2023-10-27

Connection to the online client registration test environment

How to obtain your access token

To work with API, one should first obtain their access token. To do that, please follow the steps described below:

1. Create a test account at https://passport-test.moex.com/en/registration. Obtain a MOEX Passport Token by sending request 'GET' to https://passport-test.moex.com/authenticate using Basic authentication, with the appropriate user credentials. The MOEX Passport Token value will return in cookie named 'MicexPassportCert'.

2. Write an email with header 'Testing client online registration' to help@moex.com.
   The email should contain:
   - The email address used for registration at MOEX Passport (see step 1);
   - ID of the firm you use to work at test environment INET_GATEWAY/INETCUR_GATEWAY (if any). If there is no such a firm yet, the appropriate record will be added;
   - Desired certificate type (RSA/GOST) for step 3.

   The reply email will contain:
   - Client test certificate for step 3;
   - **client_id, client_secret** for step 4;
   - Credentials to connect to INET_GATEWAY and INETCUR_GATEWAY, if needed.

3. For GOST digital signatures get the Validata of 6.0 version software from http://moex.com/s1292 (page available in Russian only):
   - Download the appropriate 32 or 64 bit version of Certificate library ZCS. Install it without enabling the TLS component, include the registry storage component if needed. Reboot.
   - Download and install Validata CSP. Start Validata CSP.
   - Start the Certificate library. When started for the first time, chose to recover from a backup and point to the Spr folder of the package received in step 2 above. Or just copy the contents of that folder to C:\Users\<USER>\AppData\Roaming\Validata\zcs\. During the following use select the test certificate. The Certificate library will prompt to insert a certificate media during the next runs. To prepare this media copy the contents of the vdkeys folder to the root of a USB stick or a virtual diskette drive.
   - Use the command line certificate utilities provided with Certificate library and stored at C:\Program Files\Validata\zpki by default. For GOST certificates use the zpki1utl tool.
   - Create a detached digital signature:
     zpki1utl.exe -profile <User> -sign -detached -data <token> -out token.p7d

For RSA digital signatures get the Validata software from http://moex.com/s1292 (page available in Russian only):
   - Download and install the appropriate 32 or 64 bit Certificate library RCS, version 6.0 or newer.
   - Start the Certificate library. When started for the first time, chose to recover from a backup and point to the Spr folder of the package received in step 2 above. Or just copy the contents of that folder to C:\Users\<USER>\AppData\Roaming\Validata\rcs\. During the following use select the test certificate. On following runs select the UserOrg.rsa

key.

- Download and unpack the command line certificate utilities from http://fs.moex.com/cdp/po/rpkiutlv6.zip. For RSA certificates use the rpki1utl* (32 or 64 bit version) tool.
- Create a detached digital signature:

  rpki1utl.exe -profile <User> -sign -detached -data <token> -out token.p7d

4. Encode the resulting token in base64 using any tool or a function in your programming language. For example, with a utility shipped with MS Windows:

   certutil -encode token.p7d token.sig

   Note that the final base64 signature must not contain any and of line symbols! So, when using a tool that doesn't have an option to disable EOLs (such as the mentioned certutil) all the EOL symbols must be stripped off.

5. Send request 'POST' to https://sso2.beta.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, obtained at step 2
   - **client_secret** – security key, obtained at step2
   - **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, obtained at step 3.
   - **signature** – MOEX Passport Token Base64 digital signature, obtained at step 4

General algorithm description of the two-factor authentication:

MicexPassportCert = Authenticate_MOEX_Passport (login, password) ;

Signature = Validata.create_EDI (**MicexPassportCert** , CLIENT_CERTIFICATE);

access_token = HTTP.post (url=https://sso2.beta.moex.com/auth/realms/SSO/protocol/openid-connect/token, parametes= {

grant_type= "password", grant_type_moex= "passport", scope "client_registration",client_id=Client_id, client_secret=Client_secret, certificate= **MicexPassportCert**, algorithm="GOST"|"RSA", signature=**Signature**

}

);

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**.

Using 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://play-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://play-apim.moex.com/client/v1/applications/{DOC_DATE}/{DOC_NUM} — Obtain request processing status, where:
   - DOC_DATE – client registration request date
   - DOC_NUM - client registration request unique identification number

   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 test environment

- API up time on test environment - 11:00 (MSK) – 16:00 (MSK)
- Maximum number of requests – 1 request per 1 second
- Test environments for registration are INET_GATEWAY (Securities market), INETCUR_GATEWAY (FX Market), T1 (Derivatives market)
- Maximum request payload size – 1 Mb

System requirements for client side

- When using VALIDATA data encryption tool:
    - Windows 7 operating system, or newer.
    - In your web browser settings, turn off protocol tls ver 1.0 support. Instead, turn on tls ver 1.1 and/or tls ver 1.2 protocol support.

To check the network availability, send telnet request to https://passport-test.moex.com and https://play-apim.moex.com/ with port settings 443/tcp

- HTTPS requests should be allowed in your local network.