FinCalc API Authentication Guide

Contents

Introduction

The Authentication API supports two flows:

1. **Authorization code flow** (standard OAuth2 specification)
2. **Client credentials flow** (custom implementation for server-to-server authentication)

Getting access to our API

Currently, there is no automated process to gain access to the FinCalc API.

To get access, please email support@fincalc.co.uk and provide the following information:

- Your name
- Company name
- Authentication type you require (see above)
- Redirect URLs needed (if using authorization code flow) (For example https://myapp.com/auth/callback)

Scopes available

Scopes are used to specify the level of access your application is requesting. When a user authorizes your app, they consent to the requested scopes.

Scope	Description
client-information	Allows access to client information.
client-financials	Grants access to view household financial data.
fact-find	Enables access to retrieve fact-find data related to a client.

Authorization code flow

This endpoint allows you to authenticate using the authorization code flow, a standard OAuth2 specification that facilitates secure user-based authentication. This flow is ideal for scenarios where your application needs to securely obtain user authorization to access their data on the FinCalc API.

How it works

- Your application redirects the user to the authorization endpoint to obtain their consent.
- After the user authorizes, they are redirected back to your redirect URI with an authorization code.
- Your application exchanges the authorization code for an access token by making a request to the token endpoint.
- Upon successful authentication, the system returns an access token and refresh token.

Step 1: Authorization request

Direct the user to the authorization endpoint to obtain an authorization code.

GET /oauth/authorize

Request

Parameter	Type	Required	Description
response_type	string	Yes	Must be ‘code’.
client_id	string	Yes	Your application's client ID.
redirect_uri	string	Yes	The URL to which the user will be redirected.
scope	string	Yes	A space-separated list of scopes for the requested access.
state	string	No	A unique, unguessable value to prevent CSRF attacks. Recommended.

Request example

```
GET /oauth/authorize?response_type=code&client_id=your-client-id&redirect_uri=https%3A%2F%2Fyourapp.com%2Fcallback&scope=client-information%20client-financials%20fact-find&state=xyz123
HTTP/1.1
Host: api.fincalc.com
```

User redirect

If the user approves the request, they will be redirected to the specified ‘redirect_uri’ with the following query parameters:

Parameter	Type	Description
code	string	The authorization code to exchange for a token.
state	string	The state value sent in the initial request (if provided).

Redirect example:

```
https://yourapp.com/callback?code=AUTH_CODE_HERE&state=xyz123
```

Error responses

HTTP Status	Error Code	Description
400	invalid_request	Missing or invalid parameters.
400	unsupported_grant_type	The authorization grant type is not supported by the authorization server.
400	invalid_scope	The requested scope is invalid, unknown, or malformed.
400	Invalid_tenant	The logged in user’s company is not a valid tenant for the provided client id.
401	access_denied	The request has not been granted permission to access the resource.
401	invalid_client	Invalid ‘client_id’, ‘client_secret’.

Error example

```
{
   "error": "invalid_request",
   "error_description": "The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed.”
   “hint”: “Cannot decrypt the authorization code”
   “message”: “The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed.”
}
```

Step 2: Token request

Once you have the authorization code, exchange it for an access token by making a request to the Token Endpoint.

POST /oauth/token

Request

Parameter	Type	Required	Description
grant_type	string	Yes	Must be ‘authorization_code’.
client_id	string	Yes	Your application's client ID.
client_secret	string	Yes	Your application's client secret.
redirect_uri	string	Yes	Must match the redirect URI used in the authorization request.
code	string	Yes	The authorization code obtained in step 1.

Headers

Header	Value	Required	Description
Content-Type	application/x-www-form-urlencoded	Yes	Indicates that the request body is formatted as key=value pairs.

Request example

```
POST /api/oauth/token
HTTP/1.1
Host: api.fincalc.com
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&client_id=your-client-id&client_secret=your-client-secret&redirect_uri=https%3A%2F%2Fyourapp.com%2Fcallback&code=AUTH_CODE_HERE
```

Response

On success, the endpoint returns a JSON object containing the access token and optional refresh token.

Field	Type	Description
token_type	string	The type of token issued (e.g., ‘Bearer’).
expires_in	number	The token's validity duration in seconds.
access_token	string	The access token to authorize subsequent requests.
refresh_token	string	A token that can be used to refresh the access token. This does not expire.

Response example

```
{
  "token_type": "Bearer",
  "expires_in": 86400,
  "access_token": "ACCESS_TOKEN_EXAMPLE”
  "refresh_token": "REFRESH_TOKEN_EXAMPLE"
}
```

Error responses

HTTP Status	Error Code	Description
400	invalid_request	Missing or invalid parameters.
400	invalid_grant	The authorization code has expired or is invalid.
400	unsupported_grant_type	The authorization grant type is not supported by the authorization server.
401	invalid_client	Invalid ‘client_id’, ‘client_secret’, or authorization code.

Error example

```
{
   "error": "invalid_request",
   "error_description": "The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed.”
   “hint”: “Cannot decrypt the authorization code”
   “message”: “The request is missing a required parameter, includes an invalid parameter value, includes a parameter more than once, or is otherwise malformed.”
}
```

Notes

- Tokens are valid for 24 hours. Use the ‘refresh_token’ to obtain new access tokens without user intervention.
- Ensure the ‘redirect_uri’ in the token request matches the one used during the authorization request.
- ‘state’ is optional but strongly recommended to mitigate CSRF attacks. Always validate it upon receiving the redirect.
- Always keep your ‘client_id’ and ‘client_secret’ secure.

Refreshing the access token

Once the access token expires, you can use the refresh token to obtain a new access token without requiring the user to reauthorize. This helps maintain a seamless user experience.

POST /oauth/token

Request parameters

Parameter	Type	Required	Description
grant_type	string	Yes	Must be ‘refresh_token’.
refresh_token	string	Yes	The refresh token obtained during the initial token request.
client_id	string	Yes	Your application's client ID.
client_secret	string	Yes	Your application's client secret.
scope	String	No	A space-separated list of scopes for the requested access.

Headers

Header	Value	Required	Description
Content-Type	application/x-www-form-urlencoded	Yes	Indicates that the request body is formatted as key=value pairs.

Request example

```
POST /api/oauth/token
HTTP/1.1
Host: api.fincalc.com
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token&refresh_token=your-refresh-token&client_id= your-client-id
&client_secret=your-client-secret&scope=your-scopes
```

Response

On success, the endpoint returns a JSON object containing the access token and optional refresh token.

Field	Type	Description
token_type	string	The type of token issued (e.g., ‘Bearer’).
expires_in	number	The token's validity duration in seconds.
access_token	string	The access token to authorize subsequent requests.
refresh_token	string	A token that can be used to refresh the access token. This does not expire.

Response example

```
{
  "token_type": "Bearer",
  "expires_in": 86400,
  "access_token": "ACCESS_TOKEN_EXAMPLE”
  "refresh_token": "REFRESH_TOKEN_EXAMPLE"
}
```

Error responses

HTTP Status	Error Code	Description
400	invalid_request	Missing or invalid parameters.

400	unsupported_grant_type	The authorization grant type is not supported by the authorization server.
400	invalid_grant	The refresh token is expired, revoked, or invalid.
401	invalid_client	Invalid client_id or client_secret.

Error example

```
{
  "error": "invalid_grant",
  "error_description": "The provided refresh token is expired, revoked, or invalid.",
  "hint": "Check if the refresh token is still valid or has been revoked.",
  "message": "The provided refresh token is expired, revoked, or invalid."
}
```

Notes

- A refresh token may expire or be invalidated if it is used multiple times or if the user revokes authorization.
- Always securely store the refresh token as it allows long-term access to the user's resources.
- When you receive a new refresh token in the response, replace the old one with the new one.

Client credentials flow

This endpoint allows you to authenticate and obtain an access token using the client credentials flow, a secure method for server-to-server authentication.

How it works

- Provide the email address associated with your account in the FinCalc app, along with your client credentials (client ID and client secret).
- Include the ‘grant_type’ parameter as ‘client_credentials’ and specify the required scope(s) as a space separated list.
- Upon successful authentication, the system will return an access token.
- This access token is valid for 24 hours.
- Once expired, you must request a new token by re-authenticating.

Authorization request

Endpoint

POST /api/oauth/token

Request

Parameter	Type	Required	Description
client_id	string	Yes	Your application's client ID.
client_secret	string	Yes	Your application's client secret.
grant_type	String	Yes	Must be ‘client_credentials’.
scope	String	Yes	A space separated list of scopes for the access token.
email	string	Yes	The email address used to log in to the FinCalc app.

Headers

Header	Value	Required	Description
Content-Type	application/x-www-form-urlencoded	Yes	Indicates that the request body is formatted as key=value pairs.

Request example

```
POST /api/oauth/token
HTTP/1.1
Host: api.fincalc.com
Content-Type: application/x-www-form-urlencoded

{
  "client_id": "your-client-id",
  "client_secret": "your-client-secret",
  "grant_type": "client_credentials",
  "scope": "client-information client-financials fact-find"
  "email": user@example.com
}
```

Response

On success, the endpoint returns a JSON object containing the access token and additional details:

Field	Type	Description
success	Boolean	Indicates whether the request was successful.
data	Object	Contains the access token details
token_type	String	The type of token issue (e.g. ‘Bearer’)
expires_in	Number	The token’s validity duration in seconds.
access_token	String	The access token to authorize subsequent requests.

Response example

```
{
  "success": true,
  "data": {
    "token_type": "Bearer",
    "expires_in": 86400,
    "access_token": "ACCESS_TOKEN_EXAMPLE"
  }
}
```

Error responses

HTTP Status	Error Code	Description
400	invalid_request	Required parameters are missing or invalid.
401	unauthorized	Invalid credentials or email address.
403	access_denied	Your account does not have the required permissions to access the endpoint.

Error example

```
{
 "error": "invalid_request",
 "error_description": "The email address or credentials provided are invalid."
}
```

Notes

- The ‘expires_in’ field specifies that the token will expire in 86400 seconds (24 hours).
- Tokens are scoped and include specific permissions (e.g., ‘client-information’, ‘fact-find’).
- If your email address changes in the FinCalc app, update the authentication process accordingly.
- Always keep your ‘client_id’ and ‘client_secret’ secure.

Using the access token

Once you obtain the access token, include it in the Authorization header of your API requests.

Example API request

```
GET /api/v2/clients
HTTP/1.1
Host: api.fincalc.com

Authorization: Bearer ACCESS_TOKEN_EXAMPLE
```

Document revision

Version	Date	Amendment	Author
1.0	18/12/2024	Initial release	DG