

**DOCUMENTATION**[Getting Started](#)**SECURITY**[Security](#)**DEVELOPMENT**[App](#)[OAuth](#)[Connect Bank](#)[Initiate Payment](#)[Batch Payments](#)[App Templates](#)[Sandbox](#)[Bank Feed](#)**SETTINGS**[Consent Control](#)[SMTP](#)[Team](#)[Webhooks](#)[Signatures](#)[API Reference](#)[Support Center](#)[Dashboard](#)

Using your app to connect bank accounts

The fastest and simplest way to get access to your clients' bank account data.

To access a bank account of your client requires your client to give a permission to do so. Here we will guide you through all the options and steps needed to connect clients' banks and access their bank account data. Before you continue you need to set up the app in your [app settings](#).

For more information on type of accounts, coverage and status please go to our [Provider Status](#) page.

It's important to understand that you need

to have a license and be regulated by the FCA if you provide account information services. However you can use our services without a need to obtain such license if you're not going to show data back to your users. If you need to show data on your end then we can offer you to be an authorized agent of Finexer.

The bank connection flow

The following steps describe entire connection flow for a user:

1. Starting on your site, the user clicks a link or button that takes them to the Finexer consent page.
2. On Finexer's website, the user selects a bank to connect (unless you've done this through the API) and is redirected to the bank consent page to authenticate themselves.
3. The user is then

redirected seamlessly back to your site, passing along either a consent identifier, or an error in case the user denied the access request.

After all above steps have been completed, your platform can make API requests and access the user's data.

Step 1: Create the consent link

To get started with your integration, you need to create a **consent** object with optional parameters listed below. To find out more about API methods please visit our [API reference](#) page.

Parameter `provider`, represents a provider ID of a **provider** object. If you leave this empty, the user will be given a choice of supported banks to connect at the consent page. If passed in the authorization URL or during creation of a consent object, the user will only need to confirm

your access request with a pre-selected bank.

Parameter `customer`, represents a customer ID of a **customer** object. You need to create first the customer object to associate it with your client and use this object in the future. If you leave this empty, the consent object will be created with the status `new`. However to send your client to the consent page, the consent object should have `pending` status. You can use **update method** to set a customer identifier.

Parameter `return_url`, a page on your website to which the user is redirected after confirming your access. If you leave this empty, we'll use one of your redirect URLs saved in the app. However if the app has no defined URLs, we'll take the user to our own confirmation page.

It's recommended if you append

`state` parameter with a unique token to the return url to prevent CSRF attacks. Also it helps you to establish a session during the entire consent journey so when you receive a callback you can determine who a consent was created for. We retain all your other parameters passed in the return url.

Once a consent object with pending status is created, redirect the user to URL supplied in

`redirect.consent_url` attribute on the consent object:

```
{
  "redirect": {
    "return_url":
    "https://yourwebsite.com/callback/{SESSION_ID}",
    "consent_url":
    "https://finexer.com/connected/consent=cst_6e5P0klsXkpaN7c"
  }
}
```

For the sake of ease we stripped out the rest of attributes on the consent object.

Consent options

By default, the consent screen opens with standard settings configured in your app settings. However you can

override some of the configurations by passing parameters in the query string:

`skip_info` (true, false by default) - when passed `true`, the first screen with beneficiary information e.g. payee, BBAN, amount will be skipped only for mobile users.

`skip_qr` (true, false by default) - when passed `true`, the screen with QR code to open a mobile app will be skipped only for desktop users.

`template` an identifier of an existing white-label template - when passed, the consent screen will use its pre-configured UI settings. For more information, check out [App Templates](#).

Step 2: User connects their bank

After you redirect the user to the supplied URL, the user is prompted to select a bank they wish to connect (unless a bank

is already set on consent object in provider attribute), the user then is taken to a selected bank's consent page (e.g. Barclays, Lloyds) to authenticate the user and approve or deny the access of our platform. Note the user can deny the access request at the consent page at Finexer as well as at the consent page at a bank. When user denies access, the status changes to `canceled` on the consent object.

Step 3: User is taken back to your site

After the user approves or denies the access request to their bank, they are redirected back to the URL that was set in a consent object. If the user approves the access request, we'll pass along in the url:

`fx_consent`, the value of which is the ID of the `consent` object and other parameters that were provided in **Step 1** (e.g. `state`).

```
https://yourwebsite.com/ca]
```

```
fx_consent=bc_mtoK6Z9ZB&sta
```

Now, your platform has access to the user's bank account data such as accounts, balance, etc.

At this stage, the **consent** object has the status

```
authorized
```

. To access the user's data you can only pass the consent ID to the API endpoints to retrieve data associated with this consent.

To retrieve accounts for a customer, you can provide the customer ID instead of the consent ID. This will return all bank accounts associated with that customer, including those from both past and recent consent authorizations.

The following example demonstrates how to retrieve bank accounts from the user that approved your request:

```
curl -X POST
https://api.finexer.com/ban
consent=bc_mtoK6Z9ZB \
-u "API_KEY"
```

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

```
https://yourwebsite.com/cal
error=access_denied&error_c
```


Revoking a consent

In cases when you or your client wish to revoke previously created consent, this can be done anytime in your dashboard or by hitting the [revoke consent](#) endpoint.

We are here to help

Need help getting started or ready to test your configuration? Contact our team at the [Support Center](#) or at [contact us](#) page. You can also email us at support@finexer.com