

Security in Vuejs 3.0 with authentication and authorization by KeyCloak Part 1

Part 1 Securing a front-end Vuejs 3.0 application



Introduction

I already did a complete article about securing [a react application with Keycloak](#).

Today we will see how to do this with the other front-end framework: Vuejs 3.0!

This article will be only a step-by-step guide, if you need to read a complete article please go [here](#)

Create our Vuejs 3.0 front-end demo

Let's go to our directory and create a new project:

```
C:\Tutorial\keycloak (main -> origin)
λ mkdir vuejs3webapikeycloak

C:\Tutorial\keycloak (main -> origin)
λ cd vuejs3webapikeycloak\

C:\Tutorial\keycloak\vuejs3webapikeycloak (main -> origin)
λ
```

We will use the cli tool and install it with npm as a global tool with the following npm command

```
npm install -g @vue/cli
C:\Tutorial\keycloak\vuejs3webapikeycloak (main -> origin)
λ npm install -g @vue/cli
[.....] / fetchMetadata: sill resolveWithNewModule
yallist@4.0.0 checking installable status
```

It takes a little while, so take a coffee or hot drink :)

We get the following result:

```
C:\Tutorial\keycloak\vuejs3webapikeycloak (main -> origin)
λ npm install -g @vue/cli
npm WARN deprecated subscriptions-transport-ws@0.11.0: The
`subscriptions-transport-ws` package is no longer maintained. We
recommend you use `graphql-ws` instead. For help migrating Apollo
software to `graphql-ws`, see
https://www.apollographql.com/docs/apollo-
server/data/subscriptions/#switching-from-subscriptions-transport-ws
For general help using `graphql-ws`, see
https://github.com/enisdenjo/graphql-ws/blob/master/README.md
npm WARN deprecated urix@0.1.0: Please see
https://github.com/lydell/urix#deprecated
npm WARN deprecated resolve-url@0.2.1:
https://github.com/lydell/resolve-url#deprecated
C:\Users\nbarlatier\AppData\Roaming\npm\vue ->
C:\Users\nbarlatier\AppData\Roaming\npm\node_modules\@vue\cli\bin\vue.
```

```
js

> @apollo/protobufjs@1.2.6 postinstall
C:\Users\nbarlatier\AppData\Roaming\npm\node_modules\@vue\cli\node_modules\@apollo\protobufjs
> node scripts/postinstall

+ @vue/cli@5.0.8
added 833 packages from 492 contributors in 159.069s
```

New major version of npm available! 6.14.4 -> 9.1.1
Changelog: <https://github.com/npm/cli/releases/tag/v9.1.1>
Run npm install -g npm to update!

```
C:\Tutorial\keycloak\vuejs3webapikeycloak (main -> origin)
```

Now let's check if we can use the CLI by getting the version:

```
λ vue -V
@vue/cli 4.5.13
```

We are ready to create our project!

```
C:\Tutorial\keycloak\vuejs3webapikeycloak (main -> origin)
λ vue create front-end
```

We get:

```
λ vue create front-end
```

```
Vue CLI v4.5.13
```

New version available 4.5.13 -> 5.0.8
Run yarn global add @vue/cli to update!

```
? Please pick a preset: (Use arrow keys)
> Default ([Vue 2] babel, eslint)
  Default (Vue 3) ([Vue 3] babel, eslint)
  Manually select features
```

I will select manually the features

Vue CLI v4.5.13

```
New version available 4.5.13 → 5.0.8
Run yarn global add @vue/cli to update!
```

```
? Please pick a preset:
  Default ([Vue 2] babel, eslint)
  Default (Vue 3) ([Vue 3] babel, eslint)
> Manually select features
```

Press the arrow down and enter

```
? Please pick a preset: Manually select features
? Check the features needed for your project:
  (*) Choose Vue version
  (*) Babel
> (*) TypeScript
  ( ) Progressive Web App (PWA) Support
  ( ) Router
  ( ) Vuex
  ( ) CSS Pre-processors
  (*) Linter / Formatter
  ( ) Unit Testing
  ( ) E2E Testing
```

I select TypeScript, we could add other features but for our demo we don't need them, Press Enter.

Then I select Vuejs 3.0 and press enter:

```
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version,
Babel, TS, Linter
? Choose a version of Vue.js that you want to start the project with
(Use arrow keys)
  2.x
> 3.x
```

Then we say no to the class-style component for our demo.

```
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version,
Babel, TS, Linter
? Choose a version of Vue.js that you want to start the project with
3.x
? Use class-style component syntax? (y/N)N
```

Then we select Yes to use Typescript and Babel at the same time

```
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version,
Babel, TS, Linter
? Choose a version of Vue.js that you want to start the project with
3.x
? Use class-style component syntax? No
? Use Babel alongside TypeScript (required for modern mode, auto-
detected polyfills, transpiling JSX)? (Y/n)
```

Then we select the default option for ESLint:

```
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version,
Babel, TS, Linter
? Choose a version of Vue.js that you want to start the project with
3.x
? Use class-style component syntax? No
? Use Babel alongside TypeScript (required for modern mode, auto-
detected polyfills, transpiling JSX)? Yes
? Pick a linter / formatter config: (Use arrow keys)
```

```
> ESLint with error prevention only
  ESLint + Airbnb config
  ESLint + Standard config
  ESLint + Prettier
  TSLint (deprecated)
```

Again we select the default option

```
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version,
Babel, TS, Linter
? Choose a version of Vue.js that you want to start the project with
3.x
? Use class-style component syntax? No
? Use Babel alongside TypeScript (required for modern mode, auto-
detected polyfills, transpiling JSX)? Yes
? Pick a linter / formatter config: Basic
? Pick additional lint features: (Press <space> to select, <a> to
toggle all, <i> to invert selection)
>(*) Lint on save
  ( ) Lint and fix on commit
```

Then we select package.json to get the config of Babel, ESLint etc

```
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version,
Babel, TS, Linter
? Choose a version of Vue.js that you want to start the project with
3.x
? Use class-style component syntax? No
? Use Babel alongside TypeScript (required for modern mode, auto-
detected polyfills, transpiling JSX)? Yes
? Pick a linter / formatter config: Basic
? Pick additional lint features: Lint on save
? Where do you prefer placing config for Babel, ESLint, etc.?
  In dedicated config files
> In package.json
```

Then we can save our settings if we want, but here let's skip it:

```
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version,
Babel, TS, Linter
? Choose a version of Vue.js that you want to start the project with
3.x
? Use class-style component syntax? No
? Use Babel alongside TypeScript (required for modern mode, auto-
detected polyfills, transpiling JSX)? Yes
? Pick a linter / formatter config: Basic
? Pick additional lint features: Lint on save
? Where do you prefer placing config for Babel, ESLint, etc.? In
package.json
? Save this as a preset for future projects? (y/N)
```

We get:

```
? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version,
Babel, TS, Linter
? Choose a version of Vue.js that you want to start the project with
3.x
? Use class-style component syntax? No
? Use Babel alongside TypeScript (required for modern mode, auto-
detected polyfills, transpiling JSX)? Yes
? Pick a linter / formatter config: Basic
? Pick additional lint features: Lint on save
? Where do you prefer placing config for Babel, ESLint, etc.? In
package.json
? Save this as a preset for future projects? No
```

Vue CLI v4.5.13

🌟 Creating project in

C:\Tutorial\keycloak\vuejs3webapikeycloak\front-end.

⚙️ Installing CLI plugins. This might take a while...

yarn install v1.22.4

info No lockfile found.

[1/4] Resolving packages...

• safe-buffer@^5.1.1

We need to wait for a while.

Vue CLI v4.5.13

New version available 4.5.13 → 5.0.8
Run yarn global add @vue/cli to update!

? Please pick a preset: Manually select features
? Check the features needed for your project: Choose Vue version, Babel, TS, Linter
? Choose a version of Vue.js that you want to start the project with 3.x
? Use class-style component syntax? No
? Use Babel alongside TypeScript (required for modern mode, auto-detected polyfills, transpiling JSX)? Yes
? Pick a linter / formatter config: Basic
? Pick additional lint features: Lint on save
? Where do you prefer placing config for Babel, ESLint, etc.? In package.json
? Save this as a preset for future projects? No

Vue CLI v4.5.13

✨ Creating project in
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-end.
⚙ Installing CLI plugins. This might take a while...

```
yarn install v1.22.4
info No lockfile found.
[1/4] Resolving packages...
[2/4] Fetching packages...
info fsevents@2.3.2: The platform "win32" is incompatible with this module.
info "fsevents@2.3.2" is an optional dependency and failed compatibility check. Excluding it from installation.
info fsevents@1.2.13: The platform "win32" is incompatible with this module.
info "fsevents@1.2.13" is an optional dependency and failed compatibility check. Excluding it from installation.
[#####] 1152/1152[3/4] Linking dependencies...
[#####]
#####-----
-----] 16410/[#####
```

success Saved lockfile.
info To upgrade, download the latest installer at
"https://yarnpkg.com/latest.msi".
Done in 171.56s.
❖ Invoking generators...
❖ Installing additional dependencies...

yarn install v1.22.4


```
[1/4] Resolving packages...
• @typescript-eslint/eslint-plugin@^4.18.0
```

I am each time both impressed and confused by the number of dependencies with front-end frameworks!

It took almost 3 minutes!

```
Vue CLI v4.5.13
✳️ Creating project in
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-end.
⚙️ Installing CLI plugins. This might take a while...

yarn install v1.22.4
info No lockfile found.
[1/4] Resolving packages...
[2/4] Fetching packages...
info fsevents@2.3.2: The platform "win32" is incompatible with this
module.
info "fsevents@2.3.2" is an optional dependency and failed
compatibility check. Excluding it from installation.
info fsevents@1.2.13: The platform "win32" is incompatible with this
module.
info "fsevents@1.2.13" is an optional dependency and failed
compatibility check. Excluding it from installation.
[#####] 1152/1152[3/4] Linking dependencies...
[#####]
#####-----] 16410/[#####

success Saved lockfile.
info To upgrade, download the latest installer at
"https://yarnpkg.com/latest.msi".
Done in 171.56s.
❓ Invoking generators...
❓ Installing additional dependencies...

yarn install v1.22.4
[1/4] Resolving packages...
[2/4] Fetching packages...
info fsevents@2.3.2: The platform "win32" is incompatible with this
module.
info "fsevents@2.3.2" is an optional dependency and failed
compatibility check. Excluding it from installation.
```

```
info fsevents@1.2.13: The platform "win32" is incompatible with this module.
info "fsevents@1.2.13" is an optional dependency and failed compatibility check. Excluding it from installation.
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Saved lockfile.
Done in 54.80s.
🔗 Running completion hooks...

💡 Generating README.md...

💡 Successfully created project front-end.
💡 Get started with the following commands:

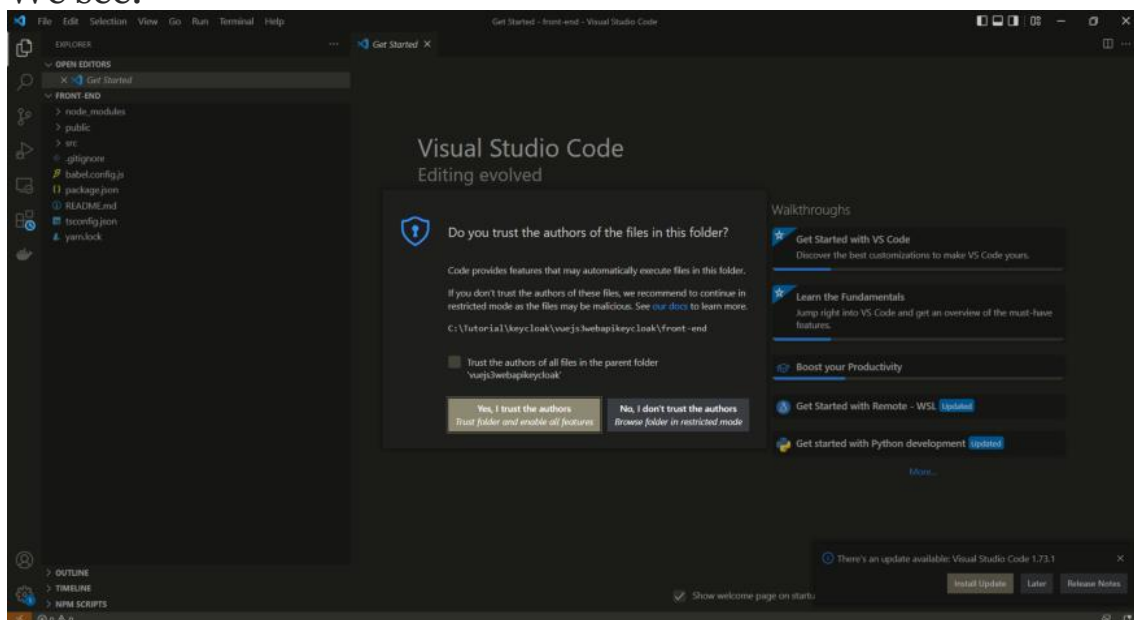
$ cd front-end
$ yarn serve
```

Let's see the project with VS Code before running it!

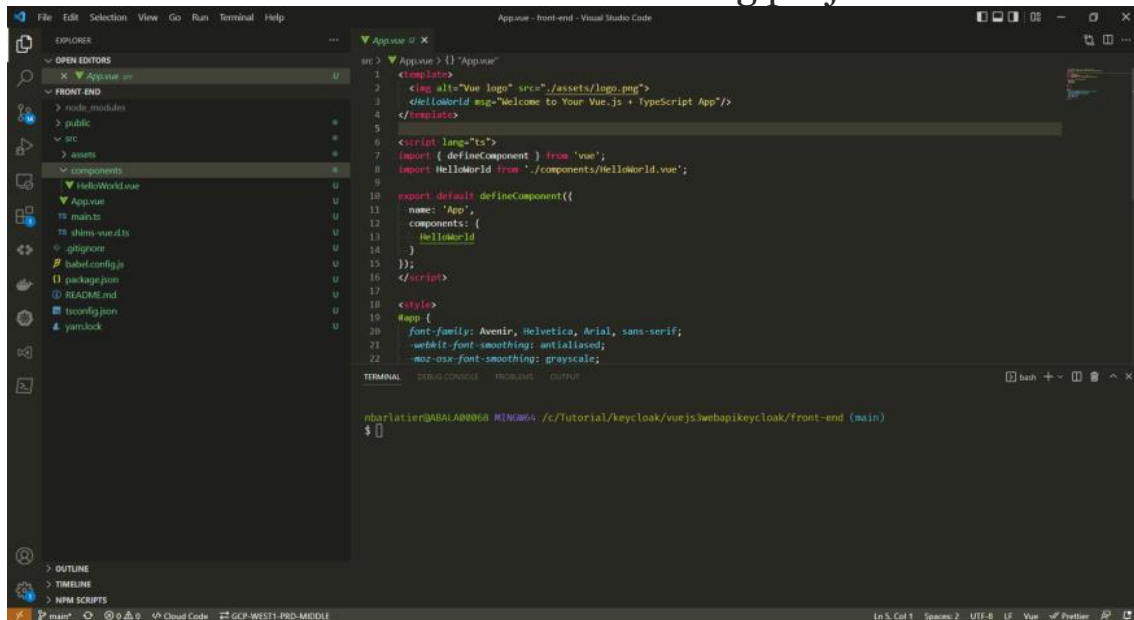
```
λ cd front-end\

C:\Tutorial\keycloak\vuejs3webapikeycloak\front-end (main -> origin)
λ code .
```

We see:



We trust our code and we see the following project:



We will not focus on Vuejs 3.0 project structure here, we will just need to know where to add our Keycloak client so we can add security to our Vuejs demo!

We will open the main.ts file :

```
import { createApp } from 'vue'
import App from './App.vue'

createApp(App).mount('#app')
```

It will be the place where we can add Keycloak!

But first, we need to make sure our application can run normally!

Let's host our app in a dev server with the following command:

```
yyyy@xxx MINGW64 /c/Tutorial/keycloak/vuejs3webapikeycloak/front-end
(main)
$ yarn serve
yarn run v1.22.4
$ vue-cli-service serve
  INFO Starting development server...
 98% after emitting CopyPlugin

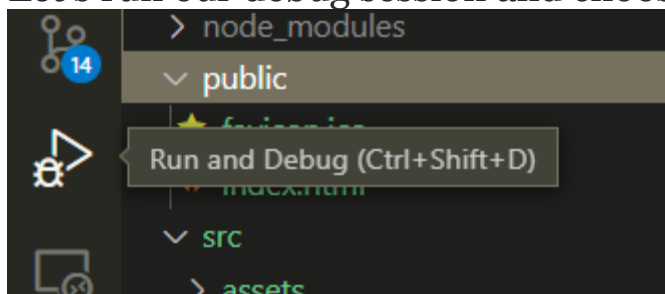
  DONE Compiled successfully in 5569ms
15:59:11

App running at:
- Local:   http://localhost:8080/
- Network: http://xxx:8080/

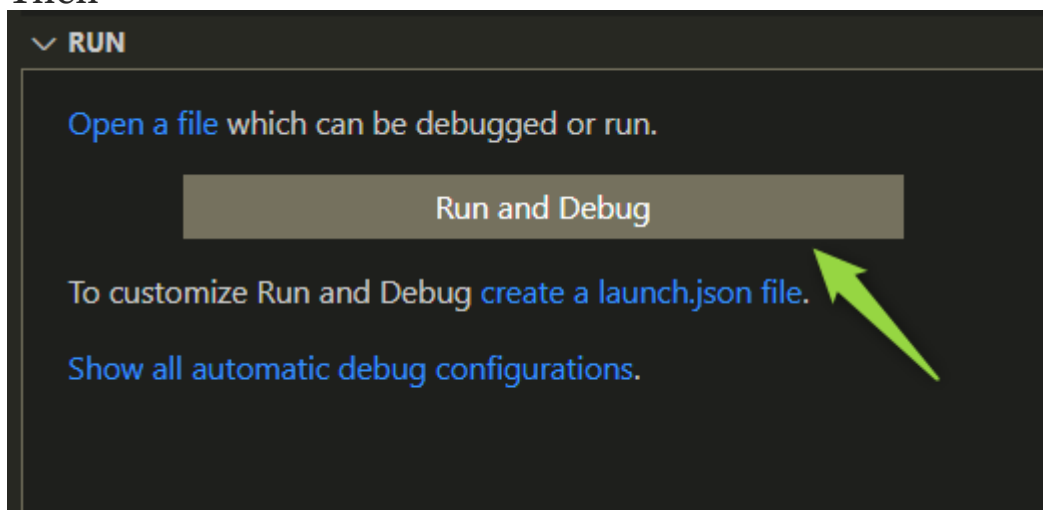
Note that the development build is not optimized.
To create a production build, run yarn build.

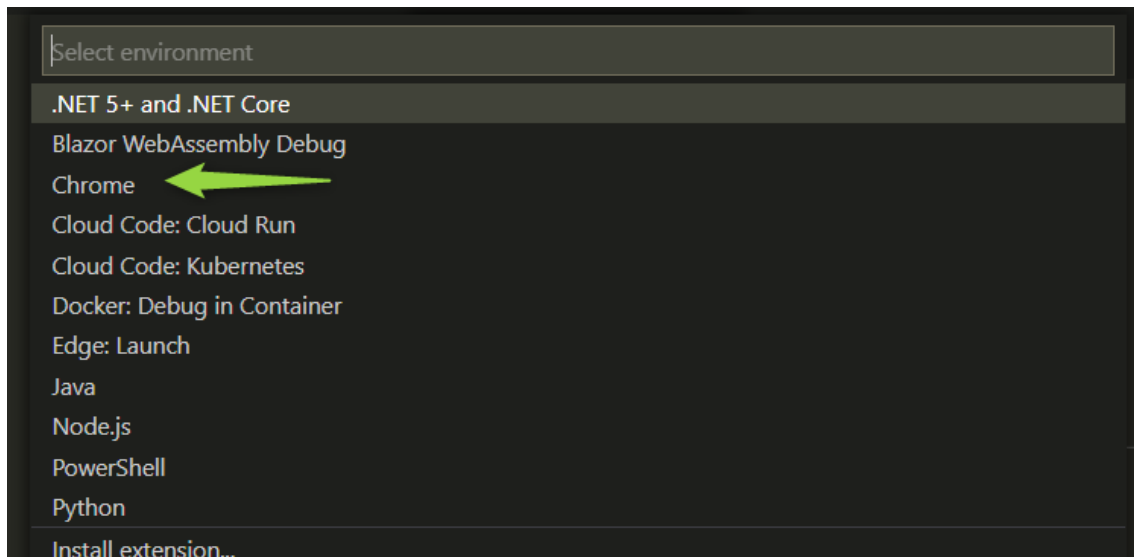
No issues found.
```

Let's run our debug session and choose chrome as our browser

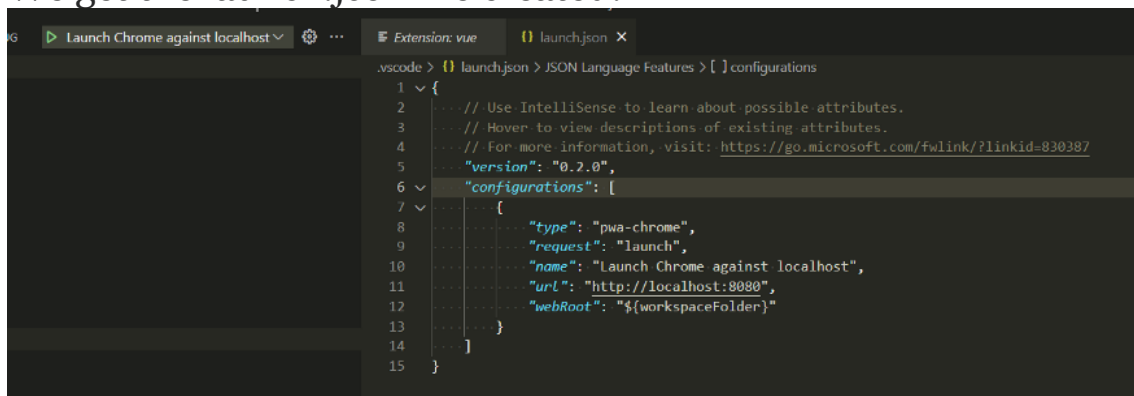


Then

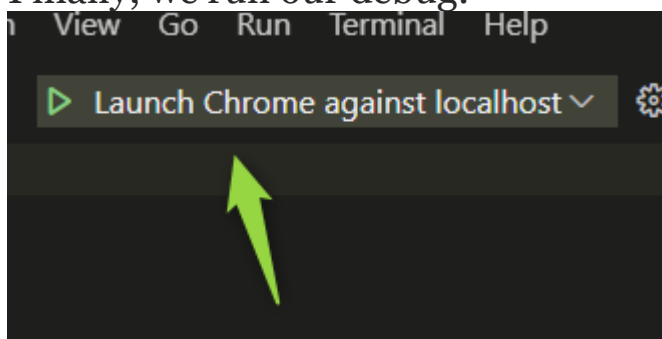




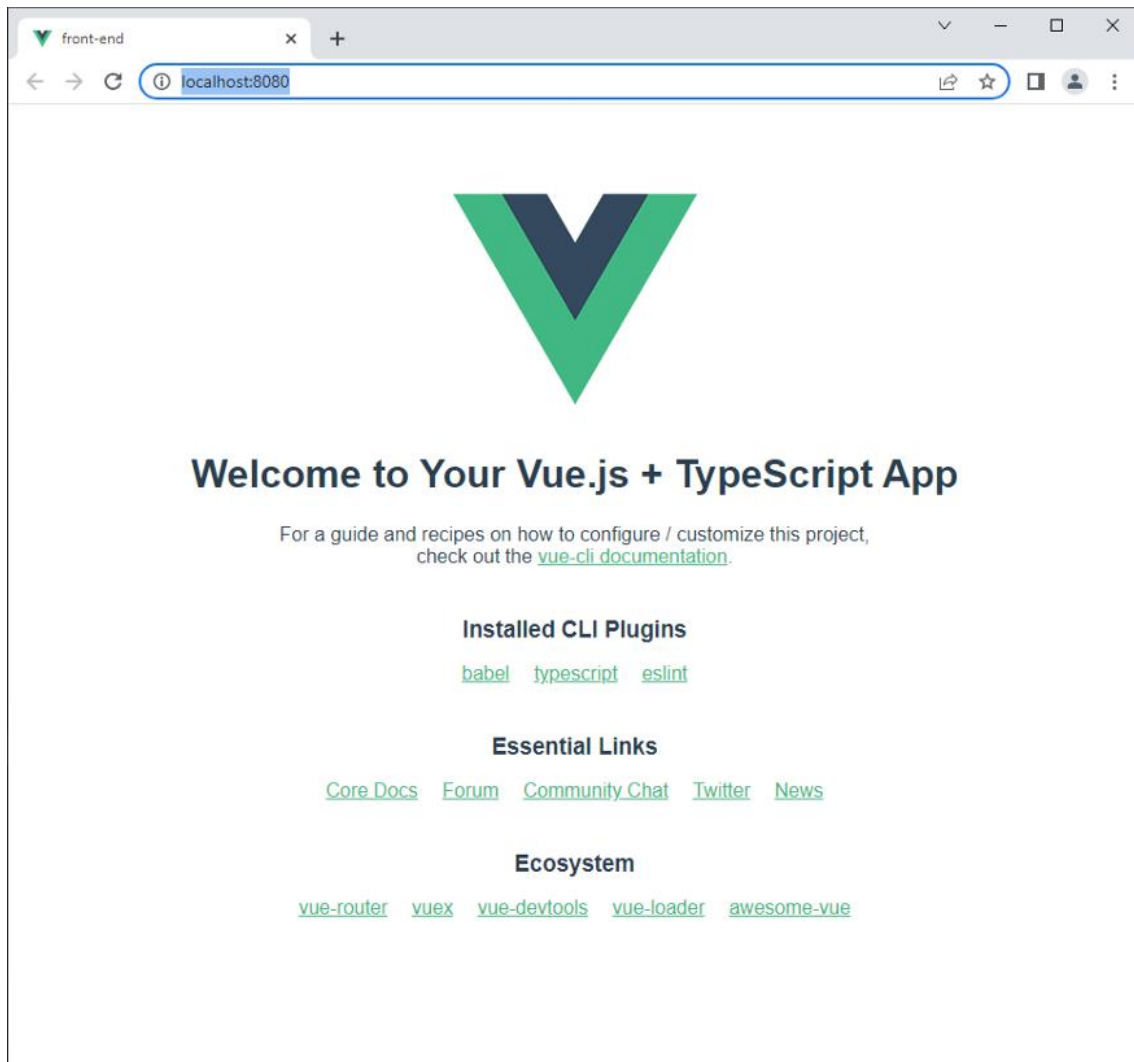
We get the launch.json file created:



Finally, we run our debug:



We see our chrome and our app!



Right now we need to add Keycloak to our Vuejs 3.0 project!

We can see the 3 CLI plugins we enabled earlier:

Installed CLI Plugins

[babel](#) [typescript](#) [eslint](#)

With their respective links:

- [babel plugin](#)
- [typescript plugin](#)

- [eslint plugin](#)

Now let's install **Keycloak-js** with npm:

We can find all the versions available on the following link:

keycloak-js

Keycloak Adapter. Latest version: 20.0.1, last published: 6 days ago. Start using keycloak-js in your project by...

www.npmjs.com

We will use Docker to run the latest stable version of Keycloak

19.0.3 :

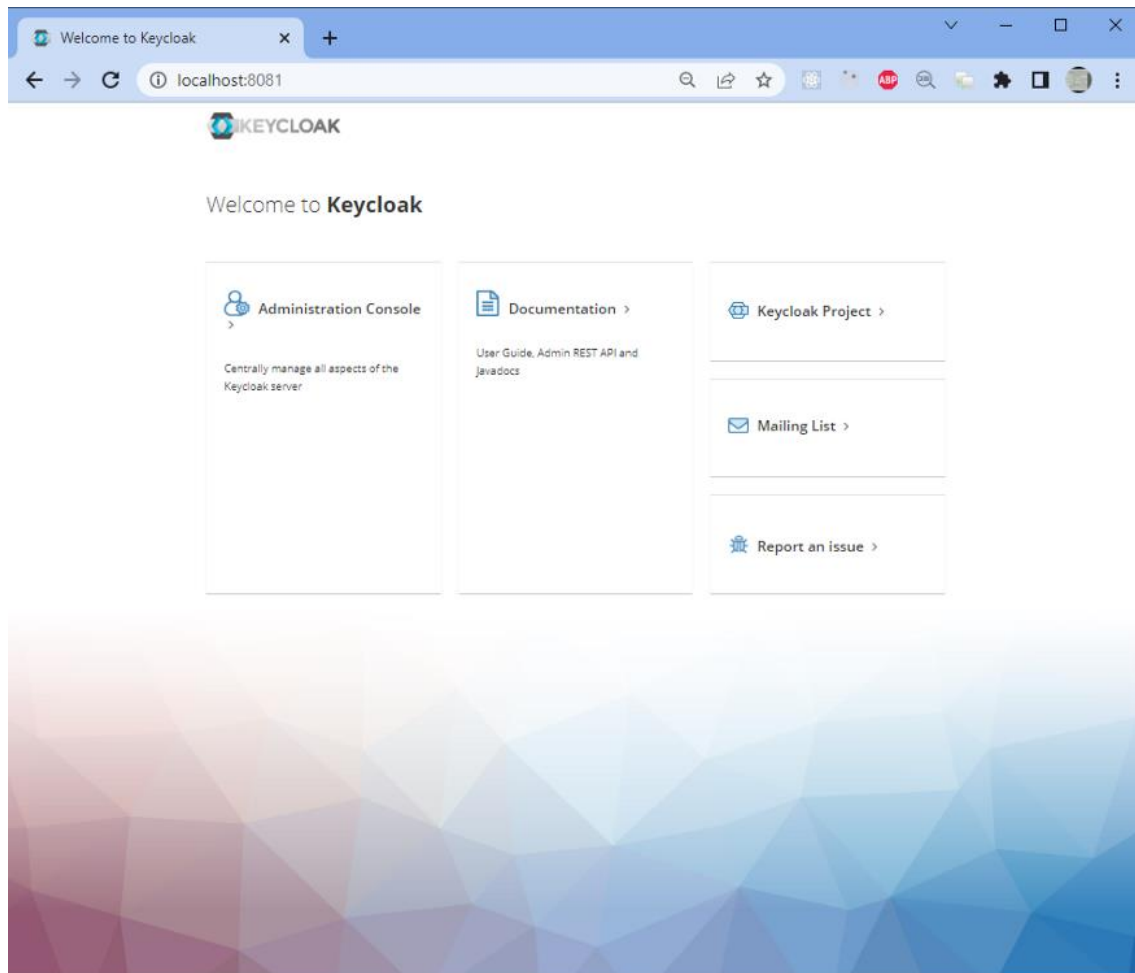
```
λ docker run -p 8081:8080 -e KEYCLOAK_ADMIN=admin -e
KEYCLOAK_ADMIN_PASSWORD=admin quay.io/keycloak/keycloak:19.0.3 start-
dev
Updating the configuration and installing your custom providers, if
any. Please wait.
2022-11-19 14:15:11,742 INFO [io.quarkus.deployment.QuarkusAugmentor]
(main) Quarkus augmentation completed in 22124ms
2022-11-19 14:15:15,287 INFO
[org.keycloak.quarkus.runtime.hostname.DefaultHostnameProvider] (main)
Hostname settings: Base URL: <unset>, Hostname: <request>, Strict
HTTPS: false, Path: <request>, Strict BackChannel: false, Admin URL:
<unset>, Admin: <request>, Port: -1, Proxied: false
2022-11-19 14:15:17,112 INFO
[org.keycloak.common.crypto.CryptoIntegration] (main) Detected crypto
provider: org.keycloak.crypto.def.DefaultCryptoProvider
2022-11-19 14:15:19,865 WARN [org.infinispan.CONFIG] (keycloak-cache-
init) ISPN000569: Unable to persist Infinispan internal caches as no
global state enabled
2022-11-19 14:15:19,988 WARN [org.infinispan.PERSISTENCE] (keycloak-
cache-init) ISPN000554: jboss-marshalling is deprecated and planned
for removal
2022-11-19 14:15:20,029 INFO [org.infinispan.CONTAINER] (keycloak-
cache-init) ISPN000556: Starting user marshaller
'org.infinispan.jboss.marshalling.core.JBossUserMarshaller'
2022-11-19 14:15:20,550 INFO [org.infinispan.CONTAINER] (keycloak-
cache-init) ISPN000128: Infinispan version: Infinispan
'Triskaidekaphobia' 13.0.9.Final
2022-11-19 14:15:21,011 INFO
[org.keycloak.connections.infinispan.DefaultInfinispanConnectionProvid
erFactory] (main) Node name: node_425607, Site name: null
2022-11-19 14:15:22,808 INFO
[org.keycloak.quarkus.runtime.storage.legacy.liquibase.QuarkusJpaUpdat
```

```
erProvider] (main) Initializing database schema. Using changelog META-INF/jpa-changelog-master.xml
2022-11-19 14:15:26,821 INFO [org.keycloak.services] (main) KC-SERVICES0050: Initializing master realm
2022-11-19 14:15:30,627 INFO [io.quarkus] (main) Keycloak 19.0.3 on JVM (powered by Quarkus 2.7.6.Final) started in 18.773s. Listening on: http://0.0.0.0:8080
2022-11-19 14:15:30,627 INFO [io.quarkus] (main) Profile dev activated.
2022-11-19 14:15:30,627 INFO [io.quarkus] (main) Installed features: [agroal, cdi, hibernate-orm, jdbc-h2, jdbc-mariadb, jdbc-mssql, jdbc-mysql, jdbc-oracle, jdbc-postgresql, keycloak, logging-gelf, narayana-jta, reactive-routes, resteasy, resteasy-jackson, smallrye-context-propagation, smallrye-health, smallrye-metrics, vault, vertx]
2022-11-19 14:15:31,254 INFO [org.keycloak.services] (main) KC-SERVICES0009: Added user 'admin' to realm 'master'
2022-11-19 14:15:31,265 WARN [org.keycloak.quarkus.runtime.KeycloakMain] (main) Running the server in development mode. DO NOT use this configuration in production.
2022-11-19 14:21:35,079 WARN [org.keycloak.events] (executor-thread-6) type=LOGIN_ERROR, realmId=cf173a2d-6041-4aaf-83fd-58fba4180d1d, clientId=vuejs, userId=null, ipAddress=172.17.0.1, error=invalid_redirect_uri, redirect_uri=http://localhost:8080/
```

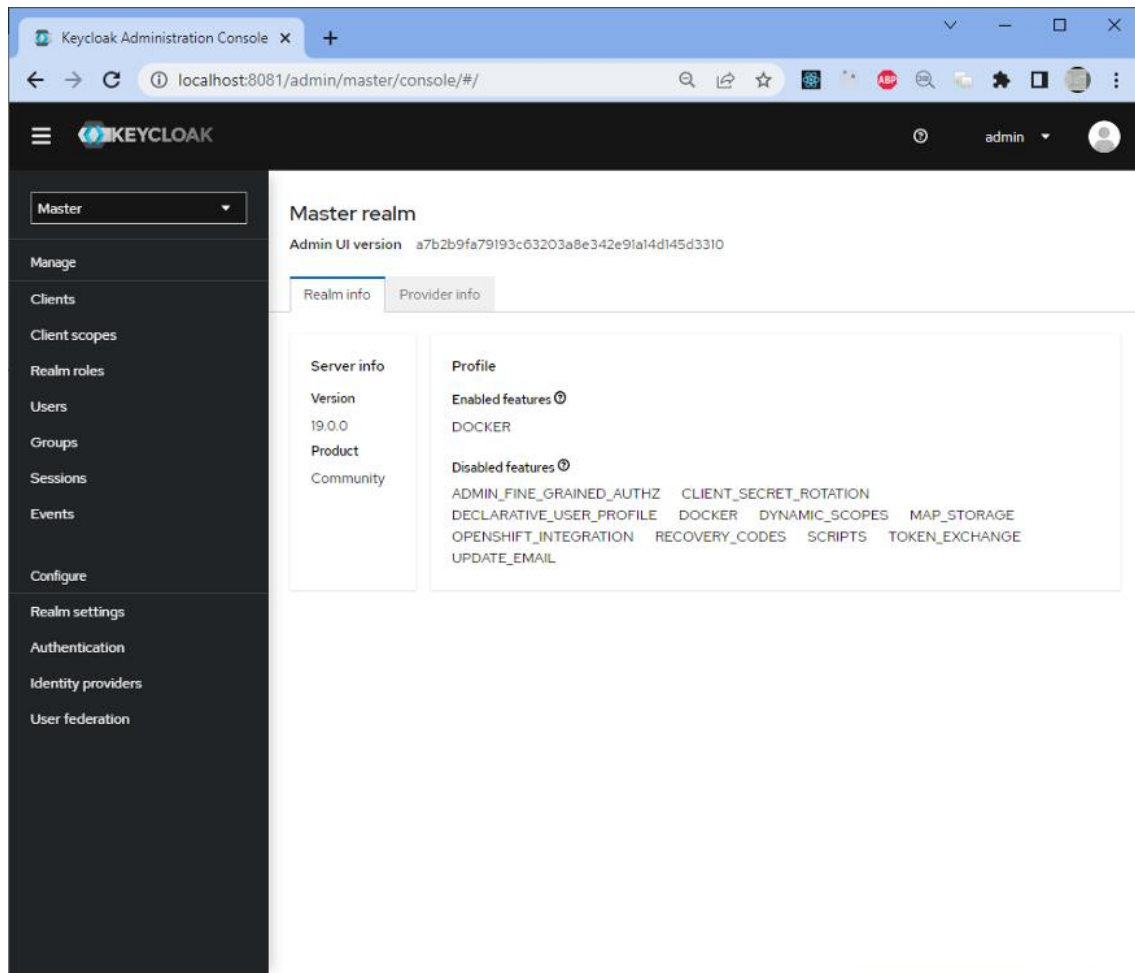
Once done, we just need to change the config options so we use the production Keycloak instance of your enterprise.

We use on our host the port 8081 because Vuejs already uses port 8080.

We get:



We log in with the admin account:



Let's add a realm "vue" and a client "vuejs":

Vue

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Identity providers

User federation

Clients > Create client

Create client

Clients are applications and services that can request authentication of a user.

1 General Settings

Client type ⓘ OpenID Connect

Client ID * ⓘ vuejs

Name ⓘ

Description ⓘ

Always display in console ⓘ ☐ Off

NextBackCancel

Clients > Create client

Create client

Clients are applications and services that can request authentication of a user.

1 General Settings

2 Capability config

Client authentication ⓘ ☐ Off

Authorization ⓘ ☐ Off

Authentication flow

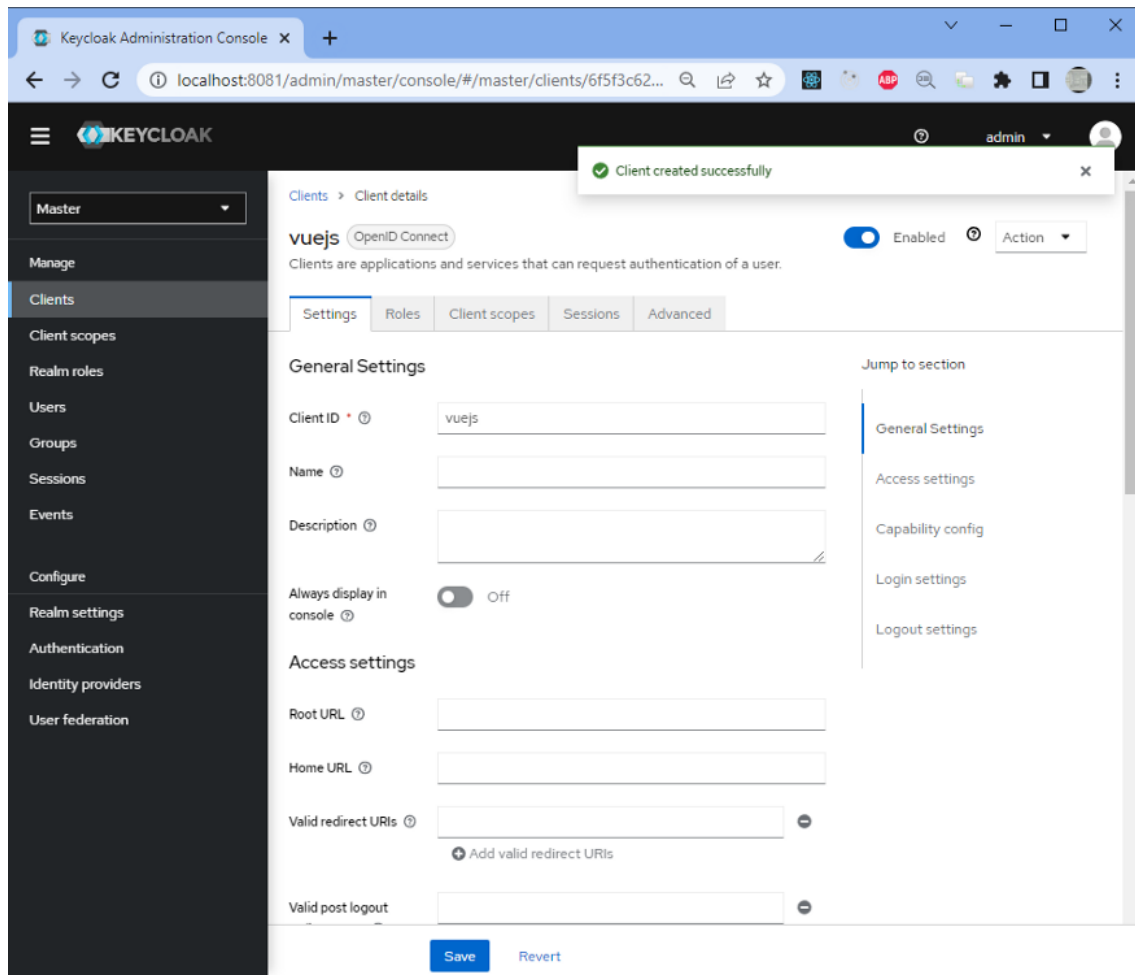
☒ Standard flow ⓘ☐ Implicit flow ⓘ☐ OAuth 2.0 Device Authorization Grant ⓘ☐ OIDC CIBA Grant ⓘ

☐ Direct access grants ⓘ☐ Service accounts roles ⓘ

Here we have the config for the OpenId Connect:

- Client authentication is Off: This defines the type of the OIDC client. When it's ON, the OIDC type is set to confidential access type. When it's OFF, it is set to **public access type. We leave it to the public access type.**
- Authorization is Off: Enable/Disable fine-grained authorization support for a client and disabled, we don't need it here. This option is available only when we use the confidential access type. As our vue app is a front-end app with javascript we can't trust it to use the confidential access type with the secret so we use the public access type.
- In the Authentication flow, we must leave "Standard flow" checked: This enables standard OpenID Connect redirect-based authentication with authorization code. In terms of OpenID Connect or OAuth2 specifications, this enables support of 'Authorization Code Flow' for this client.
- Direct access grants: we must disable this, as it provides the token against the username/providers. This option enables support for Direct Access Grants, which means that client has access to username/password of user and exchange it directly with Keycloak server for access token. In terms of OAuth2 specification, this enables support of 'Resource Owner Password Credentials Grant' for this client.

Now let's click on save.



We will focus on the access settings part now

Access Settings

Access settings

Root URL [?](#)

Home URL [?](#)

Valid redirect URIs [?](#) [Add valid redirect URIs](#)

Valid post logout redirect URIs [?](#) [Add valid post logout redirect URIs](#)

Web origins [?](#) [Add web origins](#)

Admin URL [?](#)

We will focus on :

- Valid redirect URIs: Valid URI pattern a browser can redirect to after a successful login. Simple wildcards are allowed such as 'http://example.com/*'. Relative path can be specified too such as /my/relative/path/*. Relative paths are relative to the client root URL, or if none is specified the auth server root URL is used. For SAML, you must set valid URI patterns if you are relying on the consumer service URL embedded with the login request. In our case we will use our vuejs dev URL: http://localhost:8080/*
- Valid post logout redirect URIs: Valid URI pattern a browser can redirect to after a successful logout. A value of '+' will use the list of valid redirect uris. Simple wildcards are allowed such as 'http://example.com/*'. Relative path can be specified too such as /my/relative/path/*. Relative paths are relative to the client root URL, or if none is specified the auth server root URL is used. For SAML, you must set valid URI patterns if you are relying on the consumer service URL embedded with the login request. we will put just: <http://xxx:8080> where xxx is our ip
- Web origins: Allowed CORS origins. To permit all origins of Valid Redirect URIs, add '+'. This does not include the '*' wildcard though. To permit all origins, explicitly add '*'. we will use <http://xxx:8080> and http://xxx:8080/*

Access settings

Root URL ⓘ

Home URL ⓘ

Valid redirect URIs ⓘ ⓘ
[+ Add valid redirect URIs](#)

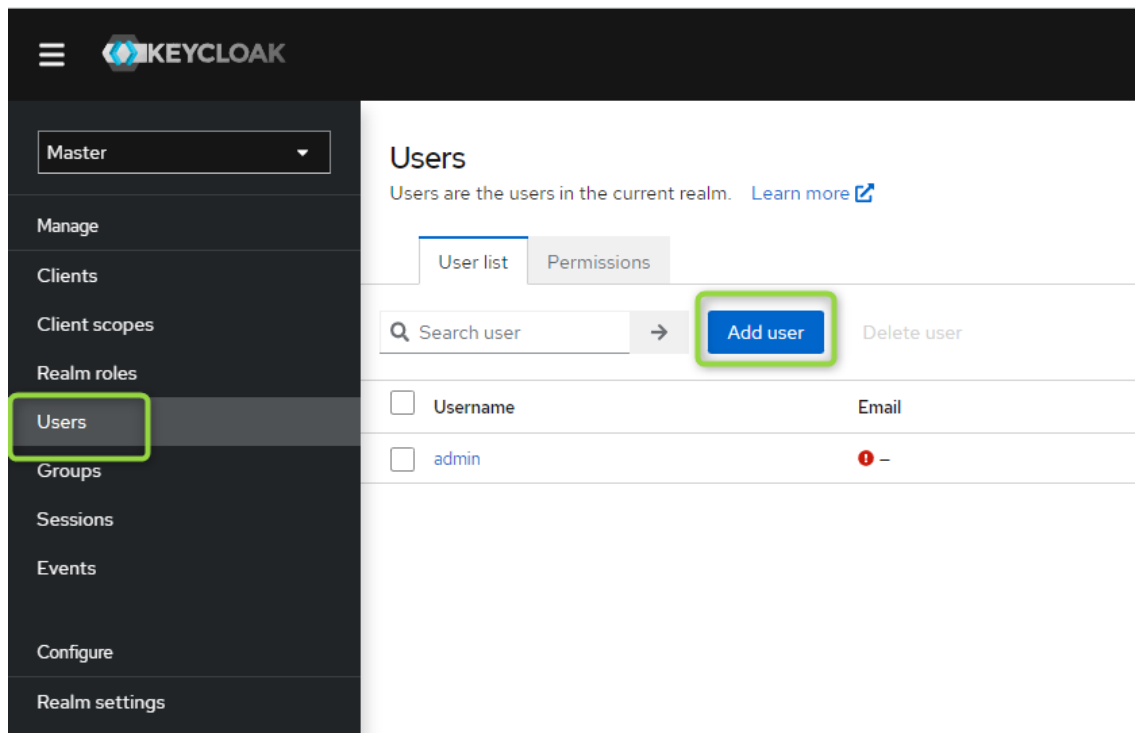
Valid post logout redirect URIs ⓘ ⓘ
[+ Add valid post logout redirect URIs](#)

Web origins ⓘ ⓘ
 ⓘ
[+ Add web origins](#)

We press on “save”

Let’s create now the user and role.

Users



The image shows the Keycloak administration console. On the left is a dark sidebar with a menu. The 'Users' option is highlighted with a green box. The main area is titled 'Users' and contains a 'User list' tab, a search bar, and an 'Add user' button (also highlighted with a green box). Below these is a table with columns for 'Username' and 'Email'. One user, 'admin', is listed with a red exclamation mark icon next to the email field.

Keycloak

Master

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Users

Users are the users in the current realm. [Learn more](#)

User list Permissions

Search user → **Add user** Delete user

<input type="checkbox"/>	Username	Email
<input type="checkbox"/>	admin	!

[Users](#) > [Create user](#)

Create user

Username *	<input type="text"/>
Email	<input type="text"/>
Email verified ⓘ	<input type="checkbox"/> Off
First name	<input type="text"/>
Last name	<input type="text"/>
Enabled ⓘ	<input checked="" type="checkbox"/> On
Required user actions ⓘ	<div>Select action ▼</div>
Groups ⓘ	<div>Join Groups</div>
<div>Create Cancel</div>	

Let's just add the username.

[Users](#) > [Create user](#)

Create user

Username *

Email

Email verified ?

☐ Off

First name

Last name

Enabled ?

☒ On

Required user actions
?

Groups ?

[Join Groups](#)

Create

Cancel

Press on “Create”

Let's add a password:

Users > User details

vueuser

Details Attributes **Credentials** Role mapping Groups Consents Identity provider links Sessions



No credentials

This user does not have any credentials. You can set password for this user.

Set password

Users > User details

vueuser

Details Attributes **Credentials** Role mapping Groups Consents Identity provider links Sessions



Set password for vueuser

Password *

Password confirmation *

Temporary ⓘ ☐ Off

Save Cancel

We turn off “Temporary”

Now let’s create the Roles of our app.

App’s Roles

Master

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Clients > Client details

vuejs (OpenID Connect)

Clients are applications and services that can request authentication of a user.

SettingsRolesClient scopesSessionsAdvanced

+

No roles for this client

You haven't created any roles for this client. Create a role to get started.

Create role

Clients > Client details > Create role

Create role

Role name *

admin

Description

Clients > Client details

vuejs (OpenID Connect)

Clients are applications and services that can request authentication of a user.

SettingsRolesClient scopesSessionsAdvanced

Q Search role by name

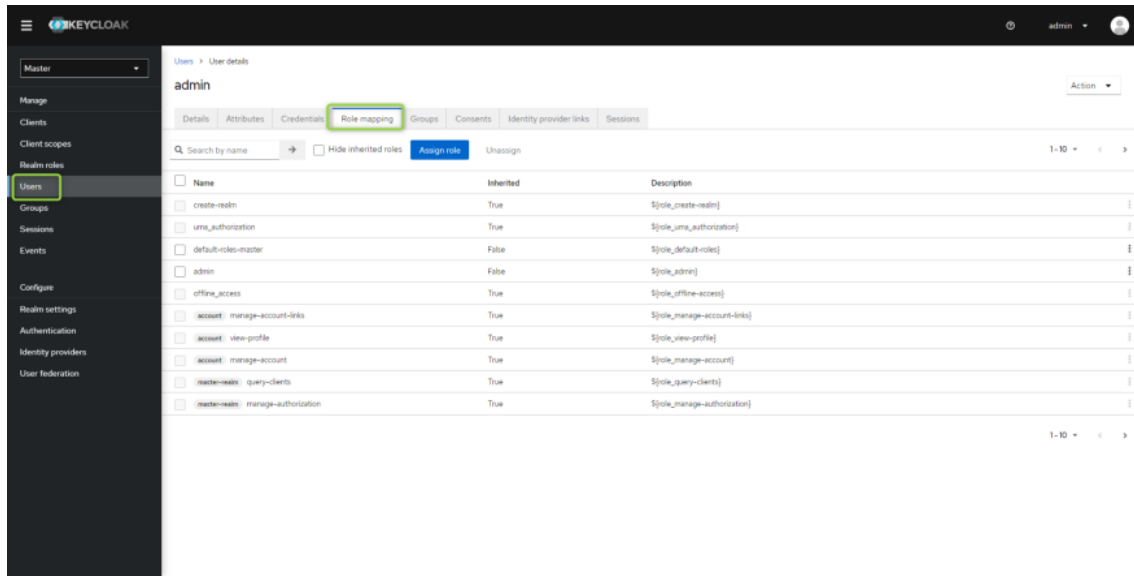
→

Create role

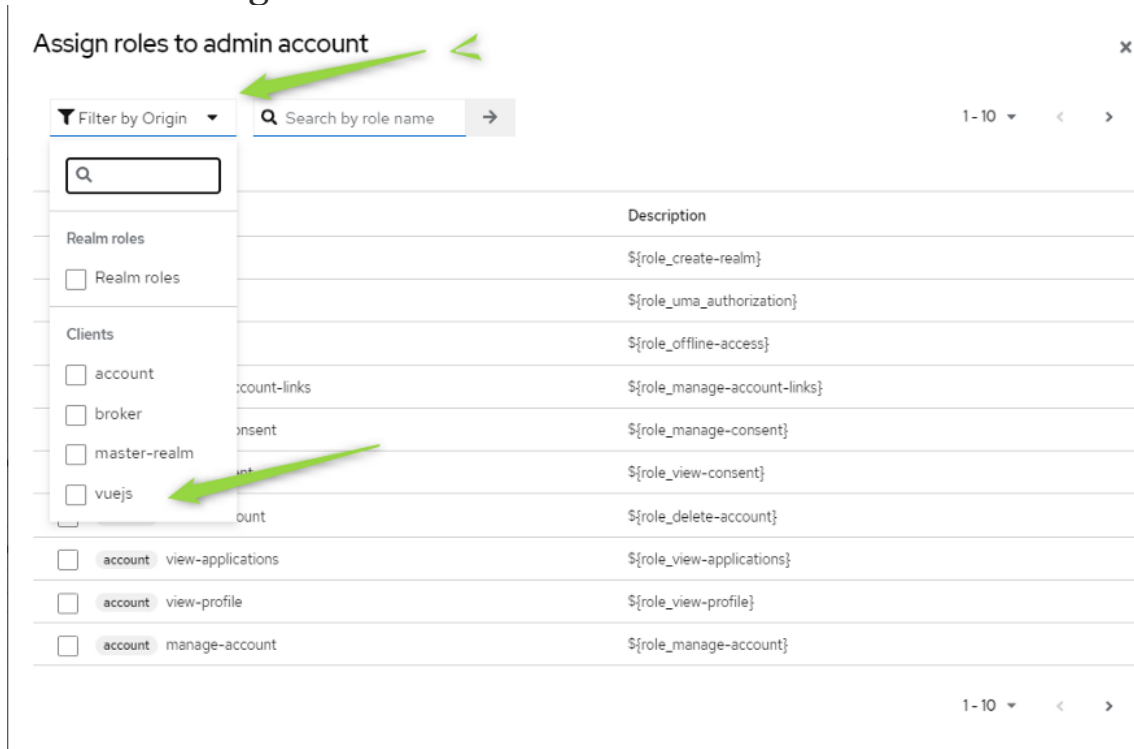
Role name	Composite
admin	False

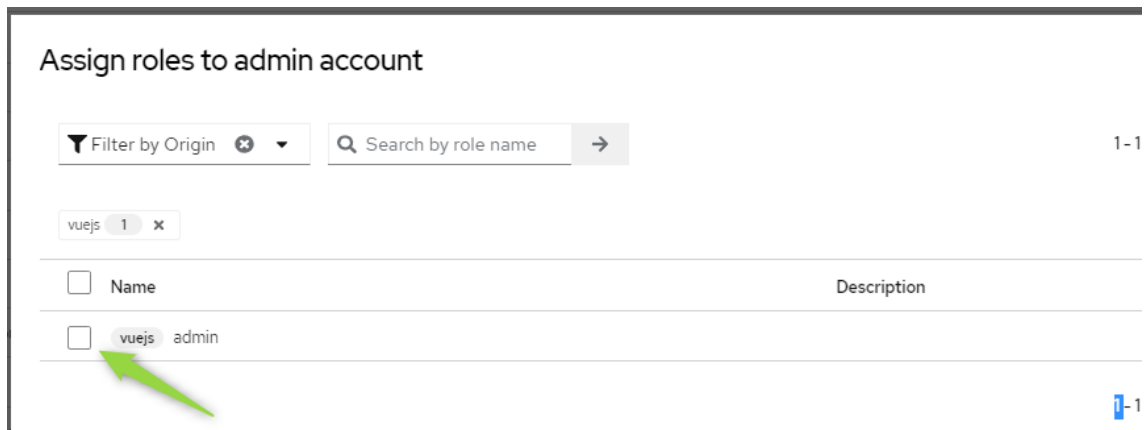
We added a simple role “admin” to our client “vuejs”.

Let’s associate this role with our user:



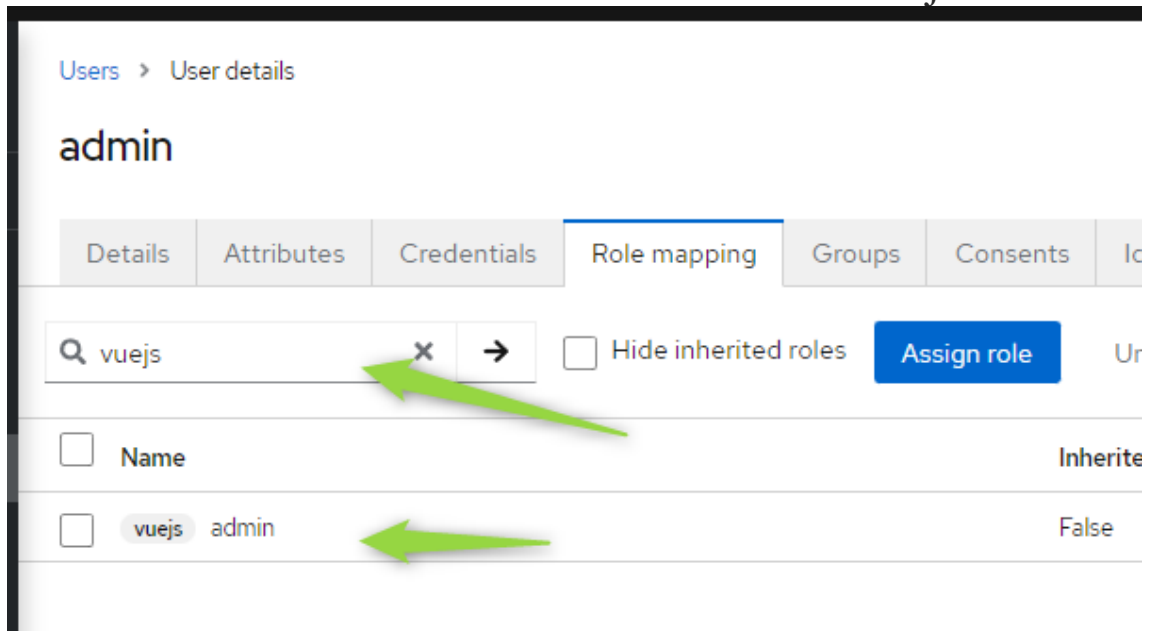
Click on “Assign Role”





Click on “Assign”

We check we have the role “admin” of the client “vuejs”:



Now we are done with Keycloak administration.

Let's go back to our vuejs app and secure it with the package keycloak-js.

Let's check the keycloak server version first:



We use version 19.0.3.

So let's get the keycloak-js npm package of the same version.

Keycloak js npm

keycloak-js

Keycloak Adapter. Latest version: 20.0.1, last published: 9 days ago. Start using keycloak-js in your project by...

[www.npmjs.com](https://www.npmjs.com/package/keycloak-js)

The versions are on the following link:

keycloak-js

Keycloak Adapter. Latest version: 20.0.1, last published: 9 days ago. Start using keycloak-js in your project by...

[www.npmjs.com](https://www.npmjs.com/package/keycloak-js)

Current Tags

Version	Downloads (Last 7 Days)	Tag
20.0.1	8,629	latest

Version History

Version	Downloads (Last 7 Days)	Published
20.0.1	8,629	9 days ago
20.0.0	3,260	15 days ago
19.0.3	26,090	a month ago
19.0.2	14,982	2 months ago
19.0.1	9,501	4 months ago
19.0.0	284	4 months ago

We will use the latest minor version of the major version **19.0.3**

keycloak-js

Keycloak Adapter. Latest version: 20.0.1, last published: 10 days ago. Start using keycloak-js in your project by...

www.npmjs.com

It gives us the command:

```
npm i keycloak-js@19.0.3
```

Let's do it now on our VS Code terminal

```
Updated crash reporting process

TERMINAL  DEBUG CONSOLE  PROBLEMS  OUTPUT

Microsoft Windows [Version 10.0.19043.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Tutorial\keycloak\vuejs3webapikeycloak\front-end>npm i keycloak-js@19.0.3
```

```
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-end>npm i keycloak-
js@19.0.3
npm WARN deprecated @hapi/topo@3.1.6: This version has been deprecated
and is no longer supported or maintained
npm WARN deprecated @hapi/hoek@8.5.1: This version has been deprecated
and is no longer supported or maintained
npm WARN deprecated @hapi/bourne@1.3.2: This version has been
deprecated and is no longer supported or maintained
npm WARN deprecated @hapi/address@2.1.4: Moved to 'npm install
@sideway/address'
npm WARN deprecated request@2.88.2: request has been deprecated, see
https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer
supported
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or
higher. Older versions may use Math.random() in certain
circumstances, which is known to be problematic. See
https://v8.dev/blog/math-random for details.
npm WARN deprecated urix@0.1.0: Please see
https://github.com/lydell/urix#deprecated
npm WARN deprecated resolve-url@0.2.1:
https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated querystring@0.2.0: The querystring API is
considered Legacy. new code should use the URLSearchParams API
instead.
npm WARN deprecated chokidar@2.1.8: Chokidar 2 will break on node
v14+. Upgrade to chokidar 3 with 15x less dependencies.
npm WARN deprecated fsevents@1.2.13: fsevents 1 will break on node
v14+ and could be using insecure binaries. Upgrade to fsevents 2.
npm WARN deprecated eslint-loader@2.2.1: This loader has been
deprecated. Please use eslint-webpack-plugin
npm WARN deprecated html-webpack-plugin@3.2.0: 3.x is no longer
supported

> yorkie@2.0.0 uninstall
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\yorkie
> node bin/uninstall.js

husky
uninstalling Git hooks
done

npm WARN rm not removing
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\.bin\rimraf.cmd as it wasn't installed
```



```

by C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\rimraf
npm WARN rm not removing
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\.bin\rimraf as it wasn't installed by
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\rimraf
npm WARN rm not removing
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\.bin\uuid.cmd as it wasn't installed by
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-end\node_modules\uuid
npm WARN rm not removing
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\.bin\uuid as it wasn't installed by
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-end\node_modules\uuid
npm WARN rm not removing
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\.bin\semver.cmd as it wasn't installed
by C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\semver
npm WARN rm not removing
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\.bin\semver as it wasn't installed by
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\semver
npm WARN rm not removing
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\.bin\json5.cmd as it wasn't installed by
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-end\node_modules\json5
npm WARN rm not removing
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\.bin\json5 as it wasn't installed by
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-end\node_modules\json5

> yorkie@2.0.0 install
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\yorkie
> node bin/install.js

setting up Git hooks
can't find .git directory, skipping Git hooks installation

```

```

> core-js@3.26.1 postinstall
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-
end\node_modules\@vue\babel-preset-app\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"

```

Thank you for using core-js (<https://github.com/zloirock/core-js>)
for polyfilling JavaScript standard library!

The project needs your help! Please consider supporting of core-js:

```

> https://opencollective.com/core-js
> https://patreon.com/zloirock
> bitcoin: bclqlea7544qtsmj2rayg0lthvza9fau63ux0fstcz

```

Also, the author of core-js (<https://github.com/zloirock>) is looking
for a good job -)

```
> ejs@2.7.4 postinstall
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-end\node_modules\ejs
> node ./postinstall.js
```

Thank you for installing EJS: built with the Jake JavaScript build tool (<https://jakejs.com/>)

npm notice created a lockfile as package-lock.json. You should commit this file.

```
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.3.2
(node_modules\watchpack\node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for
fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current:
{"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@^1.2.7
(node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for
fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current:
{"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fork-ts-checker-
webpack-plugin-v5@npm:fork-ts-checker-webpack-plugin@^5.0.11
(node_modules\@vue\cli-plugin-typescript\node_modules\fork-ts-checker-
webpack-plugin-v5):
npm WARN notarget SKIPPING OPTIONAL DEPENDENCY: No matching version
found for fork-ts-checker-webpack-plugin-v5@5.2.1.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: vue-loader-
v16@npm:vue-loader@^16.1.0 (node_modules\@vue\cli-
service\node_modules\vue-loader-v16):
npm WARN notarget SKIPPING OPTIONAL DEPENDENCY: No matching version
found for vue-loader-v16@16.8.3.
```

```
+ keycloak-js@19.0.3
added 96 packages from 38 contributors, removed 76 packages, updated
1252 packages and audited 1362 packages in 112.25s
```

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

```
found 13 vulnerabilities (2 low, 2 moderate, 7 high, 2 critical)
  run `npm audit fix` to fix them, or `npm audit` for details
```

New major version of npm available! 6.14.4 -> 9.1.1
Changelog: <https://github.com/npm/cli/releases/tag/v9.1.1>
Run npm install -g npm to update!

```
C:\Tutorial\keycloak\vuejs3webapikeycloak\front-end>
```

Pretty heavy with almost 100 packages!

Now we are ready to add the codebase which will use the js adapter.

Adding Keycloak JS TS

When we added the keycloak-js we can see it in the dependencies:

```
{
  "name": "front-end",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "serve": "vue-cli-service serve",
    "build": "vue-cli-service build",
    "lint": "vue-cli-service lint"
  },
  "dependencies": {
    "core-js": "^3.6.5",
    "keycloak-js": "^19.0.3",
    "vue": "^3.0.0"
  },
  "devDependencies": {
    "@typescript-eslint/eslint-plugin": "^4.18.0",
    "@typescript-eslint/parser": "^4.18.0",
    "@vue/cli-plugin-babel": "~4.5.13",
    "@vue/cli-plugin-eslint": "~4.5.13",
    "@vue/cli-plugin-typescript": "~4.5.13",
    "@vue/cli-service": "~4.5.13",
    "@vue/compiler-sfc": "^3.0.0",
    "@vue/eslint-config-typescript": "^7.0.0",
    "eslint": "^6.7.2",
    "eslint-plugin-vue": "^7.0.0",
    "typescript": "~4.1.5"
  },
  "eslintConfig": {
    "root": true,
    "env": {
      "node": true
    },
    "extends": [
      "plugin:vue/vue3-essential",
      "eslint:recommended",
      "@vue/typescript/recommended"
    ],
    "parserOptions": {
```

```
    "ecmaVersion": 2020
  },
  "rules": {}
},
"browserslist": [
  "> 1%",
  "last 2 versions",
  "not dead"
]
```

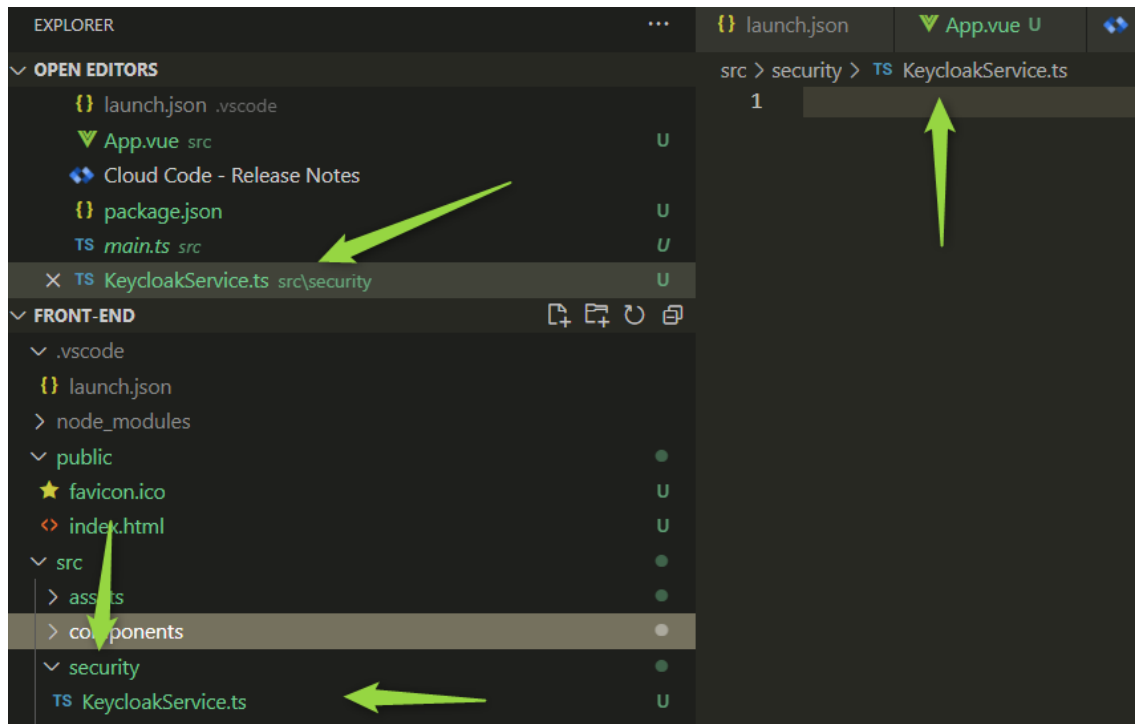
Now we can import the keycloak js adapter features in our application.

For this, we will enjoy Typescript by creating a Service with its interface.

It will make it easier to use it anywhere and change its implementation.

First, let's create a folder that we can call "security" for our keycloak service, that service instance will make it easier to use it anywhere in our React project.

Let's add the ts file called "KeycloakService.ts"



We add the following code:

- First, we **import** the Keycloak class from the keycloak js
- we create an **instance** keycloakInstance
- We create a **function** Login with the function to call if the user is authenticated. Here the function will display our main component of our React application.
To understand this part please consult the link https://www.keycloak.org/docs/latest/securing_apps/index.html#javascript_adapter

Very important to remember, the documentation says at [https://www.keycloak.org/docs/latest/securing_apps/index.html# javascript implicit flow](https://www.keycloak.org/docs/latest/securing_apps/index.html#javascript_implicit_flow)

By default, the JavaScript adapter uses the [Authorization Code](#) flow.

So we made sure that the client side can communicate with the server side by using the Authorization Code flow, the Keycloak Javascript Adapter by default uses the Authorization Code flow that we enabled for the client “MyApp” on the Keycloak server configuration

We don't have to do anything about the Flow in our code.

Now we are ready to make sure it works :) in our **main.ts** file before our application is rendered we use

```
import Keycloak from "keycloak-js";

const keycloakInstance = new Keycloak();

interface CallbackOneParam<T1 = void, T2 = void> {
  (param1: T1): T2;
}

/**
 * Initializes Keycloak instance and calls the provided callback
 * function if successfully authenticated.
 *
 * @param onAuthenticatedCallback
 */
const Login = (onAuthenticatedCallback: CallbackOneParam) => {
  keycloakInstance
    .init({ onLoad: "login-required" })
    .then(function (authenticated) {
      authenticated ? onAuthenticatedCallback() : alert("non
authenticated");
    })
    .catch((e) => {
      console.dir(e);
      console.log(`keycloak init exception: ${e}`);
    });
};

const KeyCloakService = {
  CallLogin: Login,
};
```

```
export default KeyCloakService;
```

Let's use our service keycloak in **main.ts**:

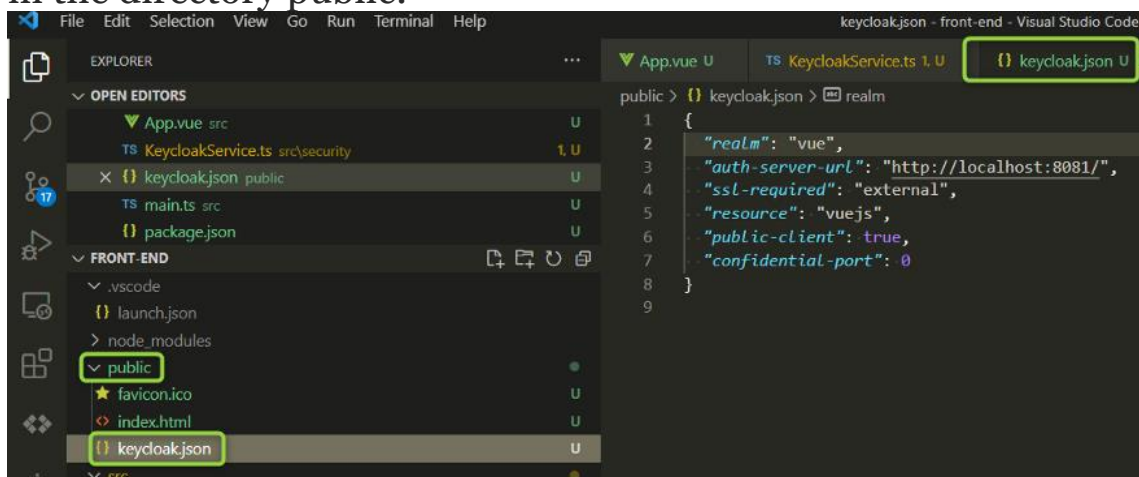
```
import { createApp } from "vue";
import App from "../App.vue";
import KeyCloakService from "../security/KeycloakService";

const renderApp = () => {
  createApp(App).mount("#app");
};

KeyCloakService.CallLogin(renderApp);
```

it's simple right!

Last thing we need to add the keycloak config file : keycloak.json in the directory public:



```
{
  "realm": "vue",
  "auth-server-url": "http://localhost:8081/",
  "ssl-required": "external",
  "resource": "vuejs",
  "public-client": true,
  "confidential-port": 0
}
```

So we can change the url of our keycloak server, give the realm and the type of client.

Now let's go to our vue application.

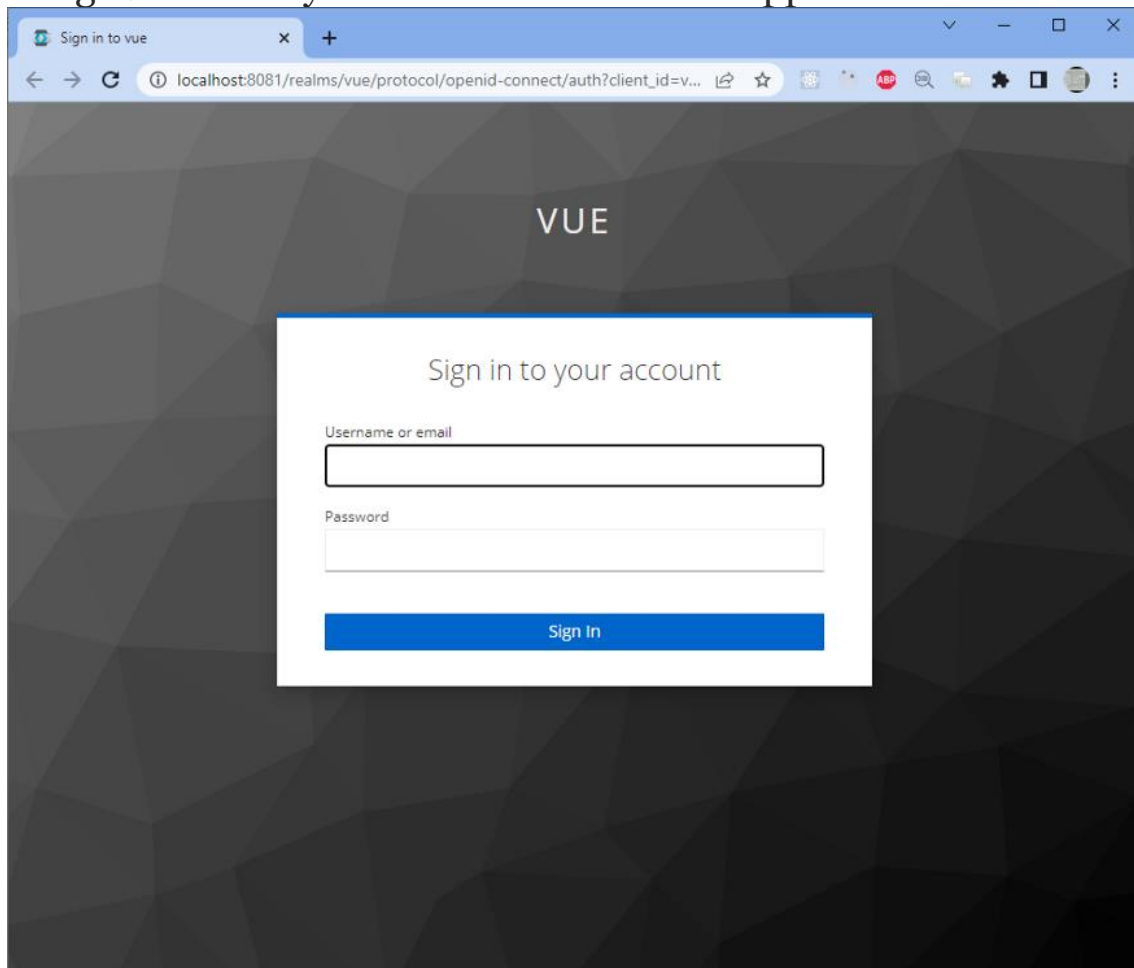
Login in Keycloak

When we open our browser from VS.Code :

<http://xxx:8080/>

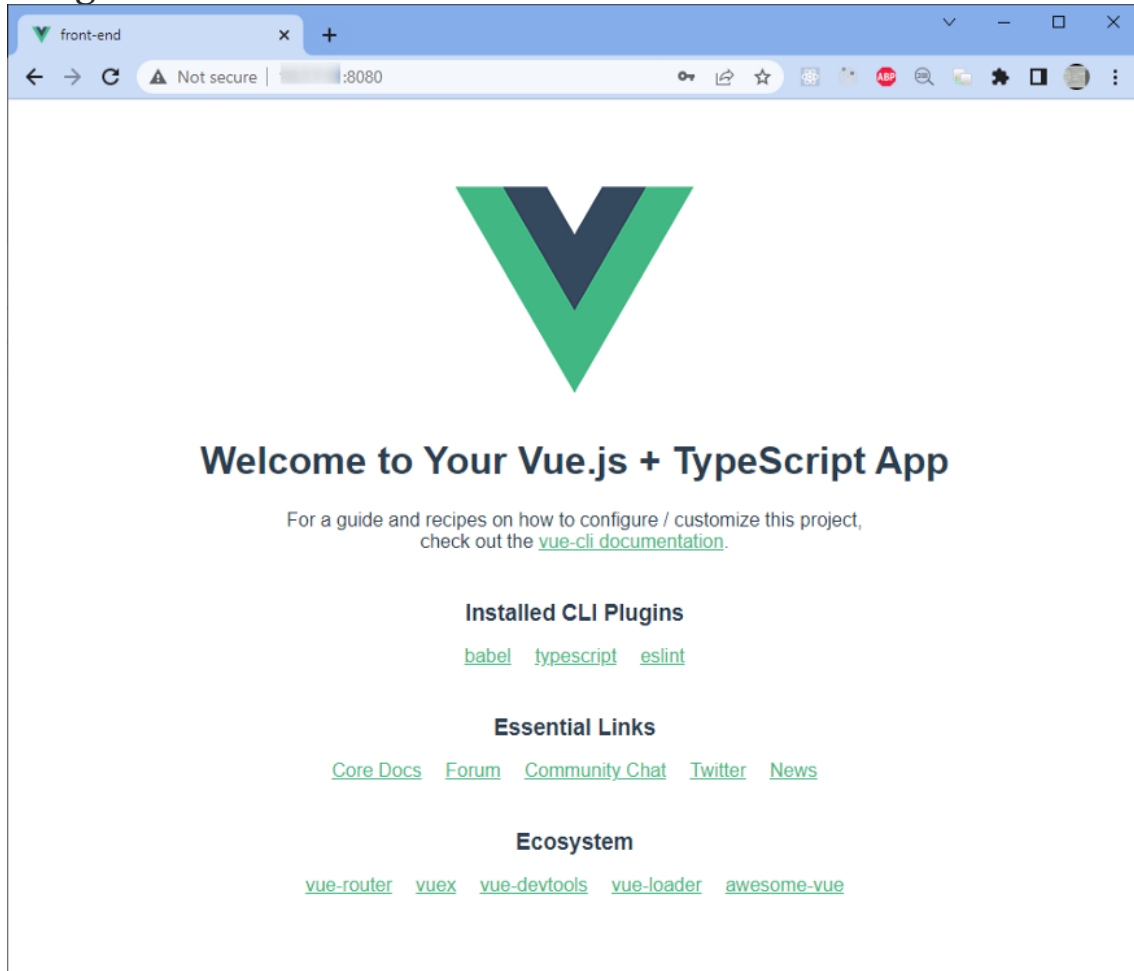
with xxx your ip address

We go back to keycloak instead of the vue app!



Let's log in!

We get back



Congrats! we managed to use Keycloak for securing our vue demo with JWT!

We can see the JWT in the chrome network dev tool:


```
id_token
:
"eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIA6ICJZbUt1RjU4aGhtOE9xYWd0T2lqTTNhQzlyY00yUmLOU091RWg2SVhSYUJRin0.eyJleHAiOjE2Njg4NzU4NDYsImVhdCmMTY2ODg3NTU0NiwiYXV0aF90aWllIjoxnNjY4ODclNDM4LCJqdGkiOiJhYmUyNGYOMSOwOGZlLTQwMTgtYWEyZi0wOTk0YmY4OWM5MDUiLCJpc3MiOiJodHRwOi8vbG9jYXRob3NOOjgwODEvcmlvbmGlzlL3ZlZSIImFlZCI6InZlZWpzIiwic3ViIjoibDRKOTFlOTgtNDVkOS00MGUYWFfkmWQtODU3NTQlMWE2ZGIWIiwidHlwIjoisSUQiLCJhenAiOiJ2dWVqcysIm5vbnNlIjoizMZhMDMzMTEtODlhMC00YWVkbGltTGwNGQTzGM3OWVhYmIxYjVjIiwic2Vzc2lvbl9zdGF0ZSI6ImM2ODdkZmY5LTNiY2ItNGMzZi05MTY1LWNkMjcwMjI4MTkxMiIsImF0X2hc2giOiI2LVZQNHFuU3NuOXNNRzhWbjNFZ2Z3IiwiYWNYIjoimCIsInNpZCI6ImM2ODdkZmY5LTNiY2ItNGMzZi05MTY1LWNkMjcwMjI4MTkxMiIsImVtYWlsZX3ZlcmlmaWVkJjpmYWxzZWsicHJlZmVycmVkdXVzZXJuYW1lIjoiaWVudDlxNCiIsImdpdmVuX25hbWUiOiIiLCJmYWlpbHlfbmFtZSI6IiJ9.AmfzbouyfYrNT916xW3iro0YbUkqh0GX2wfRIvn2e9HuWA2oEIKG6Gqa-P-Obzk8aoxZ6QInqpCWnATYtH19ziJB5i86cJB0NFznDttaCuDsIEPWRJ4vG5kX6YUpQuJLq-vunWAKzxXC2SwD6AyrTgAVnfveH0ylUbdfBF8GbDIglULYBkwOC8B3PpRqbyffRIksLf1PTURGiFbk5DM3mVY99sqYL_mOFnl__6E7g3F1y_YcxLuzGethHtVJ8hcoMR6xWzO2GY7Ny3ZoqwRr5-iwf-_vaCXW6if67E-aX69JRBaZBXH8egrgAo5c9yyXDFU_2T9m6Ry6wq8ysXA"not-before-policy
:
0refresh_expires_in
:
1800refresh_token
:
"eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIA6ICJLMWQlZWM0ZC0yZmQwLTQxNDAtYTIZzI05Mjg1YzBiZmIyY2QifQ.eyJleHAiOjE2Njg4NzczNDYsImVhdCmMTY2ODg3NTU0NiwiianRpIjoizDgxMjM0MmEtNTgyZi00N2JjLTglNzmtYjlizWZknZI5ZmNlIiwiaXNzIjoiaHR0cDovL2xvY2Fsag9zdDo4MDgxL3JlYWxtcy92dWUiLCJhdWQiOiJodHRwOi8vbG9jYXRob3NOOjgwODEvcmlvbmGlzlL3ZlZSIImFlZCI6InNlYiI6IjQ0ZDKxZTk4LTQlZDKtNDBlMilhZDFkLTglNzU0NTFhNmRiMcIsInR5cCI6IlJlLnJlc2giLCJhenAiOiJ2dWVqcysIm5vbnNlIjoizMZhMDMzMTEtODlhMC00YWVkbGltTGwNGQTzGM3OWVhYmIxYjVjIiwic2Vzc2lvbl9zdGF0ZSI6ImM2ODdkZmY5LTNiY2ItNGMzZi05MTY1LWNkMjcwMjI4MTkxMiIsInNjb3BlIjoib3BlbmlkIHByb2ZpbGUzZW1haWwiLCJzaWQOIjoijNjg3ZGZmOS0zyNmNlLTRjmMYtOTE2NSljZDI3MDIyODE5MTIifQ.bgIrYLBu1TiK5I1dv09YjXHJW3RUyTZMETYh6jflGr4"scope
:
"openid profile email"session_state
:
"c687dff9-3bcbb-4c3f-9165-cd2702281912"token_type
:
"Bearer"
```

Conclusion

It's the end of the part 1.

We learnt how to use Keycloak in dev mode and add keycloak js adapter to our vuejs demo application.

We will see in part 2 how to log out and use our JWT tokens: id token to get the user name and access token to see the claims and roles.

In part 3 we will see how to send the JWT access token to our secured ASP.NET Core Web API which will consume our access token and validate it.

Please if you like this tutorial, subscribe and don't hesitate to clap or ask questions. I will answer within 24h. Cheers!