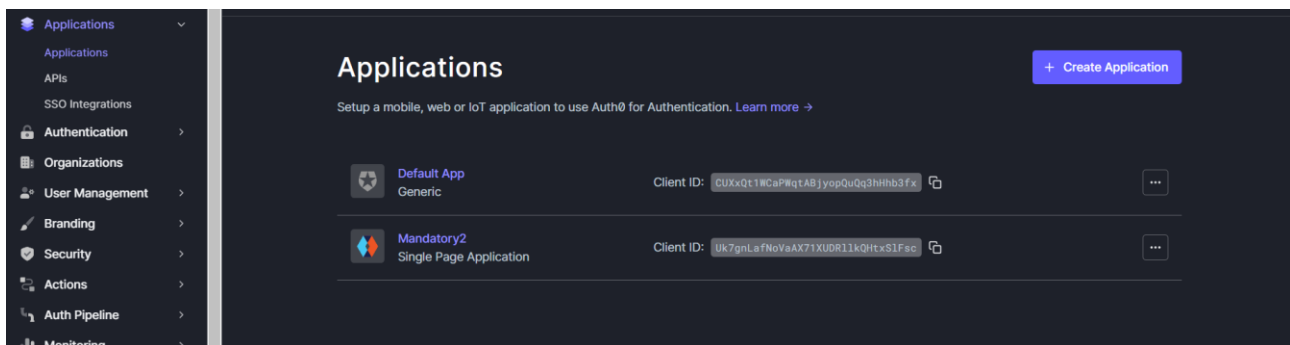


Mandatory 2 – Part 1

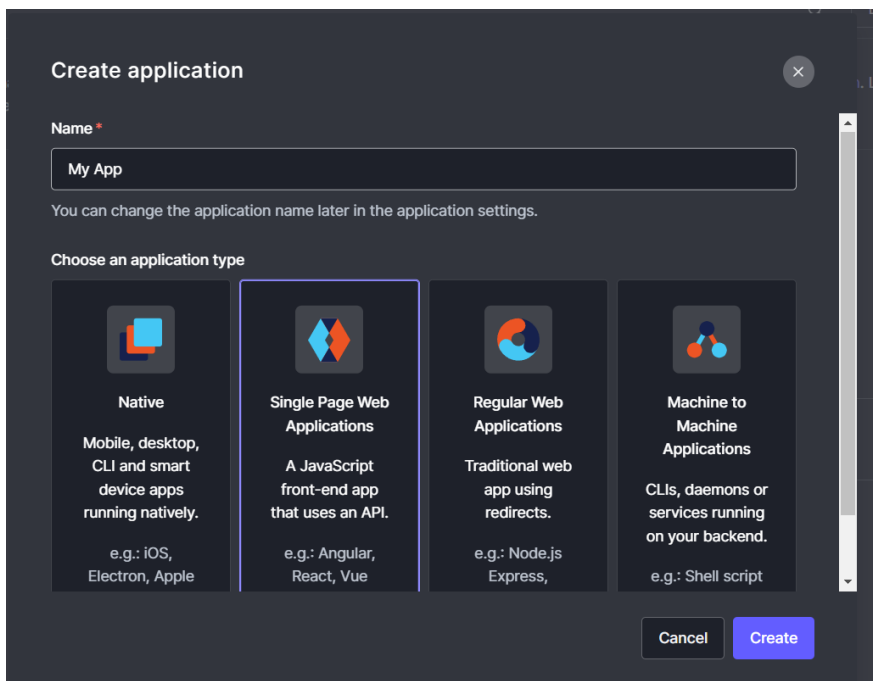
Authentication

For this project, I will be using Auth0 as authentication service. First step for using this service, is by signing up for an account on their website <https://auth0.com/>.

When signed up I am going to create an application on the website, which will be used for this project. This can be found on the side panel under “Applications”.



After clicking create, I get this pop up:



I am going to give my application a name and furthermore choose the Single Page Web Application option.

After signing up and creating my application, it is now ready to be used in my code.

To use Auth0 in my code, I am going to run the following command in my terminal:

```
npm install --save auth0
```

Now that I have it installed, I can start making my endpoints which will be used for my frontend.

I create an endpoint for both login and sign up.

```
app.get('/login', (req, res) => {  
  res.render('login');  
});  
  
app.get('/signup', (req, res) => {  
  res.render('signup');  
});
```

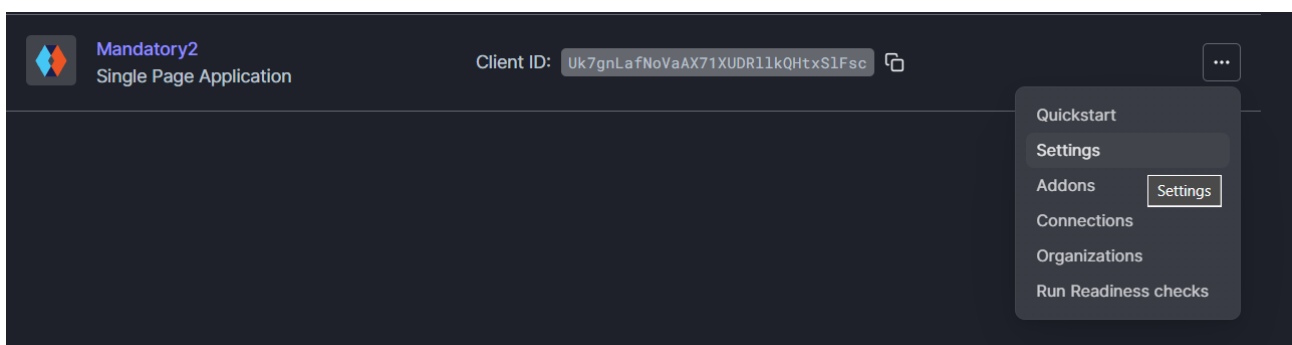
To use the authentication service with these endpoints, I will also make endpoints for the authentication login and signup.

```
app.get('/login/auth0', passport.authenticate('auth0', {  
  scope: 'openid email profile',  
  connection: 'Username-Password-Authentication',  
}), function (req, res) {  
  req.session.destroy(function (err) {  
    res.redirect('/login');  
  });  
});  
  
app.get('/signup/auth0', passport.authenticate('auth0', {  
  scope: 'openid email profile',  
  connection: 'Username-Password-Authentication',  
}), function (req, res) {  
  req.session.destroy(function (err) {  
    res.redirect('/signup');  
  });  
});
```

To use the authentication service, I will also set up my Auth0 strategy with the following code:

```
passport.use(new Auth0Strategy({
  domain: 'dev-fanu6ezuspfbskrx.us.auth0.com',
  clientId: 'Uk7gnLafNoVaAX71XUDR1lkQHTxSlFsc',
  clientSecret: 'maiFn1Gztqf13uDkTpQ-CaX6C5B41G3Snh1J0oK0xlpYrLu7yRYmTn',
  callbackURL: 'http://localhost:3000/callback'
}),
function(accessToken, refreshToken, extraParams, profile, done) {
  return done(null, profile);
});
```

The information provided can be found in my application on my Auth0 dashboard.



By clicking on settings on my application, I can find the relevant information that needs to be used in my project.

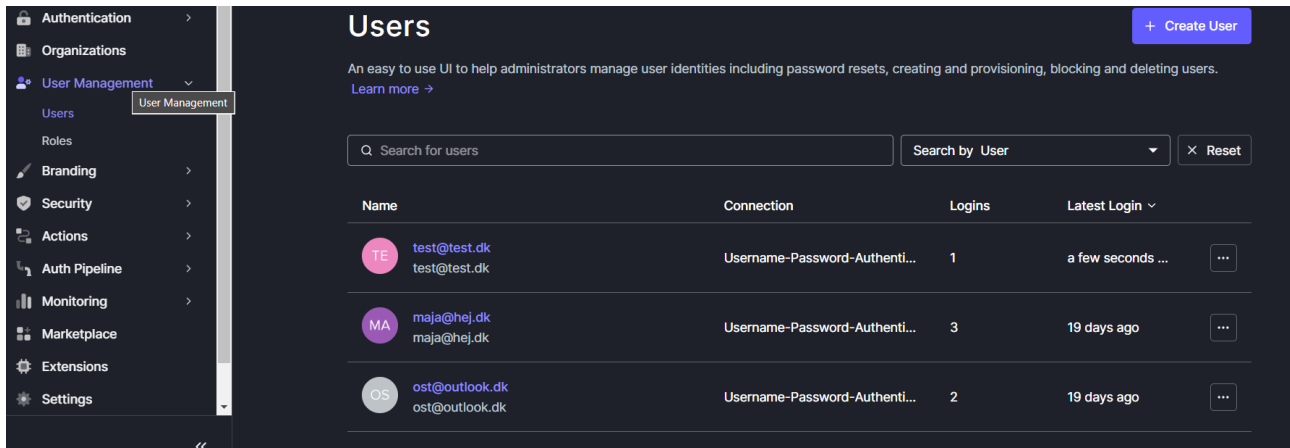
I will now start my application by running *nodemon app.js* in my terminal. The application will start on port 3000.

First, I will run the sign-up endpoint to create a user, and later login using my new created user.

When clicking sign up, I will get the following Auth0 popup:

A screenshot of an Auth0 'Welcome' popup. At the top is the Auth0 logo (a red shield with a white star). Below it, the word 'Welcome' is centered. The text 'Sign Up to dev-fanu6ezuspfbskrx to continue to Mandatory2.' is displayed. There are two input fields: 'Email address' with the value 'test@test.dk' and 'Password' with masked characters '*****'. Below the password field, a list of password requirements is shown with green checkmarks: 'At least 8 characters', 'At least 3 of the following: Lower case letters (a-z), Upper case letters (A-Z), Numbers (0-9), and Special characters (e.g. !@#%&*)'. At the bottom is a blue button labeled 'Continue'.

To confirm that the user has been signed up, I can check my current users under User Management on the side panel.



I can confirm that my user has been created. Now that my user has been created, I will try to login using that new user. There will also be done some authentication on my user, which can be seen on the following code:

```
const isAuthenticated = (req, res, next) => {
  if (req.isAuthenticated()) {
    return next();
  }
  res.redirect('/login');
};
```

If the user is authenticated, the user will be sent back to a dashboard with their information. If something went wrong, they will be redirected to the login page.

To test if my user can log in, I will now run the login endpoint and put in the information from the new user.

The user authentication worked and is now being sent to a dashboard with their information.



Welcome, test@test.dk!

