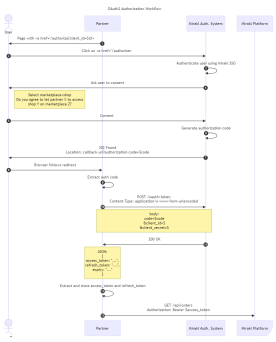
Integrating with OAuth 2 Authorization Code

As an integrator, you want to call Mirakl or Mirakl Connect APIs. You must integrate with Mirakl Authentication System using the OAuth 2 Authorization Code protocol.

OAuth 2 Authorization Code workflow diagram



About the OAuth 2 Authorization Code workflow

Mirakl Authentication System implements this OAuth 2 Authorization Code workflow in conformance with the OAuth 2 standard. Use any compliant library to access it.

To integrate with Mirakl Authentication System, you must perform the following steps.

1 - Contact Mirakl teams to get the client ID and client secret information

1. You must first contact Mirakl to explain your objectives.
   You must provide the following information:
   - The type of the integration you want to develop:
     - **Marketplace operator**: if you want to target Marketplace operators. Specify if you need access to the Marketplace information.
     - **Marketplace seller**: if you want to target users who have a store on a Marketplace or a specific store. Specify if you need access to the store list.
     - **Mirakl Connect seller**: if you want to target Mirakl Connect sellers. Specify if you need access to the company information.
   - The redirect URI:
     Provide a redirect URI (the OAuth 2 callback endpoint -including the tenant) to know from which platform the user comes. You can provide multiple redirect URIs if you need them for your integration.
     For example: `https://backoffice.partner.com/auth/mirakl?tenant=<tenant ID>`
     Users click this URL to access your application.
   - The APIs you want to call
2. Mirakl creates an account with the OAuth 2 Authorization Code feature enabled and gives you the client ID and the client secret that allow you to build the OAuth 2 Authorization Code integration.
   It provides you with the list of APIs to which you are granted access.

Client ID

This is a unique generated string, and is your unique identifier.

This data is not sensitive and will later appear in various places, like in URLs, during calls.

This data is stored in the Mirakl Authentication System. It can be retrieved after it has been issued.

Client secret

This is a unique string. This sensitive data must be handled with particular care.

This data is not stored in the Mirakl Authentication System once it has been given to you.

⚠ Store your client secret safely. If you lose it, you can get a new one from Mirakl, however, it will invalidate the active tokens.

ℹ Mirakl has always the option to revoke your authorization access at any time.

2 - Add a "Sign in with Mirakl" button to your application

You must add a "Sign in with Mirakl" button to your back office.

ℹ **"Sign in with Mirakl" button look and feel**

Mirakl recommends that you develop the "Sign in with Mirakl" button according to the following guidelines. This allows Mirakl users to easily identify in your back office how they can access your application.

>> To develop the button according to Mirakl guidelines, download the material from this link (logo and CSS) <<

ℹ View the GitHub repository that contains samples of how to obtain your access tokens.

The "Sign in with Mirakl" button is a link that has the following structure:

```
https://<url>/authorize?
    client_id=<your-client-id>              #required
    &redirect_uri=<your-redirect-uri.com>   #optional
    &audience=<filter>                      #optional
    &state=<arbitrary-value>                #optional
```

`<url>` must take one of following three values according to the target environment:

- `auth-dev.mirakl.net` for development environments

- `auth-preprod.mirakl.net` for pre-production environments
- `auth.mirakl.net` for production environments

The following query parameters are available:

client_id

Required parameter

This is a unique generated string, and is your unique identifier.

This data is not sensitive and will later appear in various places, like in URLs, during calls.

This data is stored on the Mirakl Authentication System. It can be retrieved after it has been issued.

redirect_uri

Optional parameter

This parameter indicates where a user is to be redirected after consenting to share his/her information.

If not specified, the first configured `redirect_uri` provided during the account creation is used.

You can only specify a URL that is already in your client configuration, and the values must be exact matches (for example, you cannot add extra query parameters). Refer to the `state` parameter if you need to specify dynamic values.

audience

Optional parameter

Use this parameter to only authorize certain users:

- If the integration type is Marketplace operator, `audience` can be used to filter users of a specific Marketplace (`audience=<tenant-ID>`).
- If the integration type is Marketplace seller, `audience` can be used to filter users who have a store on this Marketplace (`audience=<tenant-ID>`) or a specific store (`audience=<tenant-ID>:<shop-ID>`).
- If the integration type is Mirakl Connect seller, `audience` cannot be used.

> To know the possible values for the `audience` parameter, contact the Support Team. Ask for the list of "tenant-IDs".

For example, you can develop a button that allows sellers to connect to the ACME Marketplace only. The button can then display "Connect to ACME Marketplace".

To do so, add the `audience` parameter in the call, as follows:

```
1 https://auth.mirakl.net/authorize?client_id=<the-client-id>&
```

audience=acme-prod

state

Optional parameter

This parameter allows you to store arbitrary data and later retrieve it.

> We strongly advise you to:
>
> - encrypt the value here so that end-users are not able to alter the value.
> - store the value in a cookie under your domain name so you can check that the `state` value transmitted by Mirakl Authentication System is the state value originally provided. Refer to https://auth0.com/docs/secure/attack-protection/state-parameters

Use cases:

Secure the workflow against CSRF (Cross-Site Request Forgery) attacks

1. Generate a random and unique value.
2. Store this value as a cookie.
3. Depending on your use case, put the same value in the URL of the **"Sign in with Mirakl"** button: `https://<url>/authorize?client_id=<your-client-id>&state=<arbitrary-value>`.
4. When users consent, they are sent to the redirect URI: `https://<your-redirect-uri.com>?code=<authorization-code>&state=<same-value-you-specified>`.
5. Make sure that the value in the `state` query parameter matches the value you stored in the cookie. If these values do not match, the workflow must be aborted.

Keep information (between the steps of the workflow)

When users click the **"Sign in with Mirakl"** button, they are sent to the redirect URI:

`https://<url>/authorize?client_id=<your-client-id>&state=<arbitrary-value>`

The `<arbitrary-value>` can contain the information you want to retrieve later (for example, "userId=2347893" if you need to retrieve the user id in your system).

Combine the two methods

1. Generate a random and unique value.
2. Store this value as a cookie.

3. Store the context information in the cookie.
4. Send users to: `https://<url>/authorize?client_id=<your-client-id>&state=<arbitrary-value>`.
5. When users consent, they are sent to the redirect URI: `https://<your-redirect-uri.com>?code=<authorization-code>&state=<same-value-you-specified>`.
6. Make sure that the value in the `state` query parameter matches the cookie you stored. If these values do not match, the workflow must be aborted.
7. The cookie contains the context information you need to finish the workflow.

What happens when the user clicks the "Sign in with Mirakl" button?

When a user clicks the "Sign in with Mirakl" button, he/she is redirected to the Mirakl authentication page.

- If the user has an active Mirakl SSO session, he/she is automatically logged in.
- If the user's session has expired, the user has to fill in his/her Mirakl SSO credentials.
- If the `audience` parameter is set, Mirakl Authentication System checks the user's rights before logging the user in.

Mirakl Authentication System first selects the target resource:

- If the integration type is Marketplace operator, and if the user is present on multiple Marketplaces, the user is requested to select a target Marketplace.
- If the integration type is Marketplace seller, the user is requested to select a store on the target Marketplace.
- If the integration type is Mirakl Connect seller, nothing is required.

Then, Mirakl displays a consent dialog to the users asking them if they really want to authorize your application to access the selected resource (for example, "inventory update" and "order management" APIs).

3 - Implement the redirect URI endpoint

After users consent to sharing their information, Mirakl Authentication System generates an authorization code.

Mirakl sends a redirect to the user's browser pointing to the `redirect_uri` you provided, adding the generated authorization code as a query parameter named `code`:

```
1  302 Found
2  Location: <redirect-uri>?code=<your_authorization_code>
```

Retrieving the token

From your `redirect_uri` you must:

1. Extract the provided authorization code.
2. Call Mirakl Authentication System as follows:

```
POST https://<url>/oauth/token
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code
&client_id=<your-client-id>
&client_secret=<your-client-secret>
&code=<your-authorization-code>
&redirect_uri=<your-redirect-uri.com>
```

> **Caution**
>
> The Mirakl Authentication System does not support using query parameters.

`<url>` must take one of following three values according to the target environment:

- `auth-dev.mirakl.net` for development environments
- `auth-preprod.mirakl.net` for pre-production environments
- `auth.mirakl.net` for production environments

Fields description

Field	Description
Endpoint	POST /oauth/token
Form-body parameters	• `grant_type`: (Required) must be `authorization_code` • `client_id`: (Required) your client ID • `client_secret`: (Required) your client secret • `code`: (Required) the authorization code you retrieved previously • `redirect_uri`: (Optional) the redirect URI you provided when you created your account
Response	200 with JSON response

If a `state` parameter was specified in the initial URL, the same value will be present in the URL under the `state` parameter. You can extract this `state` value from the URL, and decode it to restore the context. The value must be compared with the value previously stored in a cookie (as explained in the `state` parameter description). If the values differ, you must abort the process.

When you call Mirakl Authentication System, Mirakl validates the client ID and client secret, and checks the authorization code.

If everything works as expected, Mirakl Authentication System generates an access token and a refresh token, and replies with a JSON structured as follows:

```
{
  "token_type": "bearer",
  "access_token": "......",
```

```
  "refresh_token": "......",
  "expires_in": 3600,
            ...
}
```

You must:

1. Store the returned values securely and permanently.
2. Use the access token to access the user's resources.

By default, Mirakl generates access tokens without TTL (Time To Live). You can ask Mirakl teams to configure an expiration duration.

Description of the returned fields:

- `token_type`: Type of the token. In this workflow, the token is always of the "bearer" type.
- `access_token`: The token you need to call the API. The access token can expire if you have been granted temporary access.
- `refresh_token`: The token you must store to generate a new `access_token` once a temporary `access_token` has expired. The refresh token never expires.
- `expires_in`: Number of seconds before the `access_token` expires. By default, there is no expiration time. You can ask MiraklMirakl teams to enable this configuration if needed.
- `resource_owner`: The email address of the user who granted the access (who clicks the **Sign in** button)

The optional fields are different depending on the integration type:

- Marketplace operator
- Marketplace seller
- Mirakl Connect seller

Optional fields for Marketplace operators

Description of the optional fields for Marketplace operators

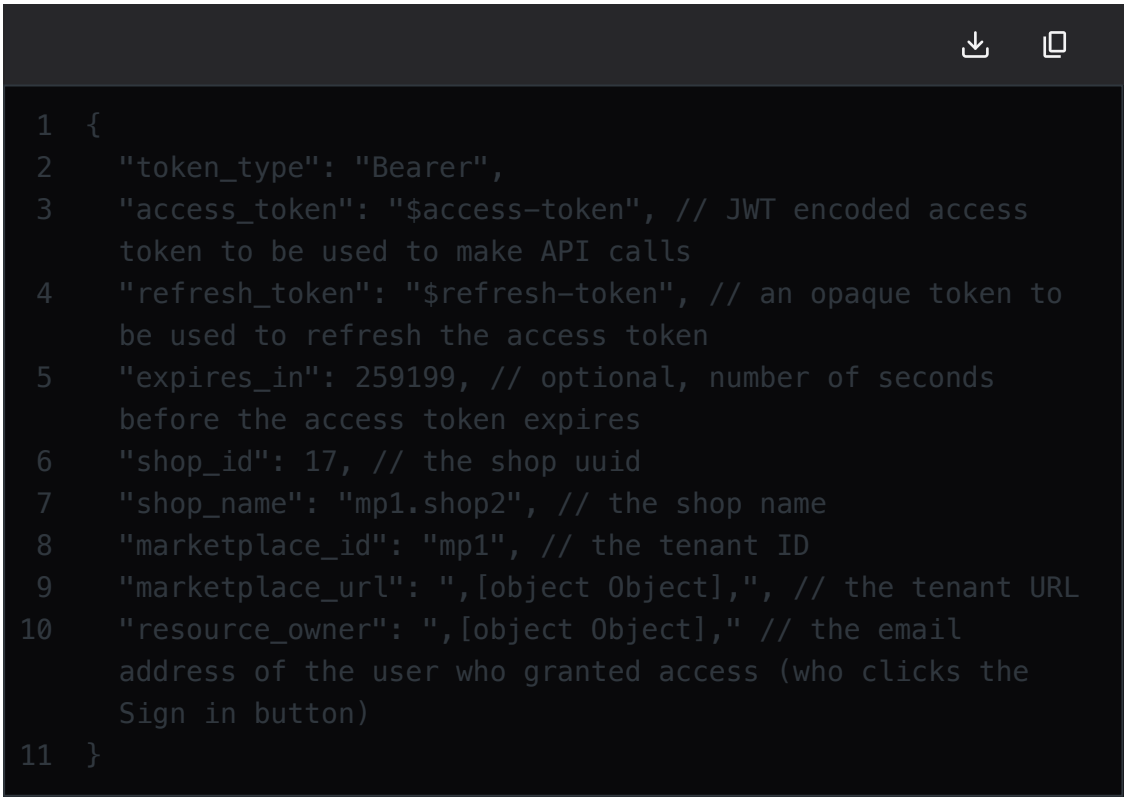
Field	Description
`marketplace_id`	The Marketplace ID (tenant ID)
`marketplace_url`	The URL of the Marketplace to which you have granted access. When integrating with Mirakl Connect, this field is not populated.

```
{
  "token_type": "Bearer",
  "access_token": "$access-token", // JWT encoded access token to be used to make API calls
  "refresh_token": "$refresh-token", // an opaque token to be used to
```

Field	Description
	refresh the access token "expires_in": 3600, // optional, number of seconds before the access token expires "marketplace_id": "mp1", // the tenant ID "marketplace_url": ",[object Object],", // the tenant URL "resource_owner": ",[object Object]," // the email address of the user who granted access (who clicks the Sign in button) }

Optional fields for Marketplace operators

Description of the optional fields for Marketplace operators

Field	Description
`shop_id`	The unique identifier of the selected store
`shop_name`	The selected store's name
`marketplace_id`	The Marketplace ID (tenant ID) When integrating with Mirakl ConnectMirakl Connect, this field is not populated.
`marketplace_url`	The URL of the Marketplace to which you have granted access When integrating with Mirakl ConnectMirakl Connect, this field is not populated.
	

Optional fields for Connect sellers

Description of the optional fields for Connect sellers

Field	Description
`company_id`	The unique identifier of the seller company
`company_type`	Always "SELLER"
`company_name`	The Mirakl Connect company name

```
{
  "token_type": "Bearer",
  "access_token": "$access-token", // JWT encoded access token to be used to make API calls
  "refresh_token": "$refresh-token", // an opaque token to be used to refresh the access token
  "expires_in": 259199, // optional, number of seconds before the access token expires
  "company_id": "c1", // the Mirakl Connect® company id
  "company_kind": "SELLER",
  "company_name": "The Shop Company", // the Mirakl Connect® company name
  "resource_owner": ",[object Object],", // the email address of the user who granted access (who clicks the Sign in button)
}
```

Using the refresh token

Even if there is no expiration date, you can use this refresh token to generate a new access token.

When you have been granted a temporary access token, you must use a refresh token to extend your access to the user API resources.

Use the `refresh_token` communicated to you and repeat the call.

```
POST https://<url>/oauth/token
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token
&client_id=<your_client_id>
&client_secret=<your_client_secret>
&refresh_token=<the_previously_issued_refresh_token>
```

Fields description

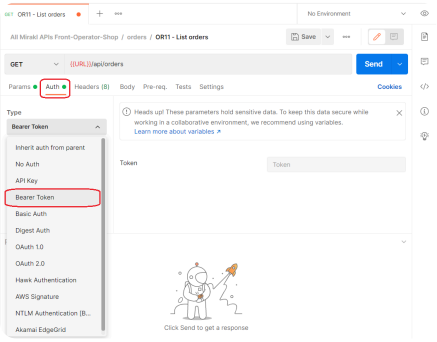
Field	Description
Endpoint	POST /oauth/token

Field	Description
Form-body parameters	• `grant_type`: Required, must be `refresh_token` • `client_id`: Required, your client ID • `client_secret`: Required, your client secret • `refresh_token`: Required, the refresh token already issued
Response	200 with JSON. You will receive another `access_token`. The response has the same format as the response for the access token.

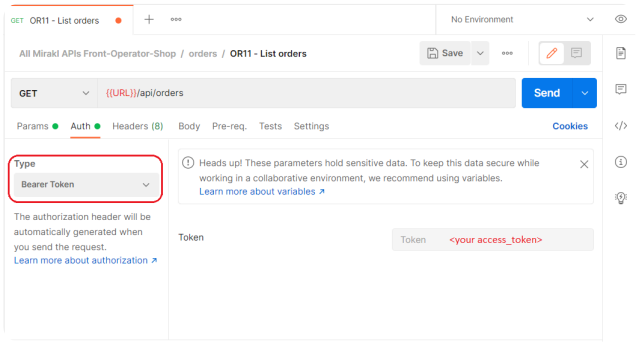
Calling Mirakl APIs

To call the APIs to which you have been granted access:

- Send an HTTP header of the "Authorization" type with the following format: `Authorization: Bearer <access-token>`
- Or, if you use Postman:
  1. Click the **Auth** tab.
  2. Select the type "Bearer Token".



  3. Set its value with the `access_token` you retrieved in the previous step.



Revoking a token

> We highly recommend allowing sellers to revoke their authorization from your platform.

This endpoint is based on OAuth 2.0 Token Revocation RFC 7009 Draft.

```
POST https://<url>/oauth/token
Content-Type: application/x-www-form-urlencoded
```

```
3  Authorization: basic <credentials>
4
5  grant_type=revoke_token
6  &token=<the_access_token_to_revoke>
7  &token_type_hint=access_token
8  &reason=<revocation_reason>
```

You can revoke any access token you own using these parameters:

Fields description

Field	Description
Endpoint	POST /oauth/token/revoke
Header	Authorization with the client ID/client secret pair in basic encoding to authenticate the call
Form-body parameters	• `token`: Required, the access token to be revoked • `token_type_hint`: Optional, only `access_token` is accepted as a value • `reason`: Optional, explain why the token is being revoked. The reason is stored in the token history.
Response	200 with JSON. Callers should ignore the JSON content.

As recommended by the specification, this endpoint returns 200 for the following cases:

- The token was already revoked.
- An invalid token is passed.

⚠ By revoking a token, you lose the access that the users granted you. Users will have to authorize you again to regain this access.