

## Keycloak Configuration

## **For**

## OJAS HIRING MANAGEMENT SYSTEM

## Ver no. 2.0

**Disclaimer: This document is classified as 'internal' and used by Ojas Innovative Technologies Pvt. Ltd. employees only until and unless it is specified otherwise.**



# Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver. No.** | **Date**  **(dd-Mmm-yyyy)** | **Name of the Author** | **Change Information** |
| 0.1 | 6-March-2024 | **Adapa Ganesh** | Initial draft version |
| 1.0 | 6-March-2024 | **Adapa Ganesh** | Baselined after receiving approval |
| 2.0 | 9-March-2024 | Nagarjuna Reddy Mekala | Implemented the Ojas template information |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver. No.** | **Date**  **(dd-Mmm-yyyy)** | **Name of the Reviewer** | **Name of the Approver** |
| 0.1 | 7-March-2024 | Venkata Ravindranath Gundugola | Venkata Ravindranath Gundugola |
| 1.0 | 7-March-2024 | Venkata Ravindranath Gundugola | Venkata Ravindranath Gundugola |
| 2.0 | 9-March-2024 | Venkata Ravindranath Gundugola | Venkata Ravindranath Gundugola |
|  |  |  |  |
|  |  |  |  |



# Template Revision History (For SEPG only)

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver. No.** | **Date**  **(dd-Mmm-yyyy)** | **Name of Author** | **Change Information** |
| 0.1 | 30-Nov-2021 | Vanima Prasad | Initial draft copy |
| 1.0 | 24-Dec-2021 | Vanima Prasad | Baselined after Receiving Approval |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver. No.** | **Date**  **(dd-Mmm-yyyy)** | **Name of the Reviewer** | **Name of the Approver** |
| 0.1 | 15-Dec-2021 | Srinivas Bollapragada | Srinivas Bollapragada |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**03-01-2022**

**Effective Date**

**(Release date of the QMS document)**



***Table of Contents***

1.0 Keycloak API Introduction 5

2.0 Keycloak Installation process…………………………………………………………….7

**Keycloak API Introduction**

[K](https://medium.com/%40imsanthiyag/introduction-to-keycloak-admin-api-44beb9011f7d)eycloak is an open-source identity and access management (IAM) solution designed to provide authentication, authorization, and user management for applications. It simplifies the process of securing web applications, mobile apps, and services through a range of features. Here's an overview of what Keycloak does:

Key Features:

1. Single Sign-On (SSO): Keycloak enables users to log in once and gain access to multiple applications without needing to re-enter credentials.
2. Identity Brokering: It allows integration with third-party identity providers like Google, Facebook, and GitHub, enabling users to log in via their existing accounts.
3. User Federation: Keycloak can connect to external user databases or directories, such as LDAP or Active Directory, allowing applications to authenticate users stored in those systems.
4. OAuth 2.0 & OpenID Connect Support: It natively supports industry-standard authentication protocols like OAuth 2.0, OpenID Connect, and SAML, making it easy to integrate with modern applications.
5. Social Login: Users can log in to applications using their social media accounts (e.g., Google, Twitter).
6. Role-Based Access Control (RBAC): It supports defining roles and permissions, enabling fine-grained access control for users.
7. Multifactor Authentication (MFA): Provides additional security by requiring users to authenticate with a second factor (e.g., an SMS code or authenticator app) in addition to their password.
8. User Self-Management: Users can manage their own accounts, change passwords, update profiles, and more, reducing administrative overhead.
9. Admin Console: Keycloak offers an intuitive web-based administration console for managing users, roles, authentication flows, and more.
10. Session Management: It tracks user sessions and supports features like session revocation and logout across applications.

Use Cases:

* Securing Web and Mobile Applications: It provides an easy way to handle user authentication and session management.
* API Security: By integrating with OAuth 2.0 and OpenID Connect, Keycloak secures APIs by issuing tokens to authenticated users.
* Single Sign-On: Enterprises can use Keycloak to implement SSO across various internal and external applications.
* Custom Authentication Flows: Developers can create custom authentication mechanisms and workflows to meet specific requirements.

Keycloak is popular among developers and organizations because it simplifies user authentication, reduces the need for custom security code, and is easily extensible for specific use cases. It is developed by Red Hat and is widely used in enterprise environments due to its flexibility and scalability.

## Keycloak Configuration:

**Operating System : Windows 10**

## Java version : 1.8

**Keycloak Version : 14.0.0**

## ------------------------------------------------------------------------------------------------------------------------------------------

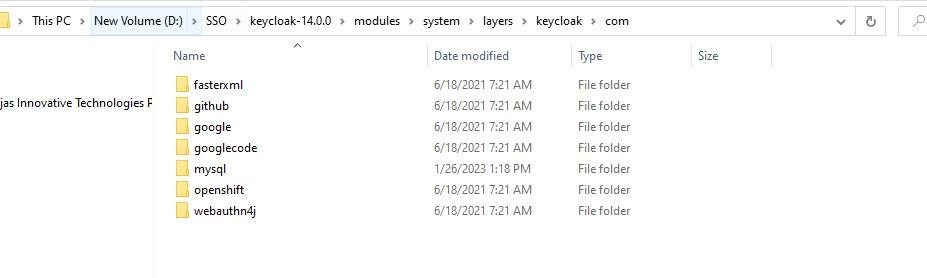
**Keycloak Installation process:**

**Keycloak datasource setup:**

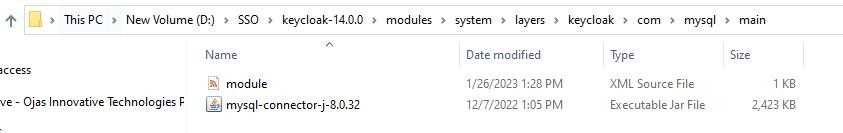
## By default it will be pointing to H2 data base, we can change it to our own database(Mysql, etc..) based on our requirement.

**Mysql :**

## Java connector setup:



Inside the above folder we need to create folder name as “mysql” -> create sub folder name as “main” Here we need to add two files,



## module.xml :

<?xml version="1.0" encoding="UTF-8"?>

<module xmlns="urn:jboss:module:1.3" name="com.mysql">

<resources>

<resource-root path="mysql-connector-j-8.0.32.jar"/>

</resources>

<dependencies>

<module name="javax.api"/>

<module name="javax.transaction.api"/>

</dependencies>

</module>

## Mysql java connector:

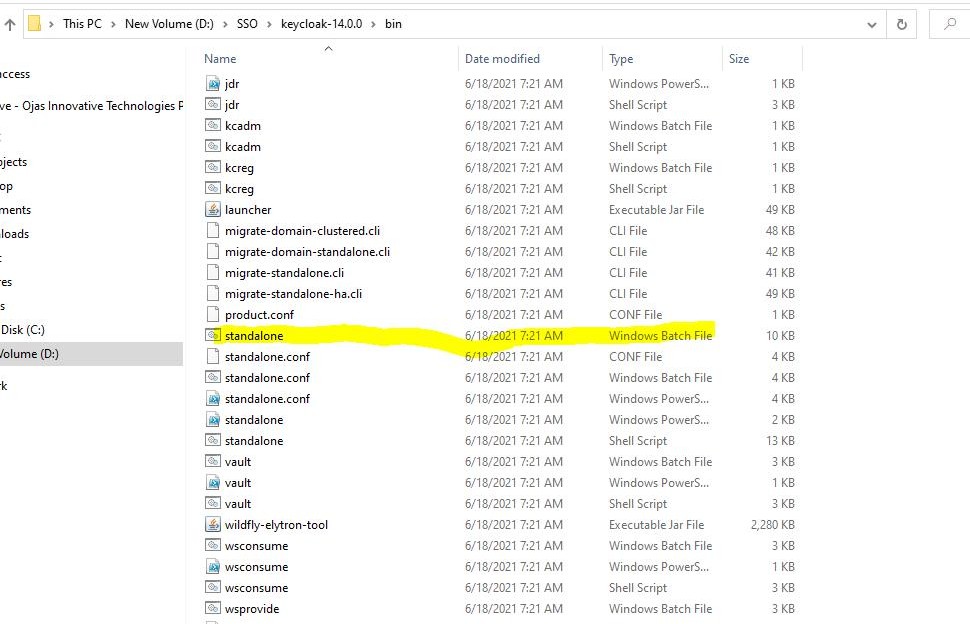
**Visit : https://dev.mysql.com/downloads/connector/j/**

We can download mysql-java-connector specific to your local database version(8.0.x, etc..) Note : while downloading select platform independent Option form the dropdown

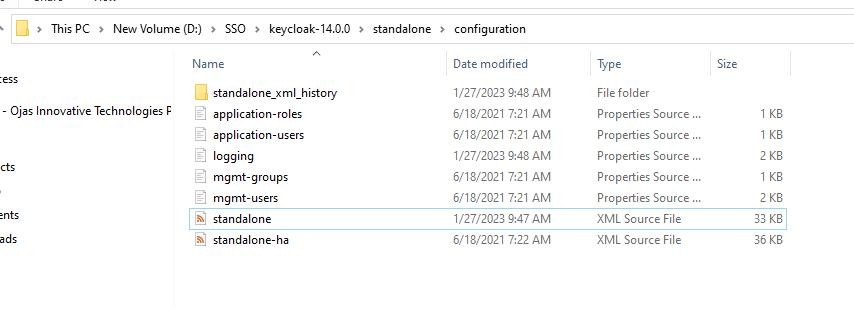
## Running Keycloak :

Note : Before running the below file, we need to create database name as “keycloak” defined in the standalone.xml file for the creation of database table data.

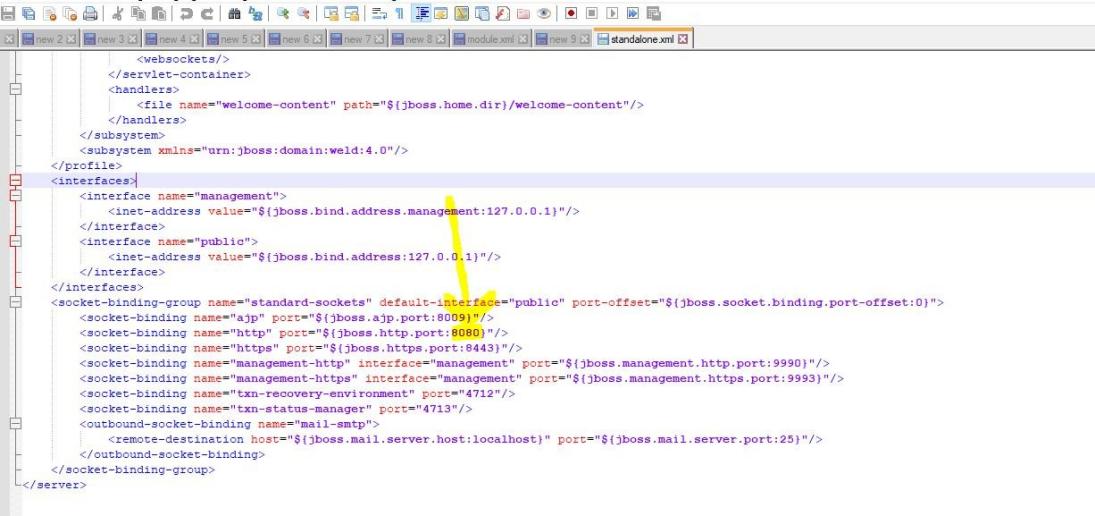
From the above path we can start Keycloak application by executing: **standalone.bat** file,



By default, the port number is: 8080, we can change it based on our requirement inside standalone.xml

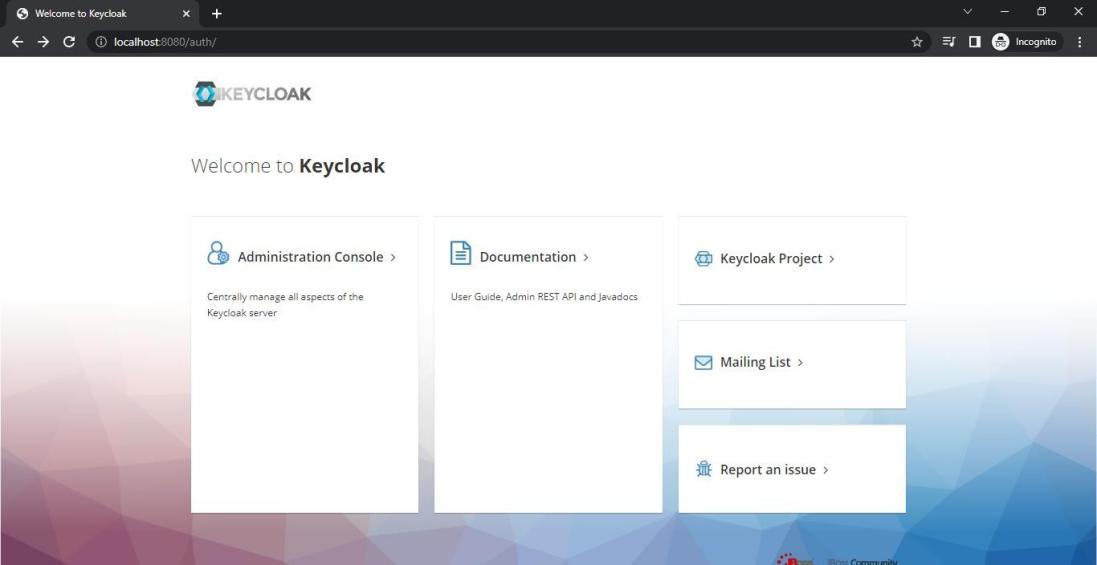


as shown below



## Keycloak Page on the browser:

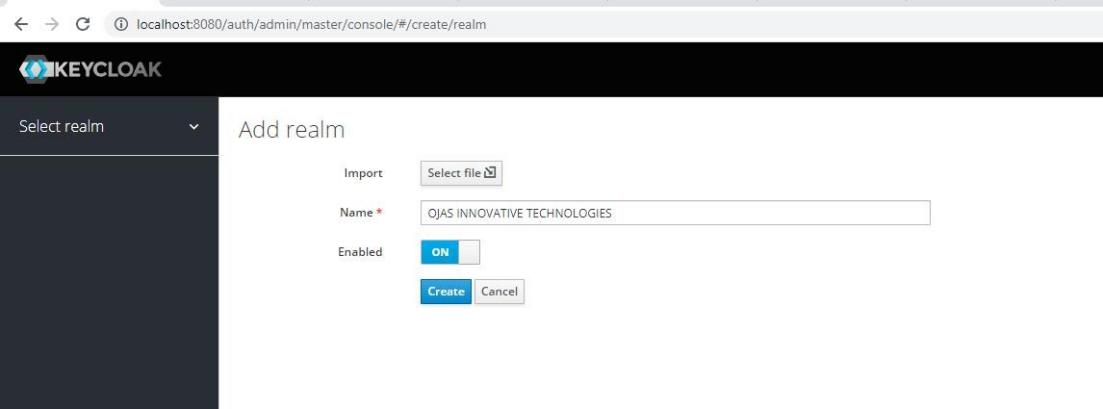
After that we can hit : http://localhost:8080 -> it will be redirected to Admin console account creation of keycloak, we can register from that and we can login into Admin console.



## Keycloak Realm creation :

After successful login into keycloak as an Admin, we can create realm

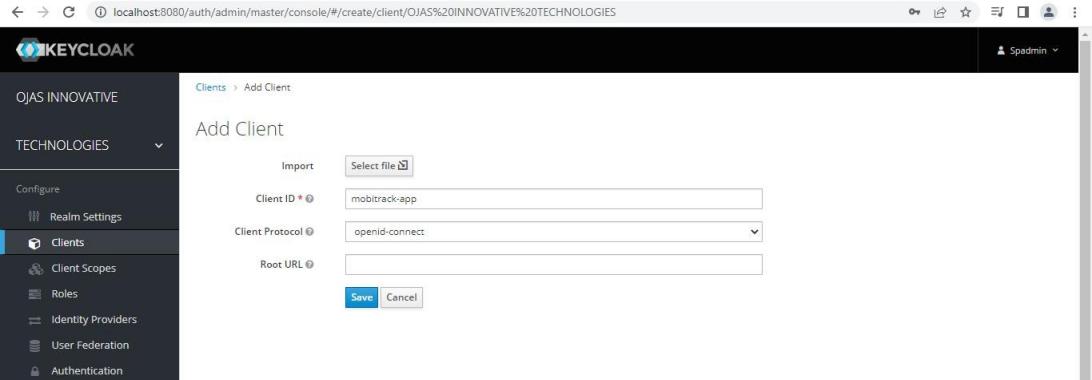
**Realm :** is central place to have **Users, credentials, Roles**



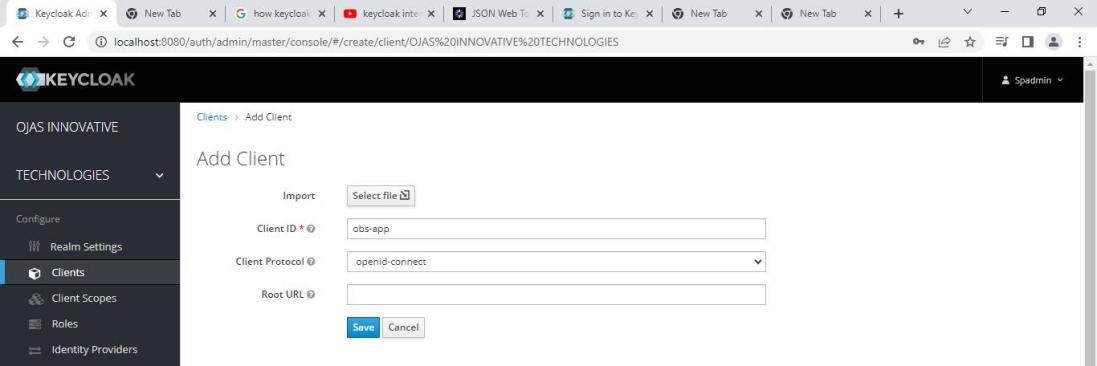
**Clients creation :** Application which wants to use keycloak is a client

Here I have take two applications, mobitrack-app and obs-app for example....

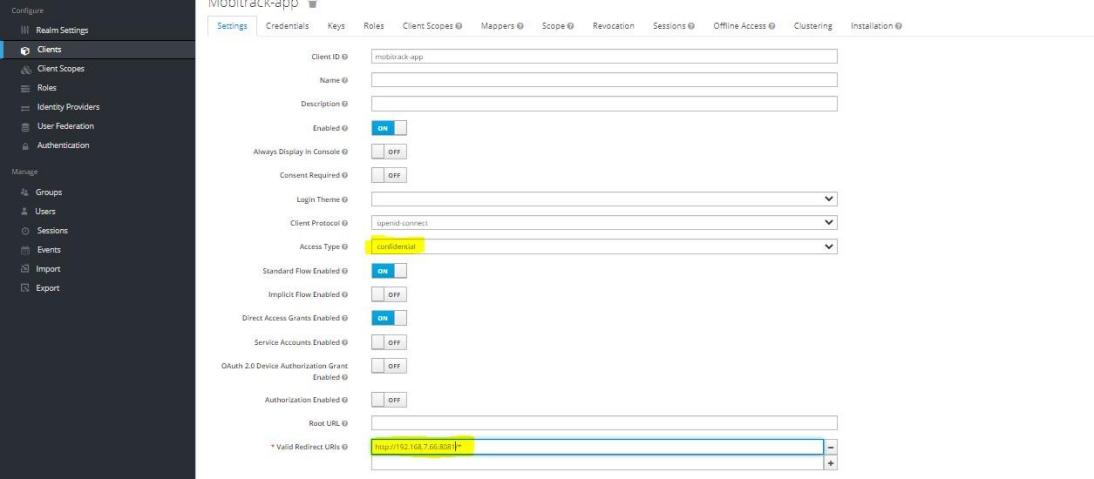
1. Mobitrack-app



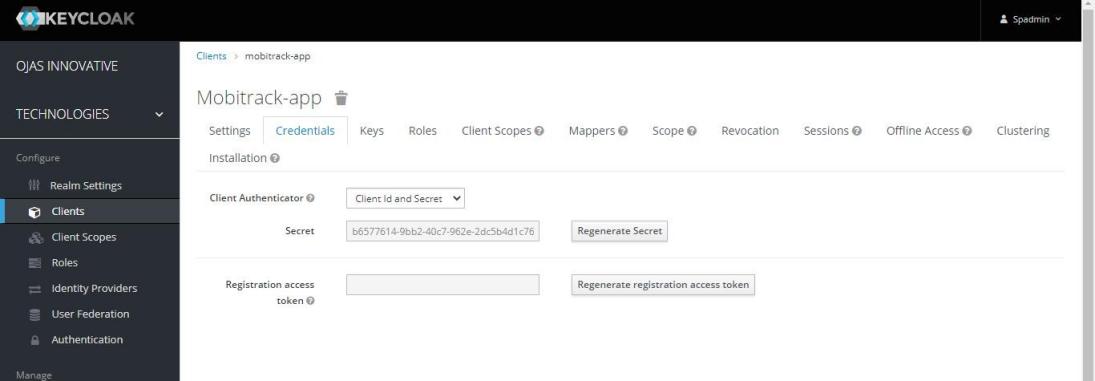
1. Obs-app



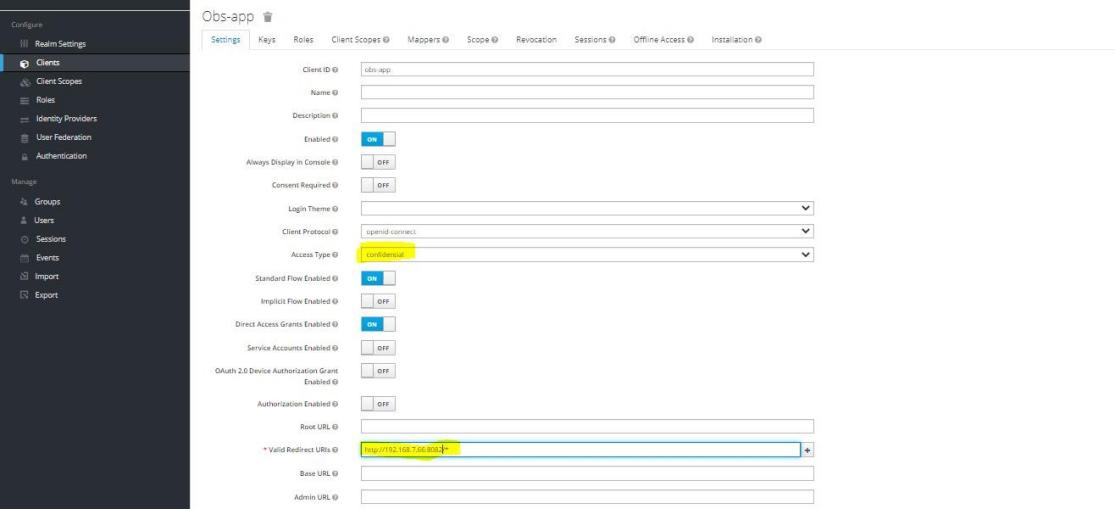
For both the applications change the access type to **confidential and redirect url of application,**



After that we can see “secret” inside credentials tab,

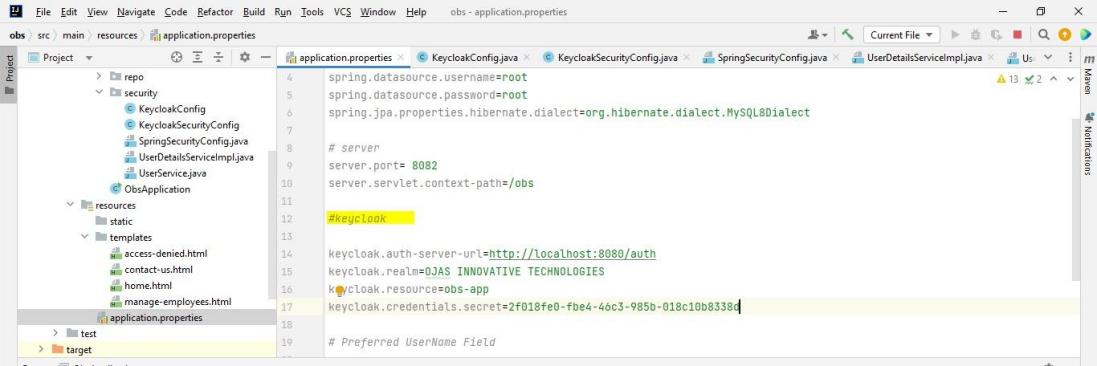


We have to do the same for obs-app,

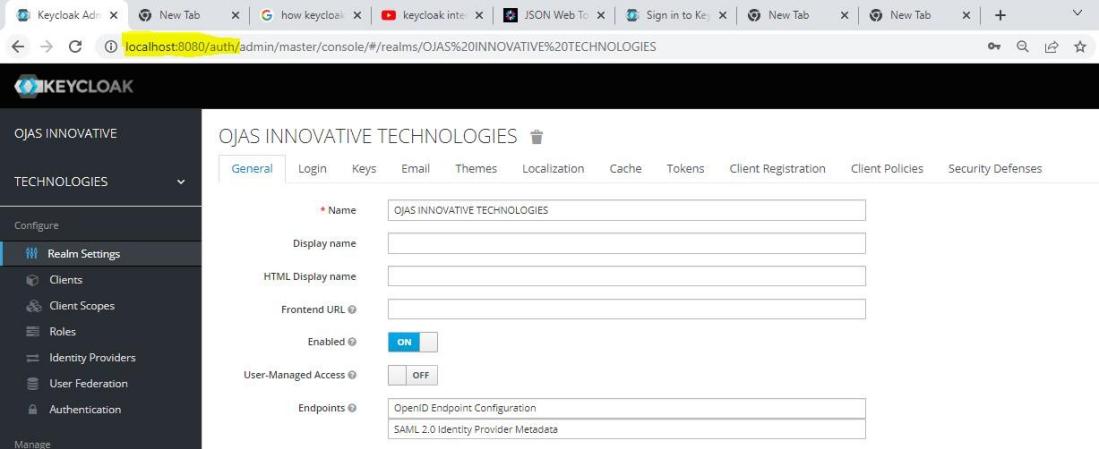


## SpringBootApp - mobitrack – keycloak setup:

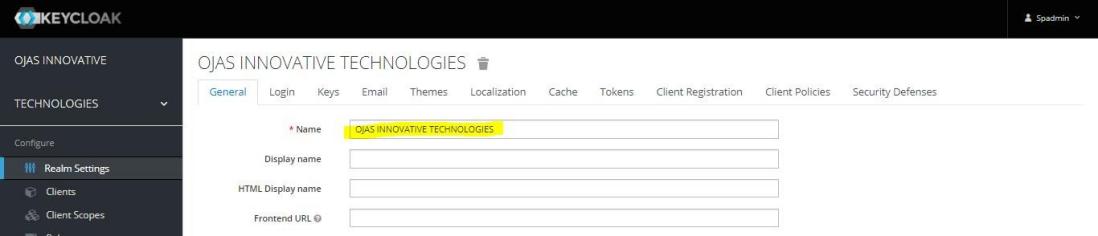
Inside application.properties file we need to set the above parameters, we have to take the configuration information from keycloak.



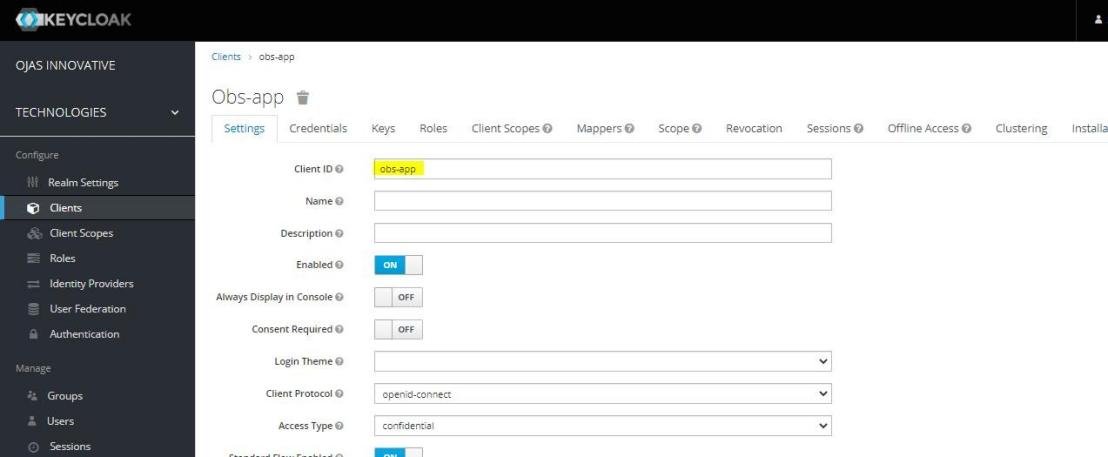
keycloak.auth-server-url:



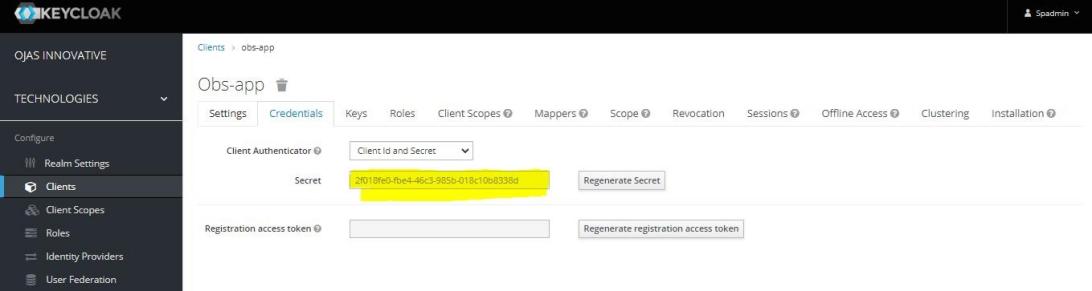
Keycloak.realm:



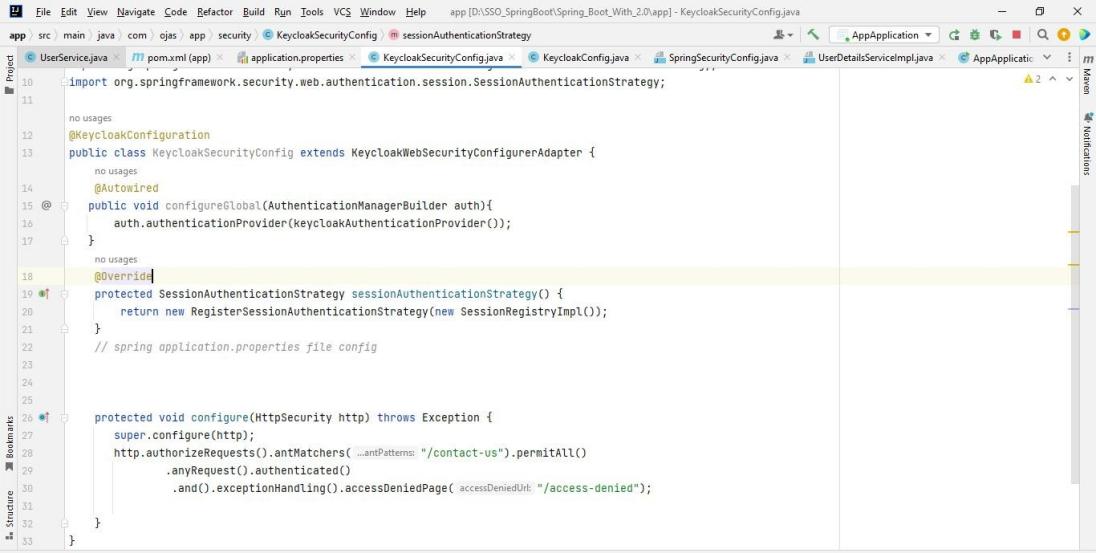
keycloak.resource:



Keycloak.credentials.secret :



We need to create one class for keycloak configuration.



And we need to create one more class for keycloak resolver as mentioned in the above screen shot.



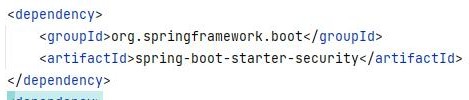


Two end points with autorization, here @PreAuthorize can handle Authorization mechanism for that we need to create roles inside keycloak with same name, here in my case I am going to create two Roles,

* + EMPLOYEE
  + MANAGER

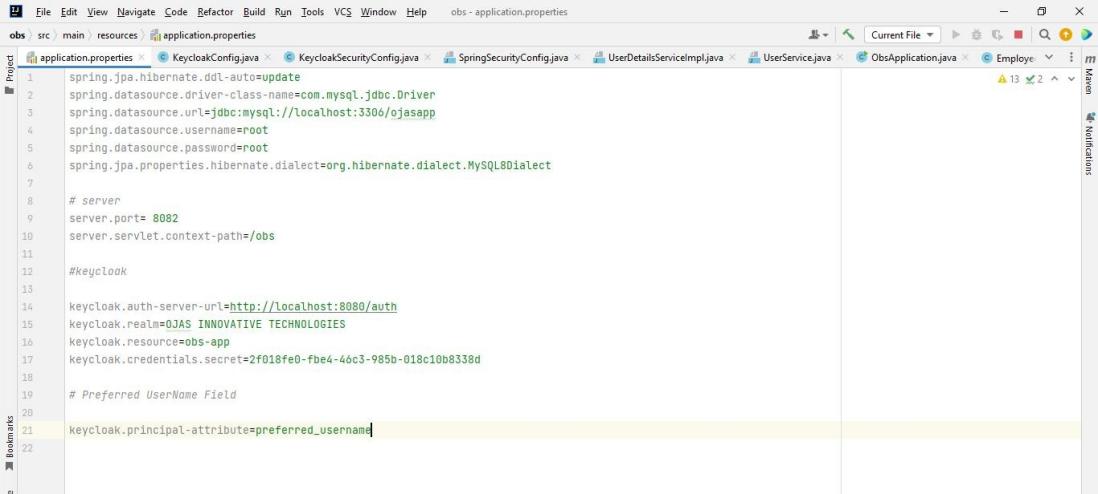
The below dependencies are required for keycloak and spring boot security



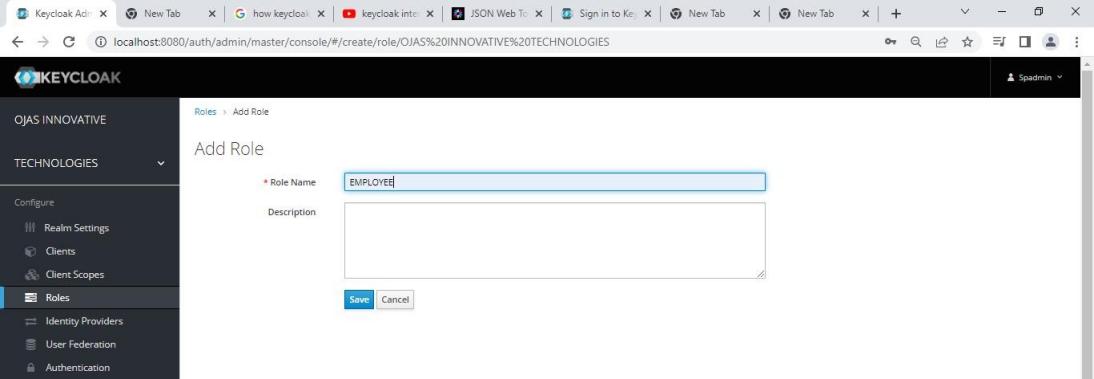


## SpringBootApp : obs–app keycloak setup:

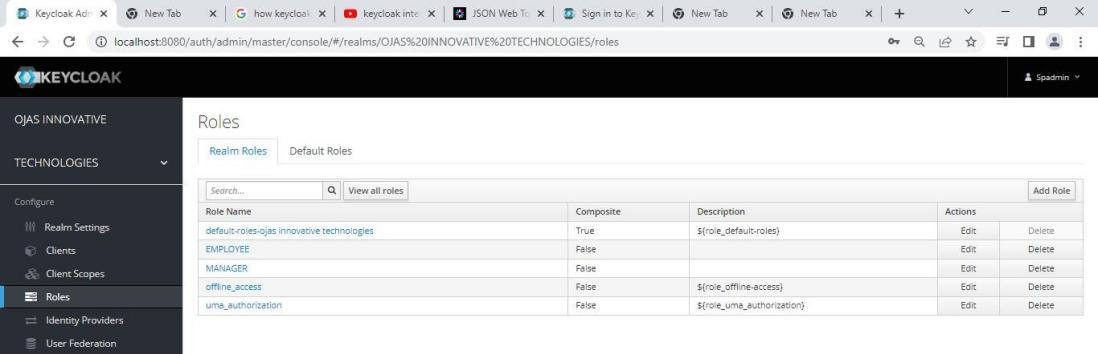
Same method for obs-app as well, only changes required in the application.properties file specific to obs-app,



## Keycloak Roles creation:

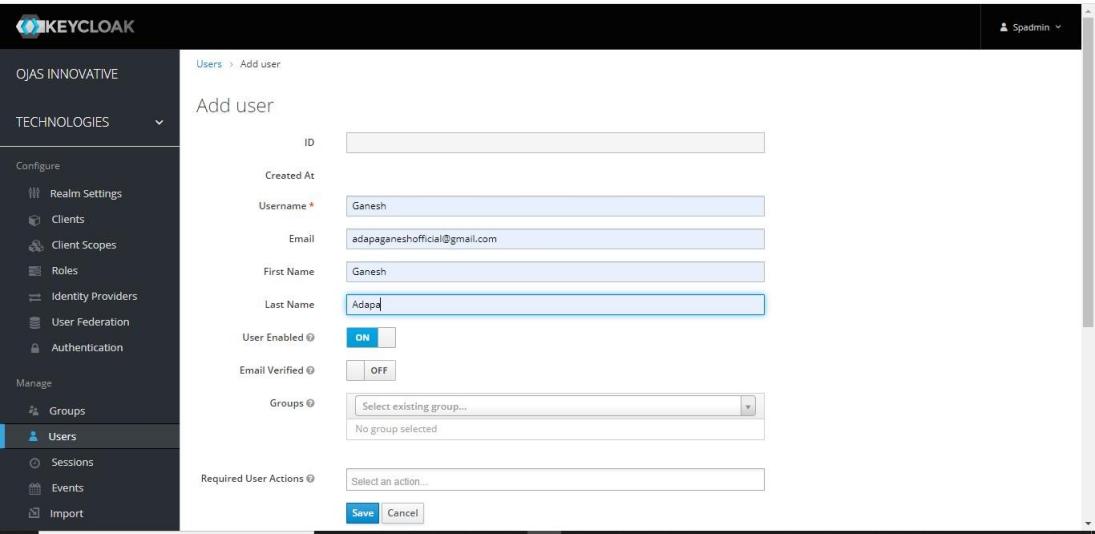


Same for MANGER,

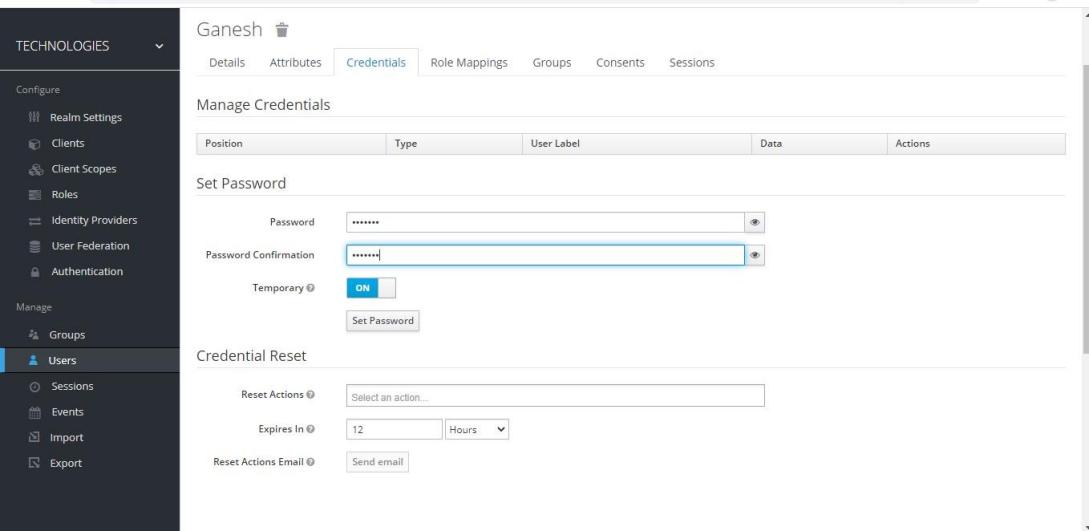


## Creating Users :

**i) Admin Login :**



Temporary password creation:

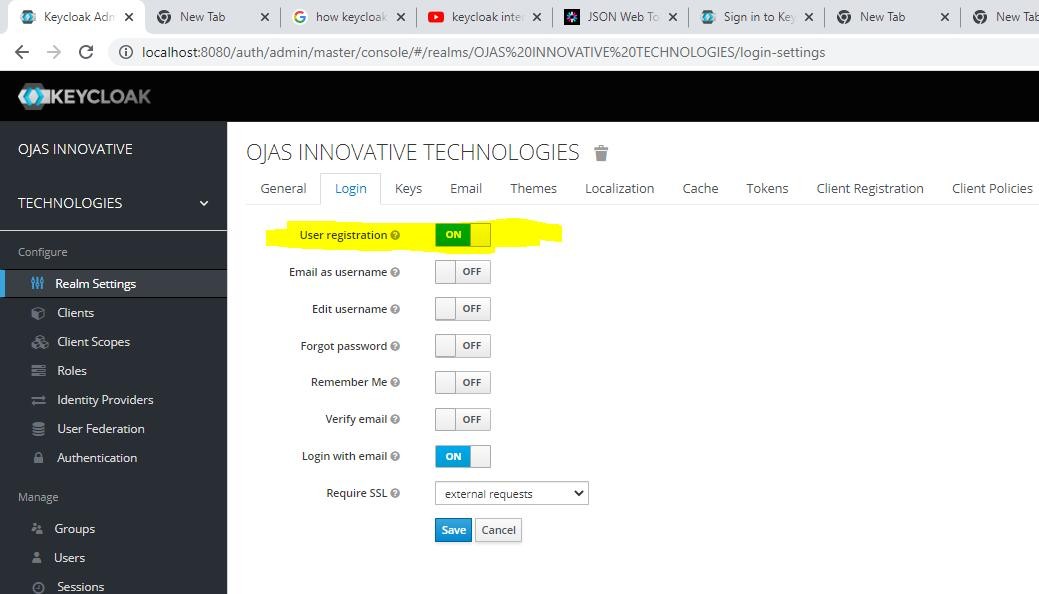


Next time the user login with that credentials he can change the password if required.

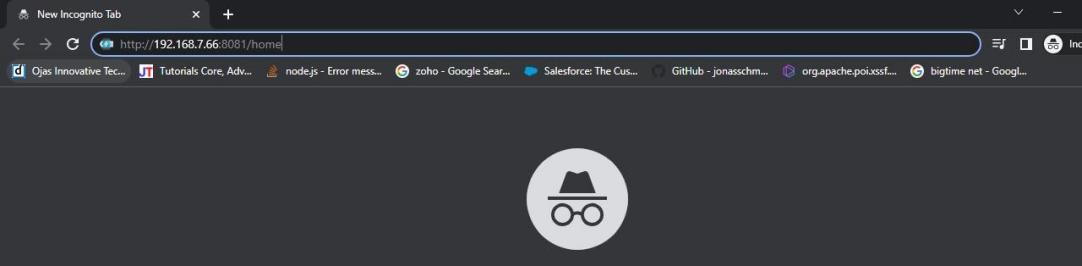
Or

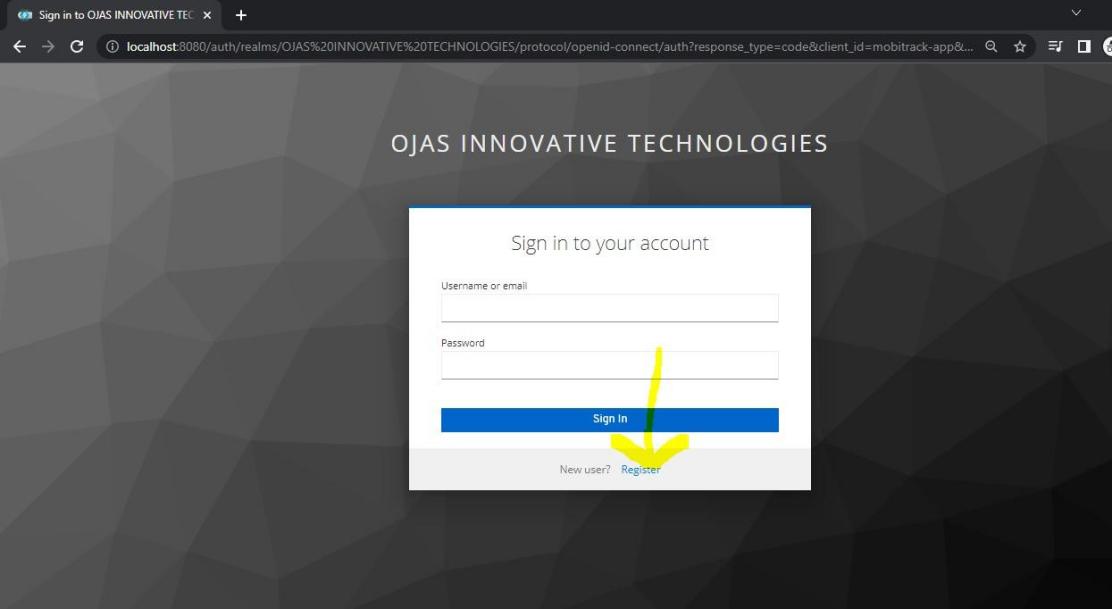
## User Self Registration:

We have to unable the User Registration, so that the user can able to register.

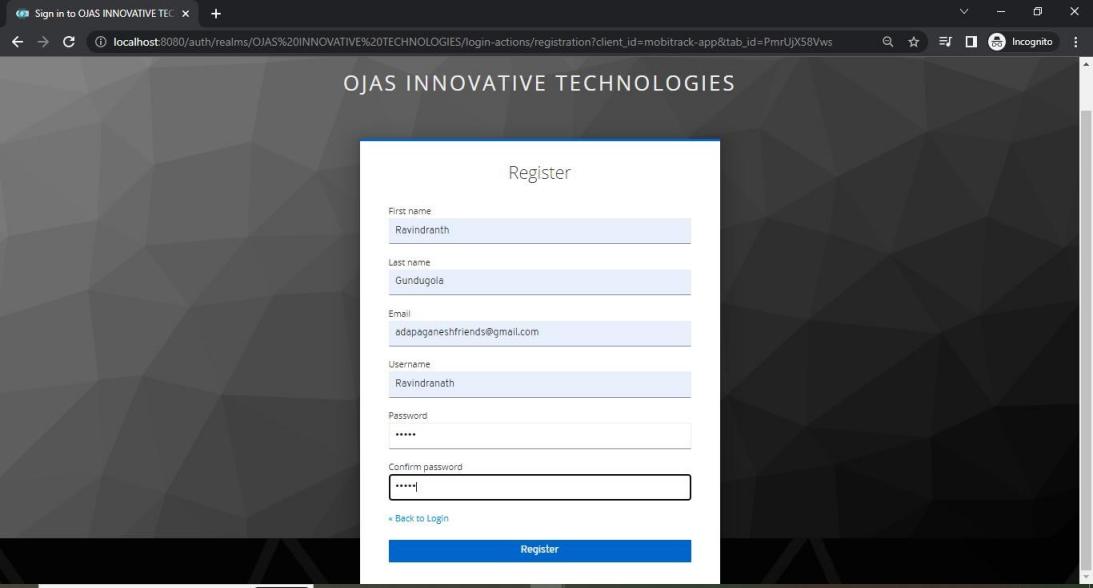


Testing the keycloak by hitting : http://192.168.7.66:8081/home

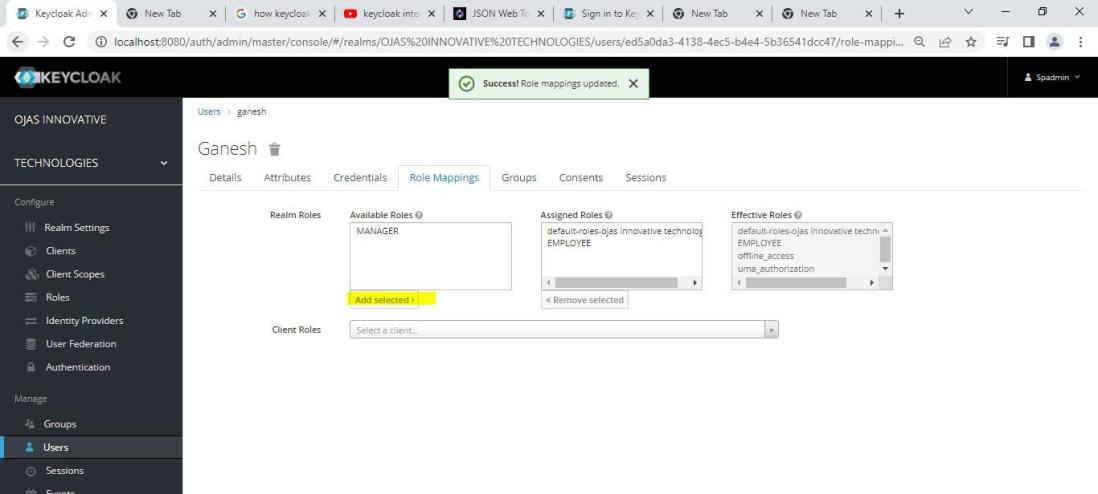


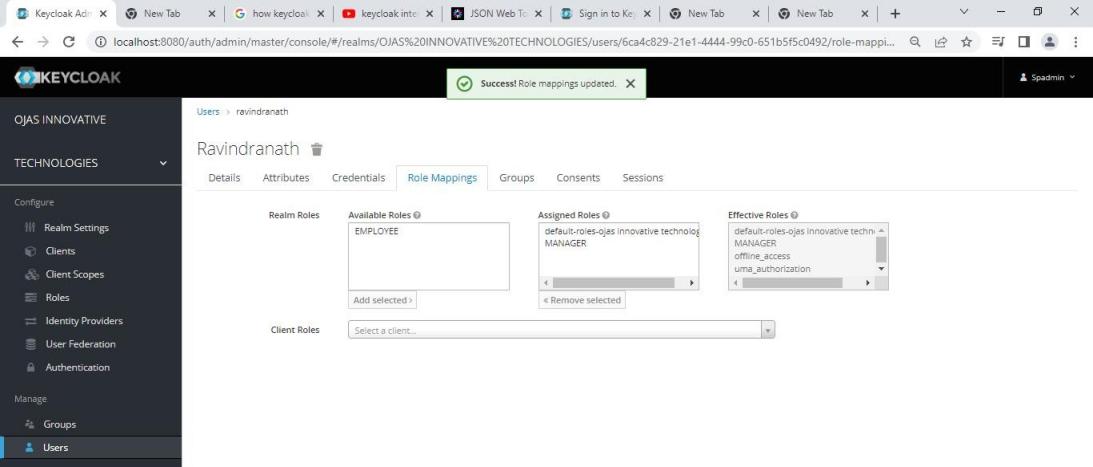


By using the above Register hyperlink, the user can able to register.



## Roles Assigning to the users:

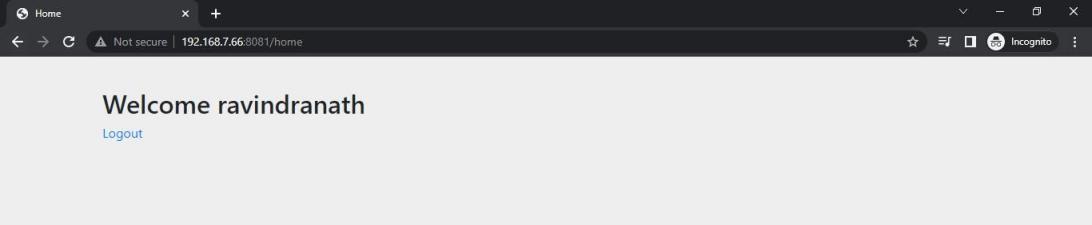
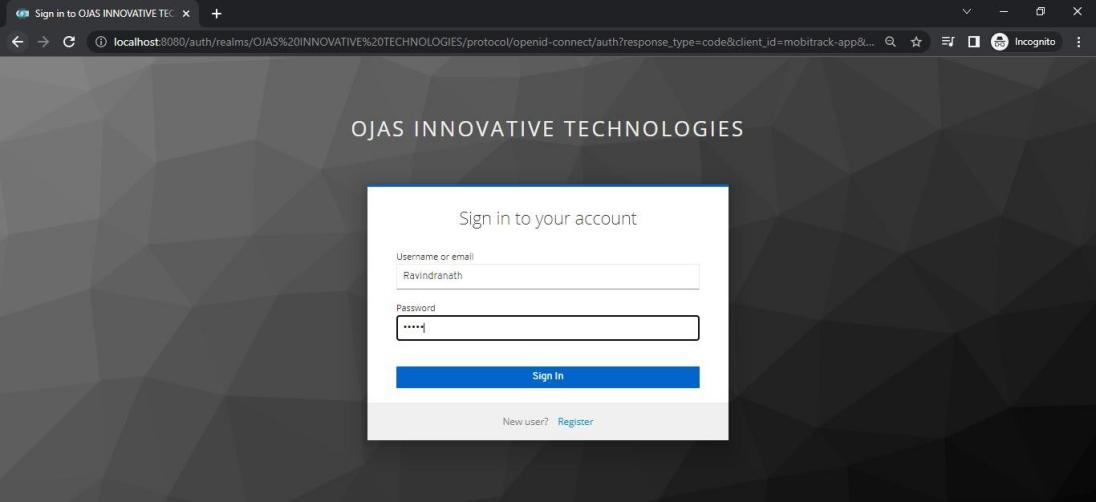


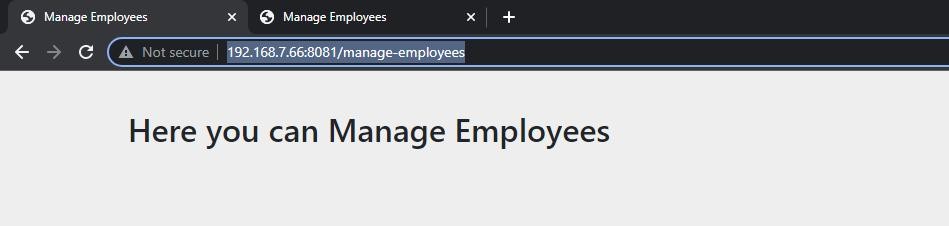


**Login using Keycloak :**

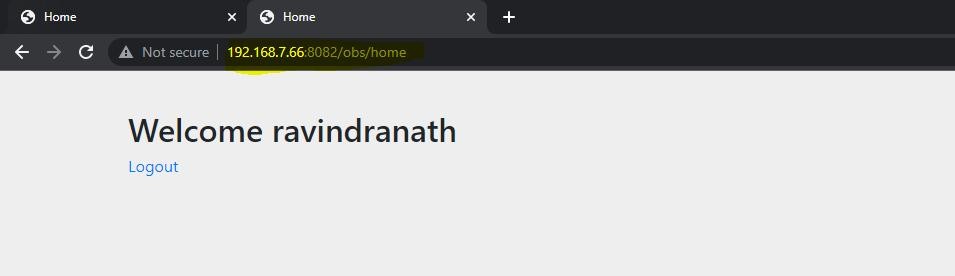
**Ravindranth – MANAGER,** Login as a Manger, he can able to access all the pages.

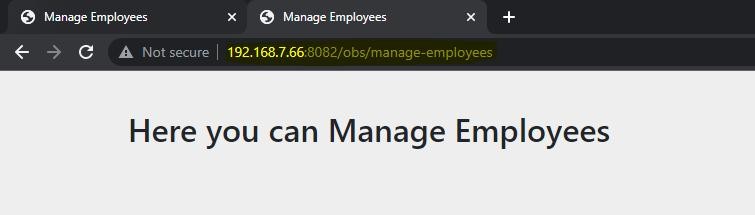
Testing the keycloak by hitting the url for : http://192.168.7.66:8081/home or http://192.168.7.66:8082/obs/home





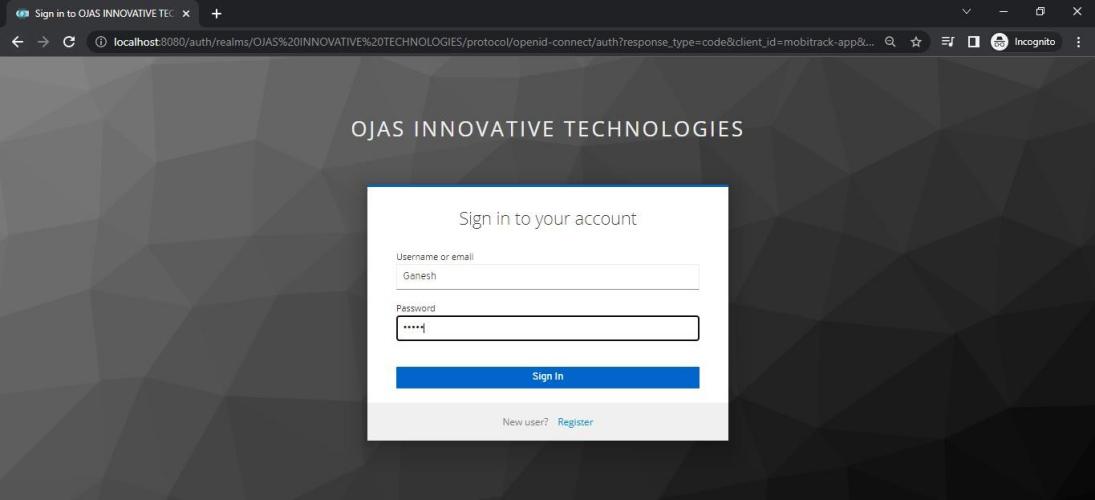
and



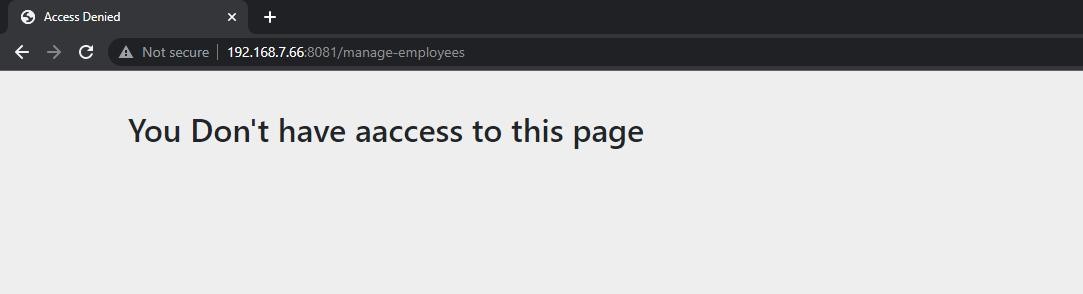
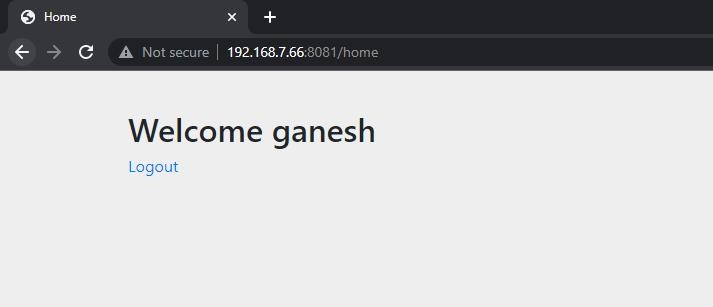


So it won’t ask again you to login for accessing obs-app, why because single sign on is working as expected.

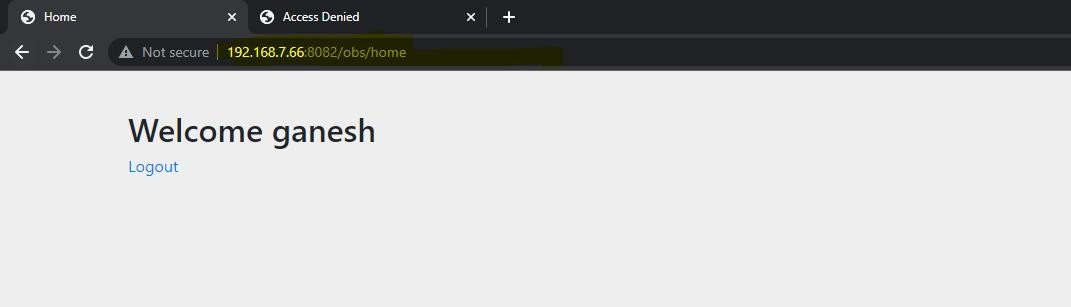
**Ganesh – EMPLOYEE,** Login as a Employee, he can able to access all the pages.

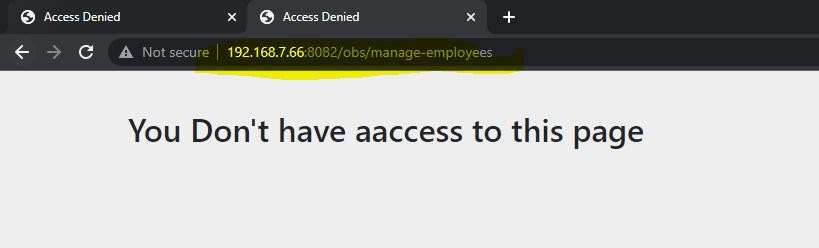


## Mobitrack-app :

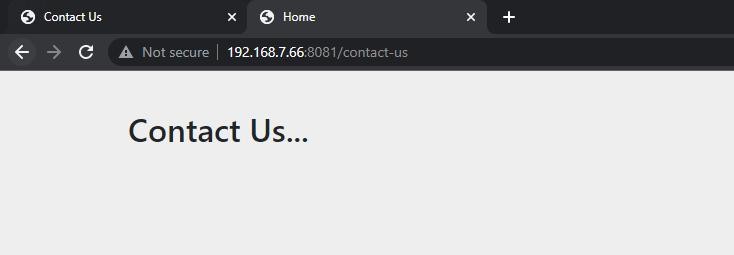
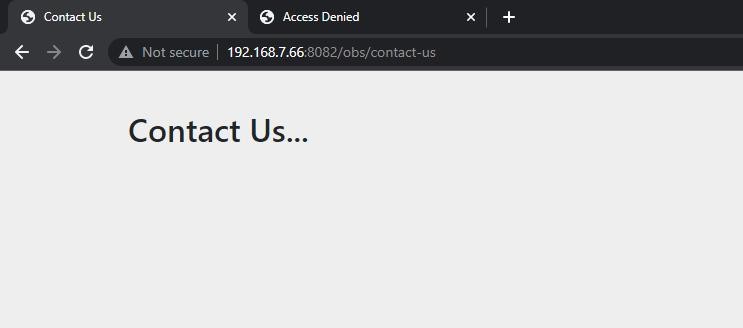


**Obs-app:**



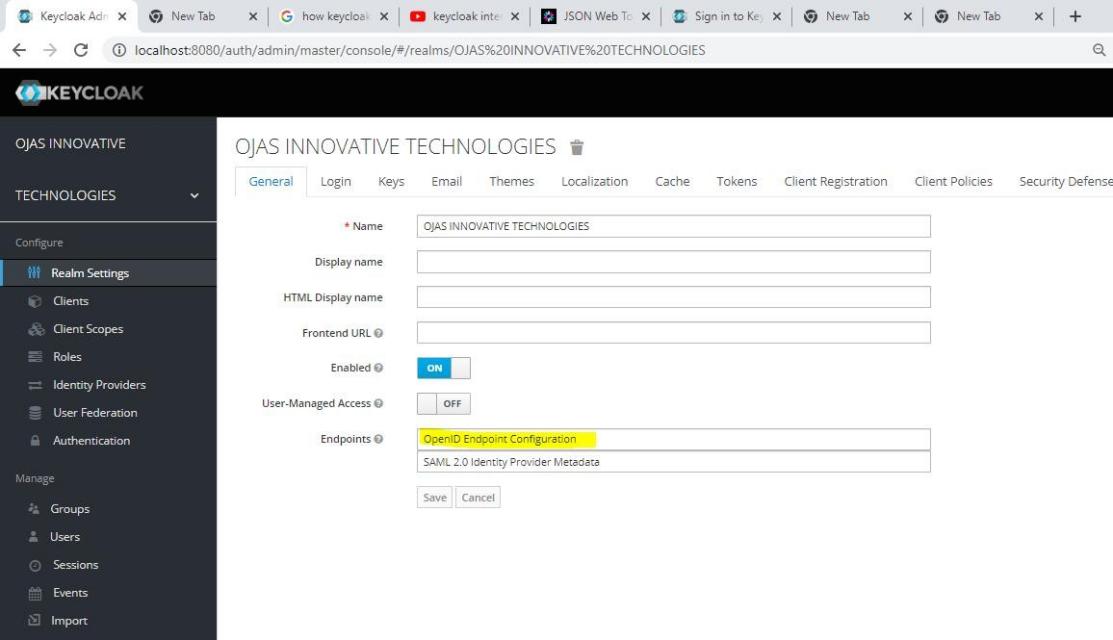


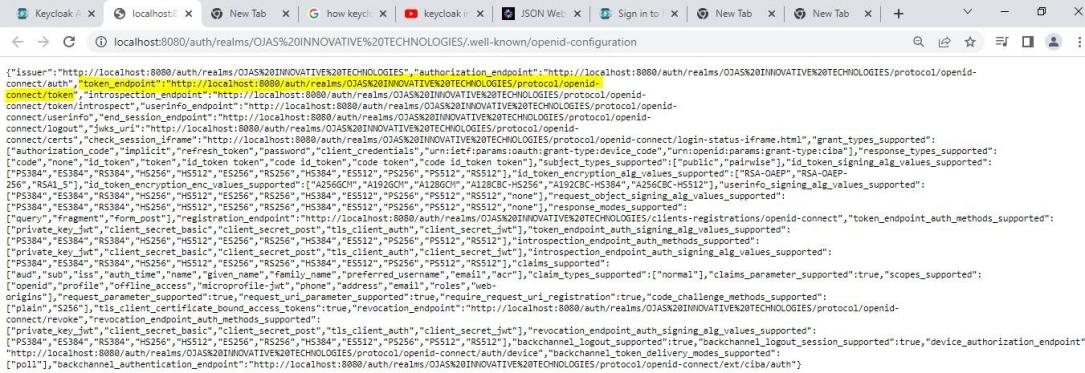
**Public pages** : contact us... everyone can access, here no authorization,



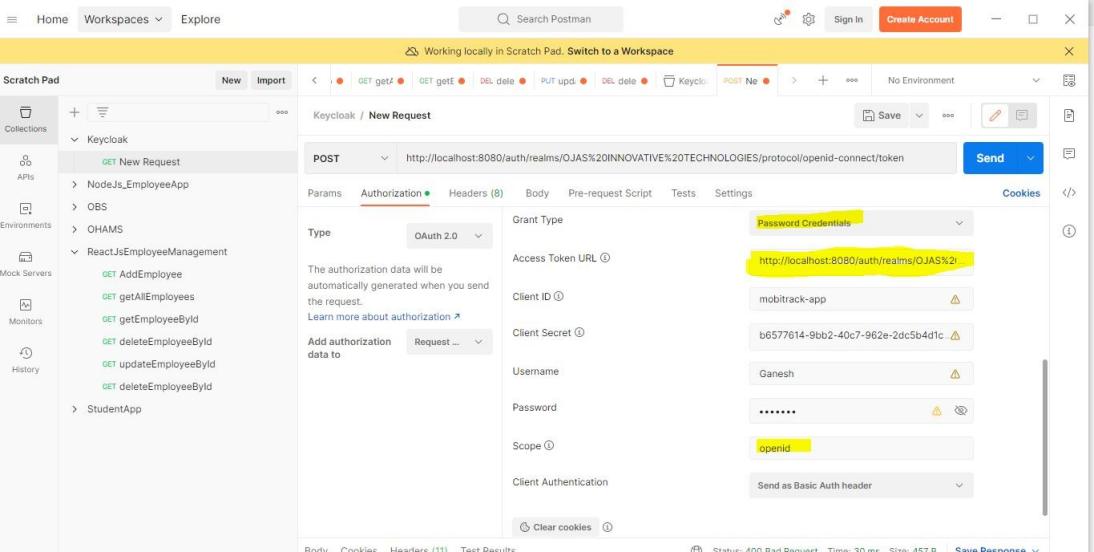
## PostMan Testing:

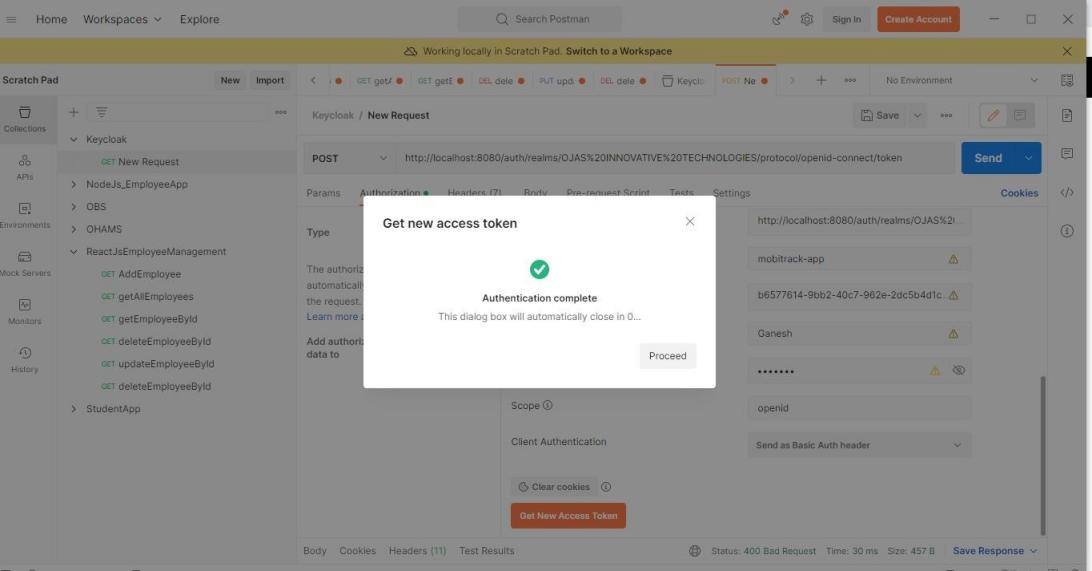
We have to go inside that Endpoints, there we can find token url,

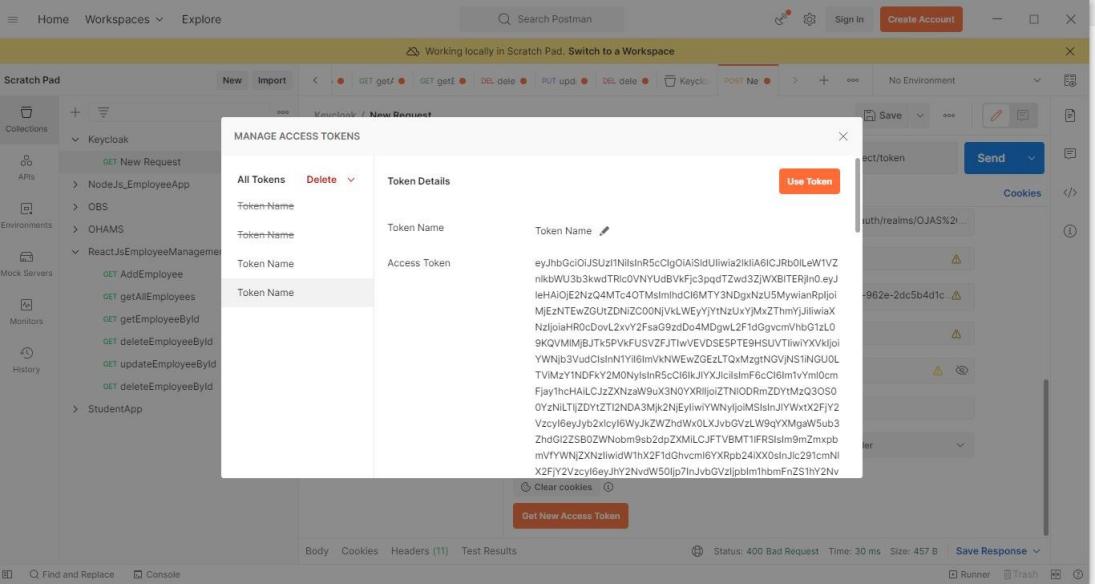




We can use the above highlighted url for testing in postman, Here we need to provide Authorization type as : OAuth2.0







After this, we can verify this in jwt decoder,

