Keycloak with Node Js and EJS

Prerequisites:

1. Docker
2. Node Js ^14
3. NPM ^6

Reference link to install Prerequisites:

1. Docker: https://docs.docker.com/engine/install/
2. Node Js and NPM: https://nodejs.org/en/download/

Setting up docker with Keycloak image:

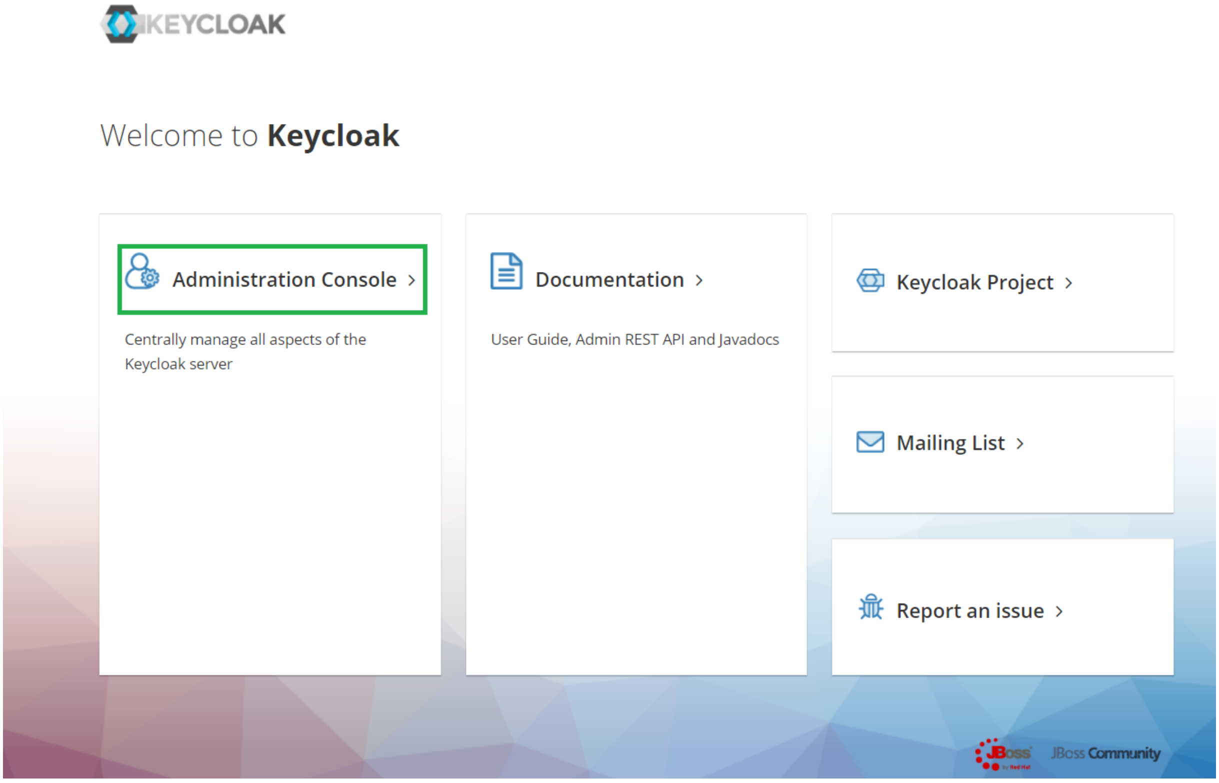
1. Open terminal and past the following command to create a container with username "admin" and password "password" with database as "H2".

```
docker run -e DB_VENDOR=H2 -e KEYCLOAK_USER=admin -e KEYCLOAK_PASSWORD=password
-p 8080:8080 --name keycloak -dit jboss/keycloak
```

2. Once the setup is completed you can access the Keycloak server from browser with this URL: http://localhost:8080/auth/

Setting up Keycloak Server:

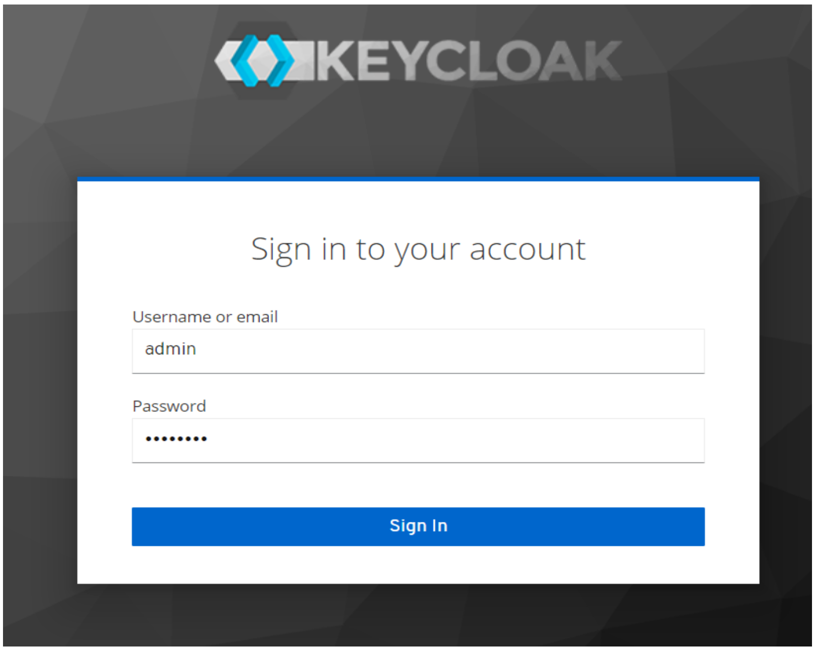
1. Click on the Administrative console:



Centrally manage all aspects of the Keycloak server

User Guide, Admin REST API and Javadocs

2. Put the same username and password which we used to create a Keycloak container in docker which is username “admin” and password “password”.

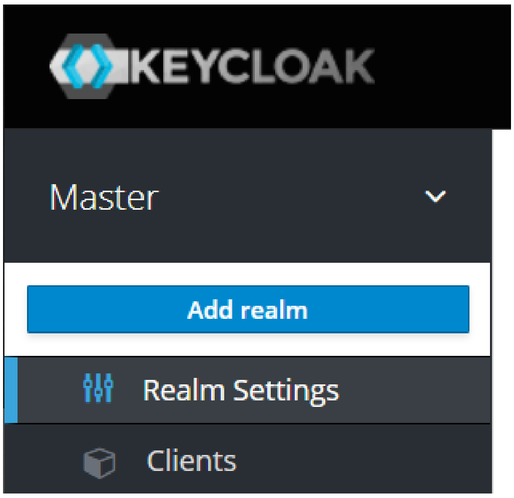


Sign in to your account

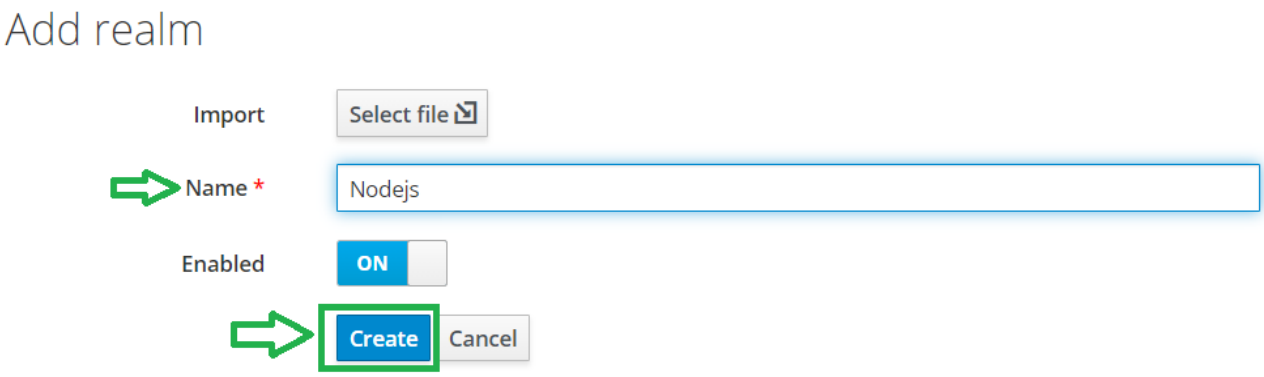
Username or email: admin

Password: ••••••••

3. Now we have to create a realm for our project. Click on drop down called “Master” and click on add realm.



4. Now give the name for the realm like “Nodejs” and click on create.



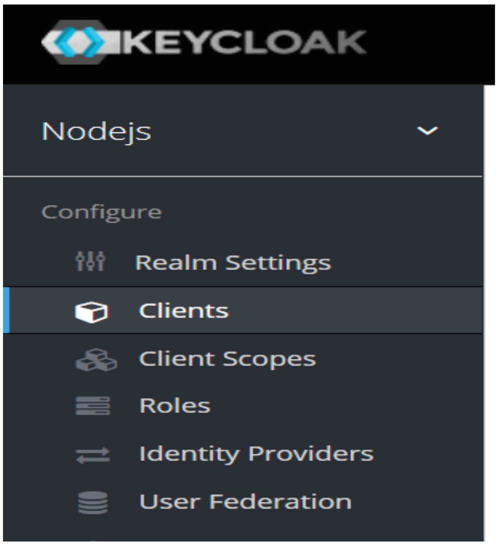
Add realm

Import: Select file

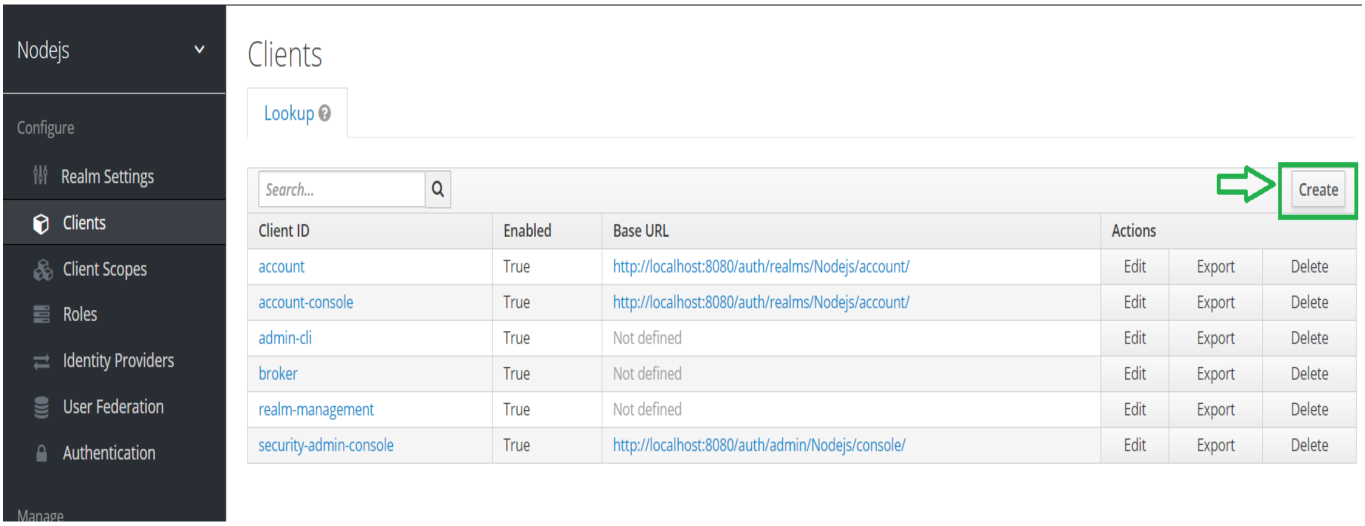
Name *: Nodejs

Enabled: ON

5. Now click on clients in the left menu bar from the Nodejs realm like below

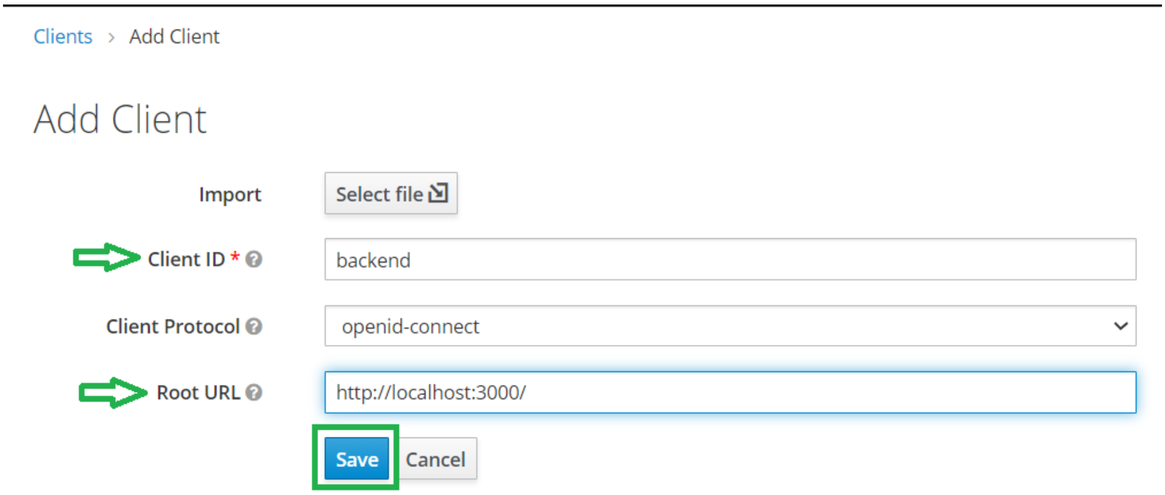


6. Click on Create button.



Client ID	Enabled	Base URL	Actions		
account	True	http://localhost:8080/auth/realms/Nodejs/account/	Edit	Export	Delete
account-console	True	http://localhost:8080/auth/realms/Nodejs/account/	Edit	Export	Delete
admin-cli	True	Not defined	Edit	Export	Delete
broker	True	Not defined	Edit	Export	Delete
realm-management	True	Not defined	Edit	Export	Delete
security-admin-console	True	http://localhost:8080/auth/admin/Nodejs/console/	Edit	Export	Delete

7. Now add the client ID as “backend”, client Protocol to “open-connect” and root URL of Node Js server which is “ http://localhost:3000/ ” and click on save.



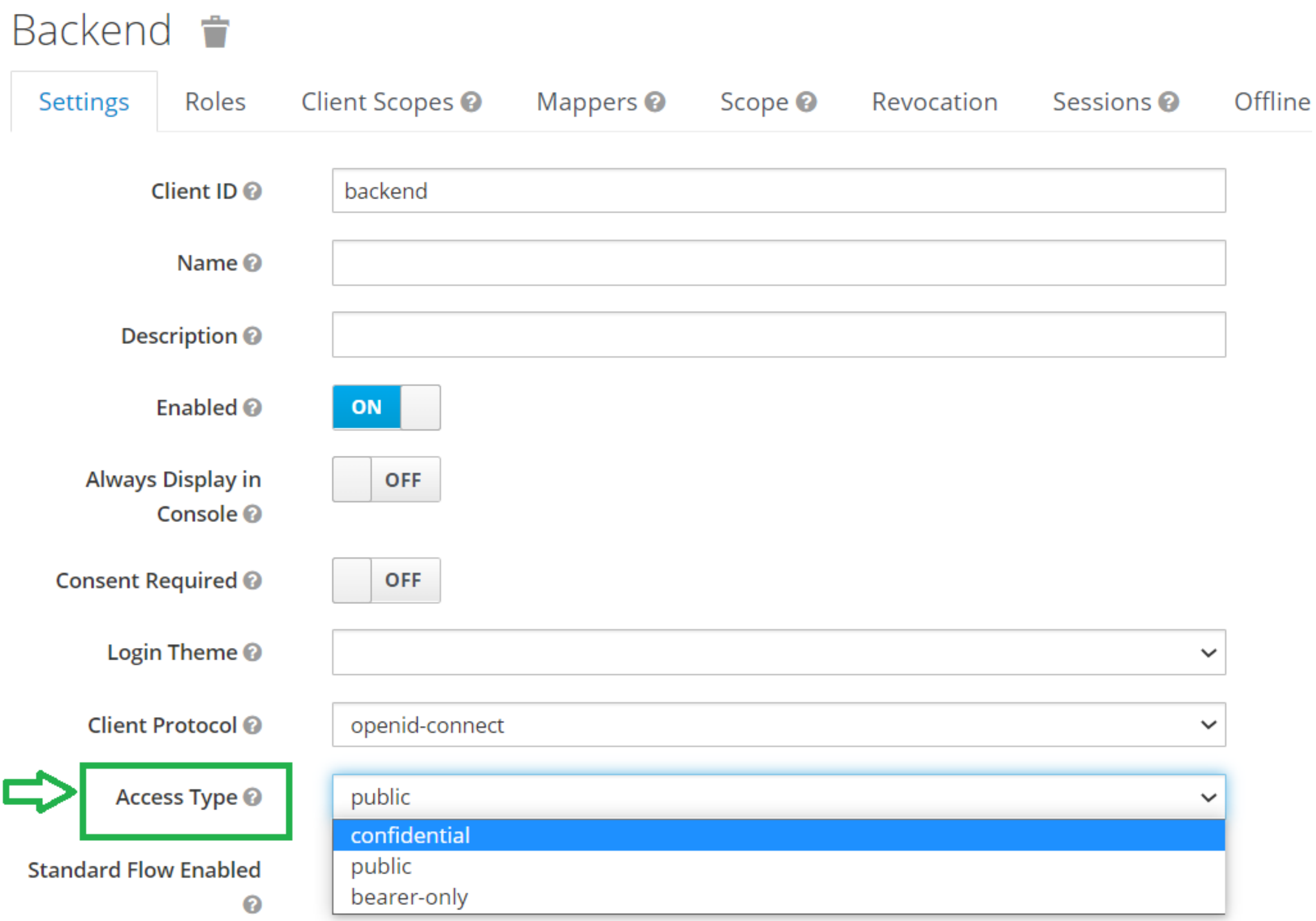
Import

Client ID * backend

Client Protocol openid-connect

Root URL http://localhost:3000/

8. Now change the access type to confidential and click on save.



Client ID: backend

Name:

Description:

Enabled: ON

Always Display in Console: OFF

Consent Required: OFF

Login Theme:

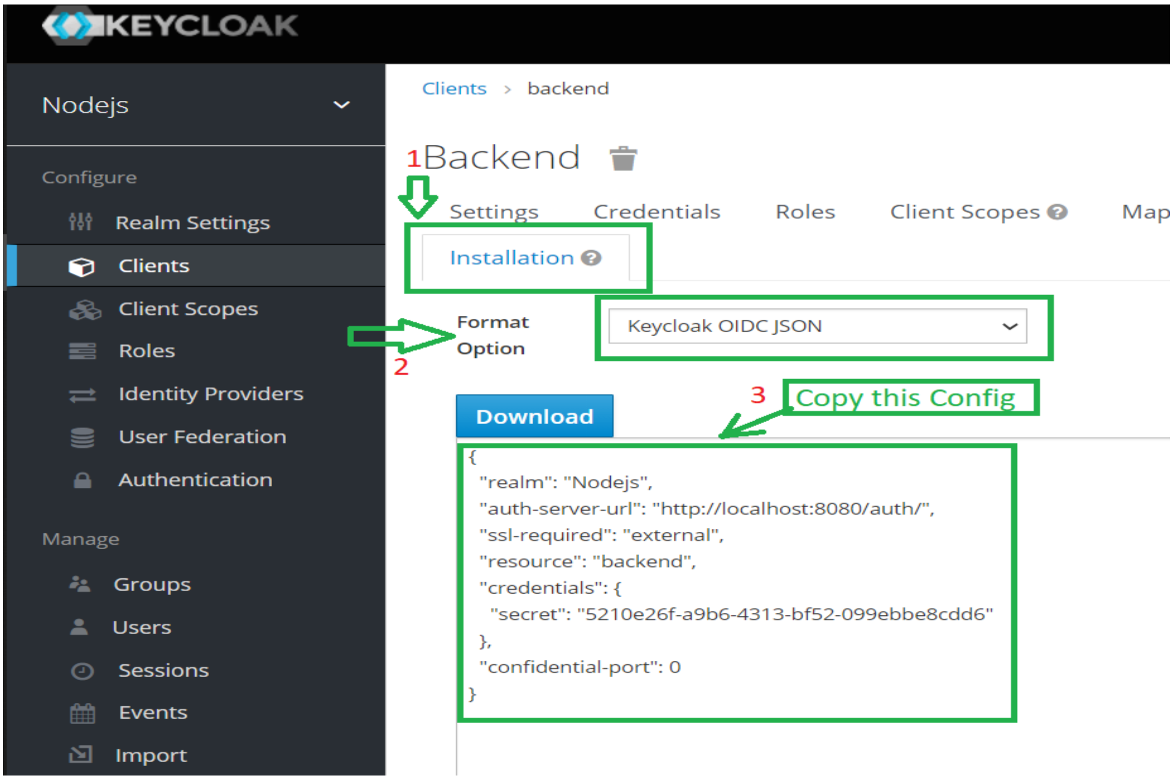
Client Protocol: openid-connect

Access Type: public

- confidential
- public
- bearer-only

Standard Flow Enabled

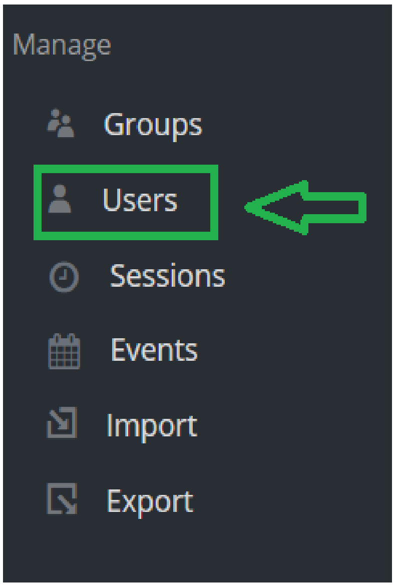
9. Now go to the Installation and select the format option as “Keycloak OIDC JSON” and copy the entire Json details and save it some where because it is used in configuring the Node JS server in upcome steps.



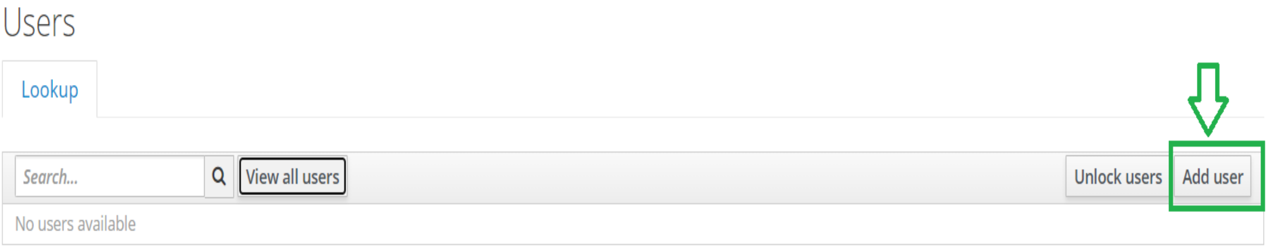
Format Option: Keycloak OIDC JSON

```
{
  "realm": "Nodejs",
  "auth-server-url": "http://localhost:8080/auth/",
  "ssl-required": "external",
  "resource": "backend",
  "credentials": {
    "secret": "5210e26f-a9b6-4313-bf52-099ebbe8cdd6"
  },
  "confidential-port": 0
}
```

10.Now we have to create users for the realm. Click on users in the menu:

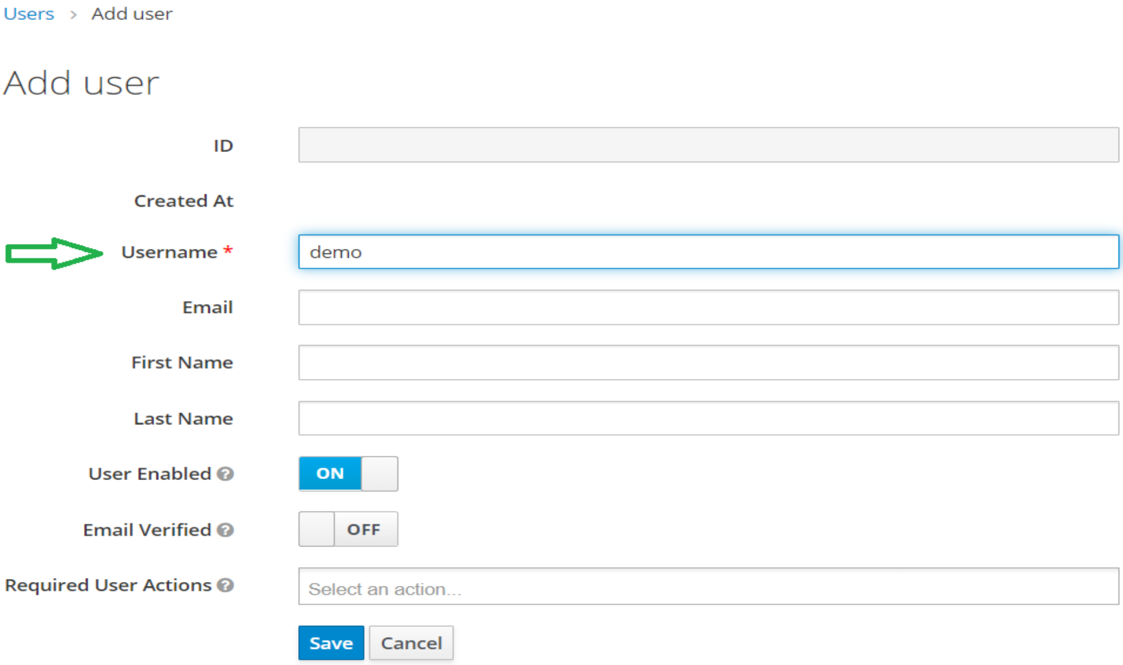


11.Click on add users:



No users available

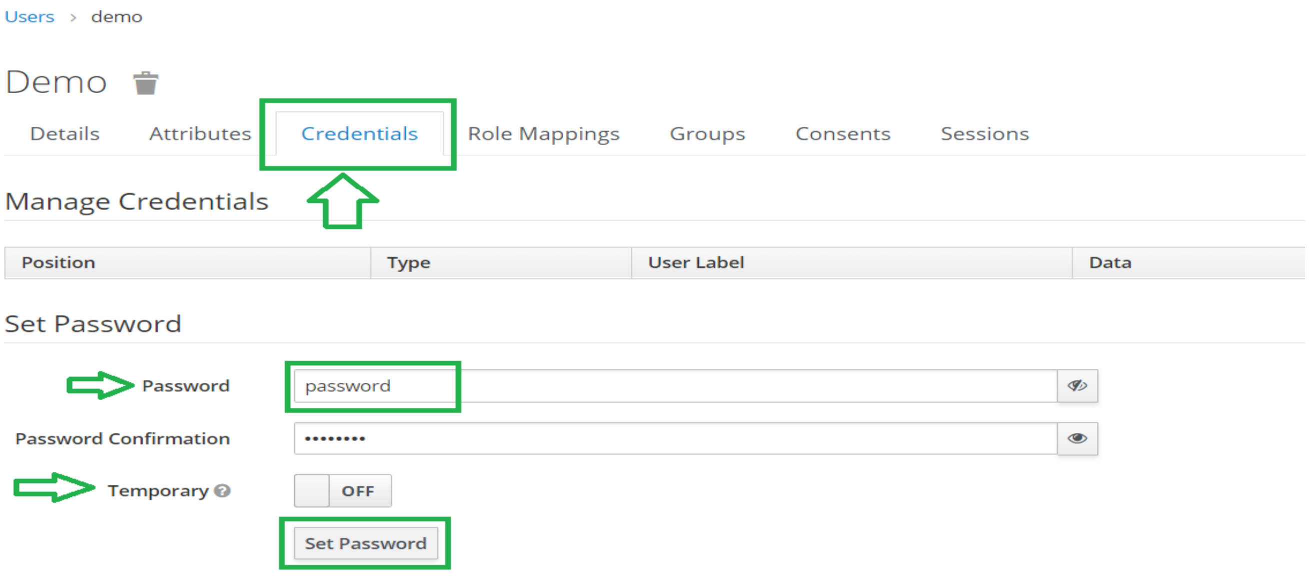
12. Add the username as “demo” and click on save. You can add other details if needed but for demo purpose we are just creating a user named “demo”.


Users > Add user

Add user

ID	
Created At	
Username *	demo
Email	
First Name	
Last Name	
User Enabled	ON
Email Verified	OFF
Required User Actions	Select an action...

13. Now click on Credentials and set the password as "password" and toggle the Temporary to off. And click on set password. It will prompt an alert, just click on set password again.



Users > demo

Demo

Manage Credentials

Position	Type	User Label	Data

Set Password

Password: password

Password Confirmation: ••••••••

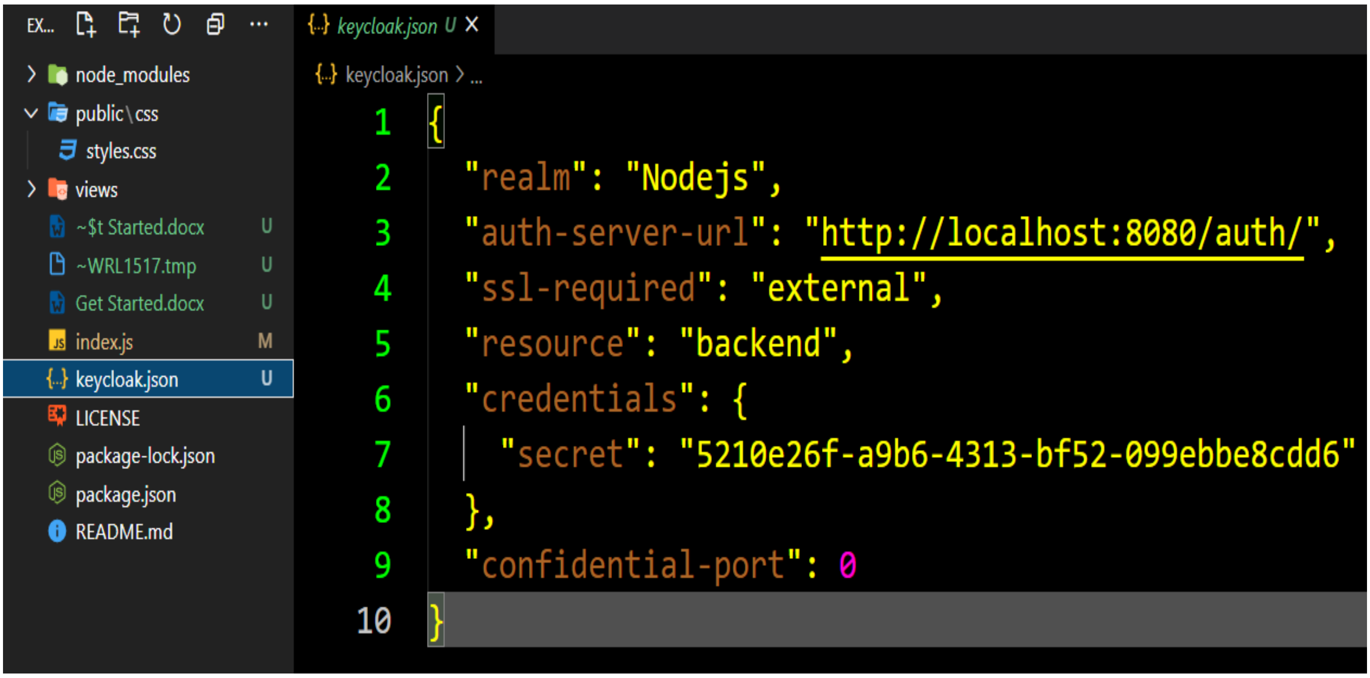
Temporary: OFF

Set Password

Now the user is been created Successfully.

Setting up Node Js server

1. Clone the repository from this link: https://github.com/Sujithk007/Keycloak-with-Node-js-and-Ejs
2. Or you can download the zip file and extract it in your local PC.
3. Inside the folder Keycloak-with-Node-js-and-Ejs, open Keycloak.json in any editor.
4. Now we have to configure the Node js server with Keycloak. To do so, we need the configuration for the client which we configured in Keycloak server.
5. From **Step 9** of **Setting up Keycloak Server** above we have the configurations.
6. Past that entire configuration in the Keycloak.json and save it.



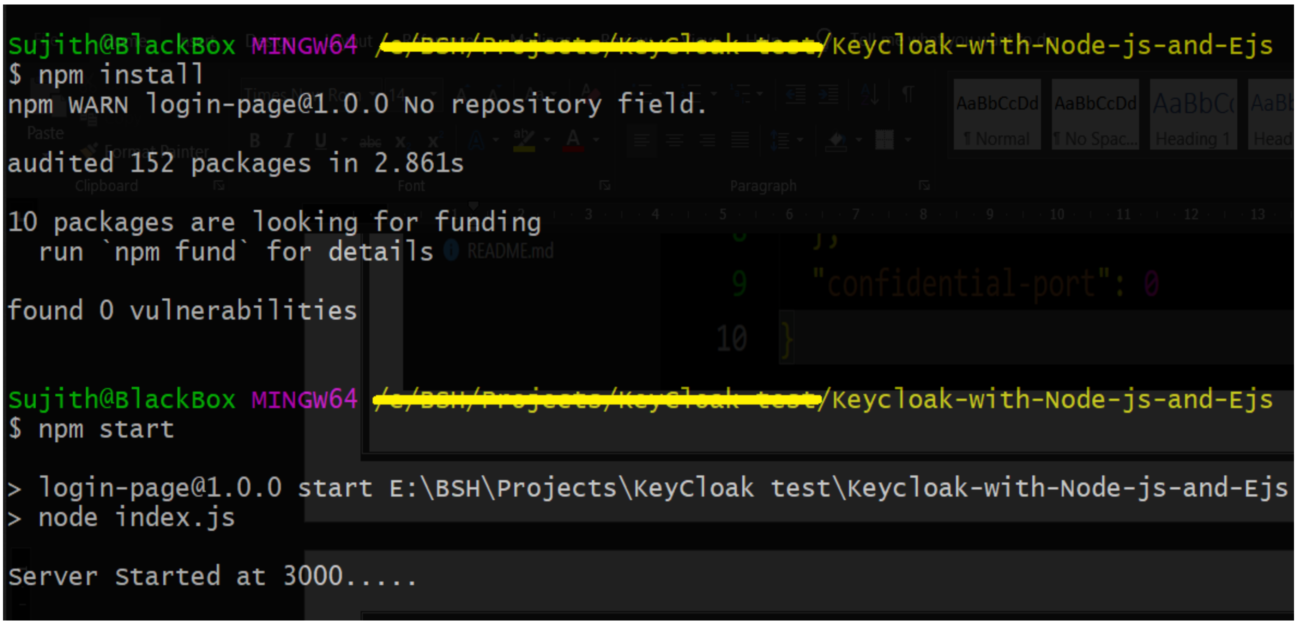
```
{
  "realm": "Nodejs",
  "auth-server-url": "http://localhost:8080/auth/",
  "ssl-required": "external",
  "resource": "backend",
  "credentials": {
    "secret": "5210e26f-a9b6-4313-bf52-099ebbe8cdd6"
  },
  "confidential-port": 0
}
```

Starting Node Js Server:

1. Inside the cloned folder start the terminal.
2. Execute the following command one by one:

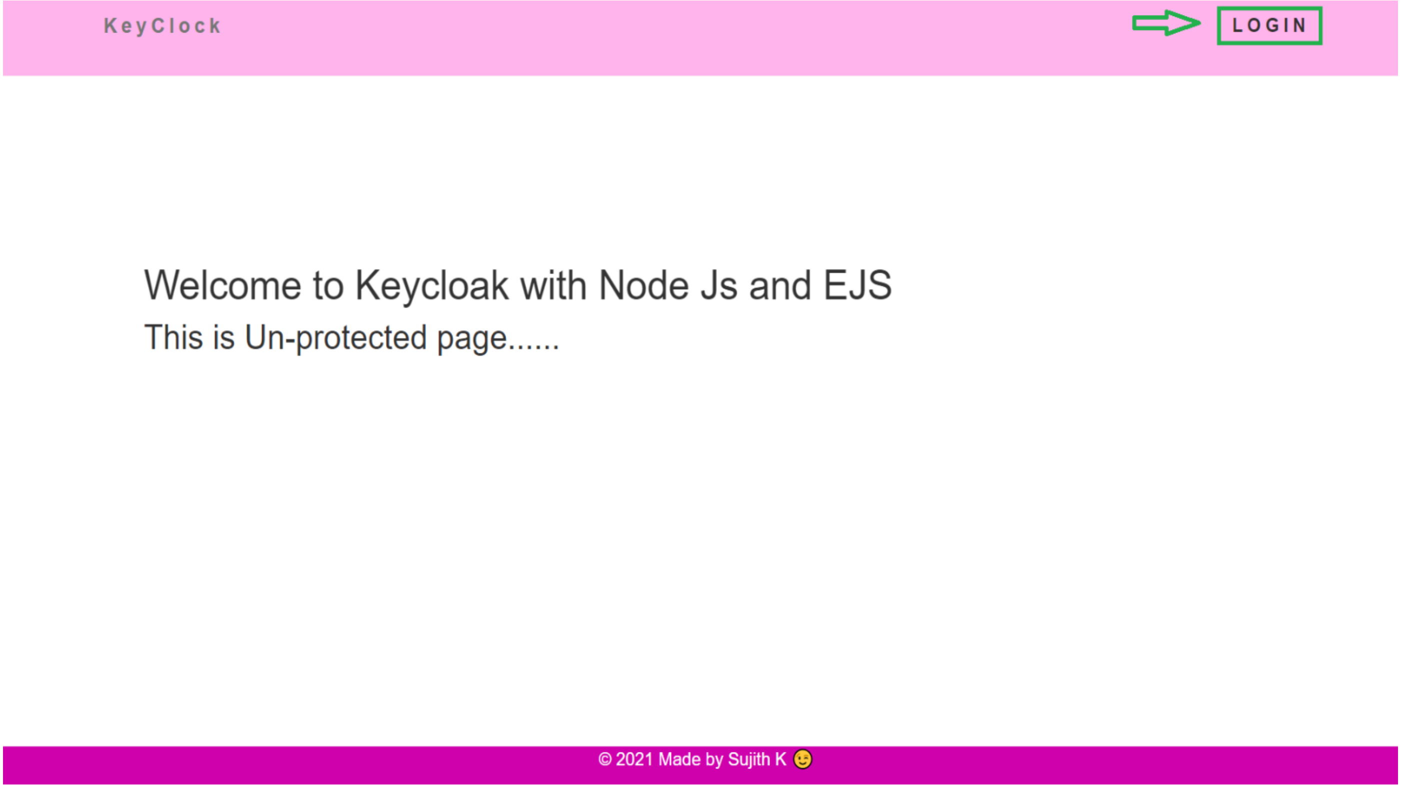
```
>> npm install

>> npm start
```

3. Like below:



```
Sujith@BlackBox MINGW64 [redacted]/Keycloak-with-Node-js-and-Ejs
$ npm install
npm WARN login-page@1.0.0 No repository field.

audited 152 packages in 2.861s

10 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities


Sujith@BlackBox MINGW64 [redacted]/Keycloak-with-Node-js-and-Ejs
$ npm start

> login-page@1.0.0 start E:\BSH\Projects\KeyCloak test\Keycloak-with-Node-js-and-Ejs
> node index.js

Server Started at 3000.....
```

4. Open browser at this URL: http://localhost:3000/ and click on login which will redirect to Keycloak login page.



KeyClock

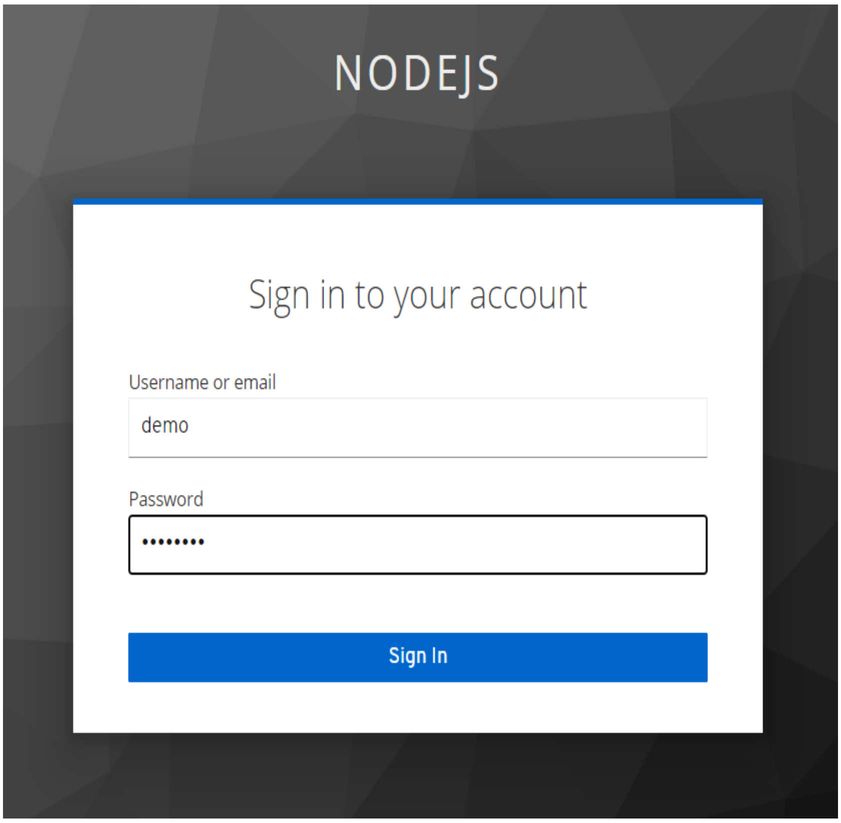
LOGIN

Welcome to Keycloak with Node Js and EJS

This is Un-protected page......

© 2021 Made by Sujith K 😉

5. Enter the username “demo” and password “password” of the user which we created.



NODEJS

Sign in to your account

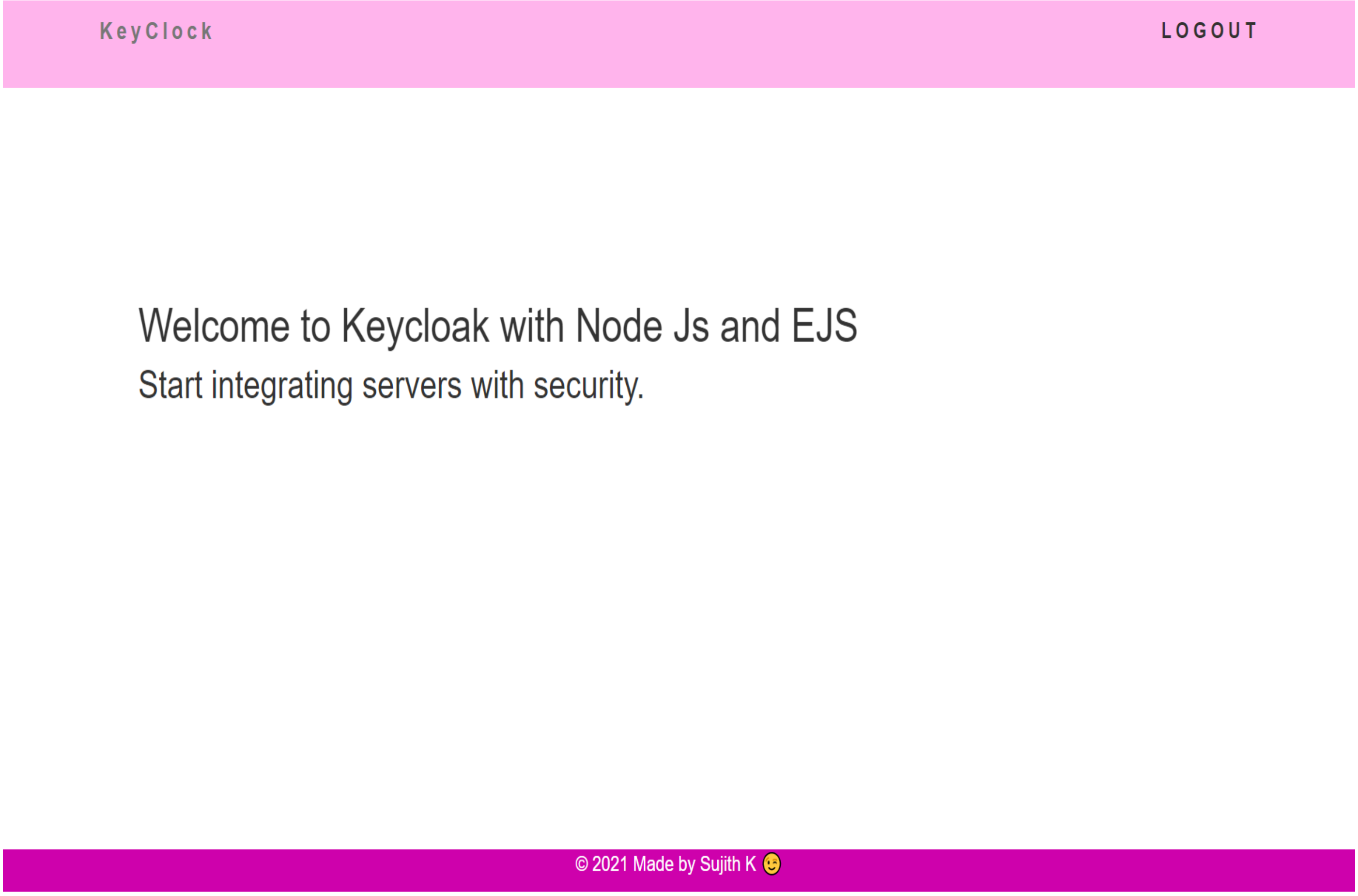
Username or email

demo

Password

••••••••

6. Once you are signed in you will be redirected to the protected page like this:

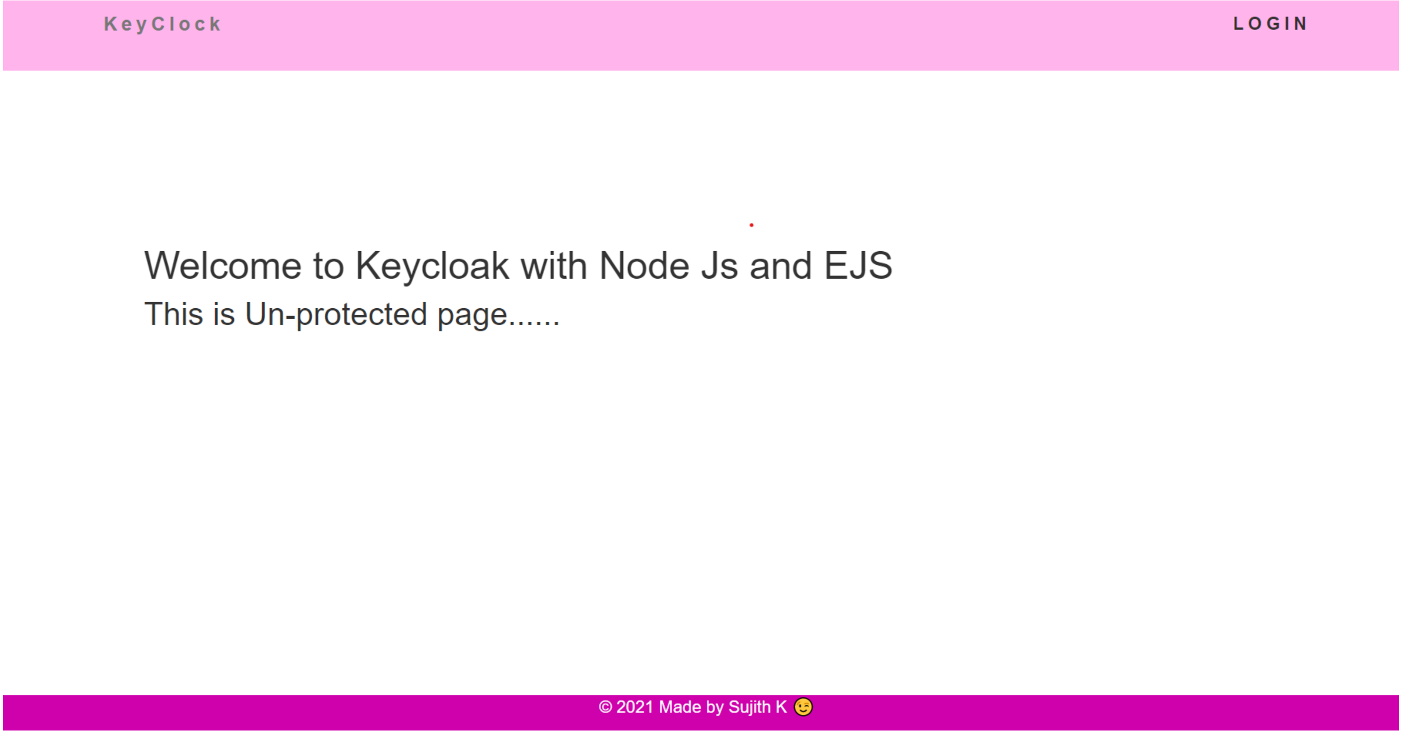


Welcome to Keycloak with Node Js and EJS

Start integrating servers with security.

© 2021 Made by Sujith K 😉

7. Once we click on logout it will redirect to unprotected page and session will be destroyed like below:



Welcome to Keycloak with Node Js and EJS

This is Un-protected page......

© 2021 Made by Sujith K 😉

Note: This code is not capable for production because there is data leak in the session memory store for this, we have to configure the Redis or MongoDB to manage the session. That can be done by adding the configurations to this code by referring the stack overflow: https://stackoverflow.com/questions/10760620/using-memorystore-in-production