finexer

API Reference Support Center

Dashboard

Using OAuth to connect Finexer accounts

The fastest and simplest way to get access to other businesses' and individuals' Finexer accounts and take an action on their behalf.

To access a Finexer account requires the account owner to give a permission to do so. Here we will guide you through all the steps needed to access data from other Finexer accounts. We use industry-standard OAuth protocol for such authorizations.

The OAuth connection flow

The following steps describe entire OAuth connection flow for a user:

1. Starting on your site, the user clicks a link or button that takes them to the Finexer OAuth flow, passing along your platform's `client_id`.
2. On Finexer's website, the user provides the necessary information and grants your platform permission to access their Finexer account.
3. The user is then redirected back to your site, passing along either an authorization code or an error in case the user canceled the flow.
4. Your site then makes a post request to Finexer's OAuth token endpoint to complete the connection and fetch the user's access token.

After all above steps have been completed, your platform can make API requests on behalf of the user using the obtained access token.

Step 1: Create the OAuth link

To get started with your integration, you need some information from your **platform settings**:

- Required parameter `client_id`, represents a unique identifier for your platform.
- Optional parameter `response_type`. If passed, a value of which must be `code`.
- Optional parameter `redirect_uri`, a page on your website to which the user is redirected with either successful or failed response. The redirect_uri must match one of redirect urls declared at your app. If redirect_uri is not passed, default redirect url is used.
- Optional parameter `scope`. You may pass multiple scopes via this parameter in CSV format or separated with space. Passed permissions must be within the range of permissions that are selected in your app.
- Optional parameter

`state`. We strongly recommend passing along a unique token as the value to prevent CSRF attacks. We'll include the state you gave us when we redirect the user back to your site.

Once you have all in hand, you're ready to create the OAuth link:

```
https://finexer.com/oauth/a
client_id=app_XBl7BcXJAXYKI
```

Step 2: User connects their account

After the user clicks the link on your site, they are taken to Finexer's website where they are prompted to accept or deny the access of your platform. In case if the user does not have an account with Finexer, they are prompted to create one and continue with authorization flow in the end.

Step 3: User is taken back to your site

After the user approves or

denies the access request to thier account, they are redirected back to your site, to one of your redirect urls specified in your platform. If the user approves the access request, we'll pass along in the url:

- `code`, the value of which is an authorization code. It's a short-lived token that can be used only ones.
- `state`, the value of which is a unique token provided in **Step 1**.

```
https://yourplatformwebsite
code={AUTHORIZATION_CODE}
```

If the user denies the access request, we'll pass instead an error in the url:

```
https://yourplatformwebsite
error=access_denied&error_
```

Step 4: Receive access token

To receive access token, exchange the authorization code received in **Step 3** for access token. The following body parameters (Content-

Type: application/x-www-form-urlencoded) are needed for this request:

- `code`, the value of which is an authorization code.
- `client_id`, represents a unique identifier for your platform.
- `client_secret`, secret API key taken from **API Key** page.
- `grant_type`, a value of which must be `authorization_code`.

```
curl -X POST
https://finexer.com/oauth/t
\
     -d code="
{AUTHORIZATION_CODE}" \
     -d client_id="
{APP_ID}" \
     -d client_secret="
{SECRET_API_KEY}" \
     -d
grant_type="authorization_c
```

Finexer responds to this request by returning a JSON object. For successful requests the following object is returned:

```
{
    "token_type":
"bearer",
    "access_token": "
{ACCESS_TOKEN}",
    "refresh_token": "
{REFRESH_TOKEN}",
    "account_id": "
{ACCOUNT_IDENTIFIER}",
    "scope":
"customer_read,customer_wr
}
```

For unsuccessful requests, we instead return an error:

```
{
    "error":
"invalid_grant",
    "error_description":
"The authorization code
was not found in the
system:
{AUTHORIZATION_CODE}"
}
```

You are done! Now, your platform has access to another Finexer account. You can retrieve data, initiate payments and manage connected account on their behalf. You should keep access token along with the refresh token safe. You can keep account identifier as a reference to the account you manage, and associate it with your client in cases when token is revoked. Use received access token in requests to the Finexer API using the "Authorization: Bearer {access_token}". The following example demonstrates how to retrieve customers from an account approved your platform:

```
curl -X GET
https://api.finexer.com/cus
\
     -H "Authorization:
Bearer {ACCESS_TOKEN}" \
```

Revoking a token

In some cases, if you or Finexer account holder wish to revoke previously granted access to their account, this can be done anytime in a few ways:

- An account holder can revoke access given to your platform from their account in **Authorized Apps** page.
- You can disconnect your platform from a managing account in **Managed Accounts** page.
- You can programmatically revoke access doing the following POST request to the API:

```
curl -X POST
https://finexer.com/oauth/r
\
     -d client_id="
{APP_ID}" \
     -d client_secret="
{SECRET_API_KEY}" \
     -d refresh_token="
{REFRESH_TOKEN}"
```

Performing the above request invalidates access and refresh tokens. For successful requests, 204 HTTP status code returned and a webhook is sent with event of the type `oauth.authorization.revoked`. For unsuccessful requests, a HTTP status code 400 is returned along with JSON object describing an error.