

Angular WebApp Security with OAuth2

(Reverse engineering Jhipster Generated Code)

arnaud.nauwynck@gmail.com

this document: [https://github.com/Arnaud-Nauwynck/Presentations/Web/
Security-OAuth2-Angular-ReverseEngineering-Jhipster](https://github.com/Arnaud-Nauwynck/Presentations/Web/Security-OAuth2-Angular-ReverseEngineering-Jhipster)

Comparison Only with the JWT Mode

This document focus on comparisons
of the default "**JWT**" mode  "**OAuth2**" Mode

For more detailed screenshots => please see document part 1

Step 1: re-generate ANOTHER Jhipster project with same name

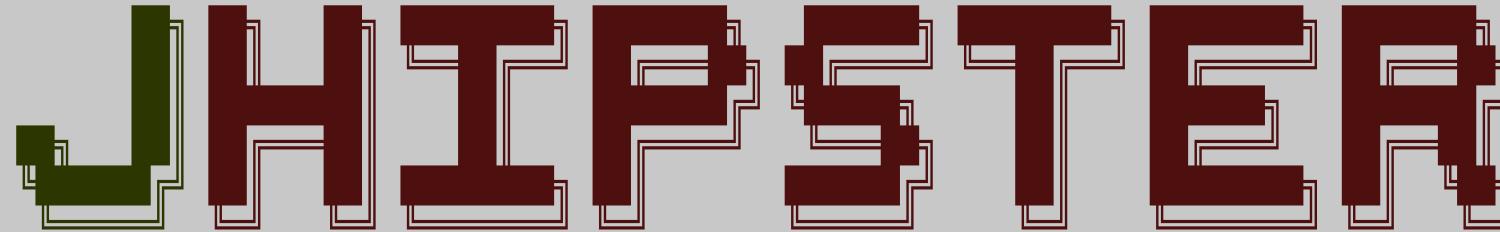
```
# check already install globally  
npm install -g generator-jhipster
```

```
# create a new empty dir  
mkdir demo-oauth2  
cd demo-oauth2
```

```
# launch jhipster generator  
jhipster
```

jhipster

```
$ jhipster
```



<https://www.jhipster.tech>

Welcome to JHipster v8.7.3

(node:19104) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)

Documentation for creating an application is at <https://www.jhipster.tech/creating-an-app/>

Application files will be generated in folder: C:\web\demo-jhipster\demo-oauth2

WARNING! Your Node version is not LTS (Long Term Support), use it at your own risk! JHipster does not support non-LTS releases
if you encounter a bug, please use a LTS version first

? What is the base name of your application? (demoOAuth2)

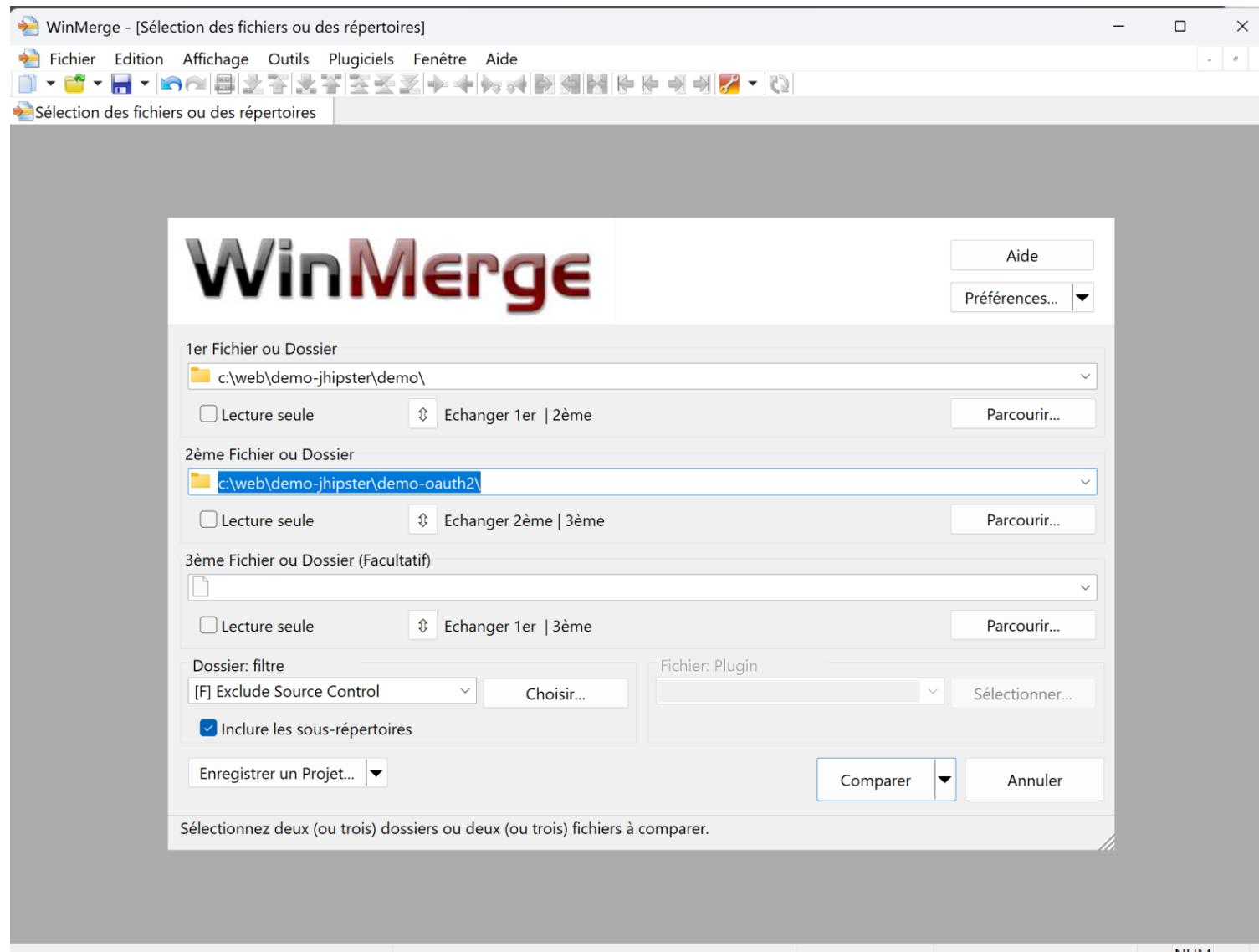
when prompted for project name (inferred for current directory name) => change to be same as "demo" project instead of "demo-oauth2"

? What is the base name of your application? demo

on question "Which type of authentication.. ?"
change from default choice #1
to choice #2: OAuth2

```
✓ What is the base name of your application? demo
✓ Which *type* of application would you like to create? Monolithic application (recommended for simple projects)
✓ What is your default Java package name? com.mycompany.myapp
✓ Would you like to use Maven or Gradle for building the backend? Maven
✓ Do you want to make it reactive with Spring WebFlux? no
? Which *type* of authentication would you like to use?
  JWT authentication (stateless, with a token)
> OAuth 2.0 / OIDC Authentication (stateful, works with Keycloak and Okta)
  HTTP Session Authentication (stateful, default Spring Security mechanism)
```

Comparing (WinMerge) Generated Apps JWT <-> OAuth2



Differences JWT <-> OAuth2

Sélection des fichiers ou des répertoires	demo\ - demo-oauth2\	c:\web\demo-jhipster\demo-oauth2\
Nom du fichier	Répertoire	Résultat de la comparaison
src	src	Les répertoires sont différents
main	src\main	Les répertoires sont différents
docker	src\main\docker	Les répertoires sont différents
java	src\main\java	Les répertoires sont différents
com	src\main\java\com	Les répertoires sont différents
mycompany	src\main\java\com\mycompany	Les répertoires sont différents
myapp	src\main\java\com\mycompany\myapp	Les répertoires sont différents
resources	src\main\resources	Les répertoires sont différents
webapp	src\main\webapp	Les répertoires sont différents
app	src\main\webapp\app	Les répertoires sont différents
account	src\main\webapp\app\account	Juste à Gauche : c:\web\demo-jhipster\demo\src...
activate	src\main\webapp\app\account\activate	Juste à Gauche : c:\web\demo-jhipster\demo\src...
password	src\main\webapp\app\account\password	Juste à Gauche : c:\web\demo-jhipster\demo\src...
password-reset	src\main\webapp\app\account\password-reset	Juste à Gauche : c:\web\demo-jhipster\demo\src...
register	src\main\webapp\app\account\register	Juste à Gauche : c:\web\demo-jhipster\demo\src...
settings	src\main\webapp\app\account\settings	Juste à Gauche : c:\web\demo-jhipster\demo\src...
account.route.ts	src\main\webapp\app\account\account.route.ts	Juste à Gauche : c:\web\demo-jhipster\demo\src...
admin	src\main\webapp\app\admin	Les répertoires sont différents
user-management	src\main\webapp\app\admin\user-management	Juste à Gauche : c:\web\demo-jhipster\demo\src...
admin.routes.ts	src\main\webapp\app\admin\admin.routes.ts	Les fichiers de texte sont différents
core	src\main\webapp\app\core	Les répertoires sont différents
auth	src\main\webapp\app\core\auth	Les répertoires sont différents
account.service.spec.ts	src\main\webapp\app\core\auth\account.service.spec.ts	Les fichiers de texte sont différents
account.service.ts	src\main\webapp\app\core\auth\account.service.ts	Les fichiers de texte sont différents
auth-jwt.service.spec.ts	src\main\webapp\app\core\auth\auth-jwt.service.spec.ts	Juste à Gauche : c:\web\demo-jhipster\demo\src...
auth-jwt.service.ts	src\main\webapp\app\core\auth\auth-jwt.service.ts	Juste à Gauche : c:\web\demo-jhipster\demo\src...
auth-session.service.ts	src\main\webapp\app\core\auth\auth-session.service.ts	Juste à Droite : c:\web\demo-jhipster\demo-oau...
user-route-access.service.ts	src\main\webapp\app\core\auth\user-route-access.service.ts	Les fichiers de texte sont différents
interceptor	src\main\webapp\app\core\interceptor	Les répertoires sont différents
auth-expired.interceptor.ts	src\main\webapp\app\core\interceptor\auth-expired.interceptor.ts	Les fichiers de texte sont différents
auth.interceptor.ts	src\main\webapp\app\core\interceptor\auth.interceptor.ts	Juste à Gauche : c:\web\demo-jhipster\demo\src...
index.ts	src\main\webapp\app\core\interceptor\index.ts	Les fichiers de texte sont différents
entities	src\main\webapp\app\entities	Les répertoires sont différents
home	src\main\webapp\app\home	Les répertoires sont différents
layouts	src\main\webapp\app\layouts	Les répertoires sont différents
login	src\main\webapp\app\login	Les répertoires sont différents

Files Only in "JWT" mode

Sélection des fichiers ou des répertoires	demo\ - demo-oauth2\	c:\web\demo-jhipster\demo\	c:\web\demo-jhipster\demo-oauth2\	Résultat de la comparaison
src		src		Les répertoires sont différents
main		src\main		Les répertoires sont différents
java		src\main\java		Les répertoires sont différents
com		src\main\java\com		Les répertoires sont différents
mycompany		src\main\java\com\mycompany		Les répertoires sont différents
myapp		src\main\java\com\mycompany\myapp		Les répertoires sont différents
resources		src\main		Les répertoires sont différents
webapp		src\main		Les répertoires sont différents
app		src\main\webapp		Les répertoires sont différents
account		src\main\webapp\app		Juste à Gauche : c:\web\demo-jhipster\demo\src...
activate		src\main\webapp\app\account		Juste à Gauche : c:\web\demo-jhipster\demo\src...
password		src\main\webapp\app\account		Juste à Gauche : c:\web\demo-jhipster\demo\src...
password-reset		src\main\webapp\app\account		Juste à Gauche : c:\web\demo-jhipster\demo\src...
register		src\main\webapp\app\account		Juste à Gauche : c:\web\demo-jhipster\demo\src...
settings		src\main\webapp\app\account		Juste à Gauche : c:\web\demo-jhipster\demo\src...
account.route.ts		src\main\webapp\app\account		Juste à Gauche : c:\web\demo-jhipster\demo\src...
admin		src\main\webapp\app		Les répertoires sont différents
user-management		src\main\webapp\app\admin		Juste à Gauche : c:\web\demo-jhipster\demo\src...
core		src\main\webapp\app		Les répertoires sont différents
auth		src\main\webapp\app\core		Les répertoires sont différents
auth-jwt.service.spec.ts		src\main\webapp\app\core\auth		Juste à Gauche : c:\web\demo-jhipster\demo\src...
auth-jwt.service.ts		src\main\webapp\app\core\auth		Juste à Gauche : c:\web\demo-jhipster\demo\src...
interceptor		src\main\webapp\app\core		Les répertoires sont différents
auth.interceptor.ts		src\main\webapp\app\core\interceptor		Juste à Gauche : c:\web\demo-jhipster\demo\src...
login		src\main\webapp\app		Les répertoires sont différents
login.component.html		src\main\webapp\app\login		Juste à Gauche : c:\web\demo-jhipster\demo\src...
login.component.spec.ts		src\main\webapp\app\login		Juste à Gauche : c:\web\demo-jhipster\demo\src...
login.component.ts		src\main\webapp\app\login		Juste à Gauche : c:\web\demo-jhipster\demo\src...
login.model.ts		src\main\webapp\app\login		Juste à Gauche : c:\web\demo-jhipster\demo\src...
i18n		src\main\webapp		Les répertoires sont différents
test		src		Les répertoires sont différents
package-lock.json				Juste à Gauche : c:\web\demo-jhipster\demo

Files Only in "OAuth2" Mode

Nom du fichier	Répertoire	Résultat de la comparaison
src	src	Les répertoires sont différents
main	src\main	Les répertoires sont différents
docker		Les répertoires sont différents
java	src\main	Les répertoires sont différents
com	src\main\java	Les répertoires sont différents
mycompany	src\main\java\com	Les répertoires sont différents
myapp	src\main\java\com\mycompany	Les répertoires sont différents
config	src\main\java\com\mycompany\myapp	Les répertoires sont différents
OAuth2Configuration.java	src\main\java\com\mycompany\myapp\config	Juste à Droite : c:\web\demo-jhipster\demo-oau...
security	src\main\java\com\mycompany\myapp	Les répertoires sont différents
oauth2	src\main\java\com\mycompany\myapp\security	Juste à Droite : c:\web\demo-jhipster\demo-oau...
AudienceValidator.java	src\main\java\com\mycompany\myapp\security\oauth2	Juste à Droite : c:\web\demo-jhipster\demo-oau...
CustomClaimConverter.java	src\main\java\com\mycompany\myapp\security\oauth2	Juste à Droite : c:\web\demo-jhipster\demo-oau...
package-info.java	src\main\java\com\mycompany\myapp\security\oauth2	Juste à Droite : c:\web\demo-jhipster\demo-oau...
web	src\main\java\com\mycompany\myapp	Les répertoires sont différents
filter	src\main\java\com\mycompany\myapp\web	Les répertoires sont différents
OAuth2RefreshTokensWebFilter.java	src\main\java\com\mycompany\myapp\web\filter	Juste à Droite : c:\web\demo-jhipster\demo-oau...
rest	src\main\java\com\mycompany\myapp\web	Les répertoires sont différents
AuthInfoResource.java	src\main\java\com\mycompany\myapp\web\rest	Juste à Droite : c:\web\demo-jhipster\demo-oau...
LogoutResource.java	src\main\java\com\mycompany\myapp\web\rest	Juste à Droite : c:\web\demo-jhipster\demo-oau...
webapp	src\main	Les répertoires sont différents
app	src\main\webapp	Les répertoires sont différents
core	src\main\webapp\app	Les répertoires sont différents
auth	src\main\webapp\app\core	Les répertoires sont différents
auth-session.service.ts	src\main\webapp\app\core\auth	Juste à Droite : c:\web\demo-jhipster\demo-oau...
login	src\main\webapp\app	Les répertoires sont différents
logout.model.ts	src\main\webapp\app\login	Juste à Droite : c:\web\demo-jhipster\demo-oau...
test	src	Les répertoires sont différents

Different Files (maybe not for auth mode)

Sélection des fichiers ou des répertoires	demo\ - demo-oauth2\	c:\web\demo-jhipster\demo\	c:\web\demo-jhipster\demo-oauth2\	Résultat de la comparaison
Nom du fichier	Répertoire			
src	src			Les répertoires sont différents
main	src\main			Les répertoires sont différents
docker	src\main\docker			Les répertoires sont différents
java	src\main\java			Les répertoires sont différents
com	src\main\java\com			Les répertoires sont différents
mycompany	src\main\java\com\mycompany			Les répertoires sont différents
myapp	src\main\java\com\mycompany\myapp			Les répertoires sont différents
config	src\main\java\com\mycompany\myapp\config			Les répertoires sont différents
SecurityConfiguration.java	src\main\java\com\mycompany\myapp\config\SecurityConfiguration.java			Les fichiers de texte sont différents
domain	src\main\java\com\mycompany\myapp\domain			Les répertoires sont différents
repository	src\main\java\com\mycompany\myapp\repository			Les répertoires sont différents
security	src\main\java\com\mycompany\myapp\security			Les répertoires sont différents
service	src\main\java\com\mycompany\myapp\service			Les répertoires sont différents
web	src\main\java\com\mycompany\myapp\service\web			Les répertoires sont différents
resources	src\main\resources			Les répertoires sont différents
webapp	src\main\webapp			Les répertoires sont différents
app	src\main\webapp\app			Les répertoires sont différents
admin	src\main\webapp\app\admin			Les répertoires sont différents
admin.routes.ts	src\main\webapp\app\admin\admin.routes.ts			Les fichiers de texte sont différents
core	src\main\webapp\app\core			Les répertoires sont différents
auth	src\main\webapp\app\core\auth			Les répertoires sont différents
account.service.spec.ts	src\main\webapp\app\core\auth\account.service.spec.ts			Les fichiers de texte sont différents
account.service.ts	src\main\webapp\app\core\auth\account.service.ts			Les fichiers de texte sont différents
user-route-access.service.ts	src\main\webapp\app\core\auth\user-route-access.service.ts			Les fichiers de texte sont différents
interceptor	src\main\webapp\app\core\interceptor			Les répertoires sont différents
auth-expired.interceptor.ts	src\main\webapp\app\core\interceptor\auth-expired.interceptor.ts			Les fichiers de texte sont différents
index.ts	src\main\webapp\app\core\interceptor\index.ts			Les fichiers de texte sont différents
entities	src\main\webapp\app\entities			Les répertoires sont différents
home	src\main\webapp\app\home			Les répertoires sont différents
home.component.html	src\main\webapp\app\home\home.component.html			Les fichiers de texte sont différents
home.component.spec.ts	src\main\webapp\app\home\home.component.spec.ts			Les fichiers de texte sont différents
home.component.ts	src\main\webapp\app\home\home.component.ts			Les fichiers de texte sont différents
layouts	src\main\webapp\app\layouts			Les répertoires sont différents
login	src\main\webapp\app\login			Les répertoires sont différents
login.service.ts	src\main\webapp\app\login\login.service.ts			Les fichiers de texte sont différents
app.routes.ts	src\main\webapp\app\app.routes.ts			Les fichiers de texte sont différents

Launching Backend:

```
mvn -Pdev,tls spring-boot:run
```

(instead of mvn -Pdev spring-boot:run)
for https:// instead of http://

```
Caused by: java.net.ConnectException: Connection refused: getsockopt
    at java.base/sun.nio.ch.Net.pollConnect(Native Method)
    at java.base/sun.nio.ch.Net.pollConnectNow(Net.java:682)
    at java.base/sun.nio.ch.NioSocketImpl.timedFinishConnect(NioSocketImpl.java:542)
    at java.base/sun.nio.ch.NioSocketImpl.connect(NioSocketImpl.java:592)
    at java.base/java.net.Socket.connect(Socket.java:751)
    at java.base/sun.net.NetworkClient.doConnect(NetworkClient.java:178)
    at java.base/sun.net.www.http.HttpClient.openServer(HttpClient.java:531)
    at java.base/sun.net.www.http.HttpClient.openServer(HttpClient.java:636)
    at java.base/sun.net.www.http.HttpClient.<init>(HttpClient.java:280)
```

Need to Launch the OAuth2 Server

<https://jhipster.tech/security>

The screenshot shows a browser window displaying the JHipster website at [https://jhipster.tech/security/](https://jhipster.tech/security). The page title is "OAuth 2.0 and OpenID Connect". The left sidebar has a "Securing your app" section selected, listing various technologies: Getting Started, Installation and set up, Create application and entities, Optional Technologies (with sub-options like Securing your app, Filtering your entities, Using Elasticsearch, Using Websockets, Doing API-First development, Using a cache, Using Oracle, Using MongoDB, Using Couchbase, and Using Neo4j). The main content area discusses OAuth 2.0 and OpenID Connect, mentions Keycloak as the default provider, and provides Docker command examples for starting Keycloak.

← → ⌂ jhipster.tech/security/

JHipster

Docs Marketplace Team Sponsors JDL Studio ✎ English ▾

Getting Started

Installation and set up >

Create application and entities >

Optional Technologies ▾

Securing your app

Filtering your entities

Using Elasticsearch

Using Websockets

Doing API-First development

Using a cache

Using Oracle

Using MongoDB

Using Couchbase

Using Neo4j

OAuth 2.0 and OpenID Connect

OAuth is a stateful security mechanism, like HTTP Session. Spring Security provides excellent OAuth 2.0 and OIDC support, and this is leveraged by JHipster. If you're not sure what OAuth and OpenID Connect (OIDC) are, please see [What the Heck is OAuth?](#)

Keycloak

Keycloak is the default OpenID Connect server configured with JHipster.

To log into your application, you'll need to have Keycloak up and running. The JHipster Team has created a Docker container for you that has the default users and roles. Start Keycloak using the following command.

```
docker-compose -f src/main/docker/keycloak.yml up
```

Alternatively, you can use `npm` as follows:

```
npm run docker:keycloak:up
```

Starting Docker (Podman + Kubernetes)

The screenshot shows the Podman Desktop application window. The left sidebar has icons for Settings, Resources (selected), Proxy, Registries, Authentication, CLI Tools, Kubernetes, and Preferences. The main area displays the following sections:

- Resources**: Shows a Podman machine running with 8 CPU(s), 1.08 TB Disk size, and 8.2 GB Memory.
- kubectl**: Describes it as a command line tool for communicating with a Kubernetes cluster's control plane using the Kubernetes API. It links to kubernetes.io.
- Kind**: Shows a kind-cluster running with an endpoint at <https://localhost:53779>.
- Compose**: Describes the Compose extension providing optional command line support for [Compose files](#) with Podman. It links to [Podman Desktop Documentation](#).

At the bottom, status bar items include "No context", "AI Lab API listening on port 10434", "v1.14.1", and notification icons.

launched using Kubernetes + Port-Forward

```
Invite de commandes - k9s  + | - | X
```

Context: kind-kind-cluster
Cluster: kind-kind-cluster
User: kind-kind-cluster
K9s Rev: v0.32.6 → v0.32.7
K8s Rev: v1.27.3
CPU: n/a
MEM: n/a

<a> Attach <f> Show PortFor...
<?> Help
<l> Logs
<p> Logs Previous
<shift-f> PortForward
<s> Shell

-----| / \-----
| < \-----
| \ / \-----\-----
| \ \ / \-----\-----
V V

IDX↑	NAME	PF	IMAGE
M1	keycloak	●	quay.io/keycloak/

Co <PortForward>

default/keycloak-68d54bfc79-l2zwq|keycloak

Exposed Ports:
keycloak::9080(http)
keycloak::9443(https)

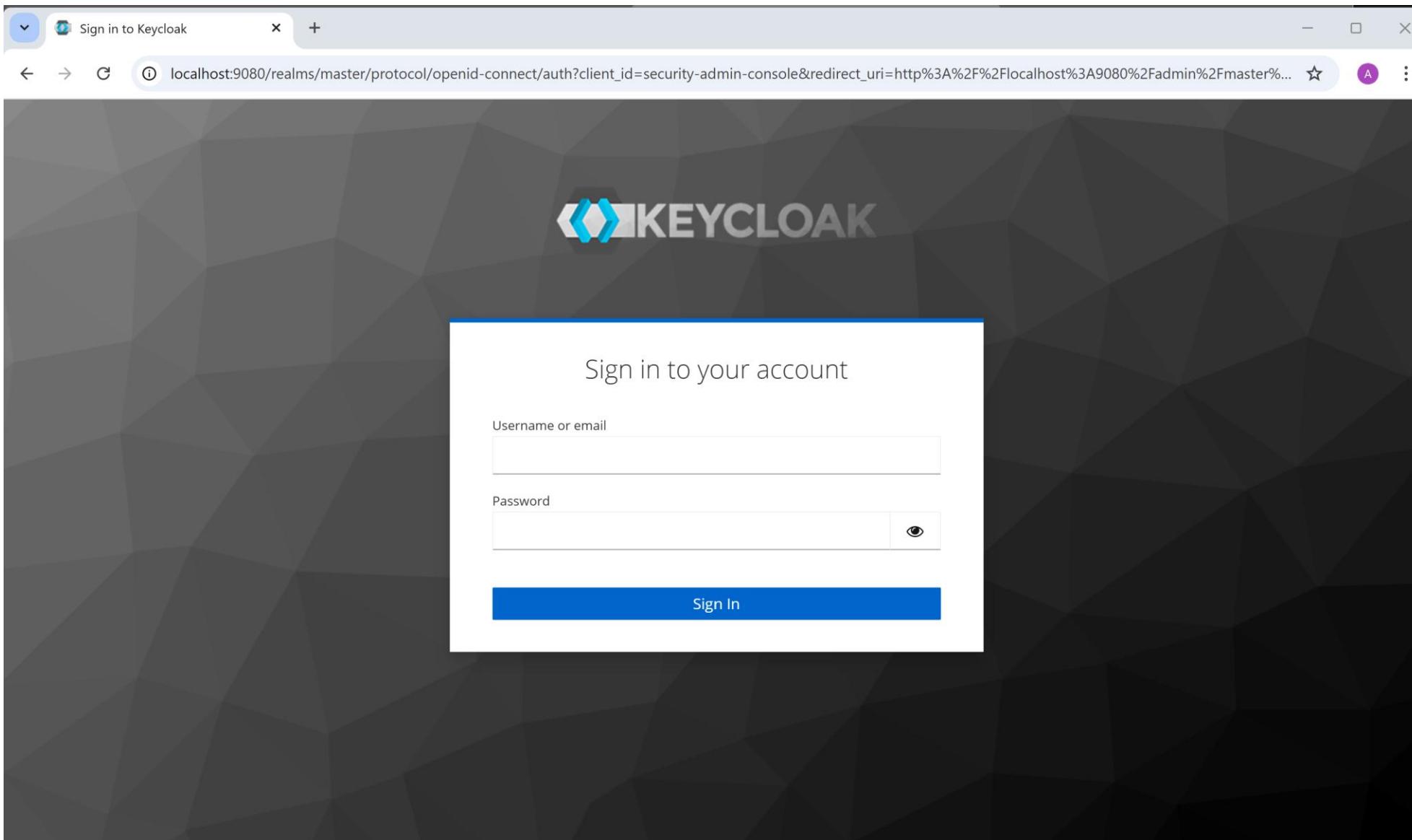
Container Port: keycloak::9080
Local Port: 9080
Address: localhost

OK Cancel

1] BES(L:R:S) CPU/R:L MEM/R:L PORTS
:off:off 0:0 0:0 http:9080

<pod> <containers>

http://localhost:9080



login admin/admin .. check "realm"

The screenshot shows the Keycloak Administration Console interface. At the top, the title bar reads "Keycloak Administration Console" and the address bar shows "localhost:9080/admin/master/console/". The top right corner displays the user "admin" and a profile icon. The left sidebar has a dark theme with white text, listing navigation options: "Keycloak" (selected), "Manage", "Clients", "Client scopes", "Realm roles", "Users", "Groups", "Sessions", "Events", "Configure", "Realm settings", "Authentication", and "Identity providers". The main content area is titled "master realm" and features a "Welcome" tab (which is selected) and "Server info" and "Provider info" tabs. The "Welcome" tab contains the text "Welcome to Keycloak" followed by a paragraph describing Keycloak's features: "Keycloak provides user federation, strong authentication, user management, fine-grained authorization, and more. Add authentication to applications and secure services with minimum effort. No need to deal with storing users or authenticating users." Below this text are four buttons: "Refer to documentation" (blue background), "View guides", "Join community", and "Read blog".

Create Realm - import jhipster-realm.json

The screenshot shows the Keycloak Administration Console interface for creating a new realm. The URL in the browser is `localhost:9080/admin/master/console/#/master/add-realm`. The top navigation bar includes the Keycloak logo, a user dropdown set to "admin", and a sidebar menu.

The main content area is titled "Create realm". It contains a brief description: "A realm manages a set of users, credentials, roles, and groups. A user belongs to and logs into a realm. Realms are isolated from one another and can only manage and authenticate the users that they control."

Below the description, there is a "Resource file" section with a file input field labeled "Drag a file here or browse to upload". The file "jhipster-realm.json" is listed with its contents:

```
1  {
2   "id": "jhipster",
3   "realm": "jhipster",
4   "displayName": "JHipster",
5   "displayNameHtml": "<div class=\"kc-logo-text\"><span>JHipster</span></div>",
6   "notBefore": 0,
7   "defaultSignatureAlgorithm": "RS256".
```

Below the file input, there is a link "Upload a JSON file".

The form fields for creating the realm are as follows:

- Realm name ***: The value is "jhipster".
- Enabled**: A toggle switch is set to "On".

At the bottom of the form are two buttons: "Create" and "Cancel".

JHipster Realm : users

The screenshot shows the Keycloak Administration Console interface. The title bar indicates the URL is `localhost:9080/admin/master/console/#/jhipster/users`. The top navigation bar includes a back/forward button, a refresh icon, a search icon, and user authentication status (admin). The left sidebar, titled 'JHipster', contains the following navigation items:

- Manage
- Clients
- Client scopes
- Realm roles
- Users** (selected)
- Groups
- Sessions
- Events
- Configure
- Realm settings
- Authentication
- Identity providers

The main content area is titled 'Users' and displays a table of users for the 'JHipster' realm. The table has columns: **Username**, **Email**, **Last name**, and **First name**. Two users are listed:

Username	Email	Last name	First name
admin	admin@localhost	Administrator	Admin
user	user@localhost	User	-

Below the table are pagination controls: '1-2' and navigation arrows. The top right of the main content area also features a 'Refresh' button.

Now can restart backend

```
$ mvn -Pdev,tls spring-boot:run
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.mycompany.myapp:demo >-----
[INFO] Building Demo 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] >>> spring-boot:3.3.5:run (default-cli) > test-compile @ demo >>>
```

```
2024-11-21T22:08:35.141+01:00  INFO 16532 --- [ restartedMain] t.j.s.ssl.UndertowSSLConfiguration      : Setting user cipher suite
order to true
2024-11-21T22:08:35.236+01:00 DEBUG 16532 --- [ restartedMain] c.m.m.w.f.OAuth2RefreshTokensWebFilter  : Filter 'OAuth2RefreshToke
nsWebFilter' configured for use
2024-11-21T22:08:35.332+01:00  INFO 16532 --- [ restartedMain] org.jboss.threads                    : JBoss Threads version 3.5
.0.Final
2024-11-21T22:08:35.442+01:00  INFO 16532 --- [ restartedMain] com.mycompany.myapp.DemoApp           : Started DemoApp in 11.057
seconds (process running for 11.636)
2024-11-21T22:08:35.458+01:00  INFO 16532 --- [ restartedMain] com.mycompany.myapp.DemoApp           :
-----
Application 'demo' is running! Access URLs:
 Local:          https://localhost:8080/
 External:        https://172.24.224.1:8080/
 Profile(s):     [dev, api-docs, tls]
-----
```

Launching Frontend: ng serve --ssl (instead of "ng serve")

```
$ ng serve --ssl
Node.js version v23.2.0 detected.
Odd numbered Node.js versions will not enter LTS status and should not be used for production.
  s://nodejs.org/en/about/releases/.
  ✓ Browser application bundle generation complete.

Initial chunk files | Names
```

```
[Browsersync] Proxying: https://localhost:4200
[Browsersync] Access URLs:
-----
      Local: https://localhost:9000
      External: https://172.24.224.1:9000
-----
        UI: http://localhost:3001
    UI External: http://172.24.224.1:3001
-----
```

Home page: <https://localhost:9000>

The screenshot shows a web browser window with two tabs: "Keycloak Administration Console" and "Welcome, Java Hipster!". The "Welcome, Java Hipster!" tab is active, displaying the JHipster homepage. The URL in the address bar is <https://localhost:9000>, which is marked as "Not secure".

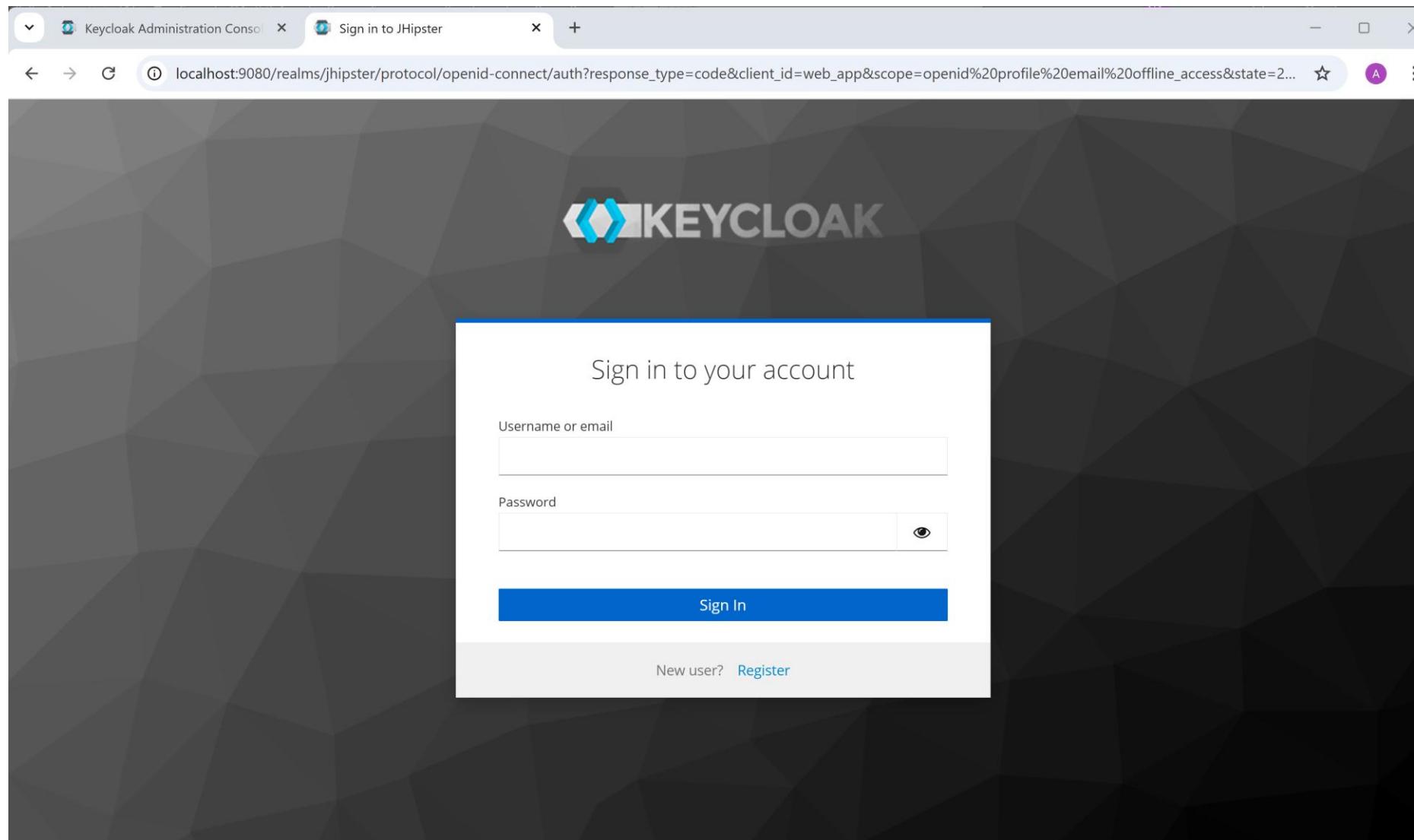
The page has a dark header with a logo and navigation links for "Home", "Language", and "Account". A red ribbon banner on the left says "Development Demo vDEV". The main content features a cartoon character with orange hair and a bow tie. The text "Welcome, Java Hipster! (Demo)" is prominently displayed, followed by "This is your homepage". A yellow callout box contains instructions for signing in with default accounts: "Administrator (login="admin" and password="admin")" and "User (login="user" and password="user"). Below this, there's a section for questions with links to the JHipster homepage, Stack Overflow, bug tracker, public chat room, and Twitter account. A note encourages users to star the project on GitHub.

If you like JHipster, don't forget to give us a star on [GitHub](#)!

This is your footer

<https://localhost:9000> using self-signed certificate => marked as "Not secure" by chrome

Click sign-in => redirect to login page of OAuth2



Http Request : check Chrome "Preserve Logs"

The screenshot shows the Chrome DevTools Network tab. At the top, there's a timeline with various request and response segments. Below the timeline is a table listing 15 network requests. The columns in the table are: Name, Status, Type, Initiator, Size, and Ti... (Time). The requests include files like oidc (status 302), auth?response_type=code&client_id=web_app&scope=openid%20profile%20email%20offline_access (status 200), and several patternfly.css and patternfly-additions.css files. The initiator column shows URLs such as :9000/oauth2/authorization/oidc and auth?response_type=code&client_id=web_app&scope=openid%20profile%20email%20offline_access. The size column shows file sizes like 1.3 kB, 7.7 kB, and 1.4 MB. The Ti... column shows times like 1.0... and 1.3... seconds.

Name	Status	Type	Initiator	Size	Ti...
oidc	302	document / Redirect		1.3 kB	1.0...
auth?response_type=code&client_id=web_app&scope=openid%20profile%20email%20offline_access&state=28Wsn3cSlrrZQzE6...	200	document	:9000/oauth2/authorization/oidc	7.7 kB	5.3...
patternfly.min.css	200	stylesheet	auth?response_type=code&client_id=web_app&scope=openid%20profile%20email%20offline_access&state=28Wsn3cSlrrZQzE6...	1.4 MB	1.3...
patternfly.min.css	200	stylesheet	auth?response_type=code&client_id=web_app&scope=openid%20profile%20email%20offline_access&state=28Wsn3cSlrrZQzE6...	183 kB	1.5...
patternfly-additions.min.css	200	stylesheet	auth?response_type=code&client_id=web_app&scope=openid%20profile%20email%20offline_access&state=28Wsn3cSlrrZQzE6...	226 kB	92...
pficon.css	200	stylesheet	auth?response_type=code&client_id=web_app&scope=openid%20profile%20email%20offline_access&state=28Wsn3cSlrrZQzE6...	789 B	1.4...
login.css	200	stylesheet	auth?response_type=code&client_id=web_app&scope=openid%20profile%20email%20offline_access&state=28Wsn3cSlrrZQzE6...	11.1 kB	1.1...
menu-button-links.js	200	script	auth?response_type=code&client_id=web_app&scope=openid%20profile%20email%20offline_access&state=28Wsn3cSlrrZQzE6...	7.8 kB	1.1...
passwordVisibility.js	200	script	auth?response_type=code&client_id=web_app&scope=openid%20profile%20email%20offline_access&state=28Wsn3cSlrrZQzE6...	954 B	1.3...
authChecker.js	200	script	auth:26	1.5 kB	1.3...
keycloak-bg.png	200	png	login.css	82.2 kB	87...
keycloak-logo-text.png	200	png	login.css	20.3 kB	1.1...
OpenSans-Light-webfont.woff	200	font	patternfly.min.css	63.5 kB	1.8...
OpenSans-Regular-webfont.woff2	200	font	patternfly.min.css	62.3 kB	1.3...
fontawesome-webfont.woff2?v=4.7.0	200	font	patternfly.min.css	77.5 kB	1.9...

At the bottom of the DevTools interface, there are summary statistics: 15 requests, 2.2 MB transferred, 2.2 MB resources, Finish: 10.54 s, DOMContentLoaded: 7.21 s, and Load: 10.70 s. Below these, there are tabs for Console, AI assistance, Rendering, and Search, along with a message center showing 'No messages'.

Request: `Https GET "/oauth2/authorization/oidc"`

The screenshot shows the Network tab in Google DevTools for a request to `https://localhost:9000/oauth2/authorization/oidc`. The request is a GET method with a status code of 302 Found. The response includes a Location header pointing to the URL `http://localhost:9080/realms/jhipster/protocol/openid-connect/auth?response_type=code&client_id=web_app&scope=openid%20profile%20email%20offline_access&state=28Wsn3cSlrrZQzE6...`.

Name: oidc

General

- Request URL: `https://localhost:9000/oauth2/authorization/oidc`
- Request Method: GET
- Status Code: 302 Found
- Remote Address: `[::1]:9000`
- Referrer Policy: strict-origin-when-cross-origin

Response Headers

- Access-Control-Allow-Origin: *
- Cache-Control: no-cache, no-store, max-age=0, must-revalidate
- Connection: close
- Content-Length: 0
- Content-Security-Policy: default-src 'self'; frame-src 'self' data; script-src 'self' 'unsafe-inline' 'unsafe-eval' https://storage.googleapis.com; style-src 'self' 'unsafe-inline'; img-src 'self' data; font-src 'self' data;
- Date: Thu, 21 Nov 2024 21:47:16 GMT
- Expires: 0
- Location: `http://localhost:9080/realms/jhipster/protocol/openid-connect/auth?response_type=code&client_id=web_app&scope=openid%20profile%20email%20offline_access&state=28Wsn3cSlrrZQzE6...`

Permissions Policy: camera=(), fullscreen=(), geolocation=(), gyroscope=(), magnetometer=(), microphone=()

15 requests | 2.2 MB transferred | 2.2 MB resources

Notice Http Status 303 + Response Header "location"

location = [http://localhost:9080/realms/jhipster/protocol/openid-connect/auth?
response_type=code
&client_id=web_app
&scope=openid%20profile%20email%20offline_access
&state=28Wsn3cSlrrZQzE6DkXKUcAAS5r573O5WLOJYA7IgSo%3D
&redirect_uri=https://localhost:9000/login/oauth2/code/oidc
&nonce=WaqexbBDkQQCIhXfMFL84RzjzTtjOyjghQUtb4YwUKs](http://localhost:9080/realms/jhipster/protocol/openid-connect/auth?response_type=code&client_id=web_app&scope=openid%20profile%20email%20offline_access&state=28Wsn3cSlrrZQzE6DkXKUcAAS5r573O5WLOJYA7IgSo%3D&redirect_uri=https://localhost:9000/login/oauth2/code/oidc&nonce=WaqexbBDkQQCIhXfMFL84RzjzTtjOyjghQUtb4YwUKs)

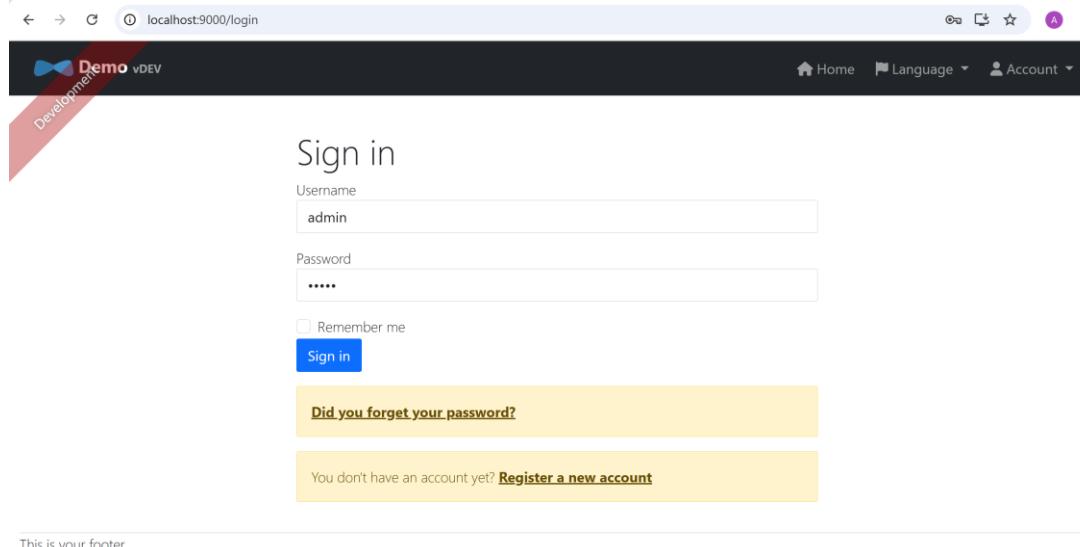
the location URL contains itself a "redirect_url" => back to origin web site after log-in

Comparison with "/login" page in JWT mode

"JWT" Mode



"OAuth2" Mode



localhost:9000/login

Sign in

Username
admin

Password
.....

Remember me

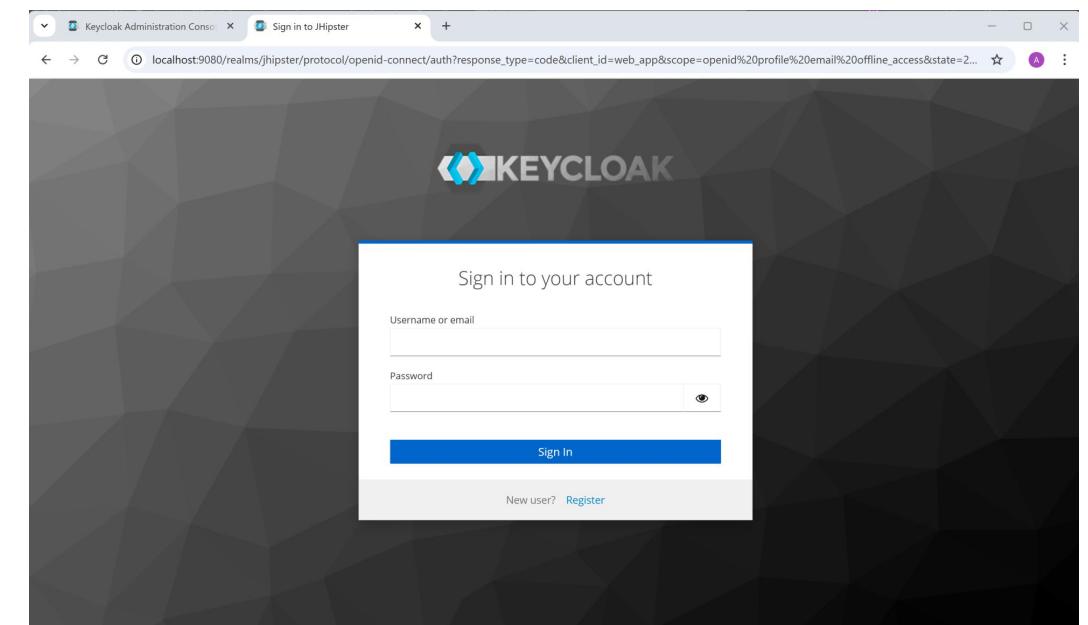
Sign in

[Did you forget your password?](#)

You don't have an account yet? [Register a new account](#)

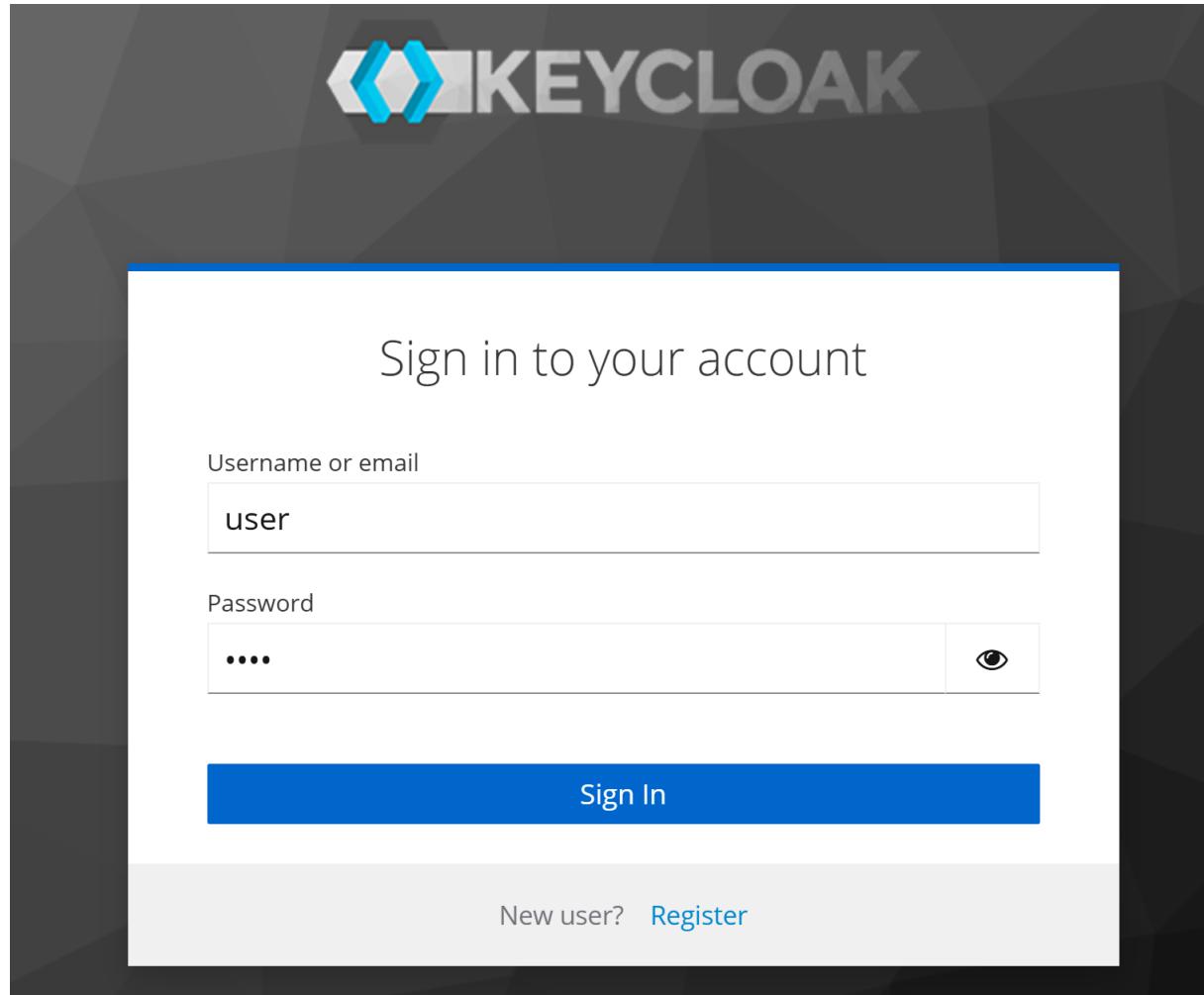
This is your footer

Login Page embedded in Webapp UI
(and user-passwords database in backend application)



Login Page externalized in OAuth2 webapp UI
(and user-passwords database in OAuth2 application)

enter user-password (from OAuth2 Realm)



Home Page after logged-in ... redirected back to WebApp

The screenshot shows a web browser window with the URL `localhost:9000` in the address bar. The page title is "Demo vDEV". The top navigation bar includes links for Home, Entities, Administration, Language, and Account. A red diagonal banner on the left says "Development". The main content area features a cartoon character with orange hair and blue overalls. The text "Welcome, Java Hipster! (Demo)" is prominently displayed, followed by "This is your homepage". A green box contains the message "You are logged in as user 'admin'.". Below this, there's a link to the JHipster homepage and other resources. At the bottom, there's a footer with the text "This is your footer".

localhost:9000

Development Demo vDEV

Home Entities Administration Language Account

Welcome, Java Hipster! (Demo)

This is your homepage

You are logged in as user "admin".

If you have any question on JHipster:

- [JHipster homepage](#)
- [JHipster on Stack Overflow](#)
- [JHipster bug tracker](#)
- [JHipster public chat room](#)
- [follow @jhipster on Twitter](#)

If you like JHipster, don't forget to give us a star on [GitHub](#)!

This is your footer

Http POST /realms/jhipster/login-actions-authenticate

Name	Value
authenticate?session_code=rPNx2WzIQYN0Uwkc...	
oidc?state=28Wsn3cSlrrZQzE6DkXKUcAAS5r573O...	
localhost	
loading.css	
styles.css	
runtime.js	
polyfills.js	
vendor.js	
main.js	
styles.js	
browser-sync-client.js?v=3.0.3	
logo-jhipster.png	
src_main_webapp_bootstrap_ts.js	
↳ ng-cli-ws	
socket.io/?EIO=4&transport=polling&t=PDGIGCQ	
socket.io/?EIO=4&transport=polling&t=PDGIGRY...	
socket.io/?EIO=4&transport=polling&t=PDGIGS2...	
↳ socket.io/?EIO=4&transport=websocket&sid=Nh...	
socket.io/?EIO=4&transport=polling&t=PDGIGXk...	
socket.io/?EIO=4&transport=polling&t=PDGIGYo...	
en.json?_=1f6947d3329a96320e6875712ddefcd3	

Headers

Request URL: http://localhost:9080/realms/jhipster/login-actions/authenticate?session_code=rPNx2WzIQYN0UwkcOC36Tp93kEhCex6zLW706FLnSn8&execution=4c9e9730-b91b-419e-b1e4-186e9633169f&client_id=web_app&tab_id=ZdHsSkZY488&client_data=eyJydSI6Imh0dHBzOi8vbG9jYWxob3N0OjkwMDAvbG9naW4vb2F1dGgyL2NvZGUvb2lkYylsInJ0ljojY29kZSlslnN0ljojMjhXc24zY1NscnJaUXpFNkRrWEtVY0FBUzVyNTczTzVXTE9KWUE3SWdTbz0ifQ

Request Method: POST

Status Code: 302 Found

Remote Address: [::1]:9080

Referrer Policy: no-referrer

Response Headers

Cache-Control: no-store, must-revalidate, max-age=0

Content-Length: 0

Content-Security-Policy: frame-src 'self'; frame-ancestors 'self'; object-src 'none';

Location: https://localhost:9000/login/oauth2/code/oidc?state=28Wsn3cSlrrZQzE6DkXKUcAAS5r573O5WLOJYA7lgSo%3D&session_state=ee8ed899-0cce-4231-8f69-2d7777271789&iss=http%3A%2F%2Flocalhost%3A9080%2Frealms%2Fjhipster&code=5e507f09-0239-43d0-ae88-a0215c7a7bda.ee8ed899-0cce-4231-8f69-2d7777271789.6e8deddb-b4d6-4e2e-b389-b397d3f74fcfd

Referrer-Policy: no-referrer

Set-Cookie: JSESSIONID=ADT-1Version-1; Path=/realms/jhipster; Max-Age=0

28 requests | 6.7 MB transferred | 6.9 MB resources

entering back to WebApp ... OAuth2 token

Location [https://localhost:9000/login/oauth2/code/oidc
?state=28Wsn3cSlrrZQzE6DkXKUcAAS5r573O5WLOJYA7IgSo%3D
&session_state=ee8ed899-0cce-4231-8f69-2d7777271789
&iss=http%3A%2F%2Flocalhost%3A9080%2Frealms%2Fhipster
&code=5e507f09-0239-43d0-ae88-a0215c7a7bda.ee8ed899-0cce-4231-8f69-2d7777271789.6e8deddb-b4d6-4e2e-b389-b397d3f74fc](https://localhost:9000/login/oauth2/code/oidc?state=28Wsn3cSlrrZQzE6DkXKUcAAS5r573O5WLOJYA7IgSo%3D&session_state=ee8ed899-0cce-4231-8f69-2d7777271789&iss=http%3A%2F%2Flocalhost%3A9080%2Frealms%2Fhipster&code=5e507f09-0239-43d0-ae88-a0215c7a7bda.ee8ed899-0cce-4231-8f69-2d7777271789.6e8deddb-b4d6-4e2e-b389-b397d3f74fc)

parameter code ... is a OAuth2 Token (not exactly a JWT format!) : 5e50----bda . ee8e---1789 . 6e8de---74fc

it does not contains a base64 characters, but also "-", like UUID

Another Request GET /login/auth2/code/oidc

The screenshot shows a browser developer tools Network tab with the Headers tab selected. A specific request is highlighted with a red box.

Name: oidc?state=28Wsn3cSlrrZQzE6DkXKUcAAS5r573O...

Headers Tab:

- Request URL:** https://localhost:9000/login/oauth2/code/oidc?state=28Wsn3cSlrrZQzE6DkXKUcAAS5r573O5WLOJYA7lgSo%3D&session_state=ee8ed899-0cce-4231-8f69-2d7777271789&iss=http%3A%2F%2Flocalhost%3A9080%2Frealm%2Fjhipster&code=5e507f09-0239-43d0-ae88-a0215c7a7bda.ee8ed899-0cce-4231-8f69-2d7777271789.6e8deddb-b4d6-4e2e-b389-b397d3f74fc
- Request Method:** GET
- Status Code:** 302 Found
- Remote Address:** [::1]:9000
- Referrer Policy:** no-referrer

Response Headers:

- Access-Control-Allow-Origin: *
- Cache-Control: no-cache, no-store, max-age=0, must-revalidate
- Connection: close
- Content-Length: 0
- Content-Security-Policy: default-src 'self'; frame-src 'self' data;; script-src 'self' 'unsafe-inline' 'unsafe-eval' https://storage.googleapis.com; style-src 'self' 'unsafe-inline'; img-src 'self' data;; font-src 'self' data;
- Date: Thu, 21 Nov 2024 21:59:41 GMT
- Expires: 0
- Location:** https://localhost:9000/
- Permissions-Policy: camera=(), fullscreen=(self), geolocation=(), gyroscope=(), magnetometer=(), microphone=(), midi=(), payment=(), sync=()

Summary: 28 requests | 6.7 MB transferred | 6.9 MB resources

OAuth2 token previously in Http Header "Location"
=> now passed as URL param

Request URL	<code>https://localhost:9000/login/oauth2/code/oidc ?state=28Wsn3cSlrrZQzE6DkXKUcAAS5r573O5WLOJYA7IgSo%3D &session_state=ee8ed899-0cce-4231-8f69-2d7777271789 &iss=http%3A%2F%2Flocalhost%3A9080%2Frealm%2Fhipster &code=5e507f09-0239-43d0-ae88-a0215c7a7bda.ee8ed899- 0cce-4231-8f69-2d7777271789.6e8deddb-b4d6-4e2e-b389- b397d3f74fcd</code>
--------------------	--

redirected to "/", with COOKIE JSESSIONID

The screenshot shows the Network tab of a browser developer tools interface. A single request to "localhost" is selected and highlighted with a red box. The Headers tab is active, displaying the following details:

- Request URL: https://localhost:9000/ (highlighted with a red box)
- Request Method: GET
- Status Code: 200 OK (highlighted with a red box)
- Remote Address: [:1]:9000
- Referrer Policy: no-referrer

Below the General section, there are sections for Response Headers (7) and Request Headers.

The Request Headers section shows:

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
- Accept-Encoding: gzip, deflate, br, zstd
- Accept-Language: en,fr-FR;q=0.9,fr;q=0.8,en-FR;q=0.7,en-US;q=0.6
- Cache-Control: max-age=0
- Connection: keep-alive

The Response Headers section shows:

- Cookie: shellInABox=3:101010; username-localhost-8888="2|1:0|10:1729931560|23:username-localhost-8888|192:eyJ1c2VybmFtZSI6ICJiODY1Y2JmMjdjMDA0MmU1OTIzMDRjM2FkMDlhZGJ2MilsICJuYW1lljogIkFub255bW91cyBFbGFyYSIsICJkaXNwbGFx25hbWUiOiAiQW5vbntb3VzIEVsYXJhliwgImluaXRpYWxzIjogIkFFliwgImNvbG9yljogbnVsbH0=|f903b8b6109cb4119762524f861461c7e4a21a4342679e640fd598be87edb95d";_xsrf=2|dc7823e1|02bf7e8cc9e6846a2ae45bbce25a0d92|1729931560; JSESSIONID=tyFkbxTleOlwm7CVxZaWe78DNJIMJvazqcg7coY_ (highlighted with a red box)

At the bottom of the Network tab, the summary is shown:

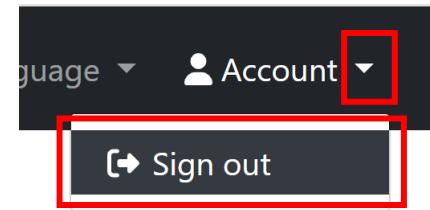
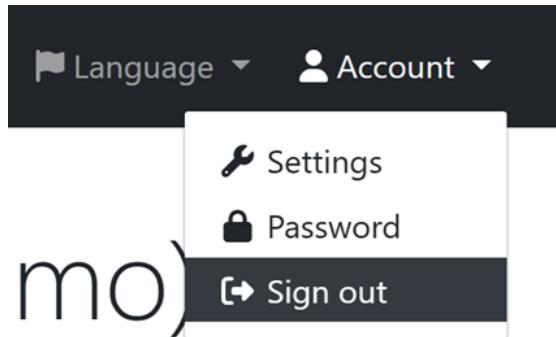
- 28 requests
- 6.7 MB transferred
- 6.9 MB resources

Different Menu for Account Sub Menu Item "Sessions"

"JWT" Mode



"OAuth2" Mode



user and password is NO MORE stored on webapp
=> no more menu to save

cf OAuth2 app instead

Click Sign-Out => 2 Requests

Name	Status	Type	Initiator
logout	302	xhr / Redirect	login.service.ts:19
localhost	200	xhr	logout

Http POST /logout

Name	Headers	Payload	Preview	Response	Initiator	Timing	Cookies
logout							
localhost							
	▼General						
	Request URL:	https://localhost:9000/api/logout					
	Request Method:	POST					
	Status Code:	302 Found					
	Remote Address:	[:1]:9000					
	Referrer Policy:	strict-origin-when-cross-origin					
	► Response Headers (18)						
	▼Request Headers	<input type="checkbox"/>	Raw				
	Accept:	application/json, text/plain, */*					
	Accept-Encoding:	gzip, deflate, br, zstd					
	Accept-Language:	en,fr-FR;q=0.9,fr;q=0.8,en-FR;q=0.7,en-US;q=0.6					
	Connection:	keep-alive					
	Content-Length:	2					
	Content-Type:	application/json					
	Cookie:	shellInABox=3:101010; username=localhost-8888="2 1:0 10:1729931560 23:username=localhost-8888 192:eyJ1c2VybmcFtZSI6ICJiODY1Y2JmMjdjMDA0MmU1OTIzMzMDRjM2FkMDlhZGI2MilsICJuYW1lIjogIkFub255bW91cyBFbGFyYSIsICJkaXNwbGF5X25hbWUiOiAiQW5vbntb3VzIEVsYXJhIwglmluaXRpYWxzIjogIkFFIiwgImNvbG9yljogbnVsbH0= f903b8b6109cb4119762524f861461c7e4a21a4342679e640fd598be87edb95d";_xsrf=2 dc7823e1 02bf7e8cc9e6846a2ae45bbce25a0d92 1729931560;JSESSIONID=tyFkbxTleOlwm7CVxZaWe78DNJIMJvazqcq7coY_; XSRF-TOKEN=51258189-b937-					
2 requests	6.4 kB transferred	5.2 kB resources					

Redirected "/"

Name	Headers	Preview	Response	Initiator	Timing	Cookies
logout						
localhost	<p>▼ General</p> <p>Request URL: https://localhost:9000/</p> <p>Request Method: GET</p> <p>Status Code: 200 OK</p> <p>Remote Address: [:1]:9000</p> <p>Referrer Policy: strict-origin-when-cross-origin</p> <p>► Response Headers (7)</p> <p>▼ Request Headers</p> <p>Accept: application/json, text/plain, */*</p> <p>Accept-Encoding: gzip, deflate, br, zstd</p> <p>Accept-Language: en,fr-FR;q=0.9,fr;q=0.8,en-FR;q=0.7,en-US;q=0.6</p> <p>Connection: keep-alive</p> <p>Cookie: shellInABox=3:101010; username=localhost-8888="2 1:0 10:1729931560 23:username=localhost-8888 192:eyJ1c2VybmFtZSI6ICJiODY1Y2JmMjdjMDA0MmU1OTIzMDFkMDlhZGI2MilsICJuYW1lIjogIkFub255bW91cyBFbGFyYSIsICJkaXNwbGF5X25hbWUiOiAiQW5vbnlb3VzIEVsYXJhIiwglmluaXRpYWxzljogIkFFIiwgImNvbG9yljogbnVsbH0= f903b8b6109cb4119762524f861461c7e4a21a4342679e640fd598be87edb95d";_xsrf=2 dc7823e1 02bf7e8cc9e6846a2ae45bbce25a0d92 1729931560;JSESSIONID=tyFkbxTleOlwm7CVxZaWe78DNJIMJvazqcg7coY_;XSRF-TOKEN=51258189-b937-4f08-8d5e-f203a8fb4680</p> <p>Host: localhost:9000</p>					

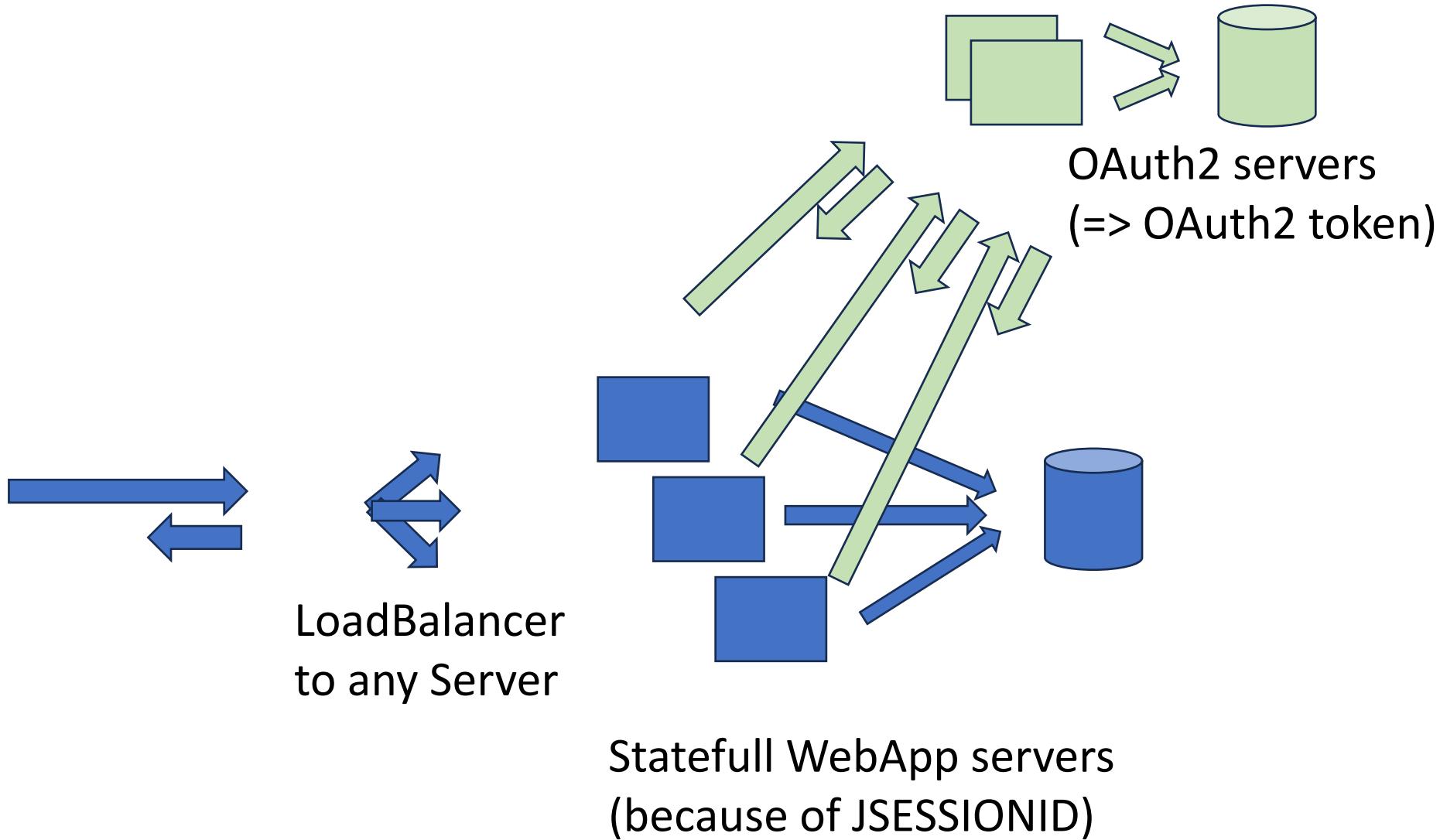
Authorisations

For Authorization (ROLE_*)
there is NO difference with the others mode JWT or Http SEssion.

Please see previous detailed document on JWT

Network & Infrastructure Deployments Options

Servers + OAuth2 Servers (Statefull for JSESSIONID- Need Database)



Take Away

- [1/4]: Debugging "JWT" security mode
- [2/4]: Debugging old-style COOKIE "jsessionId" security mode
- [3/4]: Debugging "Basic Auth"
- [4/4] Debugging the OAuth2 security mode

Security is difficult & sensitive - Leave it to experts.

Read code to learn in JHipster

Questions

arnaud.nauwynck@gmail.com