omitted if [OA_ItemGroupID] property specified

"itemType": Commodity = 0, FixedAsset = 1, Service = 2

"itemWriteOffExpenseAccount": See [AccountsList] method (use [Account] value). Can be omitted if [OA_AccountsIDExpenseAccount] property specified

"manufacturer": See [ManufacturersList] method (use [Name] value). Can be omitted if [OA_ManufacturersID] property specified "manufacturerNumber": "string",

"name": Name of commodity if [ItemType]=0, fixed asset if [ItemType]=1 or service if [ItemType]=1

"oA_AccountsID": Register parent account. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [RegisterParentAccount] property specified

"oA_AccountsIDExpenseAccount": Will be used in managarial transaction. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [ItemWriteOffExpenseAccount] property specified

"**oA_CurrenciesID**": Sets sale price currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currencies] property specified

"**oA_DebitorCreditorID**": Responsible Person. See [DebitorCreditorsList] method (use [OA_DebitorCreditorID] value). Can be omitted if [ResponsiblePerson] property specified

"**oA_ItemGroupID**": See [ItemGroupsList] method (use [OA_ItemGroupID] value). Can be omitted if [ItemGroup] property specified

"**oA_ManufacturersID**": See [ManufacturersList] method (use [OA_ManufacturersID] value). Can be omitted if [Manufacturer] property specified

"**oA_StockUnitsID**": Register minimum unit. See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified

"**oA_StockUnitsID_ExciseUnit**": See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [ExciseUnit] property specified

"**oA_StockUnitsID_SaleUnit**": Sale Unit. See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [SaleUnit] property specified

"**priceIncreasePercentage**": Will be used when forming sale price

"**registerParentAccount**": Register parent account. See [AccountsList] method (use [Account] value). Can be omitted if [OA_AccountsID] property specified

"**responsiblePerson**": Responsible Person. See [DebitorCreditorsList] method (use [Code] value). Can be omitted if [OA_DebitorCreditorID] property specified

"**salePrice**":

"**saleUnit**": See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID_SaleUnit] property specified

"**unit**": Register minimum unit. See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified

"**vatType**": Standard = 0, ZeroRate = 1, Untaxed = 2

"**writtenOff**":

"**clearItemUnits**": Clears all related units for the item

"**itemUnits**": Set OA_ItemStockUnitsID=0 to add a new related unit specification. Specify the OA_ItemStockUnitsID value to update information about related unit fields. To delete existing related unit, specify the value OA_ItemStockUnitsID and set "DeleteRow"=true

"**oA_ItemStockUnitsID**": Specify an existing OA_ItemStockUnitsID to update or remove related item information, or 0 to add a new related item

"**deleteRow**": If true, deletes related unit information for specified OA_ItemStockUnitsID value

"**measure**": Quantity of goods in the corresponding unit in relation to the minimum register unit of goods. Example: If the product is "minimum register unit"="piece" and "related unit"="box" and there are 10 items in one box, then specify "Measure"=10

"**relatedOA_StockUnitsID**": See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [RelatedUnit] property specified

"**relatedUnit**": See [StockUnitsList] method (use [Unit] value). Can be omitted if [RelatedOA_StockUnitsID] property specified

"**relatedUnitQuantity**": Quantity of goods in the corresponding unit. Example: If the product is "minimum register unit"="piece" and "related unit"="box" and there are 12 items in 3 box, then specify "Measure"=12 and "RelatedUnitQuantity"=3

"**relatedUnitQuantity**": Quantity of goods in the corresponding unit. Example: If the product is "minimum register unit"="piece" and "related unit"="box" and there are 12 items in 3 box, then specify "Measure"=12 and "RelatedUnitQuantity"=3

Examples

1. Update commodity using "oA_ItemsID" - 3005.
Add Write Off Expense Account – "7210", Item group by "oA_ItemGroupID" – 2 Commodity Group,
sale Price – "100 GEL", sale unit – "Kg"
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ItemsID": 3005,
  "currencies": "GEL",
  "itemWriteOffExpenseAccount": "7210",
  "oA_ItemGroupID": 2,
  "salePrice": 100,
  "saleUnit": "Kg",
  "clearItemUnits": false
}
```

Server response Code(200):

```
{
  "id": 3005,
  "idFieldName": "OA_ItemsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_ItemsID = 30005",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Item",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/Item

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_ItemsID": 0
}
```

Field description

[Token] and [OA_ItemsID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_ItemsID":

Examples

1. Delete item using "oA_ItemsID" – 3005 Commodity

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ItemsID": 3005
}
```

Server response Code(200):

```
{
  "id": 3005,
  "idFieldName": "OA_ItemsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51250,
    "description": "Can't delete item. A link(s) has been found!",
    "comment": "ID [3005] Code [1000001] Name [Commodity]\n\nAccount [1610 3005]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Item",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

GetItemByID

POST/api/GetItemByID

Gets Item (commodity, fixed asset or service) and its units by OA_ItemsID value.

Schema

```
{
  "token": "string",
  "oA_ItemsID": 0
}
```

Field description

[Token] and [OA_ItemsID] is necessary.

"**token**": Set [token] string from [LogIn] response value

"**oA_ItemsID**": See [ItemsList] method (use [OA_ItemsID] value)

Examples

1. Get information about item using ItemsID - 3005

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ItemsID": 3005
}
```

Server response Code(200):

```
{
  "oA_ItemsID": 3005,
  "closeDate": "0001-01-01T00:00:00",
  "code": "1000001",
  "codeContainsWeight": false,
  "comment": "",
  "currencies": "GEL",
  "dateOfExpiration": "0001-01-01T00:00:00",
  "exciseRate": 0,
  "exciseUnit": "",
  "hasDoubleRegisterUnit": false,
  "inventoryNumber": "",
  "isCloseDateNULL": null,
  "isDateOfExpirationNull": null,
  "itemGroup": "Commodity Group",
  "itemType": 0,
  "itemWriteOffExpenseAccount": "7210",
  "manufacturer": "",
  "manufacturerNumber": "",
  "name": "Commodity",
  "oA_AccountsID": 69,
  "oA_AccountsIDExpenseAccount": 180,
  "oA_CurrenciesID": 1,
  "oA_DebitorCreditorID": 0,
  "oA_ItemGroupID": 2,
  "oA_ManufacturersID": 0,
  "oA_StockUnitsID": 1,
  "oA_StockUnitsID_ExciseUnit": 1,
  "oA_StockUnitsID_SaleUnit": 1005,
  "priceIncreasePercentage": 0,
  "registerParentAccount": "1610",
  "responsiblePerson": "",
  "responsiblePersonIdentityNumber": "",
  "salePrice": 100,
  "saleUnit": "Kg",
  "unit": "Piece",
  "vatType": 0,
  "writtenOff": null,
}
```



```

"itemUnits": [
  {
    "oA_ItemStockUnitsID": 2,
    "relatedUnitQuantity": 10,
    "relatedUnit": "Kg",
    "relatedOA_StockUnitsID": 2,
    "measure": 1,
    "registerUnit": "Piece",
    "stockRegisterRate": 0.1
  }
]
}

```

Server response Code(400):

```

[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_ItemsID = 30005",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Item",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

ItemsList

POST/api/ItemsList

Gets Items (commodities, fixed assets and services). Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get a list of items (commodity, fixed asset or service) with registered parent account "1610"

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[Register parent account] IN (N'1610')"
}

```

Server response Code(200):

```

{

```

```

"Info": [
  {
    "RowCount": 1,
    "PageCount": 0,
    "PageRowCount": 0,
    "PageNumber": 0
  }
],
"Result": [
  {
    "OA_ItemsID": 3005,
    "Code": "1000001",
    "Name": "Commodity",
    "Unit": "Piece",
    "OA_StockUnitsID": 1,
    "Item type": 0,
    "OA_AccountsID": 69,
    "Register parent account": "1610",
    "OA_ItemGroupID": 2,
    "Group": "Commodity Group",
    "Sale price": 100.00000000,
    "Written off": false,
    "OA_CurrenciesID": 1,
    "Currency": "GEL",
    "OA_StockUnitsID_SaleUnit": 1005,
    "Sale unit": "Kg",
    "VAT type": 0,
    "Inventory number": "",
    "Comment": "",
    "Register parent account name": "Merchandise inventory",
    "Manufacturer number": "",
    "Excise rate": 0.00000000,
    "Price increase percentage": 0.00000000,
    "OA_AccountsIDExpenseAccount": 180,
    "Item write-off expense account": "7210"
  }
]
}

```

2. Get first 10 items from list and sort by code

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 10
  "sort": "[Code]"
}

```

3. Get the 2nd page from the entire list of items, 10 records per page

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 10,
  "pageNumber": 2
}

```

4. Get all items for further use in the filter

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0
}

```

Service

POST/api/Service

Adds new Service. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "closeDate": "2023-05-29T11:27:56.838Z",
  "code": "string",
  "comment": "string",
  "currencies": "string",
  "expenseAccountForServicePurchase": "string",
  "itemGroup": "string",
  "name": "string",
  "oA_AccountsID": 0,
  "oA_CurrenciesID": 0,
  "oA_ItemGroupID": 0,
  "oA_StockUnitsID": 0,
  "salePrice": 0,
  "unit": "string",
  "vatType": 0
}
```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"closeDate":

"code":

"comment":

"currencies": Sets sale price currency. See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified

"expenseAccountForServicePurchase": Expence account for service purchase. See [AccountsList] method (use [Account] value). Can be omitted if [OA_AccountsID] property specified "itemGroup": See [ItemGroupsList] method (use [Name] value). Can be omitted if [OA_ItemGroupID] property specified

"name": Name of service

"oA_AccountsID":Expence account for service purchase. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [RegisterParentAccount] property specified

"oA_CurrenciesID":Sets sale price currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currencies] property specified

"oA_ItemGroupID": See [ItemGroupsList] method (use [OA_ItemGroupID] value). Can be omitted if [ItemGroup] property specified

"oA_StockUnitsID": Register minimum unit. See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified

"salePrice":

"unit": Register minimum unit. See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified

"vatType": Standard = 0, ZeroRate = 1, Untaxed = 2

Examples

1. Add service name - "Tax Service", code - "5000005", currencie – "GEL", expense account for service purchase – "7425", unit –"Service"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "code": "5000005",
  "currencies": "GEL",
  "expenseAccountForServicePurchase": "7425",
  "name": "Tax Service",
  "unit": "Service"
}
```

Server response Code(200):

```
{
  "id": 4005,
  "idFieldName": "OA_ItemsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51190,
    "description": "Property not found!",
    "comment": "OA_ItemsID=0 Code=5000005\n[ExpenseAccountForServicePurchase]=7425",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Service",
    "entryPoint": "set property",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/Service

Updates Service. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_ItemsID": 0,
  "closeDate": "2023-05-29T12:04:26.517Z",
  "code": "string",
  "comment": "string",
  "currencies": "string",
  "expenseAccountForServicePurchase": "string",
  "itemGroup": "string",
  "name": "string",
  "oA_AccountsID": 0,
  "oA_CurrenciesID": 0,
  "oA_ItemGroupID": 0,
  "oA_StockUnitsID": 0,
  "salePrice": 0,
  "unit": "string",
  "vatType": 0
}
```

Field description

[Token] and [OA_ItemsID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [Login] response value

"oA_ItemsID":

"closeDate":

"code":

"comment":

"currencies": Sets sale price currency. See [CurrenciesList] method (use [Currency] value). Can be omitted if [OA_CurrenciesID] property specified

"expenseAccountForServicePurchase": Expense account for service purchase. See [AccountsList] method (use [Account] value). Can be omitted if [OA_AccountsID] property specified

"itemGroup": See [ItemGroupsList] method (use [Name] value). Can be omitted if [OA_ItemGroupID] property specified

"name": Name of service

"oA_AccountsID": Expense account for service purchase. See [AccountsList] method (use [OA_AccountsID] value). Can be omitted if [RegisterParentAccount] property specified

"oA_CurrenciesID": Sets sale price currency. See [CurrenciesList] method (use [OA_CurrenciesID] value). Can be omitted if [Currencies] property specified

"oA_ItemGroupID": See [ItemGroupsList] method (use [OA_ItemGroupID] value). Can be omitted if [ItemGroup] property specified

"oA_StockUnitsID": Register minimum unit. See [StockUnitsList] method (use [OA_StockUnitsID] value). Can be omitted if [Unit] property specified

"salePrice":

"unit": Register minimum unit. See [StockUnitsList] method (use [Unit] value). Can be omitted if [OA_StockUnitsID] property specified

"vatType": Standard = 0, ZeroRate = 1, Untaxed = 2

Examples

1. Update Tax Sservice information using "oA_ItemsID" – 4005
Add item group by "oA_ItemGroupID"– 3 Service Group
Change vat type to zero rate - 1

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ItemsID": 4005,
  "oA_ItemGroupID": 3,
  "vatType": 1
}
```

Server response Code(200):

```
{
  "id": 4005,
  "idFieldName": "OA_ItemsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51190,
    "description": "Property not found!",
    "comment": "OA_ItemsID=4005 Code=5000005\n[OA_ItemGroupID]=3",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Service",
    "entryPoint": "set property",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/Service

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_ItemsID": 0
}
```

Field description

[Token] and [OA_ItemsID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_ItemsID":

Examples

1. Delete service using "oA_ItemsID" - 4005

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ItemsID": 4005
}
```

Server response Code(200):

```
{
  "id": 4005,
  "idFieldName": "OA_ItemsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51250,
    "description": "Can't delete item. A link(s) has been found!",
    "comment": "ID [4005] Code [5000005] Name [Tax Service]\n\n\nAccept operation ID [12],[3],[5026]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Service",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

ServicesList

POST/api/ServicesList

Gets Services. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get lis of all services for further use in the filter

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 2,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
```

```

{
  "OA_ItemsID": 3008,
  "Code": "3000003",
  "Name": " Service ",
  "OA_StockUnitsID": 3,
  "Item type": 2,
  "Comment": "",
  "Sale price": 0.00000000,
  "Unit": "Service",
  "OA_AccountsID": 192,
  "Expense account for service purchase": "7425",
  "VAT type": 0
},
{
  "OA_ItemsID": 4005,
  "Code": "5000005",
  "Name": "Tax Service",
  "OA_StockUnitsID": 3,
  "Item type": 2,
  "Comment": "",
  "Sale price": 0.00000000,
  "Unit": "Service",
  "OA_AccountsID": 192,
  "Expense account for service purchase": "7425",
  "OA_ItemGroupID": 3,
  "Group": "Service Group",
  "OA_CurrenciesID": 1,
  "Currency": "GEL",
  "VAT type": 1
}
]
}

```

2. Get first 2 servises from database

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2
}

```

3. Get lis of all services and sort by name

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "sort": "[Name]"
}

```

StockUnit

POST/api/StockUnit

Adds stock unit. Returns ID value or Error descriprion Json.

Schema

```

{
  "token": "string",
  "unit": "string",
  "comment": "string"
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"unit":

"comment":

Examples

1. Add new stock unit – “Gallon”
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "unit": "Gallon"
}
```

Server response Code(200):

```
{
  "id": 2004,
  "idFieldName": "OA_StockUnitsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51133,
    "description": "Unit already exists!",
    "comment": "Unit [Gallon]\nOA_StockUnitsID [2004]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "StockUnits",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/StockUnit

Updates stock unit. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_StockUnitsID": 0,
  "unit": "string",
  "comment": "string"
}
```

Field description

[Token] and [OA_StockUnitsID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_StockUnitsID": See [StockUnitsList] method (use [OA_StockUnitsID] value)

"unit":

"comment":

Examples

1. Update stock unit name using "oA_StockUnitsID" - 2004
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_StockUnitsID": 2004,
  "unit": "Gallons"
}
```

Server response Code(200):

```
{
```



```
"id": 2004,  
"idFieldName": "OA_StockUnitsID"  
}
```

Server response Code(400):

```
[  
{  
  "errorNumber": 51051,  
  "description": "Record does not exist or read prohibited for user!",  
  "comment": "OA_StockUnitsID = 2004",  
  "command": "",  
  "errorType": "ErrorType_Error",  
  "source": "OA_BusinessObject.dll",  
  "class": "StockUnits",  
  "entryPoint": "GetProperties",  
  "resourceDLL_ForTranslation": "ORIS_Accounting",  
  "errorPointer": 0  
}  
]
```

DELETE/api/StockUnit

Returns OK or Error description Json

Schema

```
{  
  "token": "string",  
  "oA_StockUnitsID": 0  
}
```

Field description

[Token] and [OA_StockUnitsID] is necessary

"**token**": Set [token] string from [LogIn] response value

"**oA_StockUnitsID**":

Examples

1. Delete stock unit using "OA_StockUnitsID" - 2004
Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "OA_StockUnitsID":2004  
}
```

Server response Code(200):

```
{  
  "id": 2004,  
  "idFieldName": "OA_StockUnitsID"  
}
```

Server response Code(400):

```
[  
{  
  "errorNumber": 51109,  
  "description": "ORIS_Tools internal error!",  
  "comment": "OA_StockUnitsID=2004 [Gallons]\nExist relation to another table.\nRelated table  
[OA_Entries].\n[547] The DELETE statement conflicted with the REFERENCE constraint  
\n'FK_OA_Entries_OA_StockUnits'. The conflict occurred in database '\nDatabaseEN', table '\ndbo.OA_Entries',  
column 'OA_StockUnitsID'. [3621] The statement has been terminated.",  
  "command": "",  
  "errorType": "ErrorType_Error",  
  "source": "OA_BusinessObject.dll",  
  "class": "StockUnits",  
  "entryPoint": "RealDelete",  
}
```

```
"resourceDLL_ForTranslation": "ORIS_Administrate",
"errorPointer": 0
}
]
```

StockUnitsList

POST/api/StockUnitsList

Gets Stock Units. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get first 2 units from list

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 2,
      "PageCount": 0,
      "PageRowCount": 2,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_StockUnitsID": 1,
      "Unit": "Piece",
      "Comment": ""
    },
    {
      "OA_StockUnitsID": 2,
      "Unit": "Box",
      "Comment": ""
    }
  ]
}
```

2. Get all field names for further use in the filter

Request body:

```
{
```

```
"token": "... paste here [LogIn] response value ...",
"topRecordsCount": 0
}
```

3. Get a list of units sort by name
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "sort": "[Unit]"
}
```

4. Get one unit Gallons using filter
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[Unit] IN (N'Gallons)"
}
```

Currency

POST/api/Currency

Adds currency. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "currency_": "string",
  "name": "string",
  "imageString": "string"
}
```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"currency_":

"name":

"imageString":

Examples

1. Add new currency Canadian dollar – “CAD”
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "currency_": "CAD",
  "name": "Canadian dollar"
}
```

Server response Code(200):

```
{
  "id": 6,
  "idFieldName": "OA_CurrenciesID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51129,
    "description": "Currency already exists!",
    "comment": "Currency [CAD]\nName [Canadian Dollar]\nOA_CurrenciesID [6]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
  }
]
```

```

    "class": "Currency",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

PUT/api/Currency

Updates currency. Returns OK or Error description Json.

Schema

```

{
  "token": "string",
  "oA_CurrenciesID": 0,
  "currency_": "string",
  "name": "string",
  "imageString": "string"
}

```

Field description

[Token] and [OA_CurrenciesID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_CurrenciesID": See [CurrenciesList] method (use [OA_CurrenciesID] value)

"currency_":

"name":

"imageString":

Examples

1. Update currency name using "oA_CurrenciesID" - 6

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_CurrenciesID": 6,
  "name": "Canadian Dollar"
}

```

Server response Code(200):

```

{
  "id": 6,
  "idFieldName": "OA_CurrenciesID"
}

```

Server response Code(400):

```

[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_CurrenciesID = 6",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Currency",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

DELETE/api/Currency

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_CurrenciesID": 0
}
```

Field description

[Token] and [OA_CurrenciesID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_CurrenciesID":

Examples

1. Delete currency using "oA_CurrenciesID" - 6
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_CurrenciesID": 6
}
```

Server response Code(200):

```
{
  "id": 6,
  "idFieldName": "OA_CurrenciesID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "OA_CurrenciesID=6 [CAD]\nExist relation to another table.\n\nRelated table [OA_Entries].\n\n\n[547]
The DELETE statement conflicted with the REFERENCE constraint \"FK_OA_Entries_OA_Currencies\". The conflict
occurred in database \"NDatabaseEN\", table \"dbo.OA_Entries\", column 'OA_CurrenciesID'.[3621] The statement
has been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Currency",
    "entryPoint": "RealDelete",
    "resourceDLL_ForTranslation": "ORIS_Administrate",
    "errorPointer": 0
  }
]
```

CurrenciesList

POST/api/CurrenciesList

Gets Currencies. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get first currency from list

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 0,
      "PageRowCount": 1,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_CurrenciesID": 1,
      "Currency": "GEL",
      "Name": "Georgian Lari",
      "Image":
        "R0IGODIhHwAQAlcAAAAAAAAAMwAAZgAAmQAAzAAA/wArAAArMwArZgArmQArzAAr/wBVAABVMwBVZgBVmQ
        BVzABV/wCAAACAMwCAZgCAmQCAzACA/wCqAACqMwCqZgCqmqQCqzACq/wDVAADVMwDVZgDVMQDVzADV/
        wD/AAD/MwD/ZgD/mQD/zAD//zMAADMAMzMAZjMAMtMAzDMA/zMrADMrMzMrZjMrmTMrzDMr/zNVADNVmzNVZ
        jNVmTNVzDNV/zOAAADOAMzOAZjOAmTOAzDOA/zOqADOqMzOqZjOqmTOqzDOq/zPVADPVMzPVZjPVmTPVzD
        PV/zP/ADP/MzP/ZjP/mTP/zDP//2YAAGYAM2YAZmYAmWYAzGYA/2YrAGYrM2YrZmYrmWYrZGYr/2ZVAGZVM2ZV
        ZmZVmWZVzGV/2aAAGaAM2aAZmaAmWaAzGaA/2aqAGaqM2aqZmaqmWaqzGaq/2bVAGbVM2bVZmbVmWbV
        zGbV/2b/AGb/M2b/Zmb/mWb/zGb//5kAAJkAM5kAZpkAmZkAzJkA/5krAJkrM5krZpkrmZkrzJkr/5IVAjIVM5IVZpIVmZI
        VzJIV/5mAAJmAM5mAZpmAmZmAzJmA/5mqAJmqM5mqZpmqmZmqzJmq/5nVAJnVM5nVZpnVmZnVzJnV/5n/AJn/
        M5n/Zpn/mZn/zJn//8wAAMwAM8wAZswAmcwAzMwA/8wrAMwrM8wrZswrmcwrzMwr/8xVAMxVM8xVZsxVmcxVzMx
        V/8yAAMyAM8yAZsyAmcyAzMyA/8yqAMyqM8yqZsyqmqyqzMyq/8zVAMzVM8zVZszVmczVzMzV/8z/AMz/M8z/Zsz/
        mcz/zMz//8AAP8AM/8AZv8Amf8AzP8A//8rAP8rM/8rZv8rmf8rzP8r//9VAP9VM/9VZv9Vmf9VzP9V//+AAP+AM/+AZv+
        Amf+AzP+A//+qAP+qM/+qZv+qmf+qzP+q//VAP/VM//VzV/Vmf/VzP/V////AP//M//Zv//mf//zP//wAAAAAAAAAAAAAAAA
        CH/C05FVFNDQVBFMi4wAwEBAAAh+QQBAAD8ACwAAAAAAAAHwAQAAAIkgD3CSSmTBnBggcTFjSIsKEygRafQpw
        oUJpFaRQpEtM40WLHix8nSowl8SLGiiBRhQ4cmDikiRTtwoMuPHITb3tdRpsqfPnxZpQiQGtOjPncqMKr2lNKfKkzm
        FuoQZTWA0k/uo7kMFVsfFkVifwsS5UyrOfwFIDv3qFK00rm3LtgWoNePOZAwVKszLkOEwiQEBADs="
    }
  ]
}
```

2. Get all field names for further use in the filter

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0
}
```

3. Get all currencies and sort by name

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "sort": "[Currency]"
}
```

CurrencyRate

POST/api/CurrencyRate

Adds currency rate. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "oA_RatesID": 0,
  "oA_CurrenciesID": 0,
  "numberOfUnits": 0,
  "numberOfUnits2": 0,
  "rate": 0,
  "rate2": 0,
  "rateBuy": 0,
  "rateSale": 0,
  "rateDate": "2023-05-30T10:43:47.004Z"
}
```

Field description

[Token] and [OA_RatesID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_RatesID": See [CurrencyRatesList] method (use [OA_RatesID] value)

"oA_CurrenciesID": See [CurrenciesList] method (use [OA_CurrenciesID] value)

"numberOfUnits":

"numberOfUnits2":

"rate":

"rate2":

"rateBuy":

"rateSale":

"rateDate":

Examples

1. Add currency rate "1 GEL=1.9 CAD" and set on date 31.12.2023, using "oA_CurrenciesID" - 6
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_RatesID": 1,
  "oA_CurrenciesID": 6,
  "numberOfUnits": 1,
  "rate": 1.9,
  "rateDate": "2023-12-31"
}
```

Server response Code(200):

```
{
  "id": 5175,
  "idFieldName": "OA_RatesID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "OA_RatesID=5176 [CAD]\nA duplicate value cannot be inserted.\r\n[Date] = [31.12.2023 00:00:00]\r\n[OA_CurrenciesID] = [6]\r\n\r\n[2601] Cannot insert duplicate key row in object 'dbo.OA_Rates' with unique index 'IndxOA_Rates1'. The duplicate key value is (Dec 31 2023 12:00AM, 6).[3621] The statement has been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
  }
]
```

```

"source": "OA_BusinessObject.dll",
"class": "CurrencyRates",
"entryPoint": "Update",
"resourceDLL_ForTranslation": "ORIS_Administrate",
"errorPointer": 0
}
]

```

2. Add currency rate "1 GEL=2.0 CAD" and set on current date, using oA_CurrenciesID – 6, specify Buy rate – 1 and Sale rate - 2
Request body:

```

{
"token": "... paste here [LogIn] response value ...",
"oA_RatesID": 1,
"oA_CurrenciesID": 6,
"numberOfUnits": 1,
"rate": 2,
"rateBuy": 1,
"rateSale": 2
}

```

PUT/api/CurrencyRate

Updates currency rate. Returns OK or Error description Json.

Schema

```

{
"token": "string",
"oA_RatesID": 0,
"oA_CurrenciesID": 0,
"numberOfUnits": 0,
"numberOfUnits2": 0,
"rate": 0,
"rate2": 0,
"rateBuy": 0,
"rateSale": 0,
"rateDate": "2023-05-30T11:11:24.702Z"
}

```

Field description

[Token] and [OA_RatesID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_RatesID": See [CurrencyRatesList] method (use [OA_RatesID] value)

"oA_CurrenciesID": See [CurrenciesList] method (use [OA_CurrenciesID] value)

"numberOfUnits":

"numberOfUnits2":

"rate":

"rate2":

"rateBuy":

"rateSale":

"rateDate":

Examples

1. Update currency rate, add second rate "1USD=1.36 CAD" using "oA_CurrenciesID" – 6 and "oA_RatesID" - 5175

```

Request body:
{
"token": "... paste here [LogIn] response value ...",
"oA_RatesID": 5175,
"oA_CurrenciesID": 6,
"numberOfUnits2": 1,
"rate2": 1.36,
}

```



```
"rateDate": "2023-12-31"
}
```

Server response Code(200):

```
{
  "id": 5175,
  "idFieldName": "OA_RatesID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_RatesID = 5175",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "CurrencyRates",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/CurrencyRate

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_RatesID": 0
}
```

Field description

[Token] and [OA_RatesID] is necessary.

"token": Set [token] string from [LogIn] response value
"oA_RatesID":

Examples

1. Delete currency rate using RatesID - 5175
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_RatesID": 5175
}
```

Server response Code(200):

```
{
  "id": 5175,
  "idFieldName": "OA_RatesID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_RatesID = 5175",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
  }
]
```

```

    "class": "CurrencyRates",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

CurrencyRatesList

POST/api/CurrencyRatesList

Gets Currency rates. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "diapazonStartDate": "2023-05-30T12:41:49.741Z",
  "diapazonEndDate": "2023-05-30T12:41:49.741Z",
  "currenciesIDs": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string"
}

```

Field description

"**token**": Set [token] string from [LogIn] response value

"**diapazonStartDate**": Diapazon start date

"**diapazonEndDate**": Diapazon end date (to). Example: Period of 2010 year specify - "DiapazonStartDate":"2010-01-01",

"**DiapazonEndDate**":"2011-01-01"

"**currenciesIDs**":

"**topRecordsCount**": The number of records to accept. If you specify 0 you will get all records. Default=0

"**filter**":

"**sort**":

Examples

1. Get first 2 accounts from database

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2
}

```

Server response Code(200):

```

{
  "Info": [
    {
      "RowCount": 2,
      "PageCount": 0,
      "PageRowCount": 2,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_CurrenciesID": 1,
      "Currency": "GEL",
      "Name": "Georgian Lari",
      "Image":
"ROIgODIhHwAQAlcAAAAAAAAAMwAAZgAAmQAAzAAA/wArAAArMwArZgArmQArzAAr/wBVAABVMwBVZgBVmQ
BVzABV/wCAAACAMwCAZgCAmQCAzACA/wCqAACqMwCqZgCqmQCqzACq/wDVAADVMwDVZgDVmQDVzADV/
wD/AAD/MwD/ZgD/mQD/zAD//zMAADMAMzMAZjMAMTMAMzDMA/zMrADMrmzMrZjMrmTMrzDMr/zNVADNVmNVZ
jNVmTNVzDNV/zOAAADOAMzOAZjOAMTOAzDOA/zOqADOqMzOqZjOqmTOqzDOq/zPVADPVMzPVZjPVmTPVzD
PV/zP/ADP/MzP/ZjP/mTP/zDP//2YAAGYAM2YAZmYAmWYAzGYA/2YrAGYrM2YrZmYrmWYrZGYr/2ZVAGZVM2ZV

```



```
{
  "token": "string",
  "address": "string",
  "cityName": "string",
  "costCentreComment": "string",
  "costCentreGroup": "string",
  "costCentreGroupID": 0,
  "costCentreName": "string",
  "costCentreParentID": 0,
  "oA_CityID": 0,
  "oA_SalePriceTypesID": 0,
  "parentCostCentre": "string",
  "salePriceType": "string",
  "zip": "string"
}
```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"address":

"cityName": See [CitiesList] method (use [Name] value). Can be omitted if [OA_CityID] property specified

"costCentreComment":

"costCentreGroup": See [CostCentreGroupsList] method (use [Name] value). Can be omitted if [OA_CostCentreGroupID] property specified

"costCentreGroupID": See [CostCentreGroupsList] method (use [OA_CostCentreGroupID] value). Can be omitted if [CostCentreGroup] property specified

"costCentreName":

"costCentreParentID": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [ParentCostCentre] property specified

"oA_CityID": See [CitiesList] method (use [OA_CityID] value). Can be omitted if [CityName] property specified

"oA_SalePriceTypesID": See [SalePriceTypesList] method (use [OA_SalePriceTypesID] value)

"parentCostCentre": See [CostCentresList] method (use [Name] value). Can be omitted if [CostCentreParentID] property specified

"salePriceType": See [SalePriceTypesList] method (use [Description] value). Can be omitted if [OA_SalePriceTypesID] property specified

"zip":

Examples

1. Add new cost centre 'Service cost center'

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "costCentreName": " Service cost center "
}
```

Server response Code(200):

```
{
  "id": 2003,
  "idFieldName": "OA_CostCentreID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51130,
    "description": "Cost centre already exists!",
    "comment": "Name [Service cost center]\nOA_CostCentreID [2003]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "CostCentre",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

]

2. Add new cost centre 'Store' under the parent 'Service cost center' using "costCentreParentID" – 2003
Service cost center
Address - Al. Kazbegi Ave. 12-b, Tbilisi, Georgia, 0160
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "address": "Al. Kazbegi Ave. 12-b",
  "cityName": "Tbilisi",
  "costCentreName": "Store",
  "costCentreParentID": 2003,
  "zip": "0160"
}
```

PUT/api/CostCentre

Updates cost centre. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_CostCentreID": 0,
  "address": "string",
  "cityName": "string",
  "costCentreComment": "string",
  "costCentreGroup": "string",
  "costCentreGroupID": 0,
  "costCentreName": "string",
  "costCentreParentID": 0,
  "oA_CityID": 0,
  "oA_SalePriceTypesID": 0,
  "parentCostCentre": "string",
  "salePriceType": "string",
  "zip": "string"
}
```

Field description

[Token] and [OA_CostCentreID] is necessary. Specify only those fields that you want to change.

"**token**": Set [token] string from [LogIn] response value

"**oA_CostCentreID**": Set [token] string from [LogIn] response value

"**address**":

"**cityName**": See [StockUnitsList] method (use [OA_CostCentreID] value)

"**costCentreComment**": "string",

"**costCentreGroup**": See [CostCentreGroupsList] method (use [Name] value). Can be omitted if [OA_CostCentreGroupID] property specified

"**costCentreGroupID**": See [CostCentreGroupsList] method (use [OA_CostCentreGroupID] value). Can be omitted if [CostCentreGroup] property specified

"**costCentreName**":

"**costCentreParentID**": See [CostCentresList] method (use [OA_CostCentreID] value). Can be omitted if [ParentCostCentre] property specified

"**oA_CityID**": See [CitiesList] method (use [OA_CityID] value). Can be omitted if [CityName] property specified

"**oA_SalePriceTypesID**": See [SalePriceTypesList] method (use [OA_SalePriceTypesID] value)

"**parentCostCentre**": See [CostCentresList] method (use [Name] value). Can be omitted if [CostCentreParentID] property specified

"**salePriceType**": See [SalePriceTypesList] method (use [Description] value). Can be omitted if [OA_SalePriceTypesID] property specified

"**zip**":

Examples

1. Update cost centre Store using "oA_CostCentreID" - 2004 "Store", change name to 'Store N1', add cost centre group – Service, using costCentreGroupID – 1

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_CostCentreID": 2004,
  "costCentreGroupID": 1,
  "costCentreName": "Store N1"
}
```

Server response Code(200):

```
{
  "id": 2004,
  "idFieldName": "OA_CostCentreID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "OA_CostCentreID=2004 [Store N1]\nExist relation to another table.\n\nRelated table [OA_CostCentre].\n\n[547] The UPDATE statement conflicted with the FOREIGN KEY constraint \"FK_OA_CostCentre_OA_CostCentreGroup\". The conflict occurred in database \"NDatabaseEN\", table \"dbo.OA_CostCentreGroup\", column 'OA_CostCentreGroupID'.[3621] The statement has been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "CostCentre",
    "entryPoint": "Update",
    "resourceDLL_ForTranslation": "ORIS_Administrate",
    "errorPointer": 0
  }
]
```

DELETE/api/CostCentre

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_CostCentreID": 0
}
```

Field description

[Token] and [OA_CostCentreID] is necessary.

```
"token": "... paste here [LogIn] response value ...",
"oA_CostCentreID":
```

Examples

1. Delete cost centre 'Service cost centre' using "oA_CostCentreID" - 2003

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_CostCentreID": 2003
}
```

Server response Code(200):

```
{
  "id": 2003,
  "idFieldName": "OA_CostCentreID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51015,
    "description": "Can't delete record!",
    "comment": "[Service cost center] Cost Centre has sub CostCentre(s).\nFirst delete sub CostCentre(s).",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "CostCentre",
    "entryPoint": "CheckSubAccountExistance",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

CostCentresList

POST/api/CostCentresList

Gets Cost Centres. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get list of all cost centre by level 1

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[Level]=1"
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_CostCentreID": 2004,
      "Name": "Store N1",
      "Comment": "",
      "ParentID": 2003,
      "Level": 1,

```

```

    "Leaf": true,
    "Parent Cost Centre": "Service cost center",
    "OA_CostCentreGroupID": 1,
    "Cost centre group": "Service",
    "Address": "Al. Kazbegi Ave. 12-b",
    "OA_CityID": 1,
    "City": "Tbilisi",
    "Zip": "0160"
  }
}
}

```

2. Get list of all cost centre names for further use in the filter

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0
}

```

CostUnit

POST/api/CostUnit

Adds cost unit. Returns ID value or Error description Json.

Schema

```

{
  "token": "string",
  "comment": "string",
  "name": "string",
  "parentID": 0
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"comment":

"name":

"parentID": See [CostUnitsList] method (use [OA_CostUnitID] value)

Examples

1. Add "New Cost Unit"

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "name": " New Cost Unit "
}

```

Server response Code(200):

```

{
  "id": 1003,
  "idFieldName": "OA_CostUnitID"
}

```

Server response Code(400):

```

[
  {
    "errorNumber": 51132,
    "description": "Cost unit already exists!",
    "comment": "Name [ New Cost Unit ]\nOA_CostUnitID [1003]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
  }
]

```



```

    "class": "CostUnit",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]

```

PUT/api/CostUnit

Updates cost unit. Returns OK or Error description Json.

Schema

```

{
  "token": "string",
  "oA_CostUnitID": 0,
  "comment": "string",
  "name": "string",
  "parentID": 0
}

```

Field description

[Token] and [OA_CostUnitID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_CostUnitID": See [CostUnitsList] method (use [OA_CostUnitID] value)

"comment":

"name":

"parentID": See [CostUnitsList] method (use [OA_CostUnitID] value)

Examples

1. Update cost unit "oA_CostUnitID" – 1003, add parent "Total cost unit" using "oA_CostUnitID" - 1
Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_CostUnitID": 1003,
  "parentID": 1
}

```

Server response Code(200):

```

{
  "id": 1003,
  "idFieldName": "OA_CostUnitID"
}

```

DELETE/api/CostUnit

Returns OK or Error description Json.

Schema

```

{
  "token": "string",
  "oA_CostUnitID": 0
}

```

Field description

[Token] and [OA_CostUnitID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_CostUnitID":

Examples

1. Delete cost unit using oA_CostUnitID – 1003 “New Cost Unit”

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_CostUnitID": 1003
}
```

Server response Code(200):

```
{
  "id": 1003,
  "idFieldName": "OA_CostUnitID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "Exist relation to another table.\r\nRelated table [OA_Entries].\r\n\r\n[547] The DELETE statement conflicted with the REFERENCE constraint \"FK_OA_Entries_OA_CostUnit\". The conflict occurred in database \"NDatabaseEN\", table \"dbo.OA_Entries\", column 'OA_CostUnitID'.[3621] The statement has been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "CostUnit",
    "entryPoint": "RealDelete",
    "resourceDLL_ForTranslation": "ORIS_Administrate",
    "errorPointer": 0
  }
]
```

CostUnitsList

POST/api/CostUnitsList

Gets Cost Units. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get first cost unit from list

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
```

```

    "RowCount": 1,
    "PageCount": 0,
    "PageRowCount": 1,
    "PageNumber": 0
  }
],
"Result": [
  {
    "OA_CostUnitID": 1,
    "Name": "Total cost unit",
    "ParentID": 0,
    "Level": 0,
    "Leaf": false,
    "Comment": "",
    "Parent Cost Unit": null
  }
]
}

```

2. Get list of all cpst units for further use in the filter

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1
}

```

3. Getlist of cost units by level – 1

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[Level]=1"
}

```

Comment

POST/api/Comment

Adds comment. Returns ID value or Error description Json.

Schema

```

{
  "token": "string",
  "commentText": "string"
}

```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"commentText":

Examples

1. Add coment 'Service expense'

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "commentText": "Service expense"
}

```

Server response Code(200):

```

{
  "id": 2,
  "idFieldName": "OA_CommentsID"
}

```

PUT/api/Comment

Updates comment. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_CommentsID": 0,
  "commentText": "string"
}
```

Field description

[Token] and [OA_CommentsID] is necessary. Specify only those fields that you want to change.

"**token**": Set [token] string from [LogIn] response value

"**oA_CommentsID**": See [CommentsList] method (use [OA_CommentsID] value)

"**commentText**":

Examples

1. Update comment using OA_CommentsID – 2 to "Service expenses"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_CommentsID": 2,
  "commentText": "Service expenses"
}
```

Server response Code(200):

```
{
  "id": 2,
  "idFieldName": "OA_CommentsID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_CommentsID = 2",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Comments",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/Comment

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_CommentsID": 0
}
```

Field description

[Token] and [OA_CommentsID] is necessary.

"token": Set [token] string from [LogIn] response value
"oA_CommentsID":

Examples

1. Delete comment "Service expenses" using oA_CommentID - 2
Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "oA_CommentsID": 2  
}
```

Server response Code(200):

```
{  
  "id": 2,  
  "idFieldName": "OA_CommentsID"  
}
```

Server response Code(400):

```
[  
  {  
    "errorNumber": 51051,  
    "description": "Record does not exist or read prohibited for user!",  
    "comment": "OA_CommentsID = 2",  
    "command": "",  
    "errorType": "ErrorType_Error",  
    "source": "OA_BusinessObject.dll",  
    "class": "Comments",  
    "entryPoint": "GetProperties",  
    "resourceDLL_ForTranslation": "ORIS_Accounting",  
    "errorPointer": 0  
  }  
]
```

CommentsList

POST/api/CommentsList

Gets Comments. Returns [Info] and [Result] tables.

Schema

```
{  
  "token": "string",  
  "topRecordsCount": 0,  
  "filter": "string",  
  "sort": "string",  
  "pageNumber": 0  
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get list of all comments for further use in the filter
Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
}
```

```
"topRecordsCount": 0
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_CommentsID": 2,
      "Comment": "Service expenses",
      "OA_UsersID": 1
    }
  ]
}
```

2. Get first 2 accounts from database

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2
}
```

3. Get the 2nd page from the entire list of accounts, 15 records per page and sort by Comment

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 15,
  "sort": "[Comment]",
  "pageNumber": 2
}
```

TransactionType

POST/api/TransactionType

Adds transaction type. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "comment": "string",
  "containsManagerialInformation": true,
  "guid": "string",
  "name": "string"
}
```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"comment":

"containsManagerialInformation":

"guid":

"name":

Examples

1. Add new transaction type "Credit Transactions"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "name": "Credit Transactions"
}
```

Server response Code(200):

```
{
  "id": 51,
  "idFieldName": "OA_TransactionTypesID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51260,
    "description": "Transaction type name already exists!",
    "comment": "Name [Credit Transactions]nOA_TransactionTypesID [51]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "TransactionTypes",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/TransactionType

Updates transaction type. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionTypesID": 0,
  "comment": "string",
  "containsManagerialInformation": true,
  "guid": "string",
  "name": "string"
}
```

Field description

[Token] and [OA_TransactionTypesID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_TransactionTypesID": See [TransactionTypesList] method (use [OA_TransactionTypesID] value)

"comment":

"containsManagerialInformation":

"guid":

"name":

Examples

1. Update transaction type "Credit Transactions" using oA_TransactionTypesID - 51

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionTypesID": 51,
  "containsManagerialInformation": true
}
```

Server response Code(200):

```
}
```

```
"id": 51,
"idFieldName": "OA_TransactionTypesID"
}
```

Server response Code(400):

```
[
{
  "errorNumber": 51051,
  "description": "Record does not exist or read prohibited for user!",
  "comment": "OA_TransactionTypesID = 501",
  "command": "",
  "errorType": "ErrorType_Error",
  "source": "OA_BusinessObject.dll",
  "class": "TransactionTypes",
  "entryPoint": "GetProperties",
  "resourceDLL_ForTranslation": "ORIS_Accounting",
  "errorPointer": 0
}
]
```

TransactionTypesList

POST/api/TransactionTypesList

Gets Transaction Types. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get second transaction types from list

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "pageNumber": 2
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 52,
      "PageCount": 52,
      "PageRowCount": 1,
      "PageNumber": 2
    }
  ],
  "Result": [
```



```
{
  "OA_TransactionTypesID": 2,
  "Name": "Bank transfer (foreign currency)",
  "GUID": "229D16CA-96C9-4AB2-8BE9-C61FA8B724E6",
  "Comment": "",
  "Image":
  "ROIGODIhEAAQAIIEAACAAIliP//qszMzP///wAAAAAAAAAAAAACH/C05FVFNDQVBFMi4wAwEBAAAh+QQAAAAAAC
  wAAAAAEAAQAAAIxgAJCBxIcCCAggYLAjil8CBDAgwfEghAMQAAihARThTAEQDHjhI3evT4sSPBAAJGfiQpEaXKI
  SFdpwZcuPMkgICFERZsiLGkzgDDBigc2dQohp5fvyJ0GdFjVA1BgQAOW==",
  "Contains manager information": true,
  "NUM_ROW": 2
}
]
```

2. Get all field names for further use in the filter

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0
}
```

3. Get list of transaction types which is marked Contains manager information

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0,
  "filter": "[Contains manager information]<>0"
}
```

TransactionLink

PUT/api/TransactionLink

Adds, Updates or Deletes transaction link. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
  "relatedTransactions": [
    {
      "oA_TransactionsID": 0
    }
  ]
}
```

Field description

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

"relatedTransactions":

"oA_TransactionsID":

Examples

1. Add transaction link "oA_TransactionsID" - 32 : Revenue income using debtor/creditor account "1410 1", income account for supply "6110 1" and payable VAT account "3330" and payment (cash income) using cash account "1110 2" and debtor/creditor account "1410 1"
To "oA_TransactionsID" – 5050 : Supply the googs using debtor/creditor account "1410 1" and item account "1610 1"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 32,
```

```
"relatedTransactions": [
  {
    "oA_TransactionsID": 5050
  }
]
```

Server response Code(200):

```
{
  "id": 32,
  "idFieldName": "OA_TransactionLinksID"
}
```

2. Update transaction link for "oA_TransactionsID – 32" , Change related transaction To "oA_TransactionsID" – 5057

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 32,
  "relatedTransactions": [
    {
      "oA_TransactionsID": 5057
    }
  ]
}
```

3. Update transaction link for "oA_TransactionsID – 32" , Add two transaction links "oA_TransactionsID" – 5057 and "oA_TransactionsID" – 5050

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 32,
  "relatedTransactions": [
    {
      "oA_TransactionsID": 5057
    },
    {
      "oA_TransactionsID": 5050
    }
  ]
}
```

4. Delete related transaction "oA_TransactionsID" – 5050, using "oA_TransactionsID" - 32 and related transaction "oA_TransactionsID" – 5057

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_TransactionsID": 32,
  "relatedTransactions": [
    {
      "oA_TransactionsID": 5057
    }
  ]
}
```

GetTransactionLinkByID

POST/api/GetTransactionLinkByID

Gets Transaction Links by OA_TransactionsID value.

Schema

```
{
  "token": "string",
  "oA_TransactionsID": 0,
}
```

```
"relatedTransactions": [  
  {  
    "oA_TransactionsID": 0  
  }  
]
```

Field description

"token": Set [token] string from [LogIn] response value

"oA_TransactionsID":

"relatedTransactions":

"oA_TransactionsID":

Examples

1. Get list of related transactions for oA_TransactionsID - 32

Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "oA_TransactionsID": 32  
}
```

Server response Code(200):

```
{  
  "oA_TransactionsID": 32,  
  "relatedTransactions": [  
    {  
      "oA_TransactionsID": 5057  
    }  
  ]  
}
```

Manufacturer

POST/api/Manufacturer

Adds manufacturer. Returns ID value or Error description Json.

Schema

```
{  
  "token": "string",  
  "name": "string"  
}
```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"name":

Examples

1. Add manufacturer "Oris"

Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "name": "Oris"  
}
```

Server response Code(200):

```
{  
  "id": 1,  
  "idFieldName": "OA_ManufacturersID"  
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "OA_ManufacturersID=0 [Oris]\nA duplicate value cannot be inserted.\r\n[Name]\r\n\r\n[2601] Cannot insert duplicate key row in object 'dbo.OA_Manufacturers' with unique index 'IdxOA_Manufacturers1'. The duplicate key value is (Oris).[3621] The statement has been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Manufacturers",
    "entryPoint": "Update",
    "resourceDLL_ForTranslation": "ORIS_Administrate",
    "errorPointer": 0
  }
]
```

PUT/api/Manufacturer

Updates manufacturer. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_ManufacturersID": 0,
  "name": "string"
}
```

Field description

[Token] and [OA_ManufacturersID] is necessary. Specify only those fields that you want to change.

"**token**": Set [token] string from [LogIn] response value

"**oA_ManufacturersID**": See [ManufacturersList] method (use [OA_ManufacturersID] value)

"**name**":

Examples

1. Update information about manufacturer using oA_ManufacturersID – 1, change name to "LTD Oris"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ManufacturersID": 1,
  "name": "LTD Oris"
}
```

Server response Code(200):

```
{
  "id": 1,
  "idFieldName": "OA_ManufacturersID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_ManufacturersID = 1",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Manufacturers",
    "entryPoint": "GetProperties",
  }
]
```

```
"resourceDLL_ForTranslation": "ORIS_Accounting",
"errorPointer": 0
}
]
```

DELETE/api/Manufacturer

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_ManufacturersID": 0
}
```

Field description

[Token] and [OA_ManufacturersID] is necessary.

"**token**": Set [token] string from [LogIn] response value

"**oA_ManufacturersID**":

Examples

1. Delete manufacturer using oA_ManufacturersID - 1
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_ManufacturersID": 1
}
```

Server response Code(200):

```
{
  "id": 1,
  "idFieldName": "OA_ManufacturersID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_ManufacturersID = 1",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Manufacturers",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

ManufacturersList

POST/api/ManufacturersList

Gets Manufacturers. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
}
```

```
"sort": "string",
"pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get first name of manufacturer from list

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 0,
      "PageRowCount": 1,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_ManufacturersID": 1,
      "Name": "LTD Oris"
    }
  ]
}
```

2. Get the 2nd from the entire list of accounts

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "pageNumber": 2
}
```

3. Get all field names for further use in the filter

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0
}
```

4. Get one manufacturer using filter

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[Name] IN (N'LTD Oris)"
}
```

City

POST/api/City

Adds city. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "name": "string",
  "oA_CountryID": 0,
  "countryName": "string"
}
```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"name":

"oA_CountryID": See [CostUnitsList] method (use [OA_CountryID] value). Can be omitted if [CountryName] property specified

"countryName": See [CostUnitsList] method (use [Name] value). Can be omitted if [OA_CountryID] property specified

Examples

1. Add city name – "Tbilisi"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "name": "Tbilisi"
}
```

Server response Code(200):

```
{
  "id": 63,
  "idFieldName": "OA_CityID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51243,
    "description": "City already exists!",
    "comment": "Name [Tbilisi1]\nOA_CityID [63]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "City",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/City

Updates city. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_CityID": 0,
  "name": "string",
  "oA_CountryID": 0,
  "countryName": "string"
}
```

Field description

[Token] and [OA_CityID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_CityID": See [CitiesList] method (use [OA_CityID] value)

"name":

"oA_CountryID": See [CostUnitsList] method (use [OA_CountryID] value). Can be omitted if [CountryName] property specified

"countryName": See [CostUnitsList] method (use [Name] value). Can be omitted if [OA_CountryID] property specified

Examples

1. Update city by oA_CityID – 63 "Tbilisi",. Add country using "oA_CountryID – 82 "Georgia"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_CityID": 63,
  "oA_CountryID": 82
}
```

Server response Code(200):

```
{
  "id": 63,
  "idFieldName": "OA_CityID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_CityID = 63",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "City",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/City

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_CityID": 0
}
```

Field description

[Token] and [OA_CityID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_CityID":

Examples

1. Delete city "Tbilisi" using oA_CityID- 63

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_CityID": 63
}
```



```
}
```

Server response Code(200):

```
{  
  "id": 63,  
  "idFieldName": "OA_CityID"  
}
```

Server response Code(400):

```
[  
  {  
    "errorNumber": 51109,  
    "description": "ORIS_Tools internal error!",  
    "comment": "OA_CityID=63 [Tbilisi]\nExist relation to another table.\nRelated table  
[OA_DebitorCreditor].\n\n[547] The DELETE statement conflicted with the REFERENCE constraint  
\n\"FK_OA_DebitorCreditor_OA_City\". The conflict occurred in database \"NDatabaseEN\", table  
\n\"dbo.OA_DebitorCreditor\", column 'OA_CityID'.[3621] The statement has been terminated.",  
    "command": "",  
    "errorType": "ErrorType_Error",  
    "source": "OA_BusinessObject.dll",  
    "class": "City",  
    "entryPoint": "RealDelete",  
    "resourceDLL_ForTranslation": "ORIS_Administrate",  
    "errorPointer": 0  
  }  
]
```

CitiesList

POST/api/CitiesList

Gets Cities. Returns [Info] and [Result] tables.

Schema

```
{  
  "token": "string",  
  "topRecordsCount": 0,  
  "filter": "string",  
  "sort": "string",  
  "pageNumber": 0  
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get first 2 city names from list

Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "topRecordsCount": 2  
}
```

Server response Code(200):

```
{  
  "Info": [  
    {  
      "RowCount": 2,  

```

```

    "PageCount": 0,
    "PageRowCount": 2,
    "PageNumber": 0
  }
],
"Result": [
  {
    "OA_CityID": 1,
    "Name": "Tbilisi",
    "OA_CountryID": 82,
    "Country": "Georgia"
  },
  {
    "OA_CityID": 2,
    "Name": "Batumi",
    "OA_CountryID": 82,
    "Country": "Georgia"
  }
]
}

```

2. Get the 2nd page from the entire list of cities, 15 records per page

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 15,
  "pageNumber": 2
}

```

3. Get all field names for further use in the filter

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0
}

```

4. Get a list of cities whose country is Georgia and sort by name

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[Country] IN (N'Georgia')",
  "sort": "[Name]"
}

```

Country

POST/api/Country

Adds country. Returns ID value or Error description Json.

Schema

```

{
  "token": "string",
  "countryCode": "string",
  "name": "string",
  "numericCode": "string"
}

```

Field description

[Token] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"countryCode":

"name":

"numericCode":

Examples

1. Add new country "Georgia", specify code GE

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "countryCode": "GE",
  "name": "Georgia"
}
```

Server response Code(200):

```
{
  "id": 250,
  "idFieldName": "OA_CountryID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51133,
    "description": "Unit already exists!",
    "comment": "Name [Georgia]\nOA_CountryID [250]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Country",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/Country

Updates country. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_CountryID": 0,
  "countryCode": "string",
  "name": "string",
  "numericCode": "string"
}
```

Field description

[Token] and [OA_CountryID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_CountryID": See [CountriesList] method (use [OA_CountryID] value)

"countryCode":

"name":

"numericCode":

Examples

1. Update country "Georgia" using "oA_CountryID" – 250, add numeric code "268"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_CountryID": 250,
  "numericCode": "268"
}
```

Server response Code(200):

```
{
  "id": 250,
  "idFieldName": "OA_CountryID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_CountryID = 250",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Country",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/Country

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_CountryID": 0
}
```

Field description

[Token] and [OA_CountryID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_CountryID":

Examples

1. Delete country "Geprgia" using "oA_CountryID" - 250

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_CountryID": 250
}
```

Server response Code(200):

```
{
  "id": 250,
  "idFieldName": "OA_CountryID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "OA_CountryID=250 [Georgia]\nExist relation to another table.\n\nRelated table [OA_City].\n\n\n[547]
The DELETE statement conflicted with the REFERENCE constraint \"FK_OA_City_OA_Country\". The conflict
occurred in database \"NDatabaseEN\", table \"dbo.OA_City\", column 'OA_CountryID'.[3621] The statement has
been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
```

```

    "class": "Country",
    "entryPoint": "RealDelete",
    "resourceDLL_ForTranslation": "ORIS_Administrate",
    "errorPointer": 0
  }
]

```

CountriesList

POST/api/CountriesList

Gets Countries. Returns [Info] and [Result] tables.

Schema

```

{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}

```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get first 2 country names from list

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2
}

```

Server response Code(200):

```

{
  "Info": [
    {
      "RowCount": 2,
      "PageCount": 0,
      "PageRowCount": 2,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_CountryID": 2,
      "Name": "Aland Islands",
      "Country code": "AX",
      "Numeric code": "248"
    },
    {
      "OA_CountryID": 3,
      "Name": "Albania",
      "Country code": "AL",
      "Numeric code": "8"
    }
  ]
}

```

2. Get the 2nd page from the entire list of countries, 15 records per page

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 15,
  "pageNumber": 2
}
```

3. Get all field names for further use in the filter

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0
}
```

4. Get a list of countries whose name starts with G and sort by name decreasing

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[Name] LIKE N'G'",
  "sort": "[Name] DESC"
}
```

Bank

POST/api/Bank

Adds new bank. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "code": "string",
  "name": "string"
}
```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"code":

"name":

Examples

1. Add new bank "National Bank Of Georgia"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "name": "National Bank Of Georgia"
}
```

Server response Code(200):

```
{
  "id": 23,
  "idFieldName": "OA_BanksID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!"
  }
]
```

```

"comment": "OA_BanksID=0 []\nA duplicate value cannot be inserted.\r\n[Code]\r\n\r\n[2601] Cannot insert
duplicate key row in object 'dbo.OA_Banks' with unique index 'IdxOA_Banks1'. The duplicate key value is ().[3621]
The statement has been terminated.",
"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "Banks",
"entryPoint": "DBUpdateAction",
"resourceDLL_ForTranslation": "ORIS_Administrate",
"errorPointer": 0
}
]

```

PUT/api/Bank

Updates bank. Returns OK or Error description Json.

Schema

```

{
  "token": "string",
  "oA_BanksID": 0,
  "code": "string",
  "name": "string"
}

```

Field description

[Token] and [OA_BanksID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_BanksID": See [BanksList] method (use [OA_BanksID] value)

"code":

"name":

Examples

1. Update Bank code to "BNLNGE22" using "oA_BanksID – 23 “National Bank Of Georgia”

Request body:

```

{
  "token": "... paste here [LogIn] response value ...",
  "oA_BanksID": 23,
  "code": "BNLNGE22"
}

```

Server response Code(200):

```

{
  "id": 23,
  "idFieldName": "OA_BanksID"
}

```

Server response Code(400):

```

[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "OA_BanksID=23 [BNLNGE22]\nA duplicate value cannot be inserted.\r\n[Code] =
[BNLNGE22]\r\n\r\n[2601] Cannot insert duplicate key row in object 'dbo.OA_Banks' with unique index
'IdxOA_Banks1'. The duplicate key value is (BNLNGE22).[3621] The statement has been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Banks",
    "entryPoint": "DBUpdateAction",
    "resourceDLL_ForTranslation": "ORIS_Administrate",
    "errorPointer": 0
  }
]

```

DELETE/api/Bank

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_BanksID": 0
}
```

Field description

[Token] and [OA_BanksID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_BanksID":

Examples

1. Delete Bank using oA_BanksID - 23

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_BanksID": 23
}
```

Server response Code(200):

```
{
  "id": 23,
  "idFieldName": "OA_BanksID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "OA_BanksID=23 [BNLNGE22]\nExist relation to another table.\r\nRelated table  
[OA_DebitorCreditorBankAccounts].\r\n\r\n[547] The DELETE statement conflicted with the REFERENCE constraint  
\\"FK_OA_DebitorCreditorBankAccounts_OA_Banks\". The conflict occurred in database \\"DatabaseEN\", table  
\\"dbo.OA_DebitorCreditorBankAccounts\", column 'OA_BanksID'. [3621] The statement has been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Banks",
    "entryPoint": "RealDelete",
    "resourceDLL_ForTranslation": "ORIS_Administrate",
    "errorPointer": 0
  }
]
```

BanksList

POST/api/BanksList

Gets Banks. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```



```
}
```

Field description

"**token**": Set [token] string from [LogIn] response value

"**topRecordsCount**": The number of records to accept. If you specify 0 you will get all records. Default=0

"**filter**": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"**sort**": Use SQL ORDER BY string with field names

"**pageNumber**": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get the 8th bank from the entire list

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "pageNumber": 8
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 22,
      "PageCount": 22,
      "PageRowCount": 1,
      "PageNumber": 8
    }
  ],
  "Result": [
    {
      "OA_BanksID": 8,
      "Code": "TBCBGE22",
      "Name": "TBC Bank",
      "NUM_ROW": 8
    }
  ]
}
```

2. Get first 2 bank names from list

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2
}
```

3. Get all field names for further use in the filter

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0
}
```

4. Get a list of banks and sort by name decreasing

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0,
  "sort": "[Name] DESC"
}
```

5. Get one bank - National Bank Of Georgia, filter by name

```
{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[Name] IN (N'National Bank Of Georgia')"
```

Vehicle

POST/api/Vehicle

Adds vehicle. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "number": "string",
  "mark": "string",
  "trailer": "string",
  "comment": "string"
}
```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"number":

"mark":

"trailer":

"comment":

Examples

1. Add vehicle "Vehicle mark"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "number": "123456",
  "mark": "Vehicle mark"
}
```

Server response Code(200):

```
{
  "id": 1,
  "idFieldName": "OA_VehiclesID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "OA_VehiclesID=2\nA duplicate value cannot be inserted.\r\n[Number] = [123456]\r\n\r\n[2601] Cannot insert duplicate key row in object 'dbo.OA_Vehicles' with unique index 'IndxOA_Vehicles1'. The duplicate key value is (123456).[3621] The statement has been terminated.",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Vehicles",
    "entryPoint": "Update",
    "resourceDLL_ForTranslation": "ORIS_Administrate",
    "errorPointer": 0
  }
]
```

PUT/api/Vehicle

Adds vehicle. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_VehiclesID": 0,
  "number": "string",
  "mark": "string",
  "trailer": "string",
  "comment": "string"
}
```

Field description

[Token] and [OA_VehiclesID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_VehiclesID": See [VehiclesList] method (use [OA_VehiclesID] value)

"number":

"mark":

"trailer":

"comment":

Examples

1. Update vehicle number to "123-321" using oA_VehicleID – 1
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_VehiclesID": 11,
  "number": "123-321"
}
```

Server response Code(200):

```
{
  "id": 1,
  "idFieldName": "OA_VehiclesID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_VehiclesID = 1",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Vehicles",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/Vehicle

Returns OK or Error description Json

Schema

```
{
  "token": "string",
  "oA_VehiclesID": 0
}
```

Field description

[Token] and [OA_VehiclesID] is necessary.

"**token**": Set [token] string from [LogIn] response value
"oA_VehiclesID":

Examples

1. Delete vehicle using "oA_VehiclesID" - 1
Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "oA_VehiclesID": 1  
}
```

Server response Code(200):

```
{  
  "id": 1,  
  "idFieldName": "OA_VehiclesID"  
}
```

Server response Code(400):

```
[  
  {  
    "errorNumber": 51051,  
    "description": "Record does not exist or read prohibited for user!",  
    "comment": "OA_VehiclesID = 1",  
    "command": "",  
    "errorType": "ErrorType_Error",  
    "source": "OA_BusinessObject.dll",  
    "class": "Vehicles",  
    "entryPoint": "GetProperties",  
    "resourceDLL_ForTranslation": "ORIS_Accounting",  
    "errorPointer": 0  
  }  
]
```

VehiclesList

POST/api/VehiclesList

Gets . Returns [Info] and [Result] tables.

Schema

```
{  
  "token": "string",  
  "topRecordsCount": 0,  
  "filter": "string",  
  "sort": "string",  
  "pageNumber": 0  
}
```

Field description

"**token**": Set [token] string from [LogIn] response value

"**topRecordsCount**": the number of records to accept. If you specify 0 you will get all records. Default=0

"**filter**": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"**sort**": Use SQL ORDER BY string with field names

"**pageNumber**": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get all field names for further use in the filter
Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "topRecordsCount": 0  
}
```

```
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 1,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_VehiclesID": 1,
      "Number": "123-321",
      "Mark": "Vehicle mark",
      "Trailer": "",
      "Comment": ""
    }
  ]
}
```

2. Get the 2nd vehicle from the entire list

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "pageNumber": 2
}
```

3. Get first 2 vehicle names from list

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2
}
```

4. Get a list of vehicle and sort by name

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0,
  "sort": "[Mark]"
}
```

5. Get one vehicle filter by name

```
{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[Mark] IN (N'Vehicle mark!)"
}
```

SalePriceType

POST/api/SalePriceType

Adds new sale price type. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "description": "string",
  "isClosed": true
}
```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"description":

"isClosed":

Examples

1. Add sale price type – “Cost-plus pricing”

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "description": "Cost-plus pricing",
  "isClosed": false
}
```

Server response Code(200):

```
{
  "id": 1,
  "idFieldName": "OA_SalePriceTypesID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51224,
    "description": "Sale price type already exists!",
    "comment": "Sale price type = [Cost-plus pricing]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "SalePriceTypes",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/SalePriceType

Updates sale price type. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_SalePriceTypesID": 0,
  "description": "string",
  "isClosed": true
}
```

Field description

[Token] and [OA_StatusID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_SalePriceTypesID": See [SalePriceTypesList] method (use [OA_SalePriceTypesID] value)

"description":

"isClosed":

Examples

1. Update sale price type using oA_SalePriceTypesID 1 – mark as closed

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",

```

```
"oA_SalePriceTypesID": 1,
"isClosed": true
}
```

Server response Code(200):

```
{
  "id": 1,
  "idFieldName": "OA_SalePriceTypesID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_SalePriceTypesID = 1",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "SalePriceTypes",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

DELETE/api/SalePriceType

Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_SalePriceTypesID": 0
}
```

Field description

[Token] and [OA_StatusID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_SalePriceTypesID":

Examples

1. Delete sale price type using oA_SalePriceTypesID 1

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_SalePriceTypesID": 1
}
```

Server response Code(200):

```
{
  "id": 1,
  "idFieldName": "OA_SalePriceTypesID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51109,
    "description": "ORIS_Tools internal error!",
    "comment": "Exist relation to another table.\n\nRelated table [OA_Transactions].\n\n[547] The DELETE statement conflicted with the REFERENCE constraint \"FK_OA_Transactions_OA_SalePriceTypes\". The conflict
```

occurred in database \"NDatabaseEN\", table \"dbo.OA_Transactions\", column 'OA_SalePriceTypesID'.[3621] The statement has been terminated.

```
"command": "",
"errorType": "ErrorType_Error",
"source": "OA_BusinessObject.dll",
"class": "SalePriceTypes",
"entryPoint": "RealDelete",
"resourceDLL_ForTranslation": "ORIS_Administrate",
"errorPointer": 0
}
]
```

SalePriceTypesList

POST/api/SalePriceTypesList

Gets Sale Price Types. Returns [Info] and [Result] tables.

Schema

```
{
  "token": "string",
  "topRecordsCount": 0,
  "filter": "string",
  "sort": "string",
  "pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get all field names for further use in the filter

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 2,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_SalePriceTypesID": 1,
      "Description": "Cost-plus pricing",
      "Is closed": true
    },
    {
      "OA_SalePriceTypesID": 2,
      "Description": "Value pricing",
      "Is closed": false
    }
  ]
}
```



```
]
}
```

2. Get the 2nd sale price type from the entire list

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "pageNumber": 2
}
```

3. Get first 2 sale price type names from list

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 2
}
```

4. Get a list of sale price type and sort by name

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0,
  "sort": "[Description]"
}
```

5. Get list of sale price type wich isn't markes as close

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "filter": "[ViewOA_SalePriceTypes].[Is closed]=0 "
}
```

Status

POST/api/Status

Adds status. Returns ID value or Error description Json.

Schema

```
{
  "token": "string",
  "name": "string"
}
```

Field description

[Token] is necessary. Specify only those fields that you want to fill.

"token": Set [token] string from [LogIn] response value

"name":

Examples

1. Add status name – "Status Active"

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "name": "Status Active"
}
```

Server response Code(200):

```
{
  "id": 5,
  "idFieldName": "OA_StatusID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51133,
    "description": "Unit already exists!",
    "comment": "Name [Status Active]\nOA_StatusID [5]",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Status",
    "entryPoint": "CheckProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

PUT/api/Status

Updates status. Returns OK or Error description Json.

Schema

```
{
  "token": "string",
  "oA_StatusID": 0,
  "name": "string"
}
```

Field description

[Token] and [OA_StatusID] is necessary. Specify only those fields that you want to change.

"token": Set [token] string from [LogIn] response value

"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value)

"name":

Examples

1. Update status name to "Active" using oA_StatusID - 5
Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "oA_StatusID": 5,
  "name": "Active"
}
```

Server response Code(200):

```
{
  "id": 5,
  "idFieldName": "OA_StatusID"
}
```

Server response Code(400):

```
[
  {
    "errorNumber": 51051,
    "description": "Record does not exist or read prohibited for user!",
    "comment": "OA_StatusID = 5",
    "command": "",
    "errorType": "ErrorType_Error",
    "source": "OA_BusinessObject.dll",
    "class": "Status",
    "entryPoint": "GetProperties",
    "resourceDLL_ForTranslation": "ORIS_Accounting",
    "errorPointer": 0
  }
]
```

]

DELETE/api/Status

Returns OK or Error description Json

Schema

```
{  
  "token": "string",  
  "oA_StatusID": 0  
}
```

Field description

[Token] and [OA_StatusID] is necessary.

"token": Set [token] string from [LogIn] response value

"oA_StatusID": See [StatusesList] method (use [OA_StatusID] value). Can be omitted if [Status] property specified

Examples

1. Delete status using oA_StatusID – 5

Request body:

```
{  
  "token": "... paste here [LogIn] response value ...",  
  "oA_StatusID": 5  
}
```

Server response Code(200):

```
{  
  "id": 5,  
  "idFieldName": "OA_StatusID"  
}
```

Server response Code(400):

```
[  
  {  
    "errorNumber": 51109,  
    "description": "ORIS_Tools internal error!",  
    "comment": "OA_StatusID=5 [Active]\nExist relation to another table.\n\nRelated table  
[OA_Transactions].\n\n[547] The DELETE statement conflicted with the REFERENCE constraint  
\"FK_OA_Transactions_OA_StatusID\". The conflict occurred in database \"NDatabaseEN\", table  
\"dbo.OA_Transactions\", column 'OA_StatusID'. [3621] The statement has been terminated.",  
    "command": "",  
    "errorType": "ErrorType_Error",  
    "source": "OA_BusinessObject.dll",  
    "class": "Status",  
    "entryPoint": "RealDelete",  
    "resourceDLL_ForTranslation": "ORIS_Administrate",  
    "errorPointer": 0  
  }  
]
```

StatusesList

POST/api/StatusesList

Gets Statuses. Returns [Info] and [Result] tables.

Schema

```
{  
  "token": "string",  
  "topRecordsCount": 0,  
  "filter": "string",  
}
```

```
"sort": "string",
"pageNumber": 0
}
```

Field description

"token": Set [token] string from [LogIn] response value

"topRecordsCount": The number of records to accept. If you specify 0 you will get all records. Default=0

"filter": Use SQL WHERE string by field names. Get field names by launch this method with property:

[TopRecordsCount]=1

"sort": Use SQL ORDER BY string with field names

"pageNumber": Gets records by page number. Specify [TopRecordsCount] as the number of records per page

Examples

1. Get all field names for further use in the filter

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 0
}
```

Server response Code(200):

```
{
  "Info": [
    {
      "RowCount": 5,
      "PageCount": 0,
      "PageRowCount": 0,
      "PageNumber": 0
    }
  ],
  "Result": [
    {
      "OA_StatusID": 5,
      "Name": "Active"
    },
    {
      "OA_StatusID": 3,
      "Name": "Completed"
    },
    {
      "OA_StatusID": 2,
      "Name": "Confirmed"
    },
    {
      "OA_StatusID": 1,
      "Name": "In progress"
    },
    {
      "OA_StatusID": 4,
      "Name": "Rejected"
    }
  ]
}
```

2. Get the 2nd status name from the entire list

Request body:

```
{
  "token": "... paste here [LogIn] response value ...",
  "topRecordsCount": 1,
  "pageNumber": 2
}
```

3. Get first 2 status names from list

Request body:

```
{
```

```
"token": "... paste here [Login] response value ...",
"topRecordsCount": 2
}
```

4. Get a list of **status names** and sort by name
Request body:

```
{
"token": "... paste here [Login] response value ...",
"topRecordsCount": 0,
"sort": "[Name]"
}
```

5. Get information about status name **Active** using filter
Request body:

```
{
"token": "... paste here [Login] response value ...",
"filter": "[ViewOA_Status].[Name] IN (N'Active')"
}
```