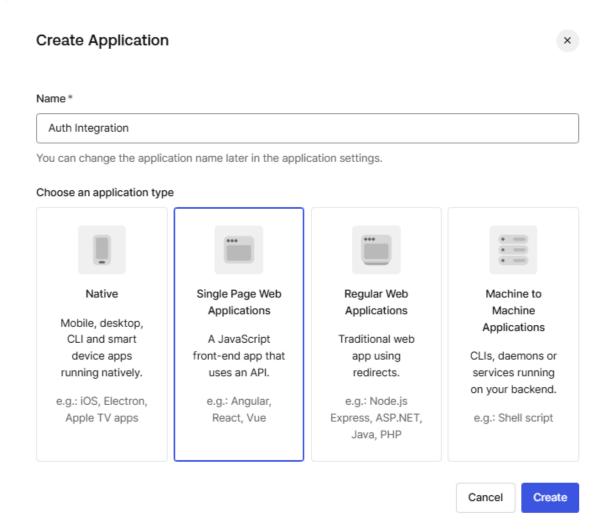
Auth Integration with Auth0:

Setup:

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



- 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

6) Init an express project

```
Auth_Integration > JS app.js > ...
      import express from "express";
      import path from "path";
  2
      import { fileURLToPath } from "url";
      const __filename = fileURLToPath(import.meta.url);
      const __dirname = path.dirname(__filename);
      const app = express();
      app.use(express.static(path.join(__dirname, "public")));
      app.get("/auth_config.json", (req, res) => {
          res.sendFile(path.join( dirname, "auth config.json"));
      });
      app.get("/", (_, res) => {
           res.sendFile(path.join( dirname, "index.html"));
      });
      app.listen(8080, () => console.log("Application running on port 8080"));
```

7) Create a configureClient, updateUI, login and logout function in the <u>script.js</u> 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 > [❷] login
      let auth0Client = null;
      const fetchAuthConfig = () => fetch("/auth_config.json");
      const configureClient = async () => {
          const response = await fetchAuthConfig();
          const config = await response.json();
          auth0Client = await auth0.createAuth0Client({
            domain: config.domain,
            clientId: config.clientId
      };
      window.onload = async () => {
          await configureClient();
          const query = window.location.search;
           if (query.includes("code=") && query.includes("state=")) {
             await auth0Client.handleRedirectCallback();
            window.history.replaceState({}, document.title, "/");
          updateUI();
        };
       const updateUI = async () => {
          const isAuthenticated = await auth@Client.isAuthenticated();
           document.getElementById("btn-logout").disabled = !isAuthenticated;
           document.getElementById("btn-login").disabled = isAuthenticated;
       };
```

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

const logout = () => {
     authOClient.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.