

Загрузка остатков через API

Дилеры могут автоматизировать загрузку остатков на сайт КЭАЗ, настроив ее через API.

Работа с ЛК

Для работы с данными из ЛК необходимо получить API ключ, он доступен в информации о компании пользователям с ролью "Администратор дилера".

Общий вид запросов

url: /restapi/office/ресурс.действие.тип

Ресурс - объект над которым будут производиться действия

Действие - стандартный набор функций: get, list, add, update, delete. Набор доступных действий различается в зависимости от конкретных ресурсов.

Тип - возвращаемый тип данных: json, xml. По умолчанию тип данных json

Данные для обработки передаются POST-запросом. Параметры передаются либо в стандартном POST-запросе, либо в формате JSON/XML. Обязательным параметром является hash, содержащий api-ключ пользователя Личного кабинета.

API ключ

Для работы с данными необходимо получить API-ключ. Он доступен на странице "Информация о компании" пользователям Личного кабинета с ролью "Администратор дилера".

Примеры запросов

Ниже приведены примеры запросов в трех форматах, данные одни и те же - api ключ и фильтр по id равному 2916

Пример запроса

POST

```
hash=jqtlkuvhik0o4ogg4wswk4ws0kwww4k&filter[id][0]=2916
```

Запрос перед отправкой необходимо кодировать через urlencode

```
hash=jqtlkuvhik0o4ogg4wswk4ws0kwww4k&filter%5Bid%5D%5B0%5D=2916
```

JSON

```
{  
  "hash": "jqtlkuvhik0o4ogg4wswk4ws0kwww4k",  
  "filter": [  
    {
```

```
    "id": 2619
  }
]
}
```

XML

```
<request>
  <hash>jqtlkuvhik0o4ogg4wswk4ws0kwww4k</hash>
  <filter>
    <id>2619</id>
  </filter>
</request>
```

Общий вид ответов

xml

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
<warehouses>
  <item>
    <code>3</code>
    <contacts>3</contacts>
    <address>Курск, ул. Луначарского 8</address>
  </item>
  <item>
    <code>11111</code>
    <contacts>test</contacts>
    <address>Курск</address>
  </item>
</warehouses>
<status>
  <code>ok</code>
  <messages>
    <item>Склад `11111` уже существует!</item>
  </messages>
</status>
</response>
```

json

```
{
  "warehouses": [
    {
```

```
"code": "3",
"contacts": "3",
"address": "\u041a\u0443\u0440\u0441\u043a, \u0443\u043b\u0438\u0446\u043d\u043e\u0435 \u0431\u043e\u0440\u044c\u0435"
},
{
"code": "11111",
"contacts": "test",
"address": "\u041a\u0443\u0440\u0441\u043a"
}
],
"status": {
"code": "error",
"messages": [
"Не корректные данные склада в строке `0`:
`{"contacts": "test", "address": "\u041a\u0443\u0440\u0441\u043a}"`"
]
}
}
```

Работа со складами контрагентов

Добавление склада

Url: /restapi/office/warehouses.add.[xml|json]

Параметры для передачи:

hash - api ключ для доступа

warehouses - массив складов состоящий из

- code - код склада контрагента
- contacts - контактные данные
- address - адрес склада, на основе этого адреса вычисляются координаты для показа ближайших к пользователю складов

Пример запроса

hash = 11111111111111111111111111111111

warehouse[0][code] = 3

warehouse[0][contacts] = test@mail.ru

warehouse[0][address] = Курск

Пример ответа

xml

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <status>
    <code>ok</code>
```

```
<messages>
  <item>Склад `111111` добавлен</item>
</messages>
</status>
</response>
```

json

```
{
  "status": {
    "code": "ok",
    "messages": [
      "Склад `111111` добавлен"
    ]
  }
}
```

Редактирование склада

Url: /restapi/office/warehouses.update.[xml|json]

Параметры для передачи:

hash - api ключ для доступа

warehouses - массив складов состоящий из

- code - код склада контрагента
- contacts - контактные данные
- address - адрес склада, на основе этого адреса вычисляются координаты для показа ближайших к пользователю складов

Структура данных аналогична добавлению склада.

Удаление склада

Url: /restapi/office/warehouses.delete.[xml|json]

Параметры для передачи:

hash - api ключ для доступа

warehouses - массив складов состоящий из

- code - код склада контрагента

Для удаления склада достаточно передать только его код.

Получение списка складов

Url: /restapi/office/warehouses.list.[xml|json]

Параметры для передачи:

hash - api ключ для доступа

Пример ответа

xml

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <warehouses>
    <item>
      <code>3</code>
      <contacts>3</contacts>
      <address>Курск, ул. Луначарского 8</address>
    </item>
    <item>
      <code>111111</code>
      <contacts>test</contacts>
      <address>Курск</address>
    </item>
  </warehouses>
  <status>
    <code>ok</code>
  </status>
</response>
```

json

```
{
  "warehouses": [
    {
      "code": "3",
      "contacts": "3",
      "address": "\u041a\u0443\u0440\u0441\u043a, \u0430\u043b\u043b. \u041b\u0443\u043d\u0430\u0447\u0430\u0440\u0441\u043a\u043e\u0433\u043e 8"
    },
    {
      "code": "111111",
      "contacts": "test",
      "address": "\u041a\u0443\u0440\u0441\u043a"
    }
  ],
  "status": {
    "code": "ok"
  }
}
```

Работа с остатками на складах контрагентов

Добавление остатков

Url: /restapi/office/remains.add.[xml|json]

Параметры для передачи:

hash - api ключ для доступа

code - код склада контрагента

products - массив остатков состоящий из

- article - код продукции

- quantity - остаток на складе

При вызове данного метода происходит очистка остатков по коду склада.

Пример запроса

```
hash = 11111111111111111111111111111111
code = test
products[0][article] = 100000
products[0][quantity] = 5
```

Пример ответа

xml

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <status>
    <code>ok</code>
    <messages>
      <item>Сохранено 1 элементов</item>
    </messages>
  </status>
</response>
```

json

```
{
  "status": {
    "code": "ok",
    "messages": [
      "\u041e\u0441\u0442\u0430\u0442\u043a\u0438",
      "\u0443\u0434\u0430\u043b\u0435\u043d\u043e",
      "\u0421\u043e\u0445\u0440\u0430\u043d\u0435\u043d\u043e 1",
      "\u044d\u043b\u0435\u043c\u0435\u043d\u0442\u043e\u0432"
    ]
  }
}
```

Удаление остатков

Url: /restapi/office/remains.delete.[xml|json]

Параметры для передачи:

hash - api ключ для доступа

code - код склада контрагента

Пример ответа

xml

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
```

```
<status>
  <code>ok</code>
</status>
</response>
```

json

```
{
  "status": {
    "code": "ok"
  }
}
```