

Примеры запросов в REST API

Последние изменения: 2024-03-26

Получение информации о базе

Для получения информации по текущей базе нужно выполнить запрос:

GET [/api/v1/BaseInfo](#)

Ответ:

```
{
"@odata.context": "http://172.19.0.30:9000/MobileSMARTS/api/v1/$metadata#BaseInfo",
"id": "rtl15",
"name": "Магазин 15, Базовый",
"folder": "C:ProgramDataCleverenceБазы Mobile SMARTSМагазин 15, Базовый",
"appId": "F42C7B5F-405C-4076-AE07-9348F189EE71",
"appName": "Магазин 15, Базовый",
"comment": null,
"allConnectionStrings": [
"https://172.19.0.30:10502/rtl15",
"Доступ к swagger:"
"https://172.19.0.30:10502/rtl15/swagger"
],
"appDescription": {
"appId": "F42C7B5F-405C-4076-AE07-9348F189EE71",
"appName": "Магазин 15, Базовый",
"desktopReqPayment": false,
"configId": "8DC2C8BA-77CE-46AD-93CB-6CFFCA7B96D7",
"serverModeSupported": true,
"onlineCallsSupported": false,
"batchModeSupported": true,
"appLink": "http://cleverence.ru/software/mobile-smarts/rtl15/",
"appSupportLink": "https://www.cleverence.ru/support/category:295/",
"aboutLicLink": "http://cleverence.ru/software/mobile-smarts/rtl15/#spec",
"appVersionsLink": "http://cleverence.ru/software/mobile-smarts/rtl15/#spec",
"comment": "Основная поставка Магазин 15",
"appVersion": "1.1.1.155",
"clientVersion": "3.0.0.100",
"panelVersion": "1.0",
"mainServerVersion": "1.0",
"appServerVersion": "3.0.0.2699",
"minPlatformVersion": "1.0",
"minPlatform35Version": "1.0"
},
"appInstanceSettings": {
"mode": "Server",
"hasServerAuth": false,
"serverSettings": {
```

```

"dataService": {
  "enabled": true,
  "port": 9000,
  "useHttps": false,
  "deviceAuth": false,
  "passwordAuth": false
},
"printService": {
  "enabled": false,
  "port": 9001,
  "useHttps": false,
  "deviceAuth": false,
  "passwordAuth": false
}
}
}
}

```

Группы пользователей

Работа с массивом

GET </api/v1/UserGroups>

```

{
  "@odata.context": "string",
  "value": [
    {
      "id": "string",
      "name": "string",
      "documentTypeNames": [
        "string"
      ],
      ": true,
      "role": 0,
      "serverSideInventory": true,
      "autorunDocumentTypeName": "string",
      "onStartHandlerName": "string",
      "onFinishHandlerName": "string"
    }
  ]
}

```

Работа с записями по идентификатору

POST [/api/v1/UserGroups \('{key}'\)](/api/v1/UserGroups ('{key}'))

PUT [/api/v1/UserGroups \('{key}'\)](/api/v1/UserGroups ('{key}'))

PATCH [/api/v1/UserGroups \('{key}'\)](/api/v1/UserGroups ('{key}'))

DELETE [/api/v1/UserGroups \('{key}'\)](/api/v1/UserGroups ('{key}'))

UserGroup {

id (*string, optional*): Id пользователя ,

name (*string, optional*): Уникальное наименование группы ,

documentTypeNames (*Array[string], optional, read only*),

batchMode (*boolean, optional*): Свойство, позволяющее задать тип обмена данными с сервером. Если true - обмен производится вручную, по запросу пользователем. Иначе, обмен будет производиться периодически, с интервалом, заданным в настройках клиентского приложения ,

role (*integer, optional*): Пользовательская роль = ['0', '1', '2', '3'],

serverSideInventory (*boolean, optional*): Указывает источник номенклатурного справочника. Если true - позиции номенклатуры будут искаться на сервере(необходимо наличие постоянной связи с сервером), иначе - справочник номенклатуры будет загружаться на терминал при операции обмена данными и использоваться локально ,

autorunDocumentTypeName (*string, optional*): Тип документа (виртуальный), который будет открыт автоматически, сразу после запуске программы на ТСД ,

onStartHandlerName (*string, optional*): Имя операции, запускающейся при входе пользователя данной группы ,

onFinishHandlerName (*string, optional*): Имя операции, запускающейся при выходе пользователя данной группы

}

Получение списка пользователей группы

GET [/api/v1/UserGroups \('{key}'\)/users](#)

Пользователи

Работа с массивом

GET [/api/v1/Users](#)

Работа с записями по идентификатору

POST [/api/v1/Users](#)

PUT [/api/v1/Users](#)

PATCH [/api/v1/Users](#)

DELETE [/api/v1/Users](#)

User {

```

id (string, optional): Id пользователя ,
name (string, optional): Имя пользователя ,
password (string, optional): (write only) Пароль пользователя ,
description (string, optional): Описание пользователя ,
barcode (string, optional): (write only) Штрихкод пользователя ,
groupId (string, optional): Имя группы в которой состоит пользователь. Подробнее смотрите
UserGroup ,
groupName (string, optional): Для совместимости со старыми обработками ,
warehouseIds (Array[string], optional, read only)
}

```

Ячейки

GET [/api/v1/Cells](#) — Список ячеек

Работа с записями по идентификатору

POST [/api/v1/Cells](#) — Добавить/редактировать ячейку

DELETE [/api/v1/Cells \('{id}'\)](#) — Удалить ячейку

GET [/api/v1/Cells \('{id}'\)](#) — Получить ячейку по идентификатору

PUT [/api/v1/Cells \('{id}'\)](#) — Добавь/редактировать ячейку по идентификатору

Обновление справочника со сбросом всех записей

POST [/api/v1/Cells/BeginOverwrite](#) — начинает процедуру пакетной выгрузки ячеек на сервер. Все позиции будут накапливаться и не будут доступны до вызова функции EndOverwrite.

POST [/api/v1/Cells/EndOverwrite](#) — завершает процедуру пакетной выгрузки номенклатуры. Только после вызова этой функции сервер завершит обработку переданных позиций номенклатуры и они попадут в справочник товаров. Старый справочник товаров при этом будет полностью очищен.

Обновление записей

POST [/api/v1/Cells/BeginUpdate](#) — начинает процедуру пакетного обновления ячеек на сервере. Все позиции будут накапливаться и не будут доступны до вызова функции EndUpdate.

POST [/api/v1/Cells/EndUpdate](#) — завершает процедуру пакетного обновления ячеек. Только после вызова этой функции сервер завершит обработку переданных позиций номенклатуры и они попадут в справочник товаров. Выгруженная номенклатура будет слита с существующим на сервере справочником.

Сброс обновления/ перезаписывания

POST [/api/v1/Cells/ResetUpdate](#) — сбрасывает процедуру пакетного обновления ячеек.

Номенклатура

Получение схемы

Получение списка всех полей номенклатуры:

[https://localhost:9000/api/v1/ProductSchema?\\$expand=allfields](https://localhost:9000/api/v1/ProductSchema?$expand=allfields)

Получение полей номенклатуры:

[https://localhost:9000/api/v1/ProductSchema?\\$expand=fields](https://localhost:9000/api/v1/ProductSchema?$expand=fields)

Получение списка фиксированных полей:

[https://localhost:9000/api/v1/ProductSchema?\\$expand=defaultFields](https://localhost:9000/api/v1/ProductSchema?$expand=defaultFields)

Работа с массивом

GET [/api/v1/Products](#) — список номенклатуры.

Работа с записями по идентификатору

POST [/api/v1/Products](#) — добавить/ редактировать номенклатуру.

DELETE [/api/v1/Products \('{id}'\)](#) — удалить номенклатуру.

GET [/api/v1/Products \('{id}'\)](#) — получить номенклатуру по идентификатору.

PUT [/api/v1/Products \('{id}'\)](#) — добавить/ редактировать номенклатуру по идентификатору.

Обновление справочника со сбросом всех записей

POST [/api/v1/Products/BeginOverwrite](#) — начинает процедуру пакетной выгрузки номенклатуры на сервер. Все позиции будут накапливаться и не будут доступны до вызова функции EndOverwrite.

POST [/api/v1/Products/EndOverwrite](#) — завершает процедуру пакетной выгрузки номенклатуры. Только после вызова этой функции сервер завершит обработку переданных позиций номенклатуры и они попадут в справочник товаров. Старый справочник товаров при этом будет полностью очищен.

Обновление записей

POST [/api/v1/Products/BeginUpdate](#) — начинает процедуру пакетного обновления номенклатуры на сервере. Все позиции будут накапливаться и не будут доступны до вызова функции EndUpdate.

POST [/api/v1/Products/EndUpdate](#) — завершает процедуру пакетного обновления номенклатуры. Только после вызова этой функции сервер завершит обработку переданных позиций номенклатуры и они попадут в справочник товаров. Выгруженная номенклатура будет слита с существующим на сервере справочником.

Сброс обновления/ перезаписывания

POST [/api/v1/Products/ResetUpdate](#) — сбрасывает процедуру пакетного обновления номенклатуры.

Обновление номенклатуры таблицей значений

POST [/api/v1/Products/BeginUploadProducts](#) — начинает выгрузку позиций номенклатуры.

POST [/api/v1/Products/AddProductToUpload](#) — добавляет в выгрузку товаров товар с упаковкой.

POST [/api/v1/Products/AddProductsToUpload](#) — добавляет в выгрузку товаров товаров с упаковками.

POST [/api/v1/Products/EndUploadProducts](#) — завершает выгрузку товаров.

Таблицы

Работа с массивом

GET [/api/v1/Tables/BiznesProcessy](#) — получить все записи таблицы.

Работа с записями по идентификатору

POST [/api/v1/Tables/BiznesProcessy](#) — редактировать/ добавить запись.

DELETE [/api/v1/Tables/BiznesProcessy \('{uid}'\)](#) — удалить запись из таблицы.

GET [/api/v1/Tables/BiznesProcessy \('{uid}'\)](#) — получить запись по идентификатору.

PATCH [/api/v1/Tables/BiznesProcessy \('{uid}'\)](#) — редактировать запись.

PUT [/api/v1/Tables/BiznesProcessy \('{uid}'\)](#) — редактировать/добавить запись по известному идентификатору.

Обновление справочника со сбросом всех записей

POST [/api/v1/Tables/BiznesProcessy/BeginOverwrite](#) — начинает процедуру пакетной выгрузки строк таблицы на сервер. Все позиции будут накапливаться и не будут доступны до вызова функции EndOverwrite.

POST [/api/v1/Tables/BiznesProcessy/EndOverwrite](#) — завершает процедуру пакетной выгрузки строк таблицы. Только после вызова этой функции сервер завершит обработку переданных позиций и они попадут в таблицу. Старое содержимое при этом будет полностью очищено.

Обновление записей

POST [/api/v1/Tables/BiznesProcessy/BeginUpdate](#) — начинает процедуру пакетного обновления строк таблицы на сервере. Все передаваемые будут накапливаться и не будут доступны до вызова функции EndUpdate.

POST [/api/v1/Tables/BiznesProcessy/EndUpdate](#) — завершает процедуру пакетного обновления строк таблицы. Только после вызова этой функции сервер завершит обработку переданных позиций и они попадут таблицу. Выгруженные позиции будут слиты с существующей на сервере таблицей.

Сброс обновления\перезаписывания

POST [/api/v1/Tables/BiznesProcessy/ResetUpdate](#) — сбрасывает процедуру пакетного обновления строк таблицы.

Документы

Получение списка типов документов

GET [/api/v1/DocTypes](#) — список типов документов.

GET [/api/v1/DocTypes \('{uni}'\)](#) — получить тип документа по идентификатору.

Получение всех полей типа документа

[https://localhost:9000/api/v1/DocTypes?\\$expand=allfields](https://localhost:9000/api/v1/DocTypes?$expand=allfields)

Работа с массивом

GET [/api/v1/Docs](#) — список документов.

POST [/api/v1/Docs](#) редактировать/ добавить документ.

Работа с записями по идентификатору

DELETE [/api/v1/Docs \('{id}'\)](#) — удалить документ.

GET [/api/v1/Docs \('{id}'\)](#) — получить документ по идентификатору.

PATCH [/api/v1/Docs \('{id}'\)](#) — редактировать документ.

PUT [/api/v1/Docs \('{id}'\)](#) — редактировать/добавить документ по известному идентификатору.

Обновление записей

При любом редактировании документа, он сразу не сохраняется в систему. Сохранение происходит, если вызвать принудительное сохранение ([EndUpdate](#)), либо через 30 сек от последнего изменения.

POST [/api/v1/Docs \('{id}'\)/EndUpdate](#) — принудительно сохраняет документ, когда все строки уже загружены (не дожидаясь сохранения через 30 сек, как указано выше).

Получение строк документа

GET [/api/v1/Docs \('{key}'\)/declaredItems](#) — получить строк документа.

POST [/api/v1/Docs \({key}\)/declaredItems](#) — редактировать/добавить строку документа.

[https://localhost:5001/api/v1/Docs \('{id}'\) ?\\$expand=currentItems](https://localhost:5001/api/v1/Docs ('{id}') ?$expand=currentItems) — получить CurentlItems строк документа по его ID.

Блокировка документа

POST [/api/v1/Docs \('{id}'\)/Block](#) — блокирует документ для совместной работы.

Тело запроса:

```
{"timeout»:1000}
```

, где timeout — время блокировки документа

POST [/api/v1/Docs \('{id}'\)/Unblock](#) — разблокирует документ для совместной работы.

Склады

Работа с массивом

GET [/api/v1/Warehouses](#) — получить список складов.

Работа с записями по идентификатору

POST [/api/v1/Warehouses](#) — добавление/ редактирование склада.

DELETE [/api/v1/Warehouses \('{id}'\)](#) — удаление склада.

GET [/api/v1/Warehouses \('{id}'\)](#) — получить конкретный склад.

PATCH [/api/v1/Warehouses \('{id}'\)](#) — редактирование существующего склада.

PUT [/api/v1/Warehouses \('{id}'\)](#) — добавление/ редактирование склада по существующему идентификатору.

Warehouse {

```

storageld (string, optional): Идентификатор склада, для хранения (не меняется) ,
id (string, optional): Уникальный идентификатор склада ,
name (string, optional): Наименование склада ,
cells (Array[Cell], optional, read only): Коллекция ячеек склада
}
Cell {
  barcode (string, optional): Штрихкод ячейки. Может быть шаблонизированным. Подробнее про
  применение шаблонов для ячеек, смотрите Руководство разработчика ,
  id (string, optional): Id ячейки, искусственный ключ ,
  name (string, optional): Наименование ячейки ,
  description (string, optional): Описание ячейки
}

```

Получение информации о подключенных устройствах

GET [/api/v1/Devices](#) — список устройств.

```

DeviceInfo {
  appInstanceId (string, optional),
  deviceId (string, optional): Уникальный идентификатор терминала. Заполняется самим
  устройством при регистрации его в системе ,
  batteryStatus (string, optional): Уровень заряда батарей ,
  lastInfoTime (string, optional): Последнее время получения информации о терминале ,
  userId (string, optional): Id пользователя, работавший с устройством в момент последнего
  получения информации о нем ,
  warehouseld (string, optional): Идентификатор склада, на котором работает пользователь ,
  documentId (string, optional): Идентификатор документа, с которым работает пользователь ,
  cellId (string, optional): Идентификатор последней ячейки, с которой работал пользователь ,
  deviceName (string, optional): Пользовательское имя устройства ,
  documentTypeName (string, optional),
  serverHostedDocument (boolean, optional),
  deviceIp (string, optional),
  userGroupId (string, optional)
}

```

Настройки

GET [/api/v1/CustomSettings](#) — список настроек

Результат:


```
{
"@odata.context": "http://localhost:9000/MobileSMARTS/api/v1/$metadata#CustomSettings",
"value": [
{
"name": "testProp",
"value": "testValue"
}
]
}
```

POST [/api/v1/CustomSettings](#) — Добавить/отредактировать настройку

Тело запроса:

```
{
"name": "testProp",
"value": "testValue"
}
```

Также можно добавлять массив настроек:

```
{
"value": [
{
"name": "testProp",
"value": "testValue"
},
{
"name": "testProp1",
"value": "testValue1"
}
]
}
```

GET [/api/v1/CustomSettings \('{name}'\)](#) — получить настройку

Тело ответа:

```
{
"@odata.context":
"http://localhost:9000/MobileSMARTS/api/v1/$metadata#CustomSettings/$entity",
"name": "testProp",
"value": "testValue"
}
```

PUT [/api/v1/CustomSettings \('{name}'\)](#) — Добавить/отредактировать настройку по известному идентификатор.

Тело запроса:

```
{
  "name": "testProp",
  "value": "testValue"
}
```

DELETE [/api/v1/CustomSettings \('{name}'\)](#) — удалить настройку

Сообщения

Действия с сообщениями:

GET [/api/v1/Messages](#) — список сообщений.

POST [/api/v1/Messages](#) — добавить/отредактировать сообщение.

GET [/api/v1/Messages \('{id}'\)](#) — получить сообщение по идентификатору.

MessageInfo {

```
  applInstanceId (string, optional),
  deviceId (string, optional): Уникальный идентификатор терминала. Заполняется самим
  устройством при регистрации его в системе ,
  batteryStatus (string, optional): Уровень заряда батарей ,
  lastInfoTime (string, optional): Последнее время получения информации о терминале ,
  userId (string, optional): Id пользователя, работавший с устройством в момент последнего
  получения информации о нем ,
  warehouseId (string, optional): Идентификатор склада, на котором работает пользователь ,
  documentId (string, optional): Идентификатор документа, с которым работает пользователь ,
  cellId (string, optional): Идентификатор последней ячейки, с которой работал пользователь ,
  deviceName (string, optional): Пользовательское имя устройства ,
  documentTypeName (string, optional),
  serverHostedDocument (boolean, optional),
  deviceIp (string, optional),
  userGroupId (string, optional)
}
```



REST API

Не нашли что искали?



Задать вопрос в техническую поддержку