

WORKSHOP STN - RED HAT SINGLE SIGN ON - ANGULAR - SPRINGBOOT

Esse tutorial tem por objetivo mostrar uma solução integrando o **SSO** com uma aplicação de backend escrita em java com Spring Boot e duas aplicações de Frontend escritas em NodeJs com Angular.

Todo o material pode ser baixado do repositório GIT <https://github.com/mgamarra/poc-keycloak-angular-springboot>

Requisitos.

- Visual Code
- java openjdk 11.0.8
- maven 3.6.3
- npm -v 6.14.4
- node v12.16.1
- docker 19.03.6

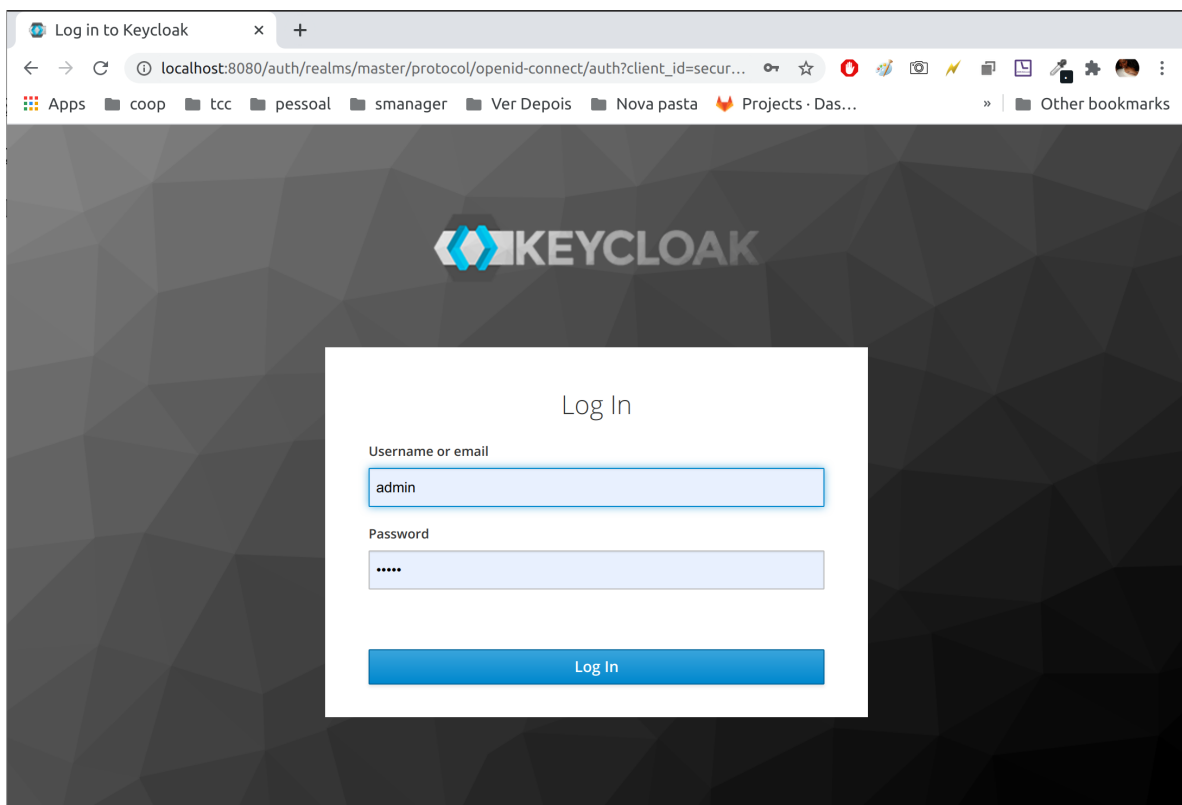
Docker

Subir SSO em uma imagem Docker.

```
docker run -p 8080:8080 -e KEYCLOAK_USER=admin -e KEYCLOAK_PASSWORD=admin --name sso -d quay.io/keycloak/keycloak:11.0.2
```

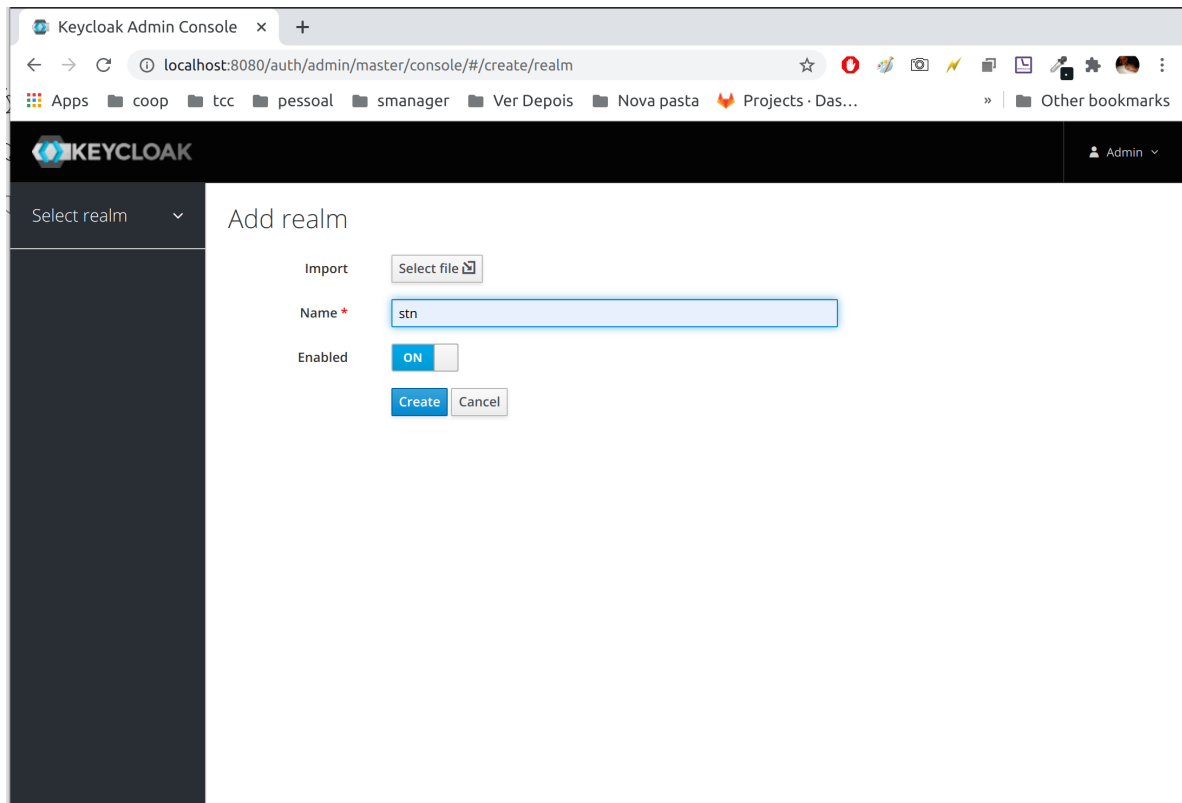
Passo 1. Criação do Realm no SSO

Acesse a console administrativa do SSO no endereço <http://localhost:8080/auth/> e faça login utilizando as credenciais de administrador, usuário **admin** e password **admin**.



Realm

No menu superior esquerdo, selecione a opção **ADD REALM** e informe o nome **stn** em **minúsculo**.



Passo 2. Criação do Client

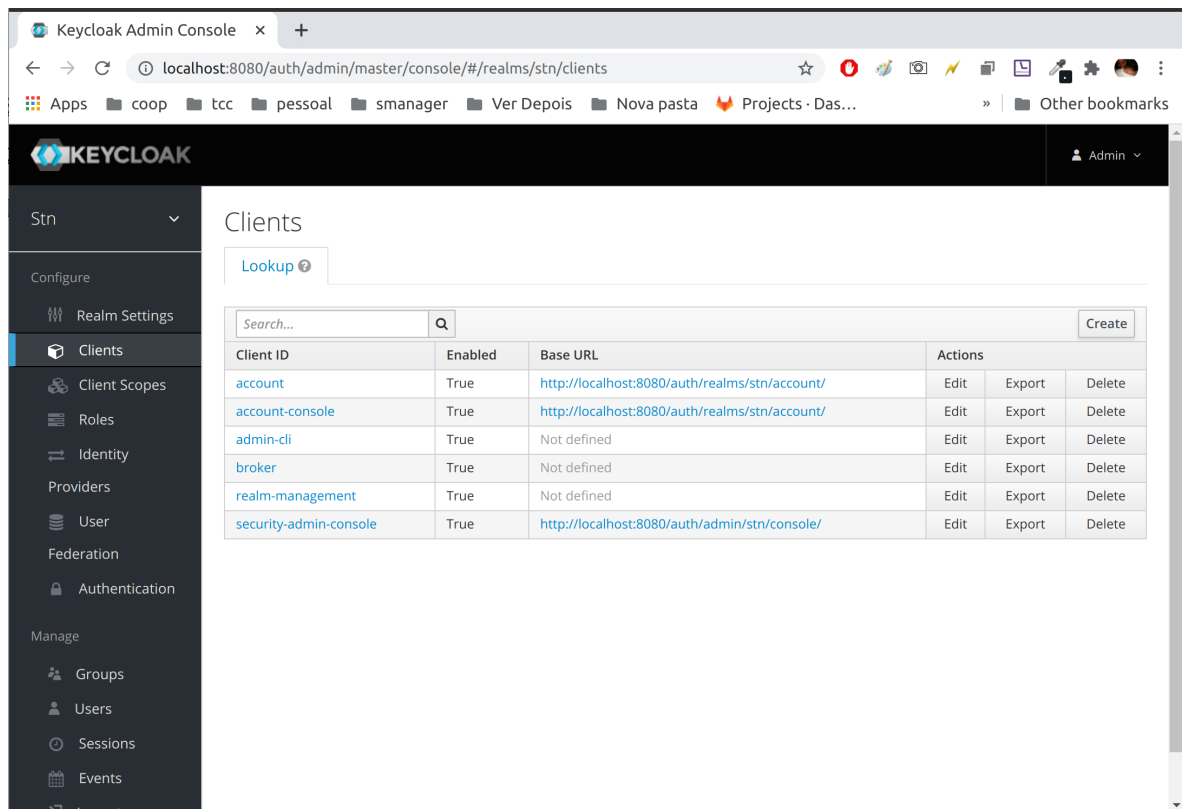
Vamos utilizar dois clients que terão objetivos distintos conforme tabela abaixo:

Client	Objetivo
spa-stn-backend	Utilizado pela aplicação de backend para autorização
spa-stn	Utilizado pela aplicação de frontend para autenticação e autorização

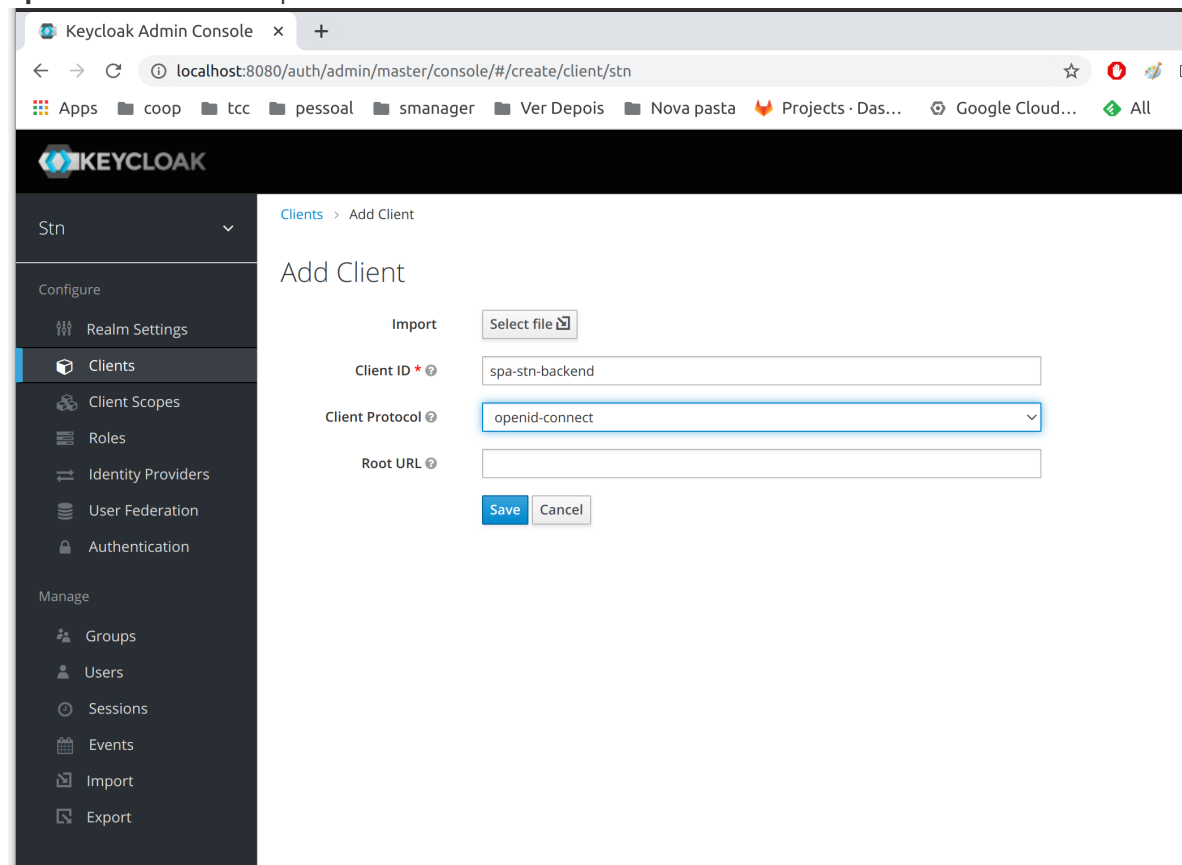
Crie o Client spa-stn-backend

Vamos criar primeiro o client **spa-stn-backend**

No menu **Clients** acesse a opção **Create**



Preencha os campos com as seguintes informações **Client ID** *spa-stn-backend* , **Client Protocol** *openid-connect* e clique em **save**



Após salvar, a tela de configuração do *client* será exibida. Certifique-se que a opção **Enabled** está configurada como **ON** e o **Access Type** como **bearer-only**.

Aplicações REST apenas validam o token, nunca iniciam o fluxo de login do SSO. Clique em salvar para concluir a etapa.

Keycloak Admin Console

localhost:8080/auth/admin/master/console/#/realms/stn/clients/4fdd9e1c-572e-49e2-9333-1f84d72db714

KEYCLOAK

Admin

Stn

Configure

- Realm Settings
- Clients**
- Client Scopes
- Roles
- Identity Providers
- User Federation
- Authentication

Manage

- Groups
- Users
- Sessions
- Events
- Import
- Export

Clients > spa-stn-backend

Spa-stn-backend

Settings Roles Client Scopes Mappers Scope Revocation Sessions Offline Access Installation

Client ID: spa-stn-backend

Name:

Description:

Enabled: ☒ ON

Consent Required: ☐ OFF

Login Theme:

Client Protocol: openid-connect

Access Type: bearer-only

Admin URL:

> Fine Grain OpenID Connect Configuration

> OpenID Connect Compatibility Modes

> Advanced Settings

> Authentication Flow Overrides

Save Cancel

Criação de **Client level Roles** para serem utilizadas pelas Aplicações, acesse a aba **Roles** da tela de configuração do **Client** e clique em **Add Role**.

Keycloak Admin Console

localhost:8080/auth/admin/master/console/#/realms/stn/clients/4fdd9e1c-572e-49e2-9333-1f84d72db714/roles

KEYCLOAK

Admin

Stn

Configure

- Realm Settings
- Clients**
- Client Scopes
- Roles
- Identity Providers
- User Federation
- Authentication

Manage

- Groups
- Users
- Sessions
- Events
- Import
- Export

Clients > spa-stn-backend

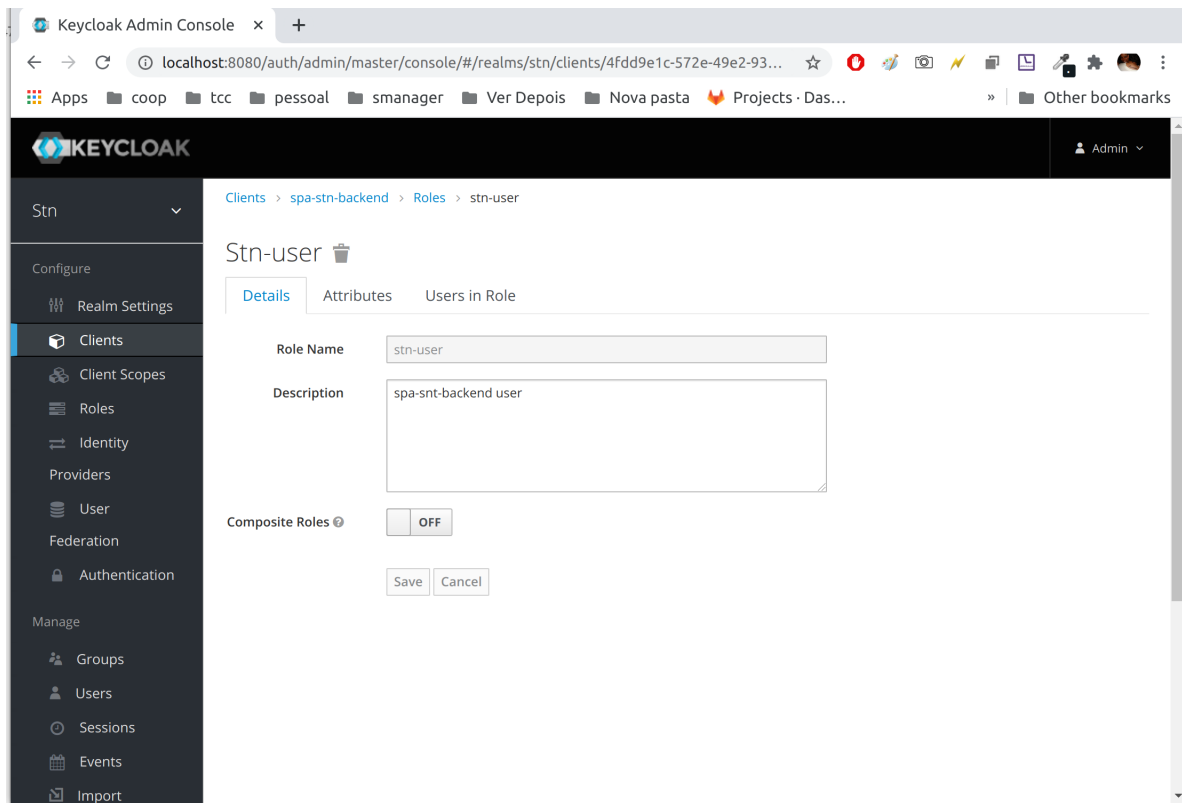
Spa-stn-backend

Settings Credentials **Roles** Revocation Clustering Installation

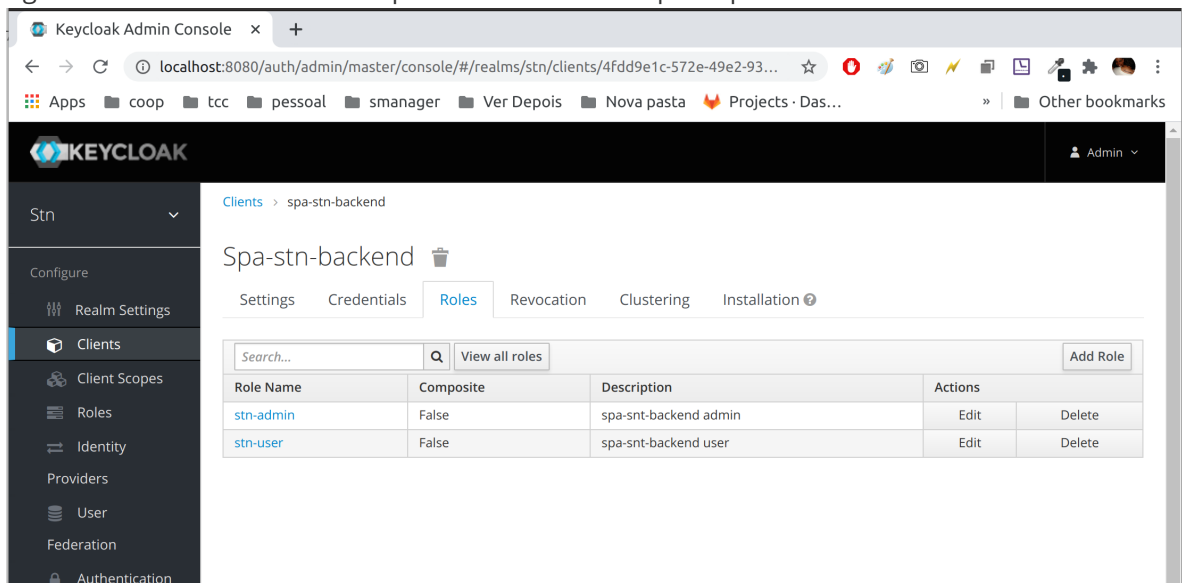
Search... View all roles Add Role

No client roles available

Role Name	Description
stn-user	spa-snt-backend user
stn-admin	spa-snt-backend admin



Agora temos duas **Client Roles** para serem utilizadas pela aplicação de backend.



Crie o Client spa-stn

Siga os passos da criação do client **spa-stn-backend** para criar o client **spa-stn** para acesso das aplicações FrontEnd

- Client ID: **spa-stn**
- Enabled: **ON**
- Client Protocol: **openid-connect**
- Access Type: **public**
- Standard Flow Enabled: **ON**
- Implicit Flow Enabled: **ON**
- Valid Redirect URI: <http://localhost:4200/>* e <http://localhost:4201/>*
- Web Origin: *

The screenshot shows the Keycloak Admin Console interface. On the left is a sidebar with a menu for 'Stn' containing options like 'Configure', 'Clients', 'Client Scopes', 'Roles', 'Identity Providers', 'User Federation', 'Authentication', 'Groups', 'Users', 'Sessions', 'Events', 'Import', and 'Export'. The main area is titled 'spa-stn' and has tabs for 'Settings', 'Roles', 'Client Scopes', 'Mappers', 'Scope', 'Revocation', 'Sessions', 'Offline Access', and 'Installation'. The 'Settings' tab is active, showing various configuration fields: Client ID (spa-stn), Name, Description, Enabled (ON), Consent Required (OFF), Login Theme, Client Protocol (openid-connect), Access Type (public), Standard Flow Enabled (ON), Implicit Flow Enabled (ON), Direct Access Grants Enabled (OFF), Root URL, Valid Redirect URIs (http://localhost:4201/, http://localhost:4200/), Base URL, Admin URL, and Web Origins (*).

Passo 3. Criação da Roles

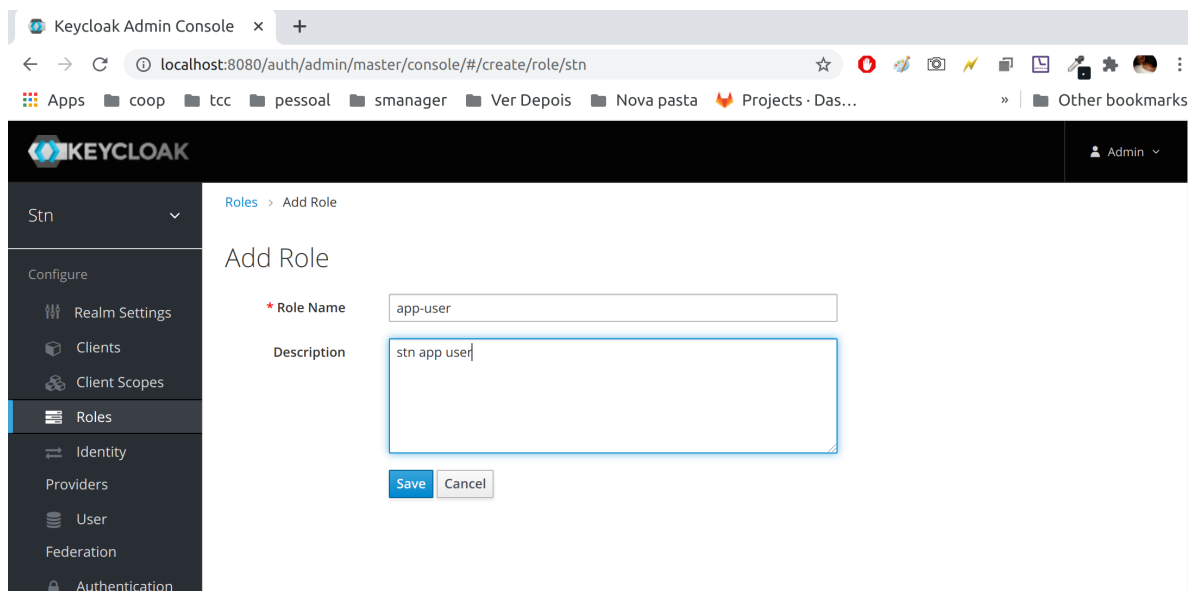
Criação de **Realm Level Roles** e associação com as **Client Level Roles** de acordo com a tabela abaixo

Realm Level Role	Client Level Role
app_user	stn-user
app_admin	stn-admin

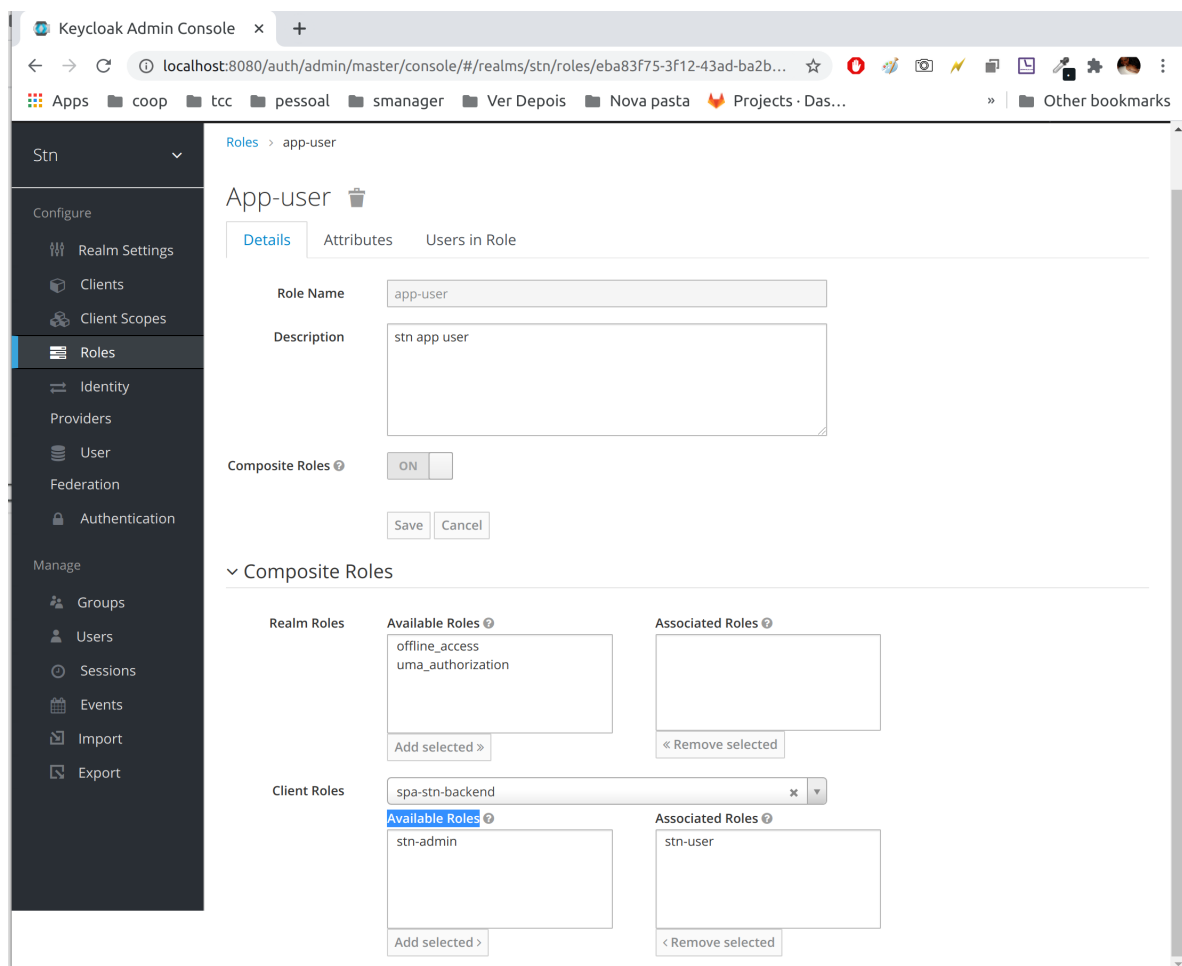
Clique em **Roles** no menu lateral esquerdo e em seguida na opção **Add Role**

The screenshot shows the Keycloak Admin Console 'Roles' page. The sidebar menu has 'Roles' selected. The main area has tabs for 'Realm Roles' and 'Default Roles'. Below the tabs is a search bar and a 'View all roles' button. A table lists existing roles: 'offline_access' and 'uma_authorization'. Each row has 'Role Name', 'Composite', 'Description', and 'Actions' (Edit, Delete). An 'Add Role' button is in the top right corner of the table area.

Preencha os campos Role Name **app-user** e Description **stn app user** e clique em salvar.



Após salvar, marque a opção **Composite Roles** como **ON** e no campo **Client Roles** selecione **spa-stn-backend**. No campo **Available Roles**, selecione **stn-user** e clique no botão **Add Selected**. O resultado deve ser igual a imagem abaixo.



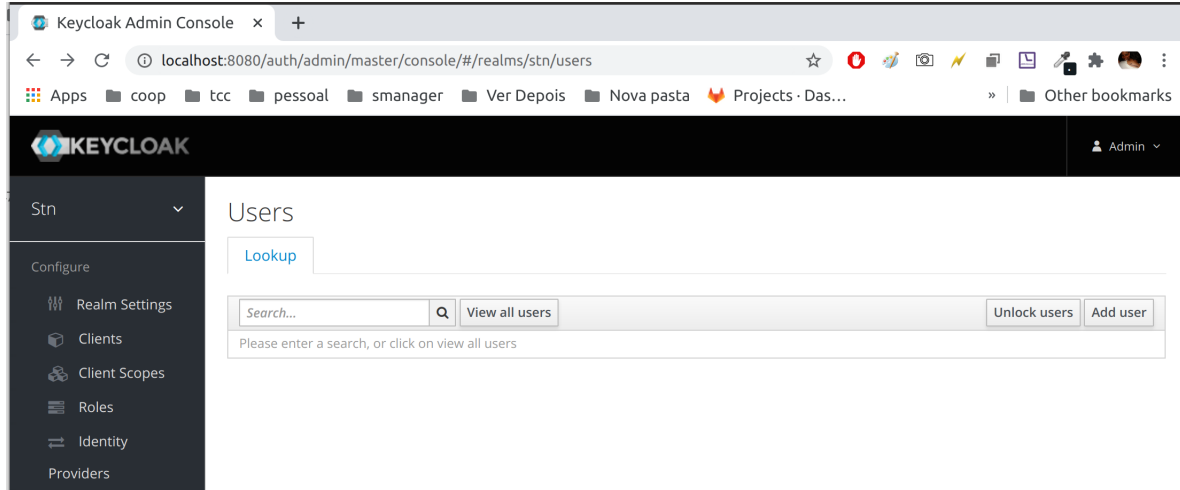
Repita o procedimento para a a Role **app_admin**

Passo 4. Criação dos Usuários

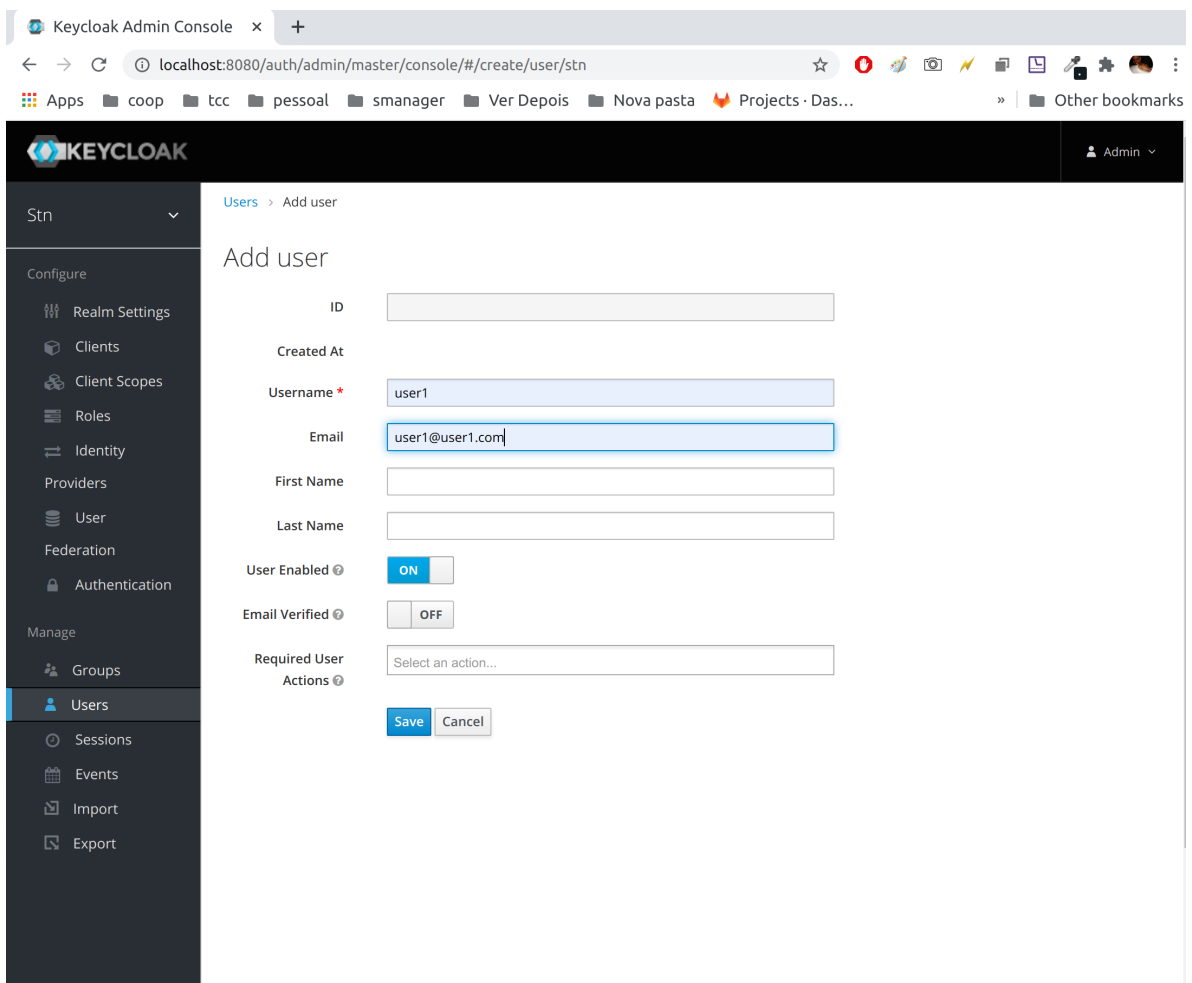
Vamos criar agora os usuários e associá-los as **Realm Level Roles** de acordo com a tabela abaixo.

Nome	Password	Roles
user1	user1	app-user
user2	user2	app-admin
user3	user3	app-admin, app-user

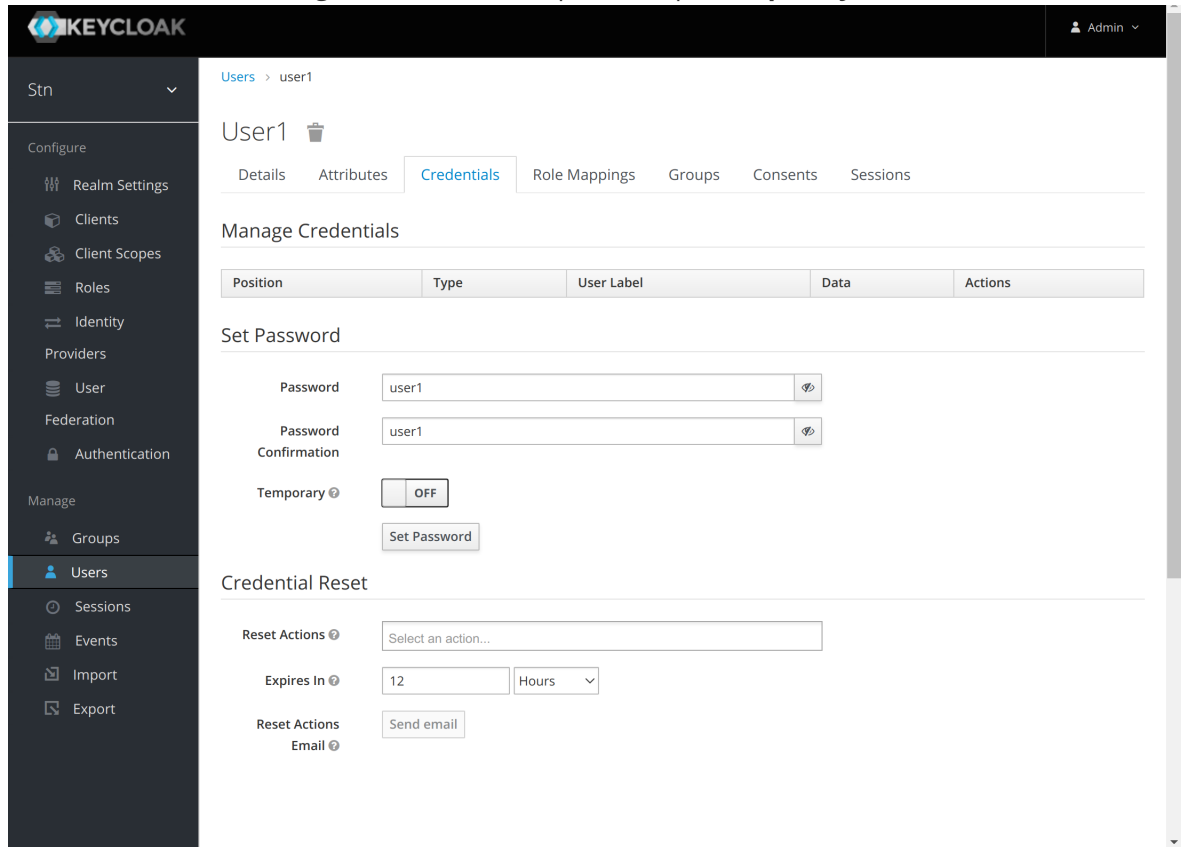
Clique em **Users** no menu lateral esquerdo e em seguida na opção **Add User**



Preencha os campos Username **user1** e Email **user1@user1.com** e clique em salvar.

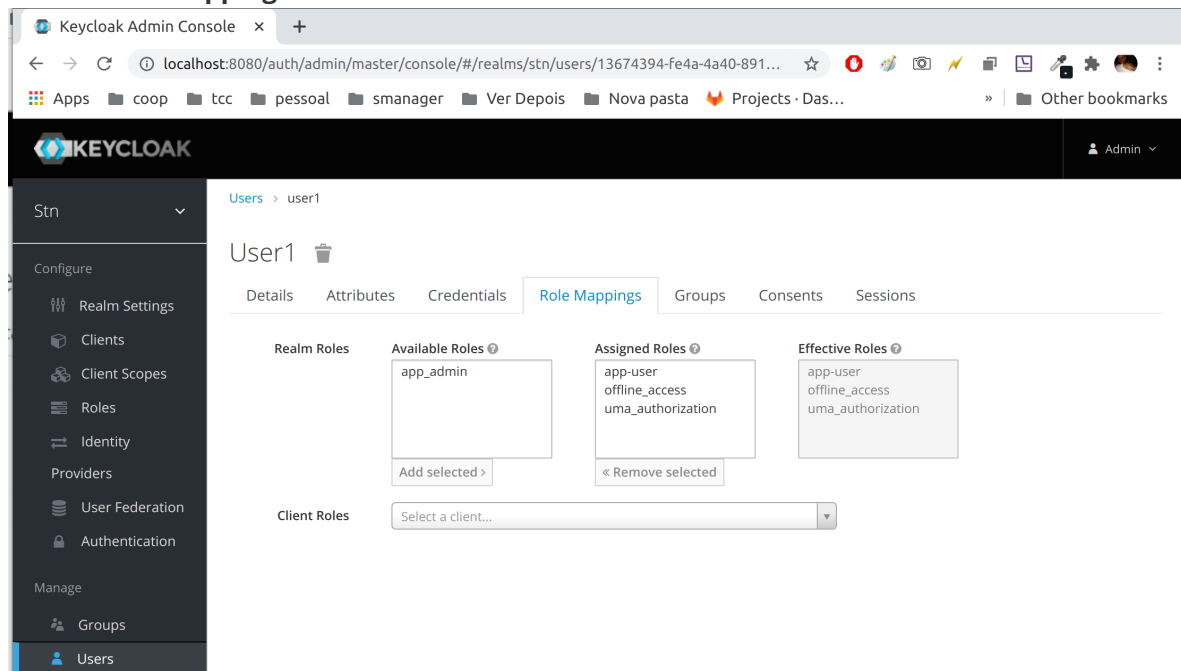


Na aba **Credentials** configure a senha e marque o campo **Temporary** como **OFF**



The screenshot shows the Keycloak Admin Console interface. On the left is a sidebar with navigation options like 'Configure', 'Manage', and 'Users'. The main area is titled 'User1' and has tabs for 'Details', 'Attributes', 'Credentials', 'Role Mappings', 'Groups', 'Consents', and 'Sessions'. The 'Credentials' tab is active, showing a table to 'Manage Credentials'. Below this, the 'Set Password' section has input fields for 'Password' (containing 'user1') and 'Password Confirmation' (also containing 'user1'). The 'Temporary' checkbox is set to 'OFF'. A 'Set Password' button is present. The 'Credential Reset' section has a 'Reset Actions' dropdown menu, an 'Expires In' field set to '12' hours, and a 'Send email' button.

Na aba **Role Mappings** selecione a role correta



This screenshot shows the 'Role Mappings' tab for user1 in the Keycloak Admin Console. It displays four columns: 'Realm Roles', 'Available Roles', 'Assigned Roles', and 'Effective Roles'. Under 'Available Roles', 'app_admin' is listed. Under 'Assigned Roles', 'app-user', 'offline_access', and 'uma_authorization' are listed. Under 'Effective Roles', 'app-user', 'offline_access', and 'uma_authorization' are listed. There are buttons for 'Add selected >' and '< Remove selected'. A 'Client Roles' section at the bottom has a dropdown menu labeled 'Select a client...'.

Repita o procedimento de criação de usuários para os outros usuários configurando de acordo com as Roles na tabela.

Passo 5. Configuração das Aplicações

Vamos agora configurar as aplicações de backend e frontend. Para esse tutorial será utilizado o Visual Code como IDE para edição das aplicações .

Faça o clone do repositório <https://github.com/mgamarra/poc-keycloak-angular-springboot>

Nesser repositório existem 3 aplicações

- heroes-app : Aplicação backend Java com SpringBoot

- heroes-backend: Aplicação frontend Angular
- keycloak-angular-tasks: Aplicação frontend Angular

Aplicação backend heroes-backend - Java com SpringBoot

As configurações da aplicação estão no arquivo **src/main/resources/application.properties** e no **pom.xml** foi adicionado a dependência do **keycloak**

A classe **com.example.controller.HeroController** possui exemplos de uso da anotação **RolesAllowed** onde são utilizadas as **Roles** configuradas no **SSO**.

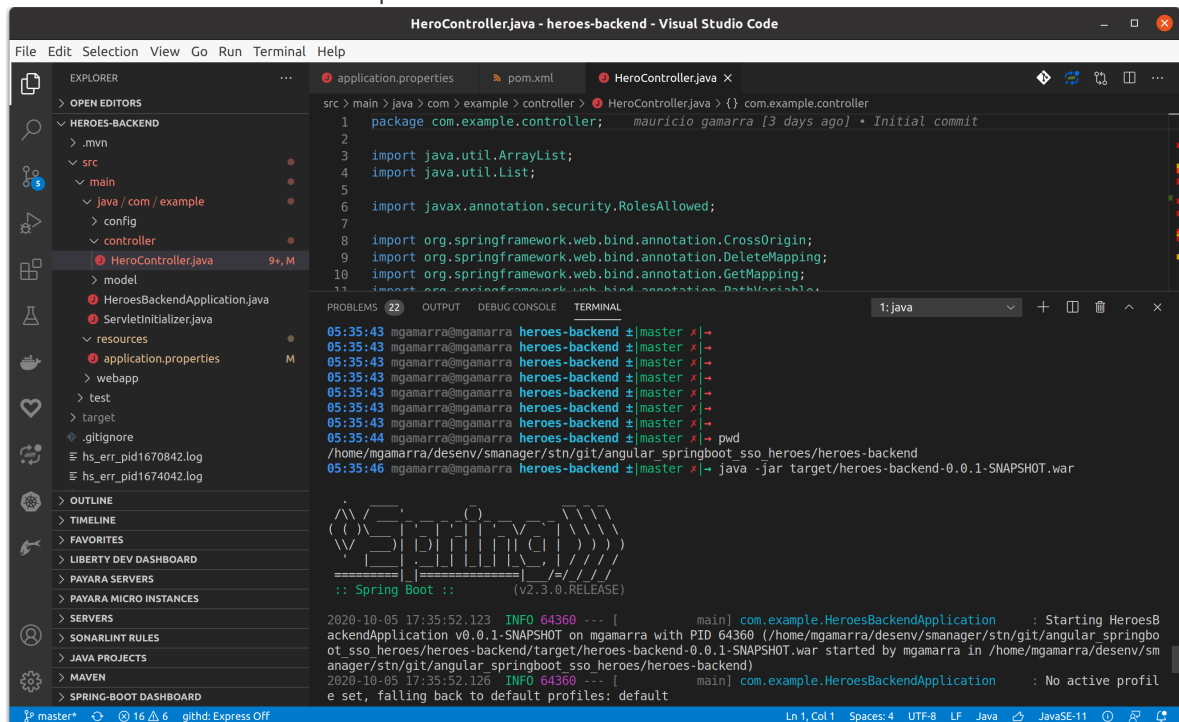
Para compilar a aplicação, execute o comando abaixo no diretório **heroes-backend**

```
./mvnw clean package -DskipTests
```

Para executar a aplicação, execute o comando abaixo no diretório **heroes-backend**

```
java -jar target/heroes-backend-0.0.1-SNAPSHOT.war
```

Executando o heroes-backend por dentro do terminal do Visual Code



Aplicação frontend keycloak-angular-tasks - Angular

As configurações da aplicação estão no arquivo **src/environments/environment.ts**

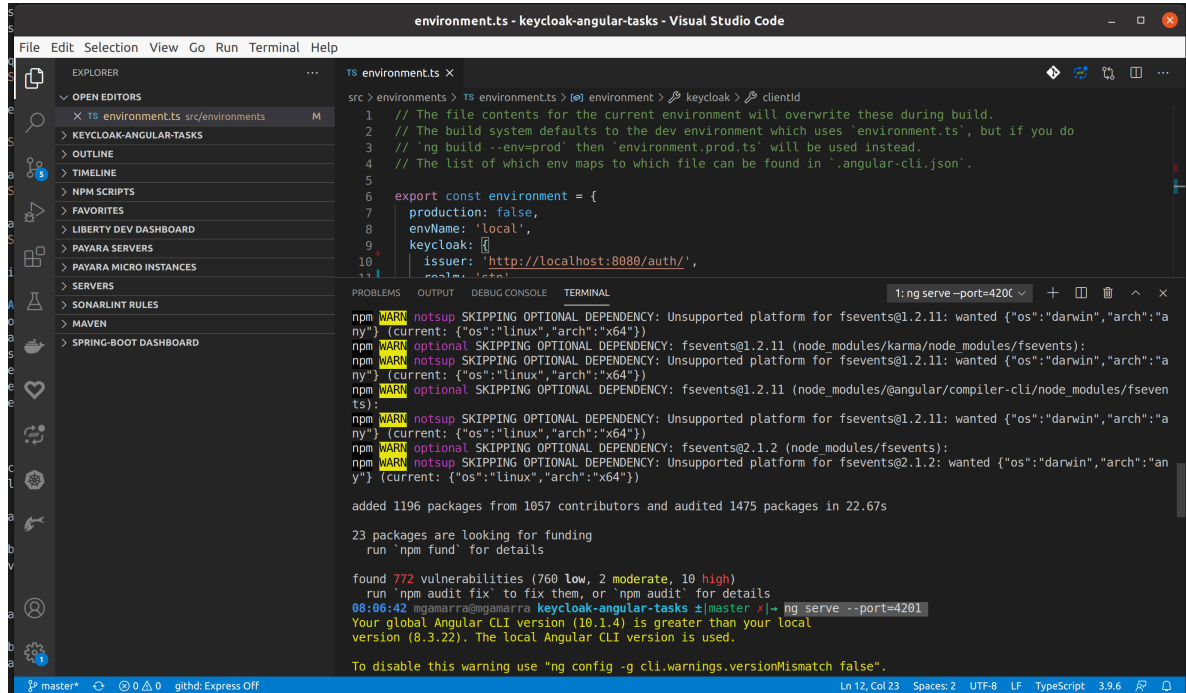
Para instalar as dependências execute o comando abaixo no diretório **keycloak-angular-tasks**

```
npm i
```

Para executar a aplicação, execute o comando abaixo no diretório **keycloak-angular-tasks**

```
ng serve --port=4200
```

Executando o keycloak-angular-tasks por dentro do terminal do Visual Code



```
environment.ts - keycloak-angular-tasks - Visual Studio Code

src > environments > TS environment.ts > environment > keycloak > clientid

1 // The file contents for the current environment will overwrite these during build.
2 // The build system defaults to the dev environment which uses 'environment.ts', but if you do
3 // 'ng build --env=prod' then 'environment.prod.ts' will be used instead.
4 // The list of which env maps to which file can be found in '.angular-cli.json'.
5
6 export const environment = {
7   production: false,
8   envName: 'local',
9   keycloak: {
10     issuer: 'http://localhost:8080/auth/',
11     realm: 'test'
12   }
13 };

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.11: wanted {"os":"darwin","arch":"a
ny"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.11 (node_modules/karma/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.11: wanted {"os":"darwin","arch":"a
ny"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.11 (node_modules/@angular/compiler-cli/node_modules/fseven
ts):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.11: wanted {"os":"darwin","arch":"a
ny"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"an
y"} (current: {"os":"linux","arch":"x64"})

added 1196 packages from 1057 contributors and audited 1475 packages in 22.67s

23 packages are looking for funding
  run 'npm fund' for details

found 772 vulnerabilities (760 low, 2 moderate, 10 high)
  run 'npm audit fix' to fix them, or 'npm audit' for details
08:06:42 mgamarra@mgamarra keycloak-angular-tasks # master $ ng serve --port=4201
Your global Angular CLI version (10.1.4) is greater than your local
version (8.3.22). The local Angular CLI version is used.

To disable this warning use "ng config -g cli.warnings.versionMismatch false".
```

Aplicação frontend heroes-app - Angular

As configurações da aplicação estão no arquivo **src/environments/environment.ts**

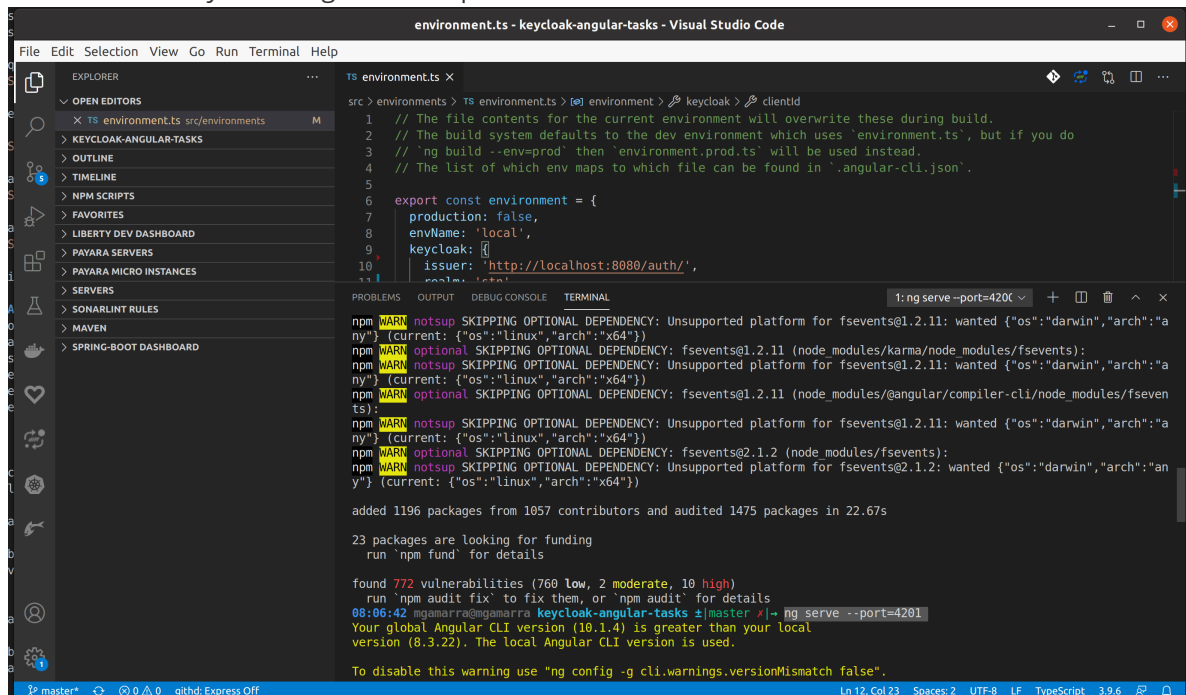
Para instalar as dependências execute o comando abaixo no diretório **heroes-app**

```
npm i
```

Para executar a aplicação, execute o comando abaixo no diretório **heroes-app**

```
ng serve --port=4201
```

Executando o keycloak-angular-tasks por dentro do terminal do Visual Code



```
environment.ts - keycloak-angular-tasks - Visual Studio Code

src > environments > TS environment.ts > environment > keycloak > clientid

1 // The file contents for the current environment will overwrite these during build.
2 // The build system defaults to the dev environment which uses 'environment.ts', but if you do
3 // 'ng build --env=prod' then 'environment.prod.ts' will be used instead.
4 // The list of which env maps to which file can be found in '.angular-cli.json'.
5
6 export const environment = {
7   production: false,
8   envName: 'local',
9   keycloak: {
10     issuer: 'http://localhost:8080/auth/',
11     realm: 'test'
12   }
13 };

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.11: wanted {"os":"darwin","arch":"a
ny"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.11 (node_modules/karma/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.11: wanted {"os":"darwin","arch":"a
ny"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.11 (node_modules/@angular/compiler-cli/node_modules/fseven
ts):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.11: wanted {"os":"darwin","arch":"a
ny"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"an
y"} (current: {"os":"linux","arch":"x64"})

added 1196 packages from 1057 contributors and audited 1475 packages in 22.67s

23 packages are looking for funding
  run 'npm fund' for details

found 772 vulnerabilities (760 low, 2 moderate, 10 high)
  run 'npm audit fix' to fix them, or 'npm audit' for details
08:06:42 mgamarra@mgamarra keycloak-angular-tasks # master $ ng serve --port=4201
Your global Angular CLI version (10.1.4) is greater than your local
version (8.3.22). The local Angular CLI version is used.

To disable this warning use "ng config -g cli.warnings.versionMismatch false".
```