

Auth Integration with Auth0:

Setup:


- 1) Go to auth0.com and sign up
- 2) Setup a Single Page application and choose what Social Connections you want to use

Create Application ✕

Name *

You can change the application name later in the application settings.


Choose an application type



Native

Mobile, desktop, CLI and smart device apps running natively.


e.g.: iOS, Electron, Apple TV apps



Single Page Web Applications

A JavaScript front-end app that uses an API.


e.g.: Angular, React, Vue



Regular Web Applications

Traditional web app using redirects.

e.g.: Node.js Express, ASP.NET, Java, PHP



Machine to Machine Applications

CLIs, daemons or services running on your backend.

e.g.: Shell script

CancelCreate

- 3) To communicate with Auth0 you'll need some credentials, which can be found on the dashboard site of the created app. Put domain and client id in a json file "auth_config.json" in the root of the project
- 4) Within the settings of the application, put <http://localhost:8080> (assuming this port is free as we are creating an express server) in "Allowed Callback URLs", "Allowed Logout URLs" and "Allowed Web Origins"
- 5) Create an index.html file in the root aswell as a /public/js folder and create a [script.js](#)

```
Auth_Integration > <> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Auth0</title>
7
8  </head>
9  <body>
10
11      <h2>Auth0</h2>
12      <button id="btn-login" disabled="true" onclick="login()">Log in</button>
13      <button id="btn-logout" disabled="true" onclick="logout()">Log out</button>
14
15      <script src="https://cdn.auth0.com/js/auth0-spa-js/2.0/auth0-spa-js.production.js"></script>
16      <script src="/js/script.js"></script>
17
18  </body>
19  </html>
20
```

6) Init an express project

```
Auth_Integration > JS app.js > ...
1  import express from "express";
2  import path from "path";
3  import { fileURLToPath } from "url";
4
5  const __filename = fileURLToPath(import.meta.url);
6  const __dirname = path.dirname(__filename);
7
8  const app = express();
9
10 app.use(express.static(path.join(__dirname, "public")));
11
12 app.get("/auth_config.json", (req, res) => {
13     res.sendFile(path.join(__dirname, "auth_config.json"));
14 });
15
16 app.get("/", (_, res) => {
17     res.sendFile(path.join(__dirname, "index.html"));
18 });
19
20 app.listen(8080, () => console.log("Application running on port 8080"));
21
```

7) Create a configureClient, updateUI, login and logout function in the [script.js](#) file and follow the documentation provided on <https://auth0.com/docs/quickstart/spa/vanillajs/interactive>. Or look at these screenshots:

Auth_Integration > public > js > JS script.js > login

```
1  let auth0Client = null;
2
3  const fetchAuthConfig = () => fetch("/auth_config.json");
4
5  const configureClient = async () => {
6    const response = await fetchAuthConfig();
7    const config = await response.json();
8
9    auth0Client = await auth0.createAuth0Client({
10     domain: config.domain,
11     clientId: config.clientId
12   });
13 };
14
15 window.onload = async () => {
16   await configureClient();
17
18   const query = window.location.search;
19   if (query.includes("code=") && query.includes("state=")) {
20
21     await auth0Client.handleRedirectCallback();
22     window.history.replaceState({}, document.title, "/");
23   }
24
25   updateUI();
26 };
27
28 const updateUI = async () => {
29   const isAuthenticated = await auth0Client.isAuthenticated();
30
31   document.getElementById("btn-logout").disabled = !isAuthenticated;
32   document.getElementById("btn-login").disabled = isAuthenticated;
33 };
34
```

```
const login = async () => {
  await auth0Client.loginWithRedirect({
    authorizationParams: {
      redirect_uri: window.location.origin
    }
  });
};

const logout = () => {
  auth0Client.logout({
    logoutParams: {
      returnTo: window.location.origin
    }
  });
};
```

You can now press the login button on <http://localhost:8080>, create an account, login and logout.

To see any user information you can use the `getUser()` function on the `auth0Client` object and then render the information in the `updateUI()` function.