GPT Action authentication

Copy page

Learn authentication options for GPT Actions.

Actions offer different authentication schemas to accommodate various use cases. To specify the authentication schema for your action, use the GPT editor and select "None", "API Key", or "OAuth".

By default, the authentication method for all actions is set to "None", but you can change this and allow different actions to have different authentication methods.

No authentication

We support flows without authentication for applications where users can send requests directly to your API without needing an API key or signing in with OAuth.

Consider using no authentication for initial user interactions as you might experience a user drop off if they are forced to sign into an application. You can create a "signed out" experience and then move users to a "signed in" experience by enabling a separate action.

API key authentication

Just like how a user might already be using your API, we allow API key authentication through the GPT editor UI. We encrypt the secret key when we store it in our database to keep your API key secure.

This approach is useful if you have an API that takes slightly more consequential actions than the no authentication flow but does not require an individual user to sign in. Adding API key authentication can protect your API and give you more fine-grained access controls along with visibility into where requests are coming from.

OAuth

Actions allow OAuth sign in for each user. This is the best way to provide personalized experiences and make the most powerful actions available to users. A simple example of the OAuth flow with actions will look like the following:

> To start, select "Authentication" in the GPT editor UI, and select "OAuth".
>
> You will be prompted to enter the OAuth client ID, client secret, authorization URL, token URL, and scope.

The client ID and secret can be simple text strings but should follow OAuth best practices.

We store an encrypted version of the client secret, while the client ID is available to end users.

OAuth requests will include the following information:

```
request={'grant_type': 'authorization_code', 'client_id': 'YOUR_CLIENT_ID',
'client_secret': 'YOUR_CLIENT_SECRET', 'code': 'abc123', 'redirect_uri':
'https://chat.openai.com/aip/g-some_gpt_id/oauth/callback'} Note:
```

https://chatgpt.com/aip/g-some_gpt_id/oauth/callback` is also valid.`

In order for someone to use an action with OAuth, they will need to send a message that invokes the action and then the user will be presented with a "Sign in to [domain]" button in the ChatGPT UI.

The `authorization_url` endpoint should return a response that looks like:

```
{ "access_token": "example_token", "token_type": "bearer", "refresh_token":
"example_token", "expires_in": 59 }
```

During the user sign in process, ChatGPT makes a request to your `authorization_url` using the specified `authorization_content_type`, we expect to get back an access token and optionally a refresh token which we use to periodically fetch a new access token.

Each time a user makes a request to the action, the user’s token will be passed in the Authorization header: (“Authorization”: “[Bearer/Basic] [user’s token]”).

We require that OAuth applications make use of the state parameter for security reasons.

Failure to login issues on Custom GPTs (Redirect URLs)?

Be sure to enable this redirect URL in your OAuth application:

#1 Redirect URL: `https://chat.openai.com/aip/{g-YOUR-GPT-ID-HERE}/oauth/callback` (Different domain possible for some clients)

#2 Redirect URL: `https://chatgpt.com/aip/{g-YOUR-GPT-ID-HERE}/oauth/callback` (Get your GPT ID in the URL bar of the ChatGPT UI once you save) if you have several GPTs you'd need to enable for each or a wildcard depending on risk tolerance.

Debug Note: Your Auth Provider will typically log failures (e.g. 'redirect_uri is not registered for client'), which helps debug login issues as well.