Moscow Exchange

Online client registration for trading at Moscow Exchange

API description

4.10.2023

Table of contents

Connecting API step-by-step

The steps required for using Moscow Exchange API:

- Create a software application to access Moscow Exchange APIs and provide its details to the support manager. Once the software application is verified and approved, we will assign it with our **client_id** and **client_secret** (the unique app ID and security key, the pair will explicitly identify your application when addressing to the API).

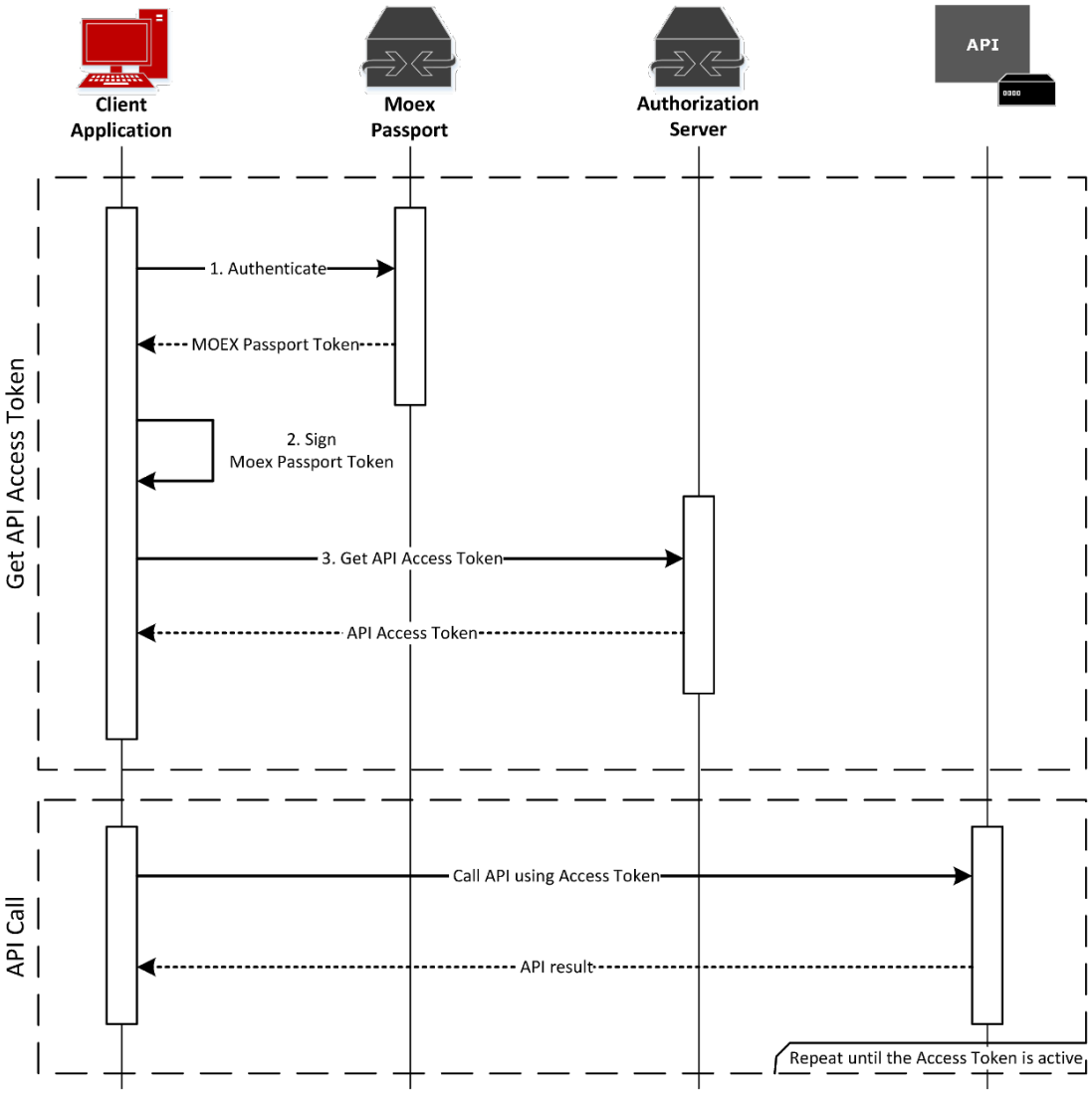
> *Please note that a user intended to work with an API should be granted with the appropriate information system usage approval. To obtain the approval, please apply to your client support manager by sending the appropriate application. In addition, the Moscow Exchange Certification Authority is to issue a token certificate registered to the user (owner of the certificate).*

- In your software application, implement protocol OAuth 2.0 and obtain the access token (for details see below) to call the API's functions.
- Once you are finished with the steps above, you are ready to use the Moscow Exchange API.

What is OAuth 2.0?

OAuth is an open authentication and authorization standard, which provides access to the server resources on behalf the resource owner (another client, or an end user). In addition, OAuth 2.0 provides possibility for end users to set up access to their server resources for third parties, without sharing their credentials.

General scheme of work with API



How to obtain the access token

To work with API, you are required to obtain the access token by following the steps described below:

1. Obtain your MOEX Passport Token by sending request 'GET' to https://passport.moex.com/en/registration using Basic authentication with the appropriate user credentials. The MOEX Passport Token value will return in a cookie named 'MicexPassportCert'.
2. Using VALIDATA data encryption tool (to generate GOST and RSA signatures), or any other data encryption tool which allows to generate RSA signatures), generate a detached digital signature for the obtained earlier MOEX Passport Token using the appropriate user certificate. For more information on work with data encryption tools please refer to http://moex.com/s1292 (available in Russian only);
3. Send 'POST' request to https://sso.moex.com/auth/realms/SSO/protocol/openid-connect/token, using the following parameter settings (all parameters should be sent using method 'application/x-www-form-urlencoded'):

- **grant_type** – password
- **grant_type_moex** – passport
- **scope** – requested access rights (for online registration, the value is *client_registration*)
- **client_id** – software application ID, provided by your manager
- **client_secret** – security key, provided by your manager
- **certificate** - MOEX Passport Token (obtained earlier at step 1)
- **algorithm** – GOST or RSA value, depending on the signature type used on generating MOEX Passport Token signature.
- **signature** – MOEX Passport Token Base64 digital signature, obtained earlier at step 2.

On successful request, the system returns a JSON object containing the following fields:

- **access_token** – API access token to be used on every API call
- **expires_in** – access token lifetime in seconds
- **refresh_expires_in** – refresh token lifetime in seconds
- **refresh_token** – refresh token. A token using to refresh your access token
- **token_type** – always *Bearer*
- **not-before-policy –** indicates if policy of not using the token before proper time of creation is active ('0' – the policy is inactive)
- **session_state –** identifier of authenticated session
- **scope –** granted access rights

In case of invalid data sent (invalid client_id, or client_secret does not match the client_id, or the digital signature sent does not match the obtained token), the system returns **HTTP Response Code 403**.

How to use access token

Once you get your access token, you are able to use the token to sign requests you send to the API.

To do that, add the following header to your requests:

Authorization: Bearer <access_token>

If your access token is invalid, or expired, the system returns **HTTP Response Code 401**. Once you get the response, you are able to repeat the request for access token following the method described above.

API Online registration description

The API is based on RESTful API, and uses the standard HTTP methods. The following two operations are now supported:

1. POST https://apim.moex.com/client/v1/applications/- Send client registration data. The request body format is identical to that of the existing client registration file format https://www.moex.com/a3361.
   In each request, please use HTTP header Content-Type with value 'application/xml'.

Return codes:

- 202 – your request has successfully registered. Please find the request processing status in HTTP header ‘Location’
- 503 – your request sent during off hours and was not registered
- 429 – maximum number of request exceeded. Please try again in 30 seconds.
- 400 – invalid request format. Please find details on the error in the reply message body.
- 500 – other errors. Please find details on the error in the reply message body.

2. GET https://apim.moex.com/client/v1/applications/{DOC_DATE}/{DOC_NUM}- Obtain request processing status, where:
   - DOC_DATE – client registration request date
   - DOC_NUM – unique identification number of the client registration request

   The reply body format is identical to that of the existing client registration file format https://www.moex.com/a3361.

Service restrictions and availability on production

API up time - 09:30:00 (MSK) – 23:30:00 (MSK)

Maximum number of requests – 1 request per 1 second

Request processing time – no longer than 5 minutes

Maximum request payload size – 1 Mb