Secure API Token Generation with OAuth and PKCE

How to use OAuth Authorization Code flow with Proof Key for Code Exchange (PKCE) to generate secure API tokens.

Document generated 03/26/2025

PDF VERSION

Tags | **API** | **Authentication** |

Applies to: Free Basic Business **Enterprise**

*This documentation is for **Stack Overflow for Teams Enterprise**. Free, Basic, and Business users can access their documentation here. Find your plan.*

Overview

This documentation is for API users who wish to implement secure and efficient authentication through the OAuth explicit flow, also known as the Authorization Code Grant type. This process involves exchanging an authorization code for an access token, a method suitable for the following applications:

- confidential clients (such as server-side applications)
- public clients (such as single-page applications, or those that run entirely on the front end)

***NOTE:** For client-side applications that run entirely on the front end, we strongly encourage the use of the PKCE extension. If you decide not to use PKCE, be sure to keep your client secret confidential.*

THIS ARTICLE APPLIES TO STACK OVERFLOW FOR TEAMS ENTERPRISE ONLY.
Other Stack Overflow for Teams users should read this article instead. Find your plan.

PKCE extension (optional)

Adding Proof Key for Code Exchange (PKCE) to the OAuth explicit flow makes it even more secure. PKCE is an extension that prevents cross-site request forgery (CSRF) and authorization code injection attacks. Though not required, we recommend using the PKCE extension for better security.

PKCE generates a code verifier string, hashes it, and provides the hash during the authentication process. This ensures secure validation and exchange of tokens, mitigating the risk of impersonation and unauthorized access. The process below will include additional steps for adding the PKCE extension.

NOTE: You'll find it helpful to have a text file or other working document open to copy/paste information as you go through this process.

Before you begin

Before you begin this process, follow the steps in the Stack Overflow for Teams API v3 article to create an API application with your desired access scope. The information you'll need to copy from your new API application depends on whether you're using PKCE. For PKCE, copy just the **Client ID**. If you're not using PKCE, copy the **Client ID** and **Secret**.

Generate a code verifier and code challenge (PKCE only)

If you're using PKCE, you need to create a code verifier and code challenge before you can generate a token. The code verifier is a random string, and the code challenge is a hashed version of the code verifier.

NOTE: *The challenge method for PKCE must be S256.*

Ping Identity has a helpful tool for generating these values here: https://developer.pingidentity.com/en/tools/pkce-code-generator.html.

Use the Ping Identity tool (or any other PKCE code generator) to generate the code verifier and code challenge strings. Copy both to your working document.

Obtain the authorization code

Before generating an API access token, you'll need to create an intermediate authorization code. You'll create this authorization code by accessing a specific URL on your site, posting several parameter values as part of the URL query string.

Your base URL is https://[your_site]/oauth. To this URL you'll add the following query parameters in this format: `?parameter=value`.

`client_id`
The Client ID value copied from your API application.

`redirect_uri`
Any redirection URI under the domain you specified on creation of the app.

`scope`
Defines the level of access an application has to a user's data. Set the scope value according to your application's requirements.

`scope` values include:

- **`read_inbox`** Access a user's global inbox.
- **`no_expiry`** The token will never expire.
- **`write_access`** Perform write operations as the user (requires read-write API application).
- **`private_info`** Access full history of a user's private actions on the site.

To specify multiple scopes, use a comma to delimit them with no spaces (for example: `scope=no_expiry,write_access`).

NOTE: *Unless you specify no_expiry, the token will expire 24 hours after creation. For security, we strongly discourage the use of the no_expiry scope unless actually required by your app.*

`code_challenge` (PKCE only)
The `code_challenge` value generated in a previous step, necessary only when using PKCE.

`code_challenge_method` (PKCE only)
Set to S256. This is only required when using PKCE.

`state` (optional, recommended for automated token generation)
A unique, random, and non-guessable string, usually created by an API integration that automates token generation. The `state` value is sent when starting an authentication request and validated when processing the response. Used to prevent cross-site request forgery (XSRF) attacks, this parameter is optional but recommended.

NOTE: *If you're creating a token manually (for development or testing, for example), you don't need to provide a state value.*

Example URLs

Without PKCE

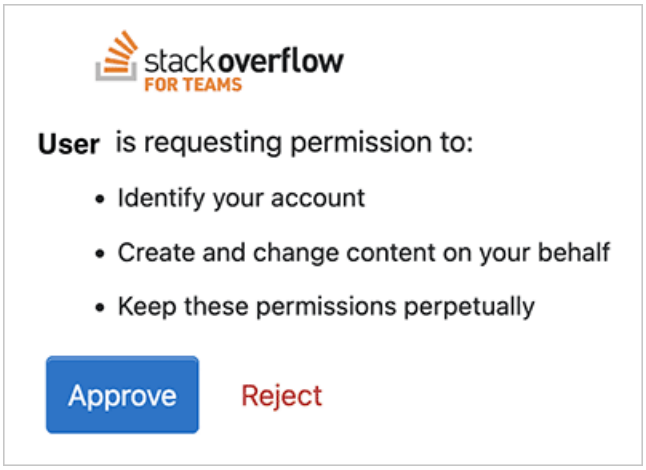
`https://[your_site]/oauth?client_id=YourClientID&scope=DesiredScope&redirect_uri=YourRedirectURI` (optional: `&state=YourState`)

With PKCE

`https://[your_site]/oauth?client_id=YourClientID&scope=DesiredScope&redirect_uri=YourRedirectURI&code_challenge=GeneratedCodeChallenge&code_challenge_method=S256` (optional: `&state=YourState`)

Authorize the application

After you visit this URL, you'll be prompted to log in to your SOE account and authorize the application.



The site will then redirect you to a page that includes an authorization code in the URL.

Example authorization URL

`https://[your_site]/oauth/login_success?code=AuthorizationCode`

Copy the authorization code (identified with "`code=`" in the URL) to your working document.

Generate the access token

The final step to generate an access token involves a POST request to https://[your_site]/oauth/access_token. Make sure the request is application/x-www-form-urlencoded and includes the following query parameters in the URL:

`client_id`
The same client ID used in the previous step.

`client_secret`
Your API application's secret, revealed on creation on the SOE API Applications page. This value is not necessary when using PKCE.

`code`
Authorization code obtained in the previous step.

`redirect_uri`
The redirect URI specified on creation of the API application in SOE.

`code_verifier`
Generated along with the PKCE code challenge (in Ping Identity, for example). Required only with PKCE.

Example access token URLs

Without PKCE

```
https://[your_site]/oauth/access_token?
client_id=YourClientID&client_secret=ClientSecret&code=Code&redirect_uri=RedirectURI
```

With PKCE

```
https://[your_site]/oauth/access_token?
client_id=YourClientID&code=Code&redirect_uri=RedirectURI&code_verifier=CodeVerifier
```

Access token response

You should receive a plain text response with your API access token, which you'll use to authenticate future requests to API v3. The response will also include the remaining seconds until the token expires, unless you specified the `no_expiry` scope for the authorization code.

Example of access token response

```
access_token='abcdefghi1234567890&expires=37845
```

Access token response (JSON-encoded)

If you append `/json` to the token URL, SOE will return access_token and expires (if applicable) wrapped in a JSON object. For example: `https://[your_site]/oauth/access_token/json?client_id=YourClientID...`.

Example of JSON-encoded access token response

```
{ access_token: 'abcdefghi1234567890', expires: 37845 }
```

If you need further support or have questions, contact your site administrator.