

Documentación detallada de la ejecución y explicación del proyecto desde SpringBoot y Java 8

1. Instalación de NPM mediante la herramienta NodeJs

<https://nodejs.org/en/>


2. Abrimos la consola de comandos y llamamos al instalador de paquetes NPM para que nos instale MAVEN
npm install maven

3. Instalamos OpenApi generator cli para ejecutar comandos de openapi

4. `npm install @openapitools/openapi-generator-cli -g`

5. Abrimos powershell si estamos en windows 10 como administrador y realizmos el siguiente comando:

set-executionpolicy unrestricted -force

 Administrador: Windows PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Windows\system32> set-executionpolicy unrestricted -force
PS C:\Windows\system32>
```

6. Nos descargamos el siguiente fichero de ejemplo para darle la información necesaria a OpenApi generator sobre nuestra API. (En nuestro caso hemos usado un fichero petstore.yaml que se encuentra en la documentación oficial de la app.



petstore.yaml

https://raw.githubusercontent.com/openapitools/openapi-generator/master/modules/openapi-generator/src/test/resources/3_0/petstore.yaml

7. Creamos una carpeta donde se vaya a alojar el proyecto con el archivo petstore.yaml contenido en ella.
8. Ejecutamos el siguiente comando dentro del directorio creado:
(unicamente modificamos el nombre del fichero .yaml a petstore.yaml o el que hayamos elegido)

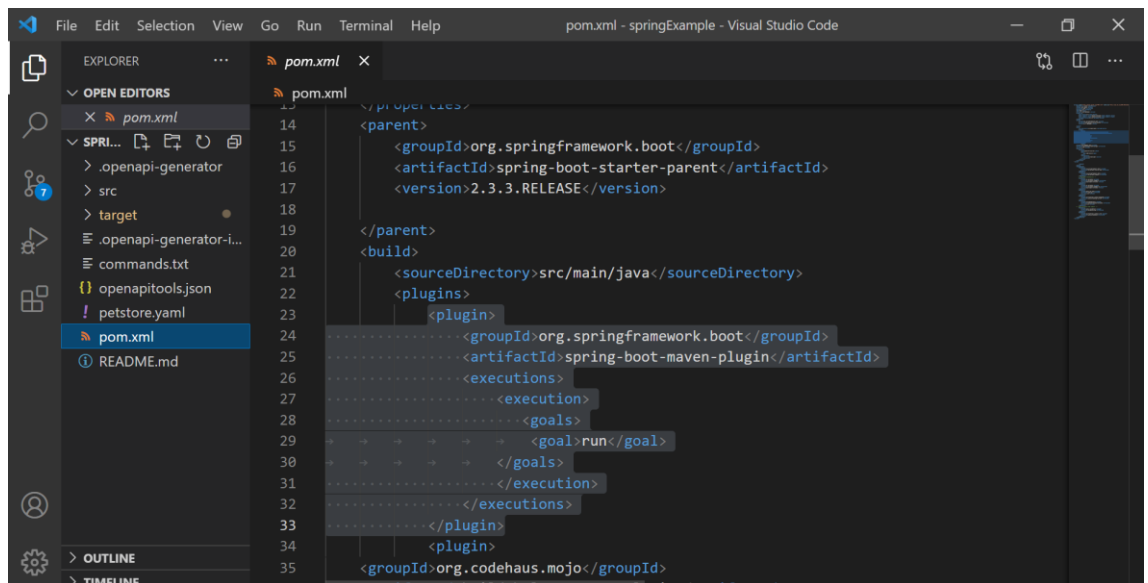
```
openapi-generator-cli generate -g spring --library
spring-boot -i petstore.yaml -o ${PWD} -p
groupId=com.redhat -p artifactId=todo -p
artifactVersion=1.0.0-SNAPSHOT -p
basePackage=com.redhat.todo -p
configPackage=com.redhat.todo.config -p
apiPackage=com.redhat.todo.api -p
modelPackage=com.redhat.todo.model -p
sourceFolder=src/main/gen -p dateLibrary=java8 -p
java8=true
```

9. Añadimos el siguiente plugin a nuestro pom.xml

```
10. <plugin>
11.   <groupId>org.codehaus.mojo</groupId>
12.   <artifactId>build-helper-maven-plugin</artifactId>
13.   <version>3.1.0</version>
14.   <executions>
15.     <execution>
16.       <phase>generate-sources</phase>
17.       <goals>
18.         <goal>add-source</goal>
19.       </goals>
20.       <configuration>
21.         <sources>src/main/gen</sources>
22.       </configuration>
23.     </execution>
24.   </executions>
25. </plugin>
```

7. Añadiremos a plugin spring-boot-maven-plugin el siguiente apartado para que nos deje ejecutar la aplicación maven

```
<plugin>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-maven-plugin</artifactId>
  <executions>
    <execution>
      <goals>
        <goal>run</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```



8. Abrimos la terminal y ejecutamos el siguiente comando dentro del directorio de nuestro proyecto para limpiar el proyecto e instalar todas sus dependencias modificadas:

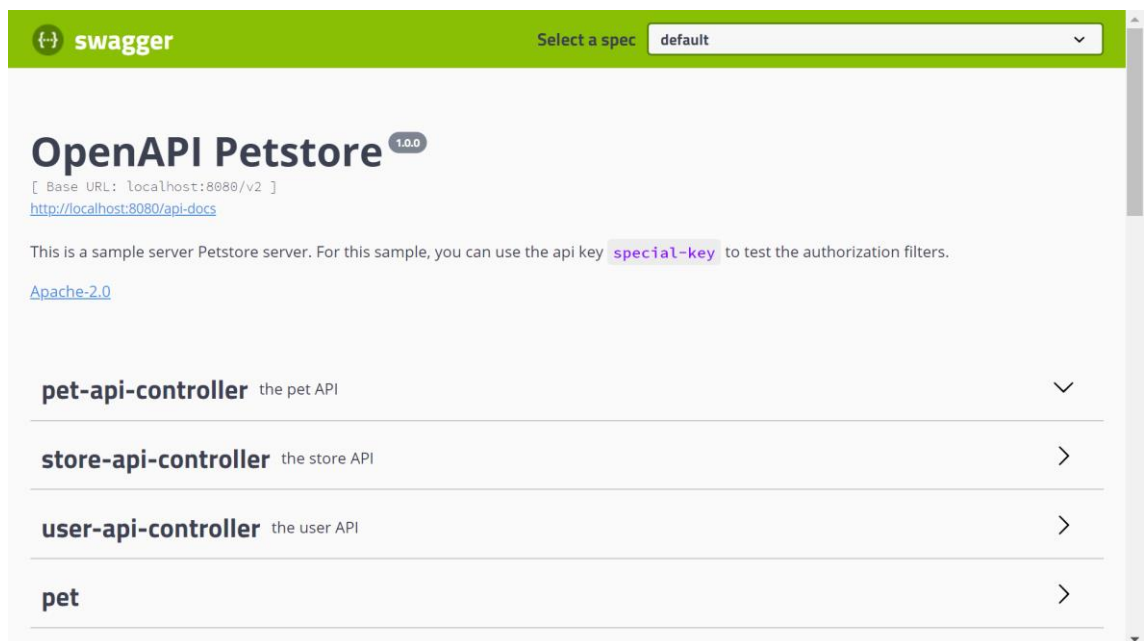
```
mvn clean install
```

9. No cierre la terminal ahora toca ejecutar el siguiente comando para que se quede la aplicación que hemos creado ejecutandose y poder acceder a ella desde cualquier navegador:

```
mvn spring:boot run
```

10. Abrimos el navegador (preferentemente firefox por el poco consumo de recursos y escribimos la siguiente URL):

<http://localhost:8080/swagger-ui.html#/>



11. Aquí nos encontraremos la documentación necesaria para hacer peticiones CRUD a nuestras apis

pet-api-controller

the pet API

>

store-api-controller

the store API

>

user-api-controller

the user API

>

pet

>

POST

/pet

Add a new pet to the store

🔒

PUT

/pet

Update an existing pet

🔒

GET

/pet/{petId}

Find pet by ID

🔒

POST

/pet/{petId}

Updates a pet in the store with form data

🔒

Inlusive se pueden ampliar y probar

pet

>

POST

/pet

Add a new pet to the store

🔒

Parameters

Try it out

Name

Description

pet required

Pet object that needs to be added to the store

(body)

Example Value | Model

<?xml version="1.0" encoding="UTF-8"?>
<Pet>
 <category>
 <id>0</id>
 <name>string</name>
 </category>
 <id>0</id>
 <name>doggie</name>
 <photoUrls>string</photoUrls>
 <status>available</status>
 <tags>
 <id>0</id>
 <name>string</name>
 </tags>
</Pet>

12. Modelos especificados:

Models

>

Category

>

description:
id
name
}

A category for a pet
integer(\$int64)
string

File

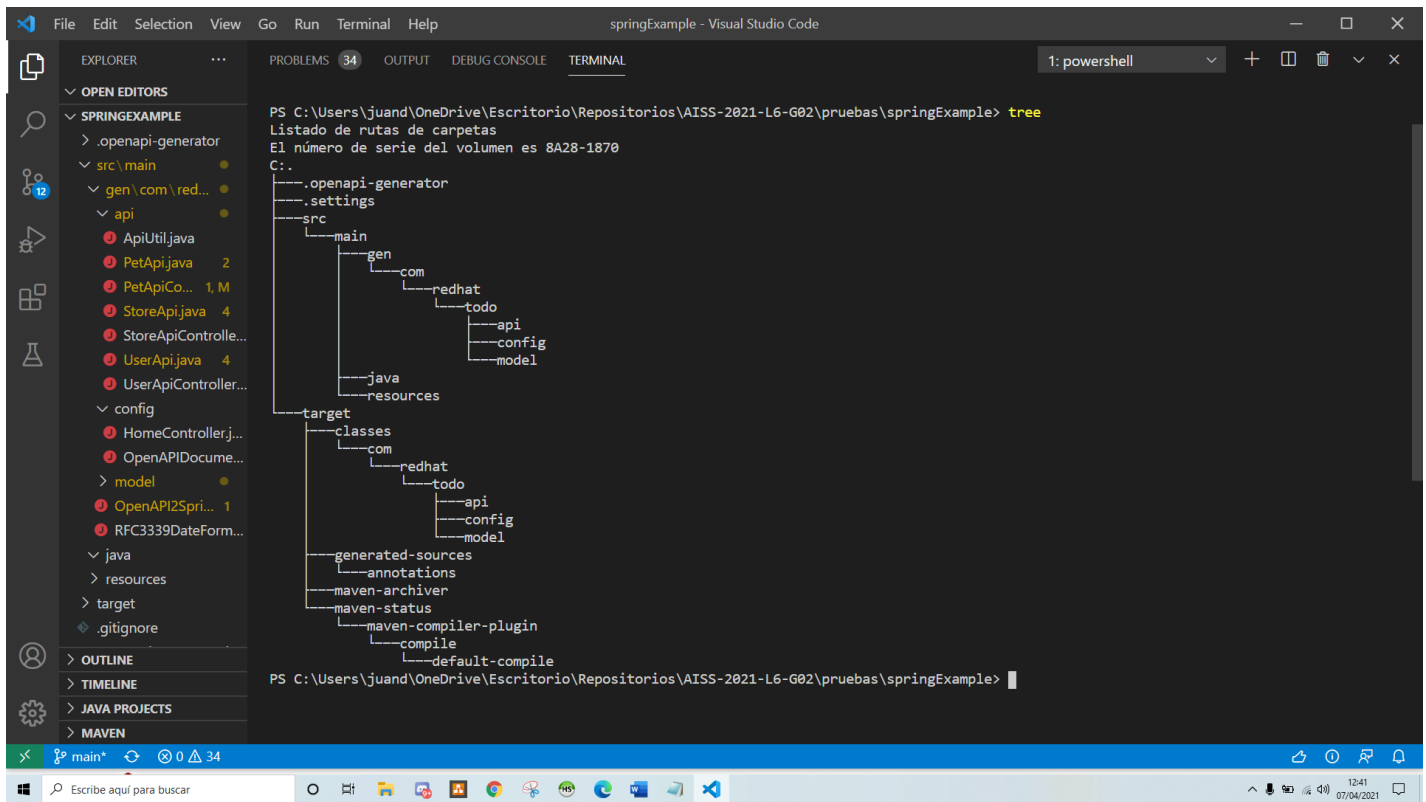
>

absolute
absoluteFile
absolutePath
canonicalFile
canonicalPath
directory
executable
file
freeSpace
hidden
lastModified
name
}

boolean
> {...}
string
> {...}
string
boolean
boolean
boolean
integer(\$int64)
boolean
integer(\$int64)
string

IMPORTANTE: este proyecto usa [api key](#)

13. Estructura autogenerada de nuestro proyecto



The screenshot shows the Visual Studio Code interface with the 'springExample' project open. The Explorer panel on the left displays the project structure, including the 'src/main/gen/com/redhat/todo' directory. The Terminal panel on the right shows the output of the 'tree' command, which lists the directory structure of the project. The output is as follows:

```
PS C:\Users\juand\OneDrive\Escritorio\Repositorios\AISS-2021-L6-G02\pruebas\springExample> tree
Listado de rutas de carpetas
El número de serie del volumen es 8A28-1870
C:.
|-- .openapi-generator
|-- .settings
|-- src
|   |-- main
|   |   |-- gen
|   |   |   |-- com
|   |   |   |   |-- redhat
|   |   |   |   |   |-- todo
|   |   |   |   |   |   |-- api
|   |   |   |   |   |   |-- config
|   |   |   |   |   |   |-- model
|   |   |-- java
|   |   |-- resources
|   |-- target
|       |-- classes
|       |   |-- com
|       |   |   |-- redhat
|       |   |   |   |-- todo
|       |   |   |   |   |-- api
|       |   |   |   |   |-- config
|       |   |   |   |   |-- model
|       |-- generated-sources
|       |-- annotations
|       |-- maven-archiver
|       |-- maven-status
|       |-- maven-compiler-plugin
|       |-- compile
|       |-- default-compile
|-- .gitignore
-- .gitignore

PS C:\Users\juand\OneDrive\Escritorio\Repositorios\AISS-2021-L6-G02\pruebas\springExample>
```

14. RECOMENDACIÓN TESTING:

Recomendamos encarecidamente el uso de postman para probar las apis.

<https://www.postman.com/>