

Reverse Engineering Jhipster SpringBoot Security

arnaud.nauwynck@gmail.com

this presentation:

Links..

<https://garnier.wf/spring-security-workshop-slides/>

Devoxx France 2022

Spring Security - Décodage et démystification



Devoxx FR - 2022/04/20

Daniel Garnier-Moiroux



Step 1: create Jhipster app

```
$ jhipster
INFO! Using bundled JHipster

JHIPSTER

https://www.jhipster.tech

Welcome to JHipster v7.8.1

Application files will be generated in folder: C:\arn\test-jhipster

Documentation for creating an application is at https://www.jhipster.tech/creating-an-app/
If you find JHipster useful, consider sponsoring the project at https://opencollective.com/generator-jhipster

WARNING! Your Node version is not LTS (Long Term Support), use it at your own risk! JHipster does not support non-LTS releases,
.
? Which *type* of application would you like to create? (Use arrow keys)
> Monolithic application (recommended for simple projects)
  Gateway application
  Microservice application
```

Choose App Type > Security Type

```
? Which *type* of application would you like to create? Monolithic application (recommended for simple projects)
? What is the base name of your application? myapp
? Do you want to make it reactive with Spring WebFlux? No
? What is your default Java package name? com.mycompany.myapp
? Which *type* of authentication would you like to use? (Use arrow keys)
> 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)
```

End Choices (Db, Angular, Maven, ..)

```
? Which *type* of database would you like to use? SQL (H2, PostgreSQL, MySQL, MariaDB, Oracle, MSSQL)
? Which *production* database would you like to use? PostgreSQL
? Which *development* database would you like to use? H2 with disk-based persistence
? Which cache do you want to use? (Spring cache abstraction) No cache - Warning, when using an SQL database, this will disable the Hibernate 2nd level cache!
? Would you like to use Maven or Gradle for building the backend? Maven
? Do you want to use the JHipster Registry to configure, monitor and scale your application? No
? Which other technologies would you like to use? WebSockets using Spring Websocket
? Which *Framework* would you like to use for the client? Angular
? Do you want to generate the admin UI? Yes
? Would you like to use a Bootswatch theme (https://bootswatch.com/)? Default JHipster
? Would you like to enable internationalization support? No
? Please choose the native language of the application English
? Besides JUnit and Jest, which testing frameworks would you like to use?
? Would you like to install other generators from the JHipster Marketplace? No
```

Start Generating

```
G n ration d'une paire de cl s RSA de 2 048 bits et d'un certificat auto-sign  (SHA256withRSA) d'une validit  de 99 999 jours  
pour : CN=Java Hipster, OU=Development, O=com.mycompany.myapp, L=, ST=, C=  
  
KeyStore 'C:\arn\test-jhipster\src\main\resources\config\tls\keystore.p12' generated successfully.
```

Generating a RSA Key pair, for exposing https instead of http

(but self-signed certificate)

Finished Generating

```
Application successfully committed to Git from C:\arn\test-jhipster.  
  
If you find JHipster useful consider sponsoring the project https://www.jhipster.tech/sponsors/  
  
Server application generated successfully.  
  
Run your Spring Boot application:  
./mvnw (mvnw if using Windows Command Prompt)  
  
Client application generated successfully.  
  
Start your Webpack development server with:  
npm start  
  
> myapp@0.0.1-SNAPSHOT clean-www  
> rimraf target/classes/static/app/{src,target/}  
  
Congratulations, JHipster execution is complete!  
Sponsored with ❤ by @oktadev.  
  
arnaud@DESKTOP-2EGCC8R /cygdrive/c/arn/test-jhipster  
$
```

Created Files

```
create .prettierrc
create .prettierignore
create package.json
  force .yo-rc.json
  force .yo-rc-global.json
create .gitattributes
create .editorconfig
create .gitignore
create sonar-project.properties
create .husky\pre-commit
create .lintstagedrc.js
create mvnw
create mvnw.cmd
create .mvn\jvm.config
create .mvn\wrapper\maven-wrapper.jar
create .mvn\wrapper\maven-wrapper.properties
create npmw
create npmw.cmd
create src\main\resources\banner.txt
```

(truncated ...) total files :

```
$ find . | wc -l
105964
```


Lot of npm files (in node_modules/***) ... but ok .gitignore

```
$ git status
On branch master
nothing to commit, working tree clean
```

```
$ git log
commit 86ba746ae24834926b6b79e85cecef20481d44ab (HEAD -> master)
Author: Arnaud Nauwynck <arnaud.nauwynck@gmail.com>
Date: Thu May 19 14:13:09 2022 +0200

    Initial version of myapp generated by generator-jhipster@7.8.1
```

```
$ cat .gitignore
#####
# Project Specific
#####
/src/main/webapp/content/css/main.css
/target/classes/static/**
/src/test/javascript/coverage/

#####
# Node
#####
/node/
node_tmp/
node_modules/
npm-debug.log.*
/.awcache/*
/.cache-loader/*
```

```
$ ls -1
README.md
angular.json
checkstyle.xml
jest.conf.js
mvnw
mvnw.cmd
ngsw-config.json
node_modules
npmw
npmw.cmd
package-lock.json
package.json
pom.xml
sonar-project.properties
src
target
tsconfig.app.json
tsconfig.json
tsconfig.spec.json
webpack
```

```
$ find . -maxdepth 1 -type d
.
./.devcontainer
./.git
./.husky
./.mvn
./node_modules
./src
./target
./webpack
```

Src Files

```
$ find src/main | wc -l  
464
```

```
$ find src/main/java | wc -l  
92
```

```
$ find src/main/webapp | wc -l  
322
```

```
$ find src/main/resources | wc -l  
29
```

```
$ find src/main/docker | wc -l  
20
```

Created Files - Categories

```
$ find . -maxdepth 1 -type d
.
./.devcontainer
./.git
./.husky
./.mvn
./node_modules
./src
./target
./webpack
```

src/
(in git)

```
$ find ./src -maxdepth 2
./src
./src/main
./src/main/docker
./src/main/java
./src/main/resources
./src/main/webapp
./src/test
./src/test/java
./src/test/resources
```

src/main
(in artifact delivery)

src/test
(during test only)

src/main/webapp

Technical layers organization:
account, admin, config, core,
entities, home, layouts, login, shared

```
$ find ./src/main/webapp/app -maxdepth 1
./src/main/webapp/app
./src/main/webapp/app/account
./src/main/webapp/app/admin
./src/main/webapp/app/app-routing.module.ts
./src/main/webapp/app/app.constants.ts
./src/main/webapp/app/app.module.ts
./src/main/webapp/app/config
./src/main/webapp/app/core
./src/main/webapp/app/entities
./src/main/webapp/app/home
./src/main/webapp/app/layouts
./src/main/webapp/app/login
./src/main/webapp/app/shared
```

src/main/java

Technical layers organization:
aop, config, domain, managment,
security, service, web

```
$ find ./src/main/java -maxdepth 4
./src/main/java
./src/main/java/com
./src/main/java/com/mycompany
./src/main/java/com/mycompany/myapp
./src/main/java/com/mycompany/myapp/aop
./src/main/java/com/mycompany/myapp/ApplicationWebXml.java
./src/main/java/com/mycompany/myapp/config
./src/main/java/com/mycompany/myapp/domain
./src/main/java/com/mycompany/myapp/GeneratedByJHipster.java
./src/main/java/com/mycompany/myapp/management
./src/main/java/com/mycompany/myapp/MyappApp.java
./src/main/java/com/mycompany/myapp/repository
./src/main/java/com/mycompany/myapp/security
./src/main/java/com/mycompany/myapp/service
./src/main/java/com/mycompany/myapp/web
```

Created Files, src/main/java

Created Files, src/main/java

