

# Авторизация в системе через REST API

Последние изменения: 2024-03-26

Если в системе включена авторизация, то для начала работы с API необходимо пройти авторизацию в системе.

Реализованы несколько вариантов авторизации:

- BASIC авторизация.
- Авторизация методом GET по адресу `/api/session` с получением token.
- Авторизация методом POST по адресу `/api/session` с получением token.

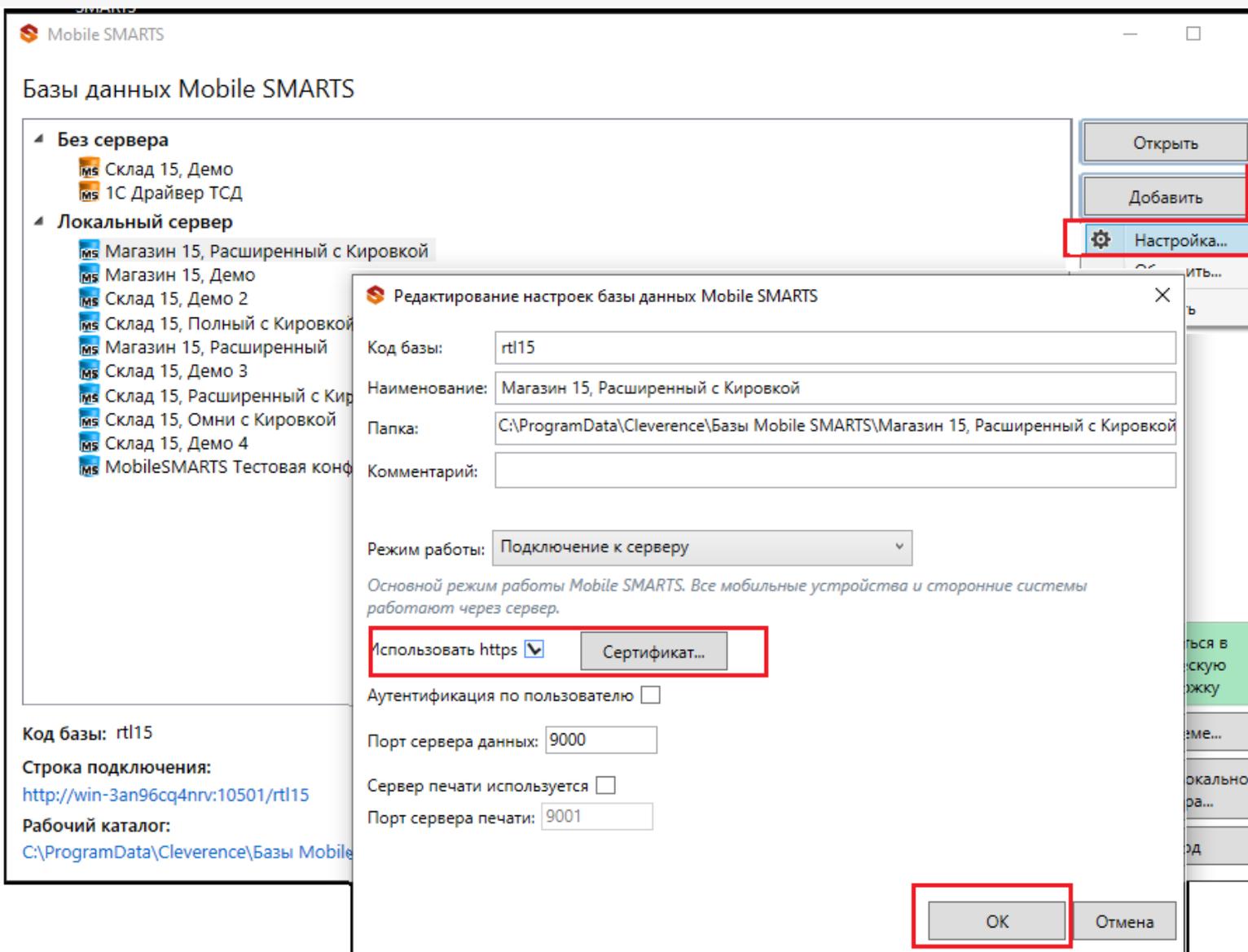
**REST API** работает только если при работе с ТСД используется сервер Mobile SMARTS. При прямом **подключении ТСД к компьютеру через кабель/кредл** или при обмене с учетной системой через папку использовать REST API не получится.

В базе Mobile SMARTS должна быть включена **авторизация**, тогда для выполнения HTTP-запросов необходимо будет каждый раз авторизовываться, есть 2 варианта авторизации:

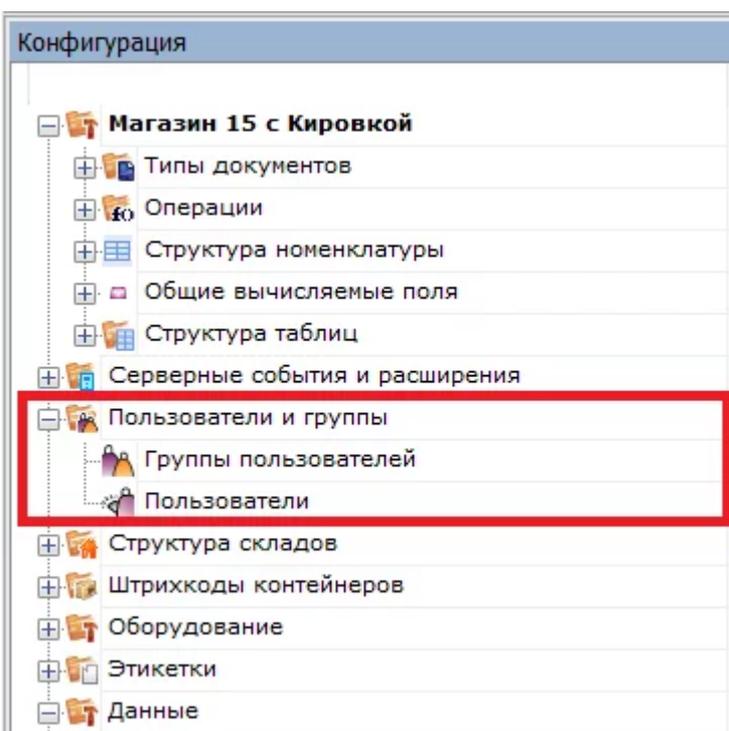
- BASIC авторизация — в этом случае при каждом HTTP-запросе нужно будет отправлять логин и пароль;
- авторизация с использованием токена — при первом HTTP-запросе передаются логин и пароль, а сервер MS возвращается токен — уникальный идентификатор сессии (access token), который при последующих HTTP-запросах можно будет использовать вместо передачи логина и пароля. Срок действия токена (т.е. сессии) ограничен, поэтому для обновления токена при первом запросе сервер Mobile SMARTS дополнительно возвращает «токен обновления» (refresh token) — он нужен для получения нового токена после истечения срока действия текущего токена.

Обратите внимание. Для защиты передаваемых данных рекомендуется использовать режим с включенной аутентификацией и доступом по https.

**Для работы через протокол https** необходимо указать это в настройках базы данных, для этого зайдите в менеджер базы, справа кнопки «Добавить» нажмите на выпадающий список и выберите пункт «Настройки», далее в окне «Редактирование настроек» поставьте галочку «Использовать http» и нажмите «ОК».



Для работы в **режиме с включенной аутентификацией** необходимо завести пользователей, задать им логины и пароли. В панели управления Mobile SMARTS узел «Пользователи и группы» содержит данные о пользователях и группах, в которых они состоят.



Инструкция по конфигурации настроек пользователей и групп пользователей, как определить роль пользователей и список типов документов, доступных для обработки пользователям такой группы описана в одноименной [статье на сайте](#).

## Пример запроса на авторизацию приложения методом POST

1. Для начала в настройках базы Mobile SMARTS включите авторизацию/аутентификацию по пользователю и нажмите на кнопку «OK»

Редактирование настроек базы данных Mobile SMARTS

Код базы: 15911b0e-830f-433b-8d8c-048988df6cdf

Наименование: MS-5748 Магазин 15

Папка: D:\Work\MS-5748\Clever\CLEVER\01fd935c-ce47-4cb6-a3c7-7ce03b12bd29

Комментарий:

Режим работы: Подключение к серверу

Основной режим работы Mobile SMARTS. Все мобильные устройства и сторонние системы работают через сервер.

Использовать https  Сертификат...

**Аутентификация по пользователю** ▾

Порт сервера данных: 51434

Сервер печати используется

Порт сервера печати: 51435

OK Отмена

2. Пример запроса на авторизацию <http://localhost:51434/api/v1/session?username={username}&password={password}> в ответ приходит Access\_token

```
v1/D
78b7

nam

nam
Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize HTML
1 "Access_token": "a856aa43587cf22dabc638005b035873", "Token_type": "bearer", "Expires_in": 2147483647, "Refresh_token":
2 "e4c0f0ce16ae352e474f78e03d295441"
```

3. В последующих запросах в заголовке указываем

{Authorization: Bearer {Access\_token}}

Если не указать Authorization, в ответ приходит статус ошибки 401 — Unauthorized

The screenshot shows a Postman request configuration for a GET request to `http://localhost:51434/MobileSMARTS/api/v1/Docs['new_07400ce3-5f35-46a6-816a-c87e278b71f5']`. The Headers tab is active, showing a list of headers. The 'Authorization' header is present with the value `Bearer a856aa43587cf22dabc638005b035873`. However, the 'Status' field in the bottom right corner is highlighted in red and displays `Status: 401 Unauthorized`, indicating that the request failed because the Authorization header was not correctly processed or was missing in the context of the test.

| KEY   | VALUE                                   | DESCRIPTION |
|---|---|-------------|
| <input checked="" type="checkbox"/> Postman-Token   | <calculated when request is sent>       |             |
| <input checked="" type="checkbox"/> Host            | <calculated when request is sent>       |             |
| <input checked="" type="checkbox"/> User-Agent      | PostmanRuntime/7.26.8                   |             |
| <input checked="" type="checkbox"/> Accept          | */*                                     |             |
| <input checked="" type="checkbox"/> Accept-Encoding | gzip, deflate, br                       |             |
| <input checked="" type="checkbox"/> Connection      | keep-alive                              |             |
| <input type="checkbox"/> Authorization              | Bearer a856aa43587cf22dabc638005b035873 |             |
| Key   | Value                                   | Description |

4. При успешной авторизации приходит ответ и статус 202 — OK

The screenshot shows the same Postman request configuration as above, but now the 'Status' field in the bottom right corner is highlighted in red and displays `Status: 202 OK`, indicating a successful response. The 'Authorization' header is checked and its value is visible in the table below.

| KEY   | VALUE                                   | DESCRIPTION |
|---|---|-------------|
| <input checked="" type="checkbox"/> Postman-Token   | <calculated when request is sent>       |             |
| <input checked="" type="checkbox"/> Host            | <calculated when request is sent>       |             |
| <input checked="" type="checkbox"/> User-Agent      | PostmanRuntime/7.26.8                   |             |
| <input checked="" type="checkbox"/> Accept          | */*                                     |             |
| <input checked="" type="checkbox"/> Accept-Encoding | gzip, deflate, br                       |             |
| <input checked="" type="checkbox"/> Connection      | keep-alive                              |             |
| <input checked="" type="checkbox"/> Authorization   | Bearer a856aa43587cf22dabc638005b035873 |             |
| Key   | Value                                   | Description |

## BASIC авторизация

Данный вид авторизации чаще всего используется браузером для доступа к функциям API.

При использовании данного метода необходимо в заголовке каждого запроса указывать:

```
Authorization: Basic {login}:{password}
```

Допускается base64 при формировании строки {login}:{password}

## Авторизация методом GET

Еще один способ авторизации — отправить GET запрос по адресу /api/v1/session, при этом в url запросе указать параметры login и password:

```
https://localhost:9000/api/v1/session?username=${Username}&password=${Password}
```

Ответ сервера:

```
{
  Access_token:"123123123",
  Token_type:"bearer",
  Expires_in:86400,
  Refresh_token:"321321321",
}
```

## Авторизация методом POST

Авторизация по логину и паролю происходит путем отправки POST запроса на сервер, в результате которого возвращается access\_token и refresh token в формате JSON.

Пример запроса:

```
POST /oauth/token HTTP/1.1
Host: mobilesmares.ru/api/session
Content-Type: application/x-www-form-urlencoded
```

```
grant_type=password&client_id=123&client_secret=user&username=user@domain.ru&password=123456
```

Ответ сервера:

```
{
  Access_token:"123123123",
  Token_type:"bearer",
  Expires_in:86400,
  Refresh_token:"321321321",
}
```

## Восстановление после окончания срока действия сессии

Восстановление после окончания срока действия сессии происходит путем отправки refresh\_token на сервер, в результате приходит новый access\_token и refresh\_token

Пример:

```
POST /oauth/token HTTP/1.1
```

Host: mobilesmares.ru/api/session  
Content-Type: application/x-www-form-urlencoded

grant\_type:refresh\_token&client\_id=123&client\_secret=user&refresh\_token=321321321

Ответ:

HTTP/1.1 200 OK  
Content-Type: application/json

```
{  
  Access_token:"99999",  
  Token_type:"bearer",  
  Expires_in:86400,  
  Refresh_token:"789789789",  
}
```

## Вызов функций с использованием token

Для того чтобы обратиться к функциям (если не используется Basic авторизация), для которых необходима авторизация, необходимо в заголовке Authorization передавать токен:

Authorization: Bearer <token>

Иначе сервер вернет ошибку авторизации 401.

 интеграция, REST API

Не нашли что искали?



Задать вопрос в техническую поддержку