OAuth 2.0

# Предварительно

Версия платформы не ниже **7738**.

Версия модуля БД **std:security** – не ниже **7675**

# Общая архитектура

Используется протокол авторизации OAuth 2.0.

Все запросы выполняются от имени какого-нибудь пользователя. Чтобы выполнить любой запрос нужно сначала получить токен доступа. URL для получения токена: /oauth2/token.

После получения токена можно выполнять разрешенные запросы к API версии 2.0.

# Настройка web.config

Для работы с OAuth 2.0 нужно настроить параметры авторизации в файле web.config приложения.

Сначала надо добавить описание секции (в начало файла)

<configuration>

<configSections>

<section name="oauth2" type="A2v10.Web.Mvc.OAuth2.Oauth2Section, A2v10.Web.Mvc" />

</configSections>

Далее добавляем саму секцию:

<oauth2 tokenEndpoint="/oauth2/token"

allowInsecureHttp="true"

expireTimeSpan="00:20:00">

</oauth2>

Атрибуты секции:

* tokenEndpoint – url авторизации относительно корня сайта. Может быть любым. По соглашению именно такой, как указано в примере. В дальнейшем считаем, что мы его не меняли.
* allowInsecureHttp – разрешен ли доступ по http (без “s”). Устанавливаем ТОЛЬКО в отладочных средах!
* expireTimeSpan – время жизни токена (часы:минуты:секунды).

# Получение токена

Получение токена выполняется путем POST запроса по адресу {host}/oauth2/token.

ContentType: application/x-www-form-urlencoded

Тело запроса:

* grant\_type: client\_credentials
* client\_id: <идентификатор клиента>
* client\_secret: <секрет клиента>

Система ищет соответствующего пользователя в таблице и возвращает токен доступа.

* Status: 200 OK
* ContentType: application/json;charset=UTF-8

Пример ответа:

{

"access\_token":"n19YIuxM3rVt………………………………HQ",

"token\_type": "bearer",

"token\_expired": 1199

}

Токен будет получен только в том случае, если запрос приходит с указанного IP и соответствующего домена. Настройка доменов и IP выполняется в админке.

# Выполнение запросов

Токен авторизации задается в заголовке запроса Authorization. Префикс токена: Bearer.

Обращение выполняется по пути: /apiv2/{request\_data}. Как правило, все параметры запросов передаются в URL, а результат возвращается в виде JSON.

Сами запросы описываются в части request\_data соответсвующего URL. Предпоследний сегмент request\_data рассматривается как имя действия, а предыдущие сегменты – как путь к файлу model.json в папке \_apiv2 приложения. Последний сегмент рассматривается как идентификатор сущности.

Например, вызов /apiv2/documents/waybill/get/21 найдет действие get в файле /\_apiv2/documents/waybill/model.json, передаст ему 21 в качестве параметра с именем Id и вернет сериализованный результат.

Обратите внимание, что если идентификатор не требуется (например для списка документов), его все равно нужно передавать (можно просто передать 0).