

Configuring Local Swagger Auth

In this example we are using the following values:

Environment URL: <https://ostiasolutions-agtbaad4aabam.scalekit.dev>

Client ID: `spac_124490605639238658`

Use your own values.

1. Add the security scheme to your OpenAPI spec (`openapi.yaml` or `swagger.json`):

```
components:
  securitySchemes:
    scalekit_pkce:
      type: oauth2
      flows:
        authorizationCode:
          authorizationUrl: <Environment URL>/oauth/authorize
          tokenUrl: <Environment URL>/oauth/token
          scopes:
            profile: User profile
            email: User email
            offline_access: Refresh token

security:
  - scalekit_pkce: [openid, profile, email]
```

In our case the values are:

authorizationUrl: `'https://ostiasolutions-agtbaad4aabam.scalekit.dev/oauth/authorize'`

tokenUrl: `'https://ostiasolutions-agtbaad4aabam.scalekit.dev/oauth/token'`

The "security" block can be declared globally, at the top level of the file, or on per-endpoint basis.

2. Configure `SwaggerUIBundle` initialisation:

```
const ui = SwaggerUIBundle({
  url: './swagger.yaml',
  dom_id: '#swagger-ui',
  oauth2RedirectUrl: 'http://localhost:8080/oauth2-redirect.html',
  persistAuthorization: true,
});
```

```
ui.initOAuth({
  clientId: 'spac_124490605639238658',
  scopes: 'openId profile email',
  scopeSeparator: ' ',
  usePkceWithAuthorizationCodeGrant: true,
});
```

Example:

1. In a separate folder create **index.html**:

```
<!DOCTYPE html>
<html>
<head>
  <title>Swagger UI</title>
  <link rel="stylesheet" href="https://unpkg.com/swagger-ui-dist@4.19.0/swagger-ui.css"/>
</head>
<body>
<div id="swagger-ui"></div>
<script src="https://unpkg.com/swagger-ui-dist@4.19.0/swagger-ui-bundle.js"></script>
<script>
  const ui = SwaggerUIBundle({
    url: './swagger.yaml',
    dom_id: '#swagger-ui',
    oauth2RedirectUrl: 'http://localhost:8080/oauth2-redirect.html',
    persistAuthorization: true,
  });

  ui.initOAuth({
    clientId: 'spac_124490605639238658',
    scopes: 'openId profile email',
    scopeSeparator: ' ',
    usePkceWithAuthorizationCodeGrant: true,
  });
</script>
</body>
</html>
```

2. Add **swagger.yaml** with the correct security scheme

3. Add **swagger-ui.css** ([file](#))


4. Create **oaut2-redirect.html** ([file](#)) with the corresponding **oaut2-redirect.js**

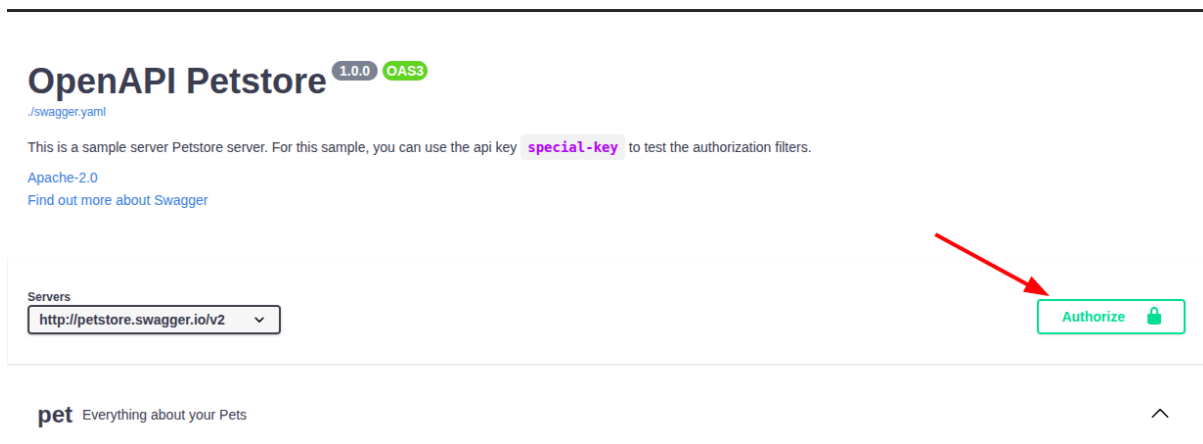
All the files can be taken from [here](#).

Open the index.html on some port. You can use a simple npx server:

In the terminal open the folder location with the files above and run the following command:

```
"npx http-server . -p 8080 --cors"
```

Open the Swagger UI page in your browser on the port localhost:8080 and click the "Authorize 



In the dialog that appears, confirm the **client_id** is pre-filled and click "Authorize". Leave **client_secret** empty. A login window will open.

Available authorizations x

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes.

API requires the following scopes. Select which ones you want to grant to Swagger UI.

scalekit_pkce (OAuth2, authorizationCode with PKCE)

This API uses OAuth 2

Authorization URL: <https://ostiasolutions-agtbaad4aabam.scalekit.dev/oauth/authorize>

Token URL: <https://ostiasolutions-agtbaad4aabam.scalekit.dev/oauth/token>

Flow: authorizationCode with PKCE

client_id:

client_secret:

Scopes: [select all](#) [select none](#)

- openid
openID
- profile
User profile
- email
User email

Log in with your credentials. Swagger UI will handle the PKCE flow automatically — no **client secret** or manual token needed.

Once logged in, close the dialog. Swagger UI will automatically attach the JWT as a *Bearer token* to all API requests you make from the UI.

⚠ Note: When the token expires, you'll will start getting 401 *Unauthorized* responses on API calls and will need to click **Authorize** again and log in (Scalekit SSO will be still active, so it won't even prompt for credentials again — just silently issue a new token).