# Аутентификация в API к системе клиринга ОТС деривативов

## Последовательность действий для получения доступа к API

- Создайте приложение для работы с API системы клиринга ОТС деривативов (далее ОТС Система) и запросите своего клиентского менеджера о присвоении приложению client_id и client_secret (уникальный идентификатор вашего приложения и ключ безопасности, в совокупности однозначно идентифицирующие ваше приложение при обращении к API), отправив менеджеру информацию о своем приложении. После проверки и утверждения созданного вами приложения, данному приложению будут выданы учетные данные client_id и client_secret для доступа к запрошенным API Биржи.

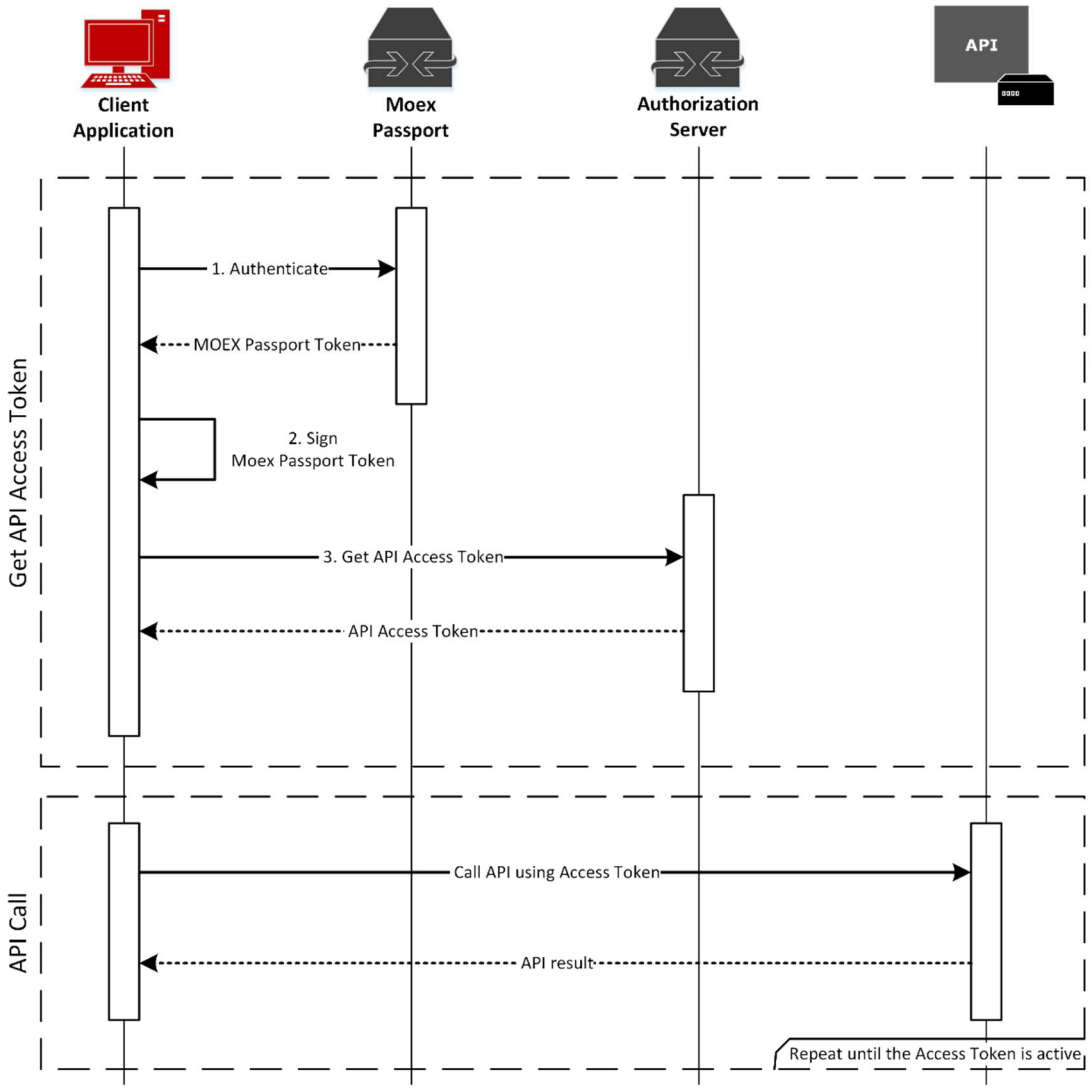
> *Необходимо учитывать, что у пользователя (учётная запись MOEX Passport), от имени которого планируется работать с API, должно быть соответствующее разрешение на использование ОТС системы. Данное разрешение предоставляется через своего клиентского менеджера, путем отправки заявки на доступ пользователя к системе. Участнику клиринга также должен быть выпущен (Удостоверяющим центром Московской биржи) сертификат электронного ключа на имя этого пользователя (владелец сертификата).*

- Реализуйте в своем приложении поддержку протокола OAuth 2.0 и алгоритм получения токена доступа (**access_token** – см. ниже), который вы впоследствии будете использовать при вызове функций API.
- Теперь вы готовы использовать API ОТС системы.

## Что такое OAuth 2.0?

*OAuth - открытый стандарт аутентификации и авторизации. OAuth предоставляет метод доступа клиентов к ресурсам сервера от имени владельца ресурса (такого, как другой клиент или конечный пользователь). Для конечных пользователей он также обеспечивает процесс авторизации доступа третьих сторон к ресурсам их сервера без совместного использования их учетных данных.*

## Общая схема работы с API




## Получение токена доступа

Для аутентификации при вызове методов API необходимо первоначально получить токен доступа, для чего надо выполнить несколько шагов:

1. Получить MOEX Passport Token, выполнив GET запрос по адресу https://passport-test.moex.com/authenticate , используя Basic аутентификацию с учетными данными пользователя, от имени которого предполагается работа с API. Значение Moex Passport Token будет возвращено в куке MicexPassportCert;
2. Используя API СКЗИ Валидата для формирования подписи в форматах ГОСТ, создать отсоединенную электронную цифровую подпись полученного на предыдущем шаге MOEX Passport Token сертификатом пользователя, от имени которого предполагается работы с API. Всю необходимую информацию по работе с СКЗИ вы можете найти на http://moex.com/s1292;
3. Выполнить POST запрос по адресу https://play-api.moex.com/auth/oauth/v2/token, использовав следующие параметры (параметры должны передаваться с использованием метода “application/x-www-form-urlencoded”):
    - **grant_type** – passport

- **scope** – идентификатор API, к которому запрашивается доступ (для системы клиринга ОТС деривативов значение равно *spfi*)
- **client_id** – идентификатор приложения, выданный вашим персональным менеджером
- **client_secret** – ключ безопасности, выданный вашим персональным менеджером
- **certificate** - MOEX Passport Token, полученный на первом шаге
- **algorithm** – GOST
- **signature** – электронная подпись MOEX Passport Token, сформированная на втором этапе, в Base64 кодировке

Если запрос выполнится успешно, вы получите JSON объект со следующими полями:

- **access_token** – токен доступа, который должен передаваться при каждом вызове API
- **token_type** – всегда имеет значение *bearer*
- **expires_in** – время жизни токена доступа в секундах
- **scope** – идентификатор API, для которого действителен полученный токен доступа

В случае, если переданные данные не являются валидными (например, приложение с таким client_id отсутствует, client_secret не соответствует client_id или же переданная электронная подпись не соответствует переданному MOEX Passport токену) результатом буден HTTP Response Code 403

## Использование токена доступа

Теперь, когда у вас есть токен доступа, все, что вам нужно сделать, это использовать его для подписания запросов, отправленных в API.

Вы делаете это, добавляя следующий заголовок к вашим запросам:

Authorization: Bearer <access_token>

В случае, если используемый токен доступа не является валидным или время его жизни истекло, в ответ вы получите HTTP Response Code 401.
При получении ответа с данным кодом ошибки, вы можете повторно запросить токен доступа так, как это описано ранее.