Creating Your First Spring Boot App

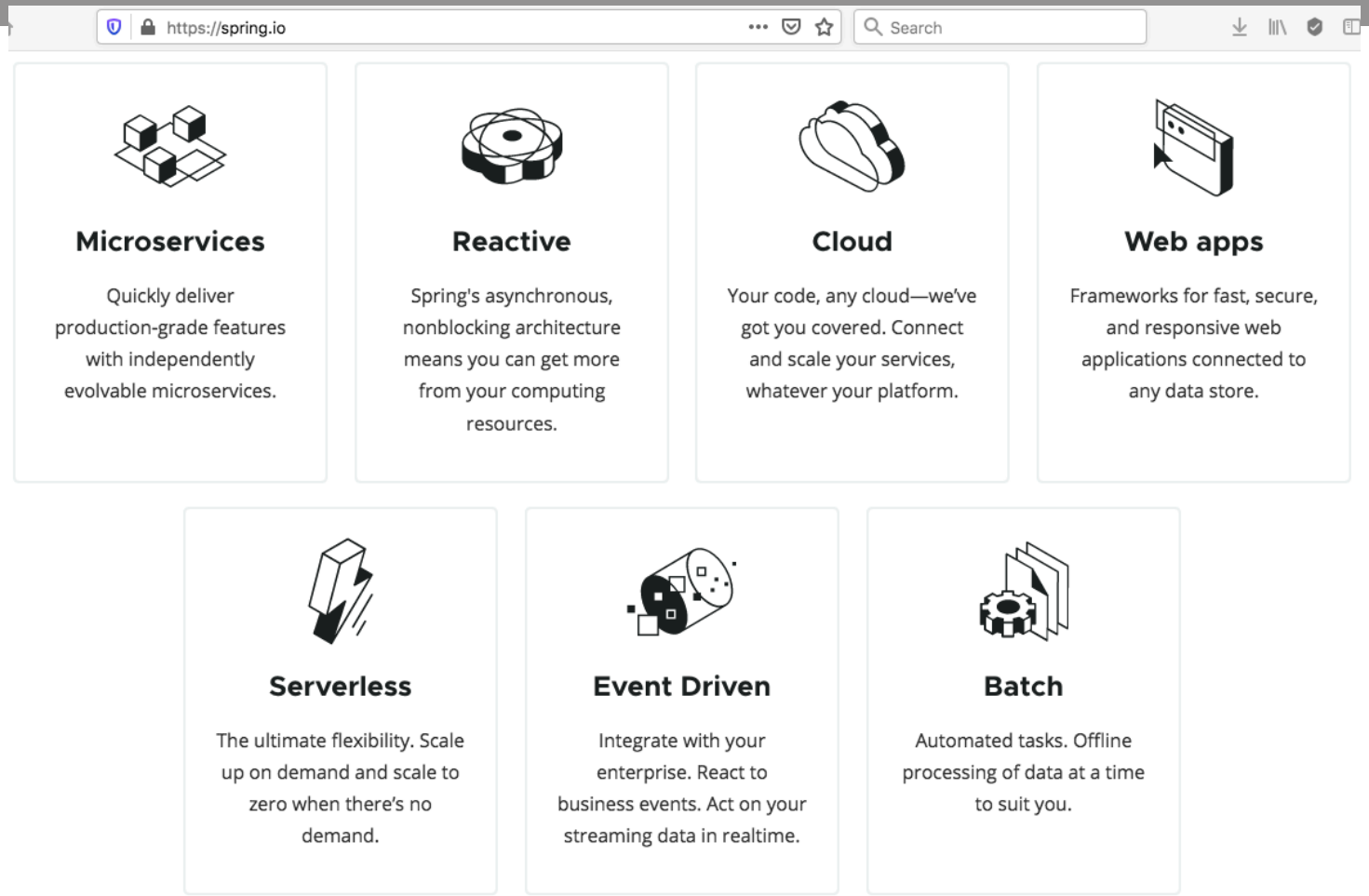
Introducing Spring Boot 2

TOCs

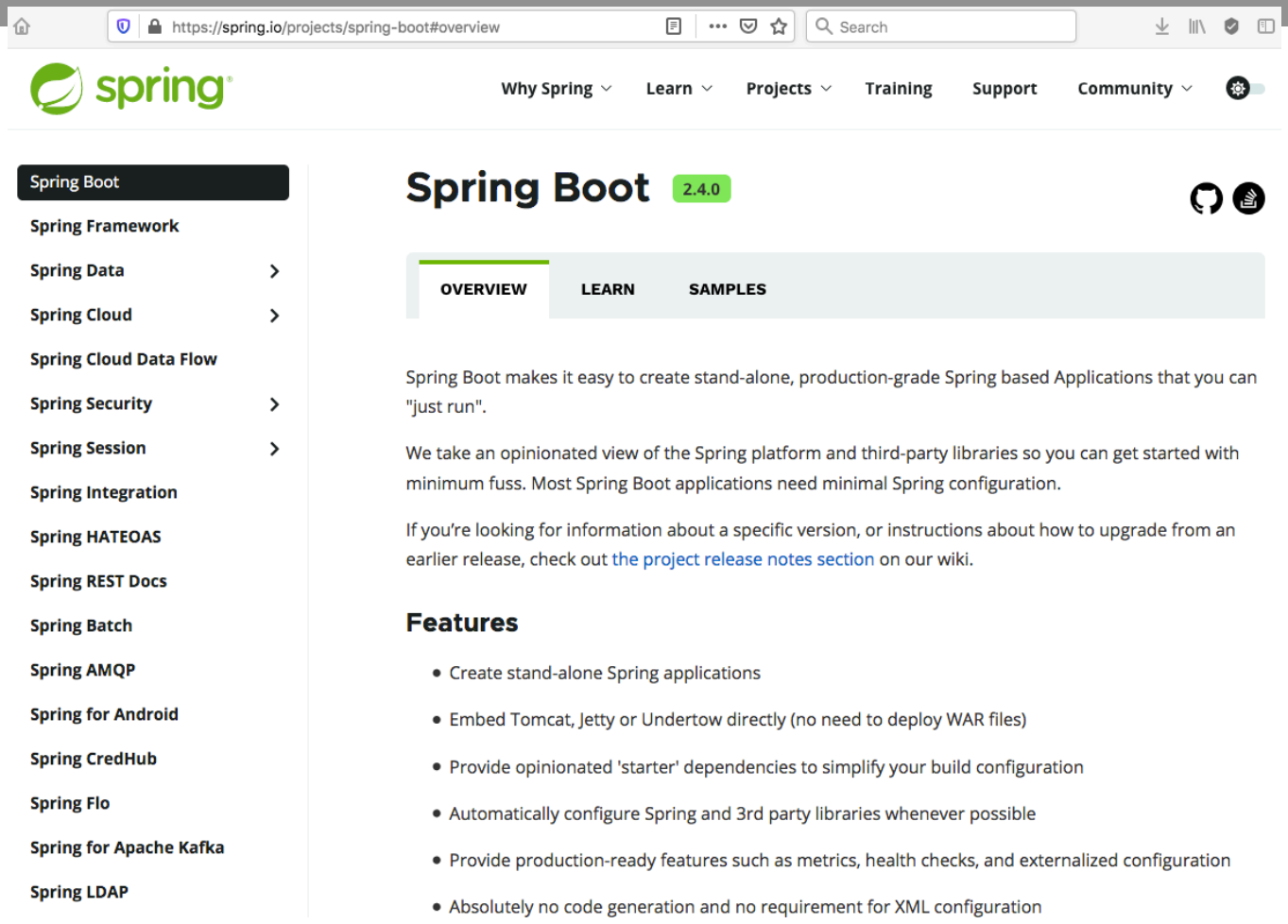
HTL Villach
Future Inside

- what Spring can do – since years, by projects
- web app Spring Initialzr – dependencies like Web incl. Tomcat
- IntelliJ CE vs Ultimate
- best practices for a simple project structure
- Spring Boot Starters supporting dependence injections
- starting Spring Boot Applications
- maven, eclipse, ..

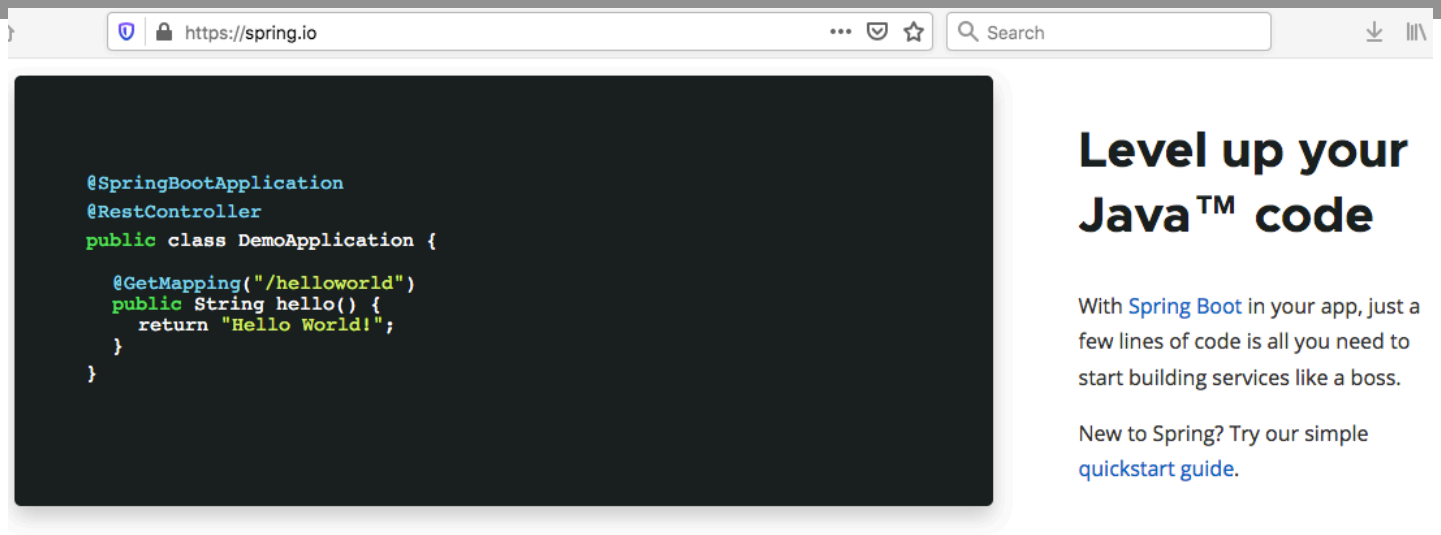
What Spring can do: Ecosystem beside Oracle



Spring Projects



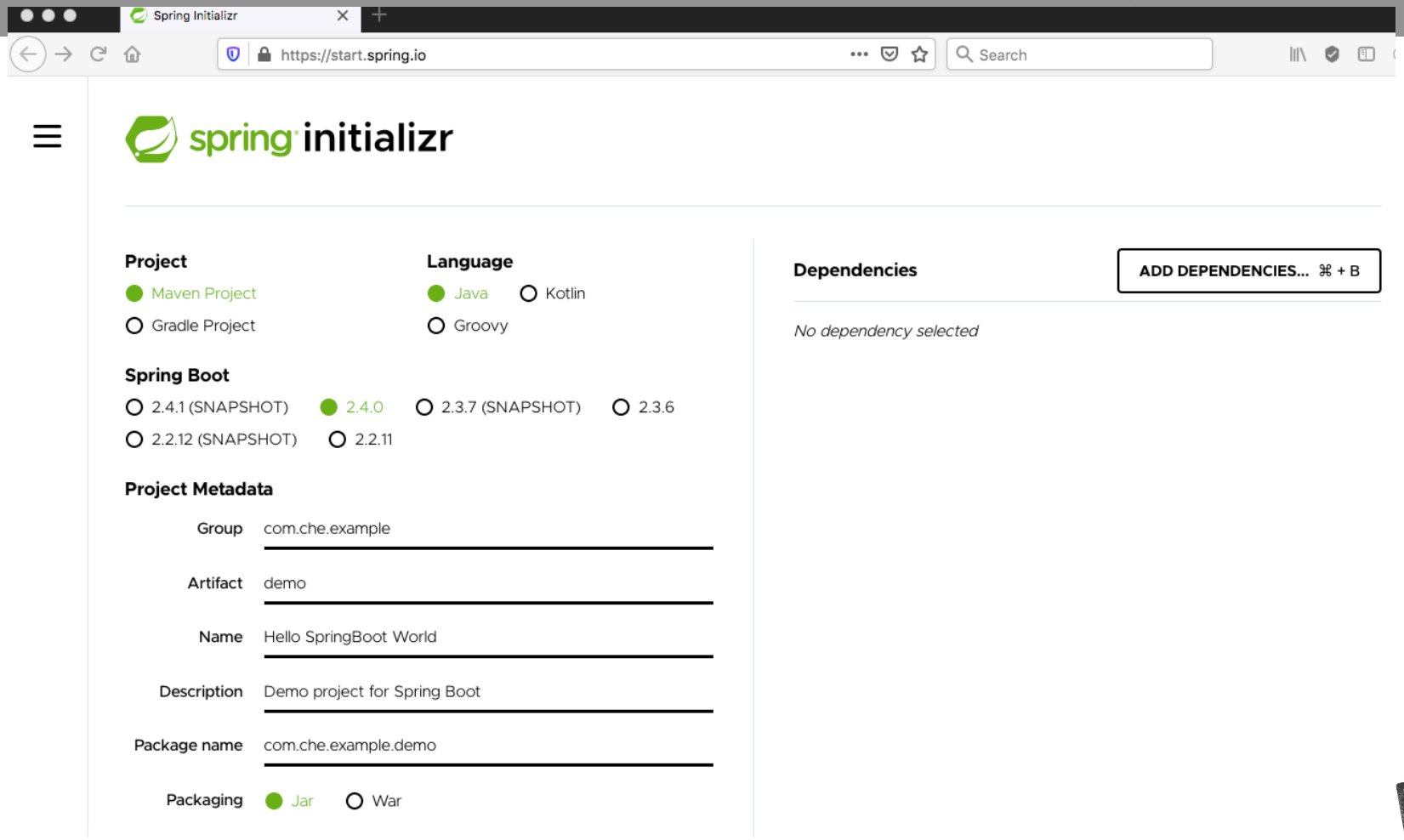
It’s not really new, but ..



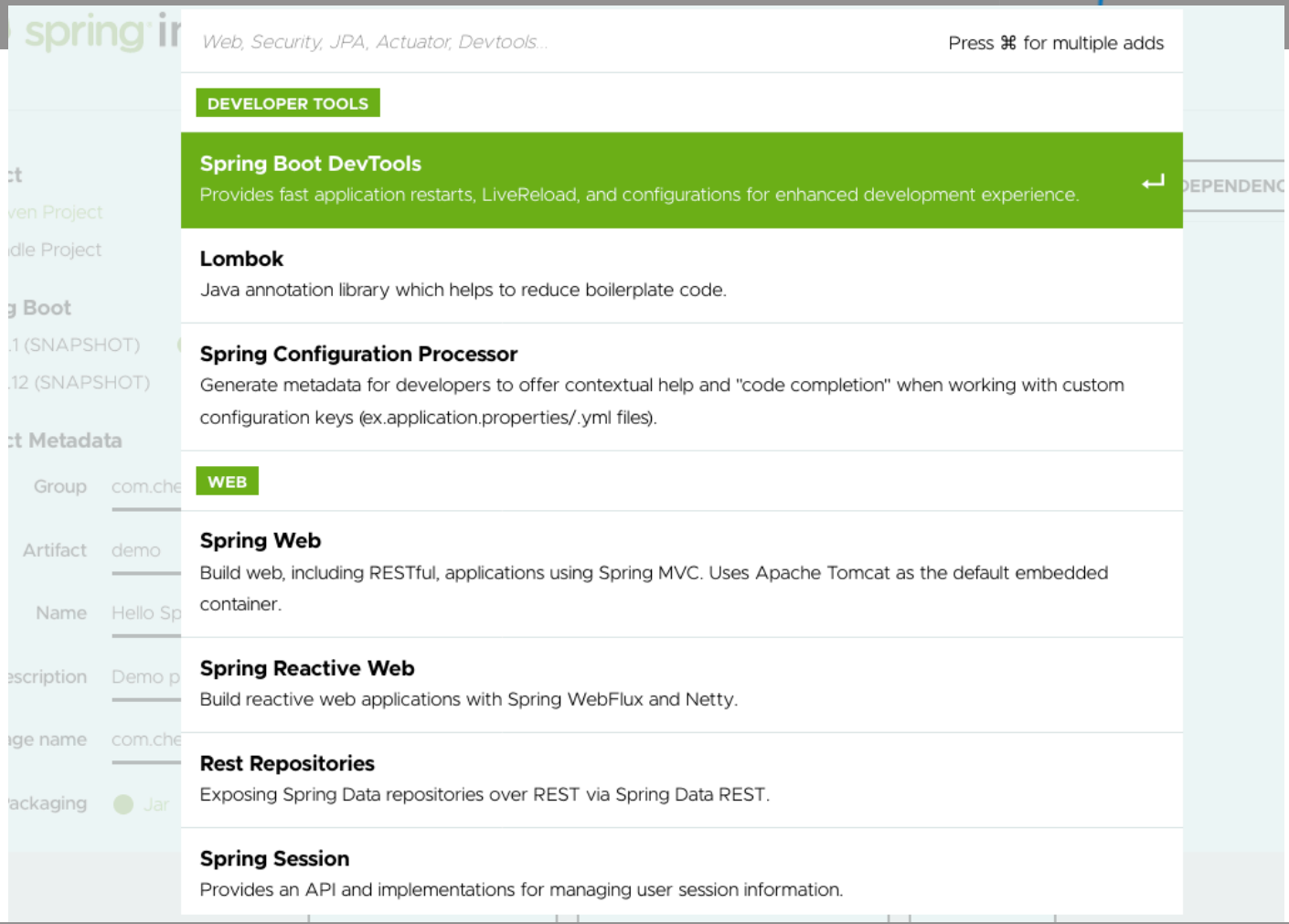
“Originally [Netflix's Java] libraries and frameworks were built in-house. I'm very proud to say, as of early 2019, we've moved our platform almost entirely over to Spring Boot.”

TAYLOR WICKSELL, SENIOR SOFTWARE ENGINEER, NETFLIX
Watch now

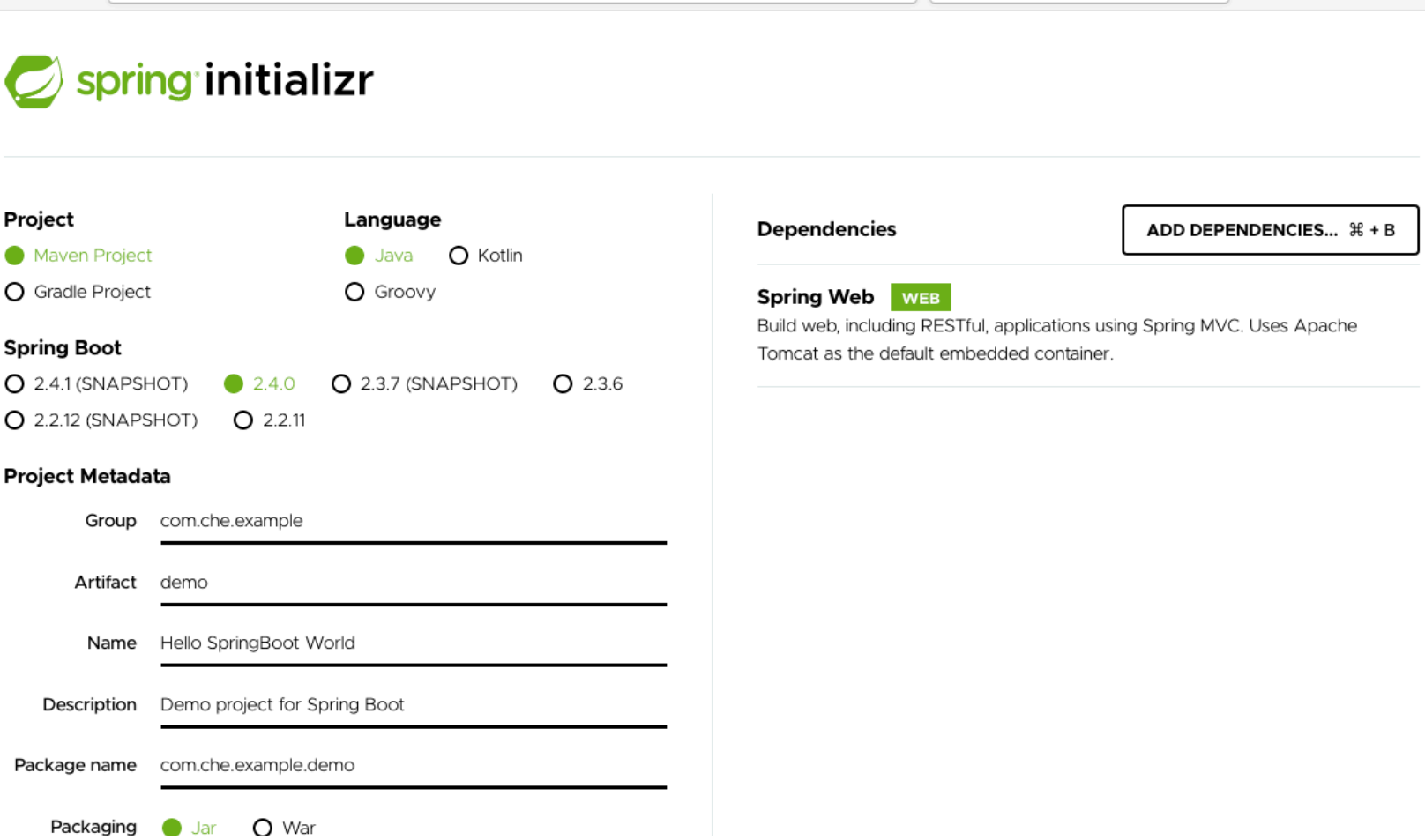
Use Spring Intializr with Java, Maven



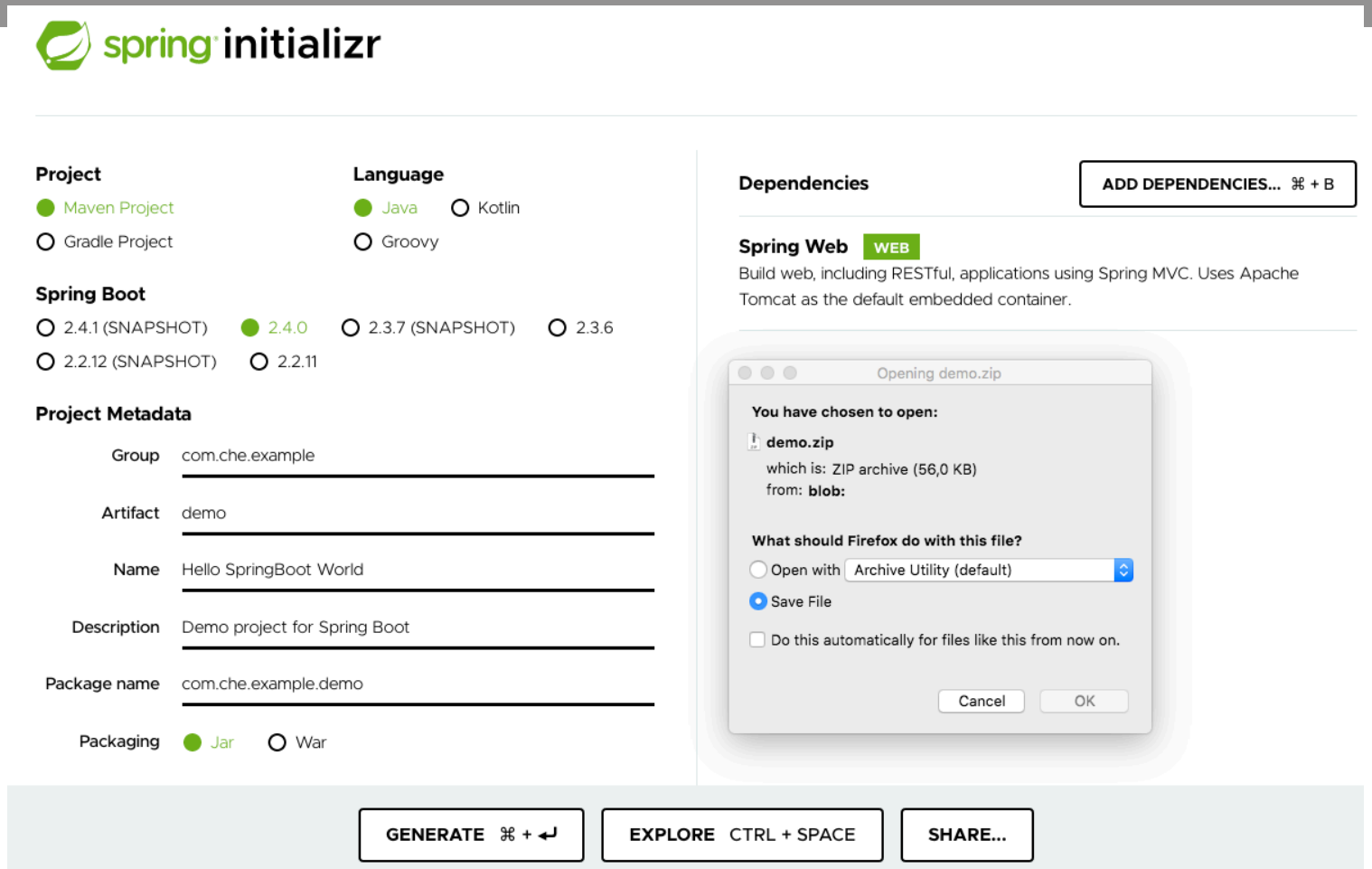
Add dependencies ..



Like Web incl. Tomcat

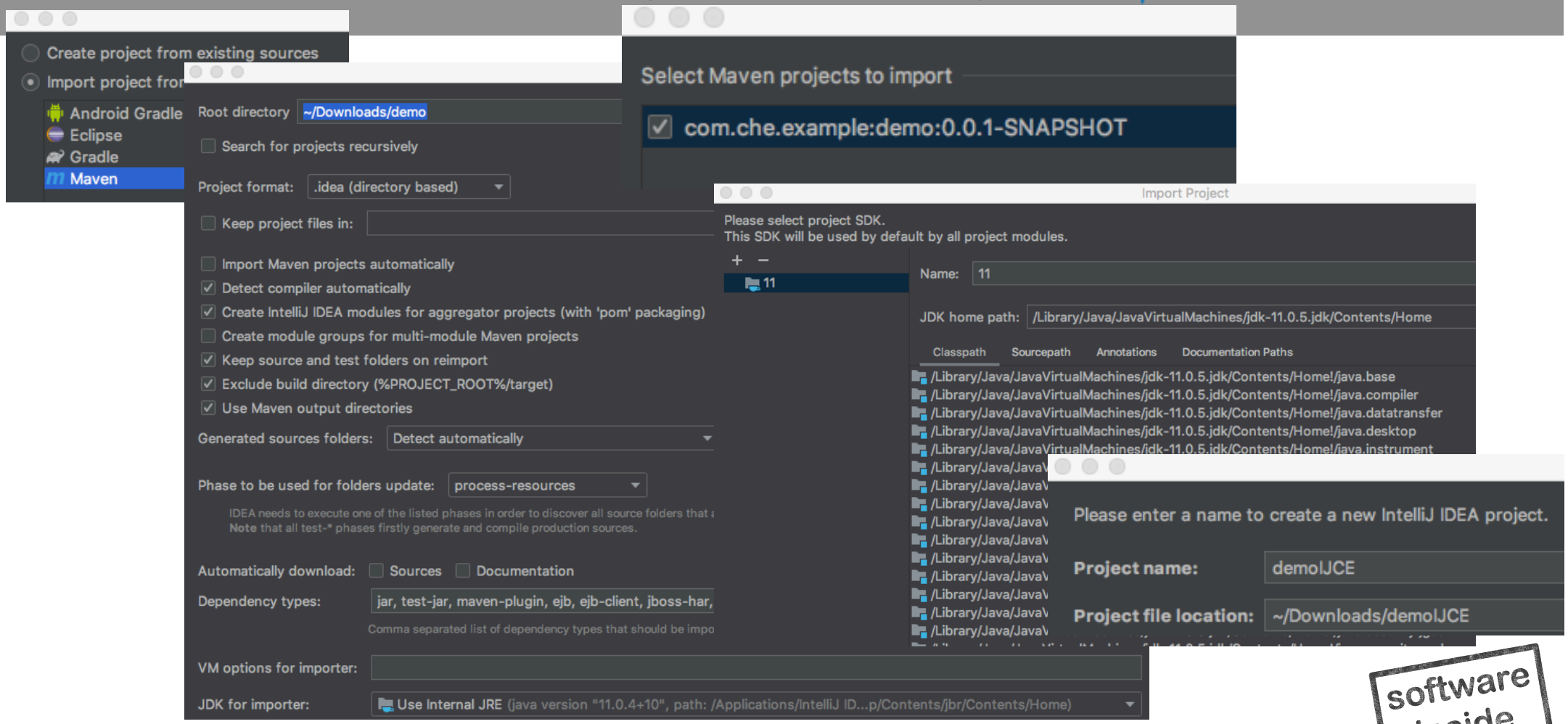


Press Generate button



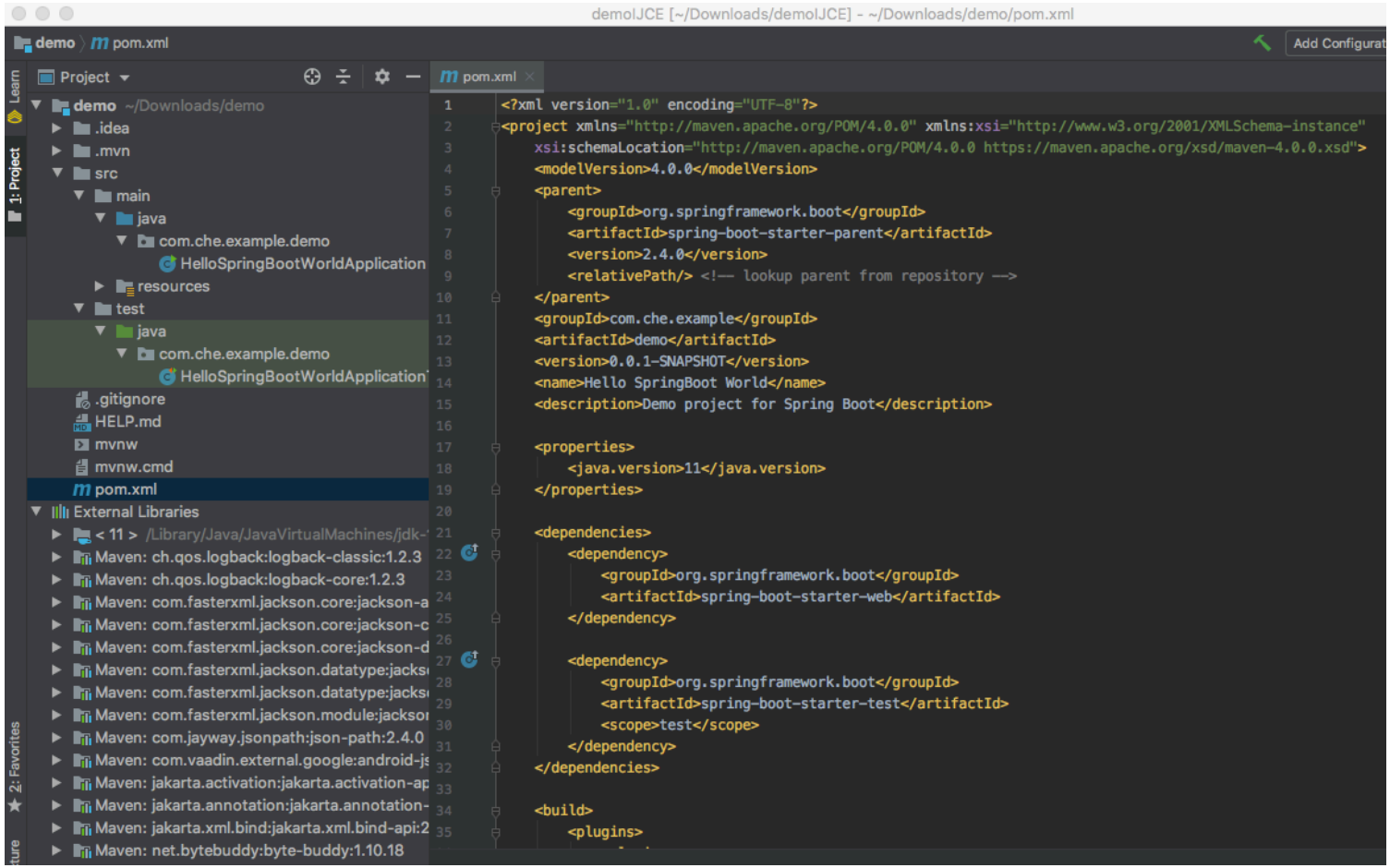
Import extracted zip by Community Edition

HTL Villach
Future Inside



The result is a POM file..

HTL Villach
Future Inside

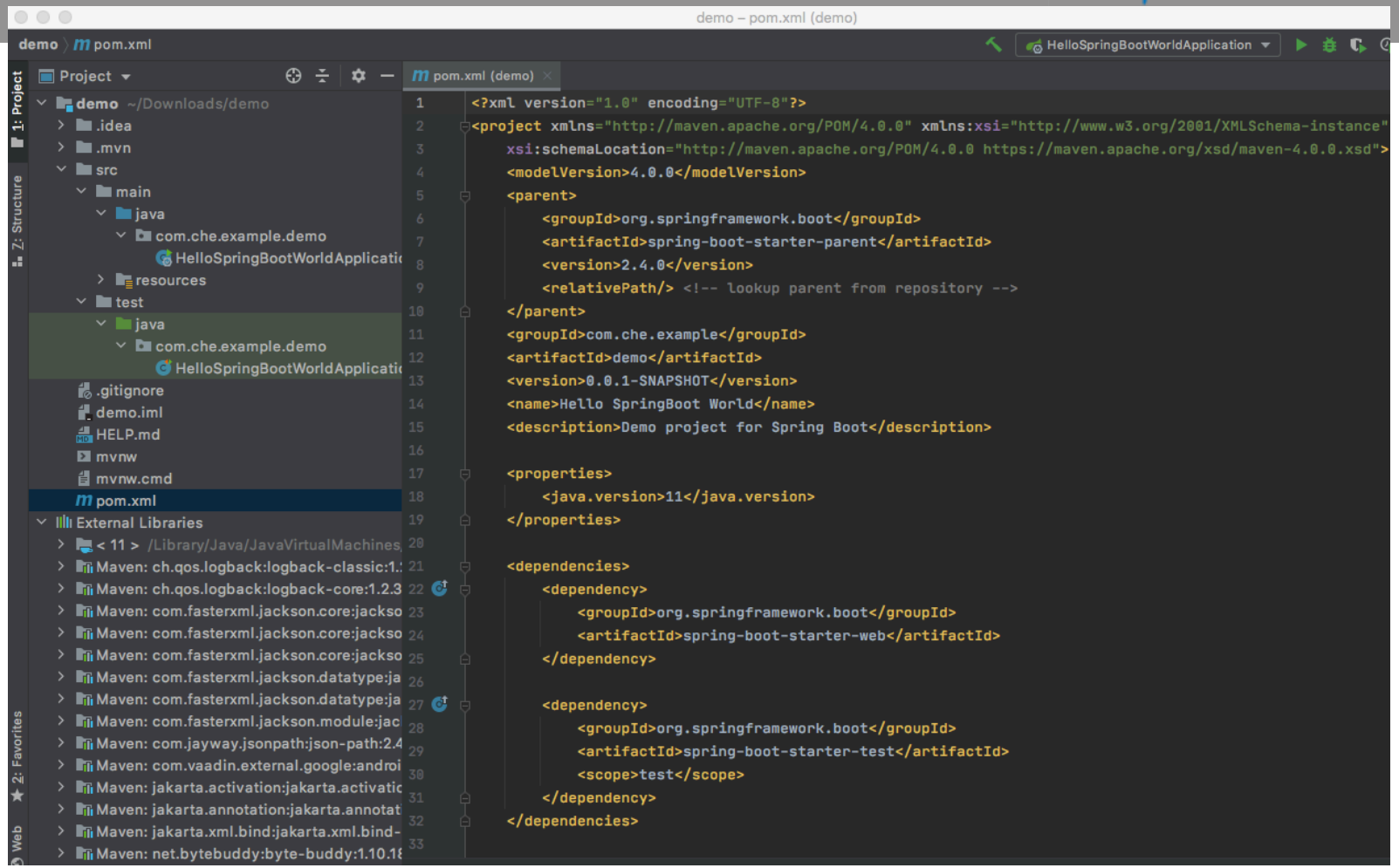


or re-extracted zip by Ultimate

HTL Villach
Future Inside

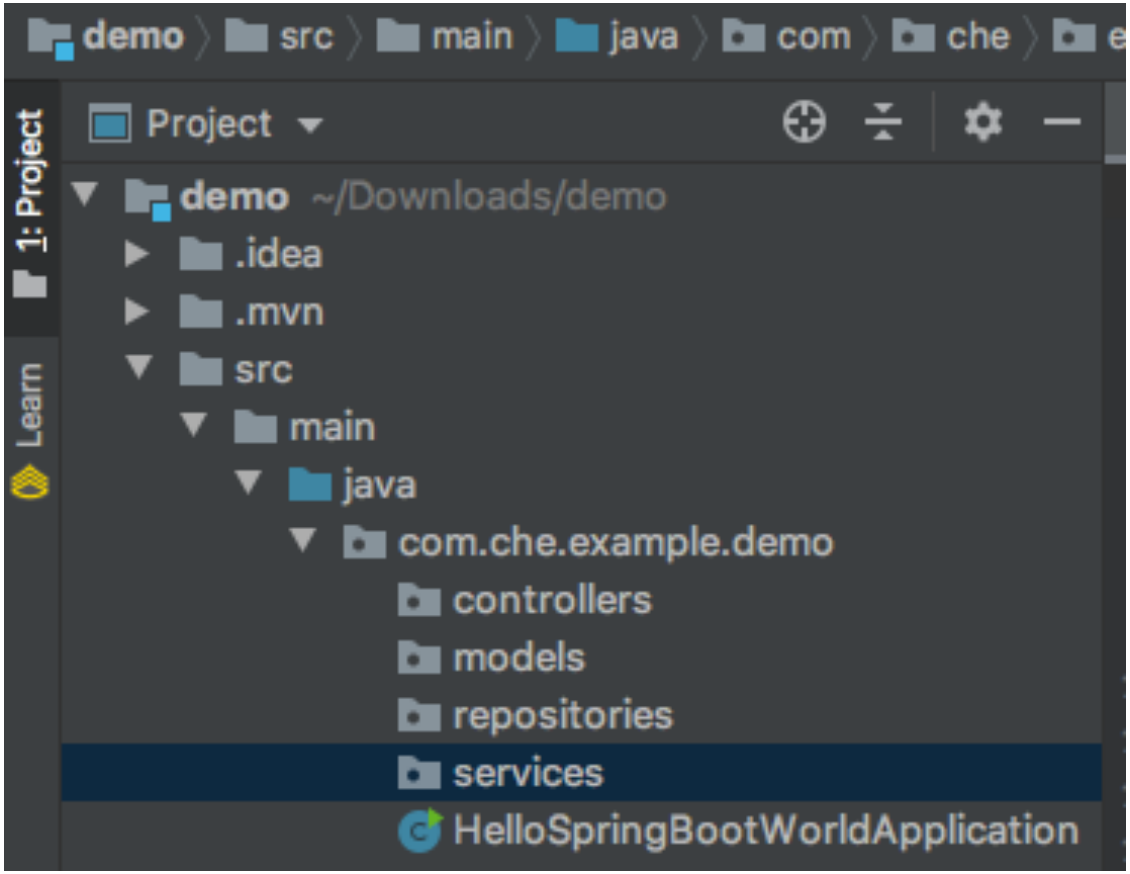
software inside

result is the same



Create best-practices packages on

HTL Villach
Future Inside



- controllers for the API
- models for JPA Entities or persistence information
- repositories for JPA access
- services for logic based code

software inside

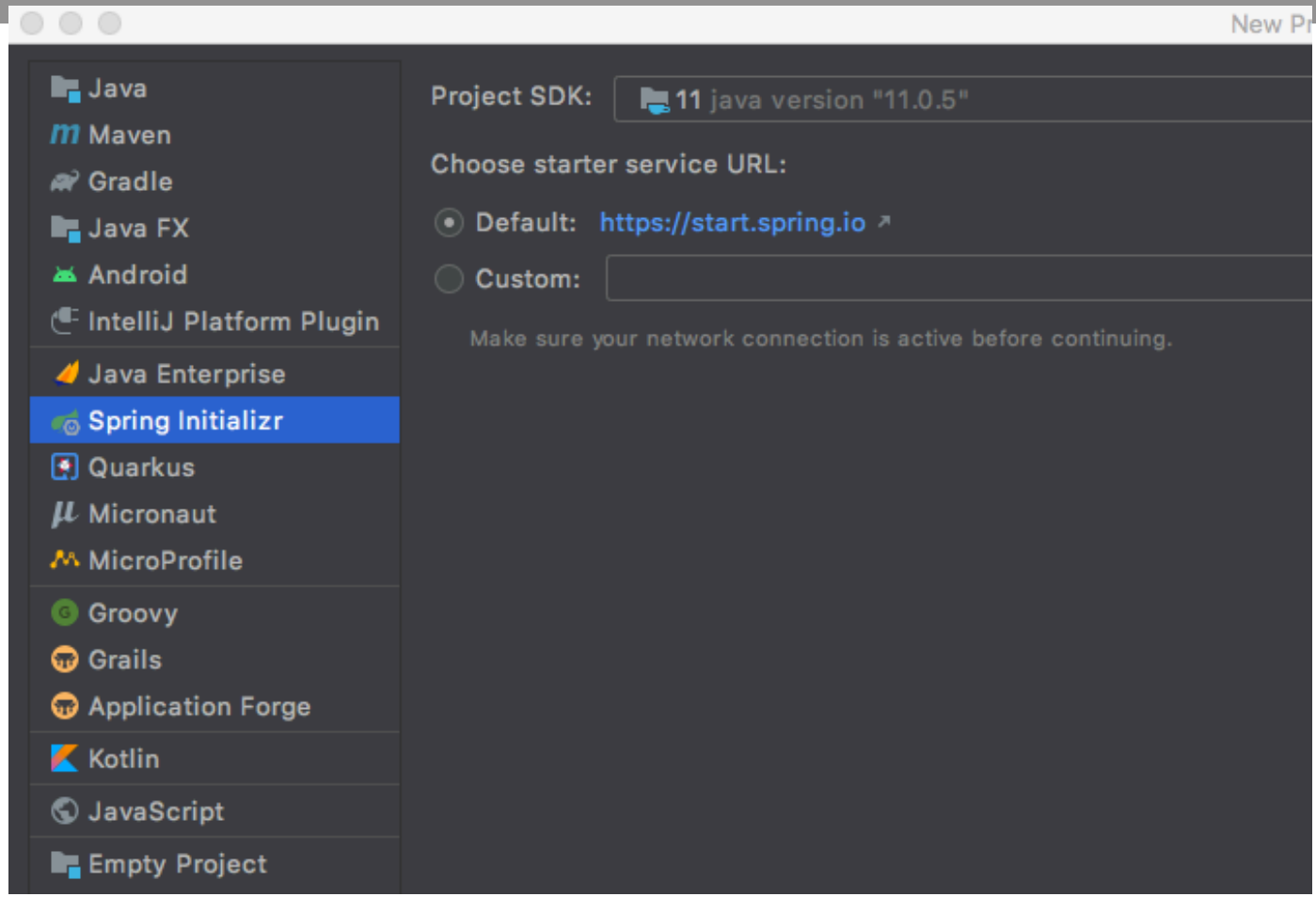
In InteliJ Ultimate create a new project

HTL Villach
Future Inside

software inside

Choose the Initializr wizard

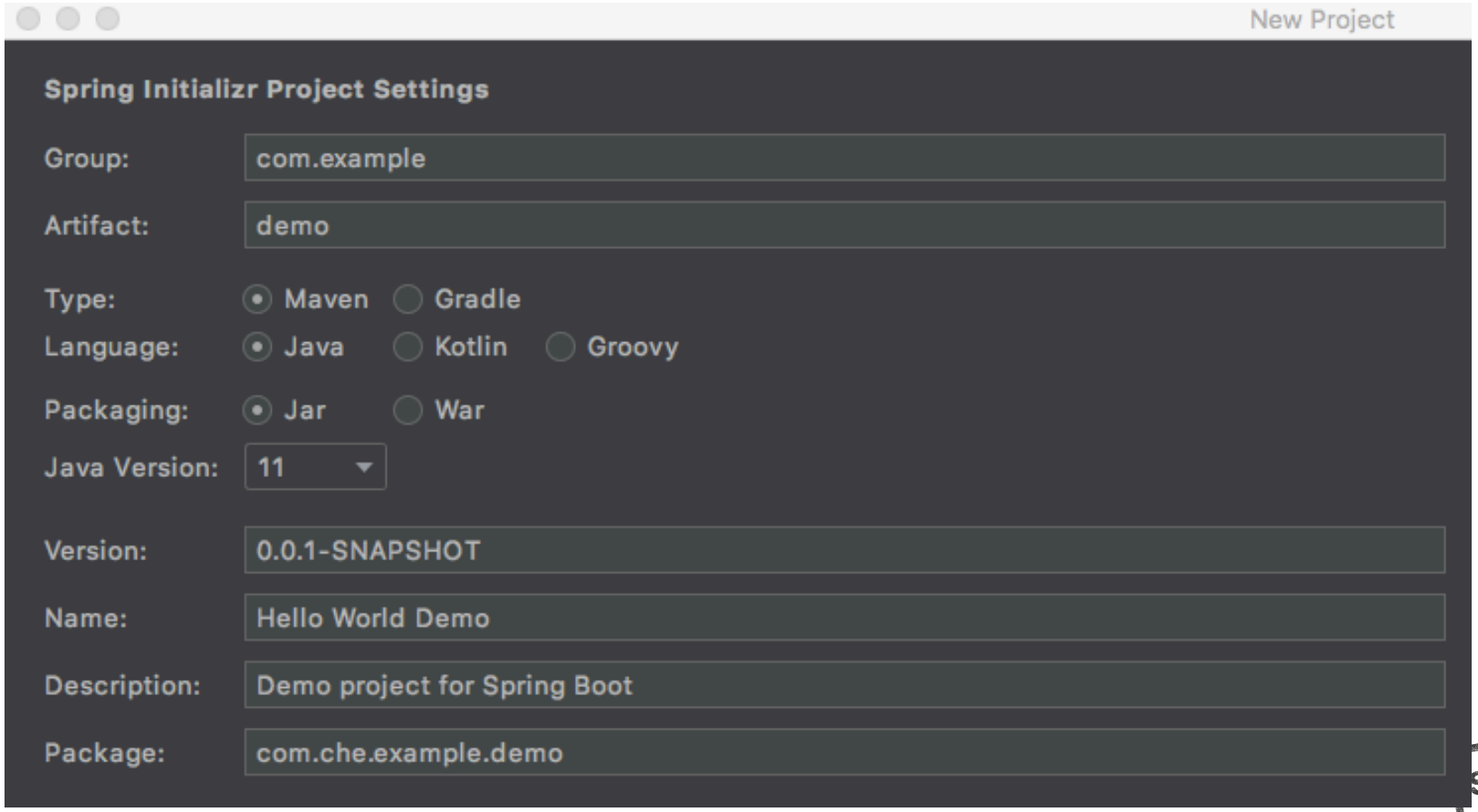
HTL Villach
Future Inside



software inside

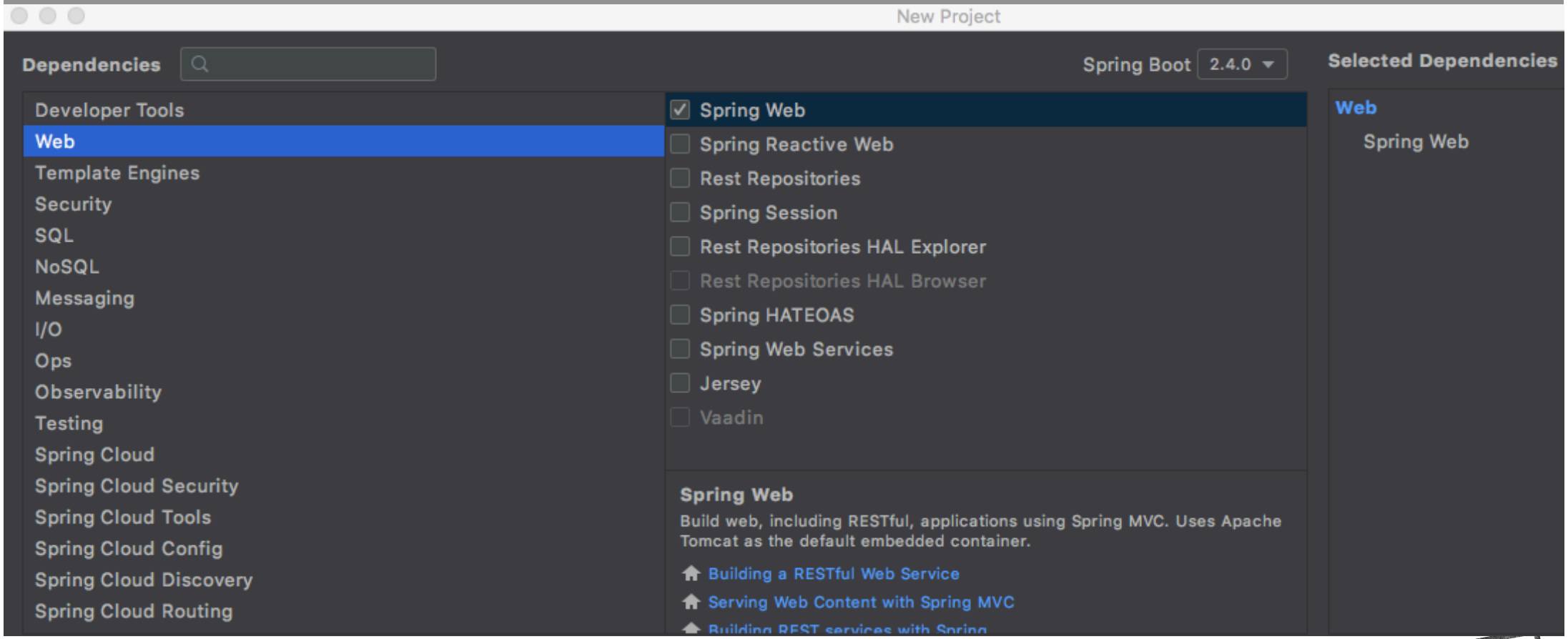
configure your project

HTL Villach
Future Inside

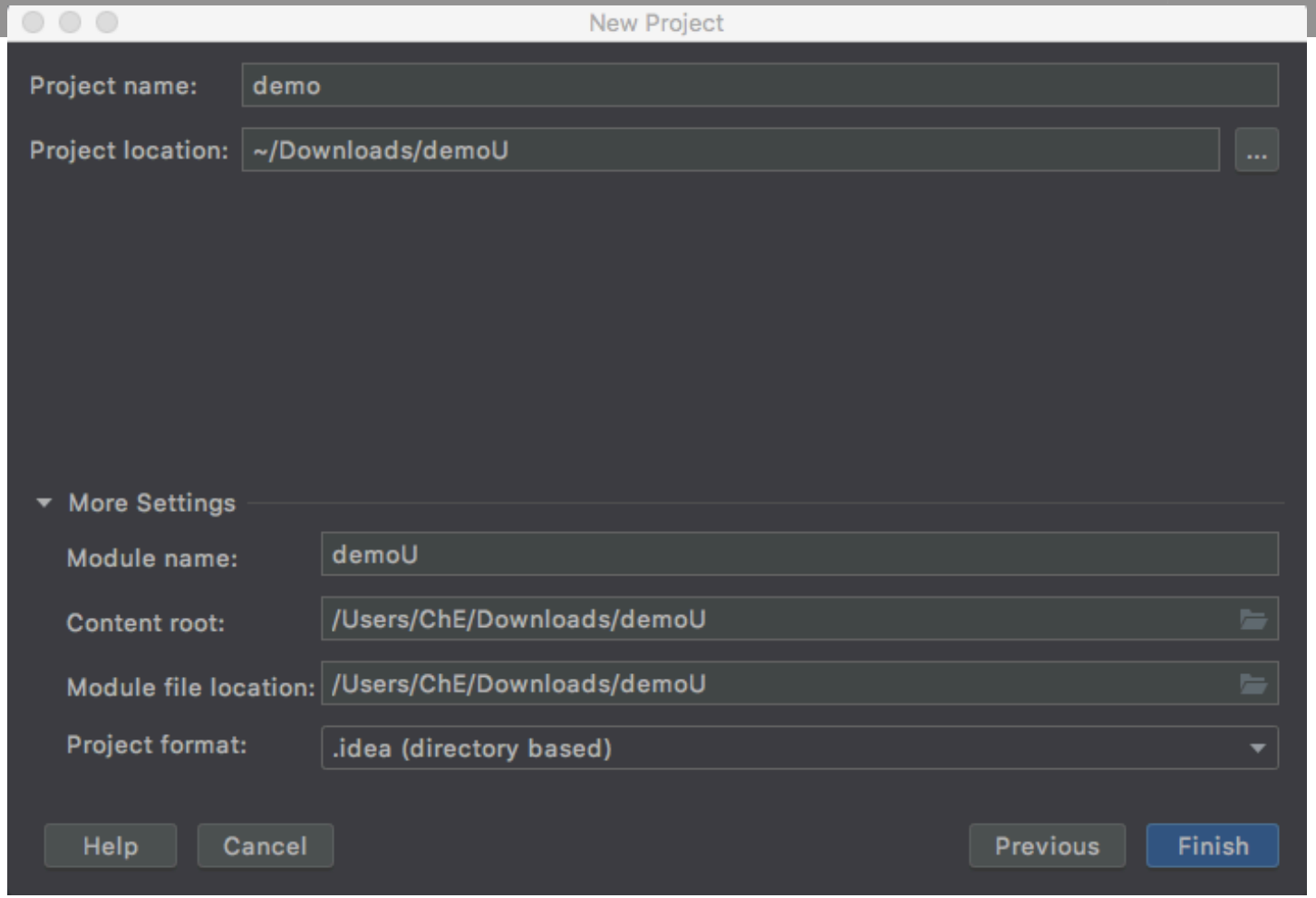


software inside

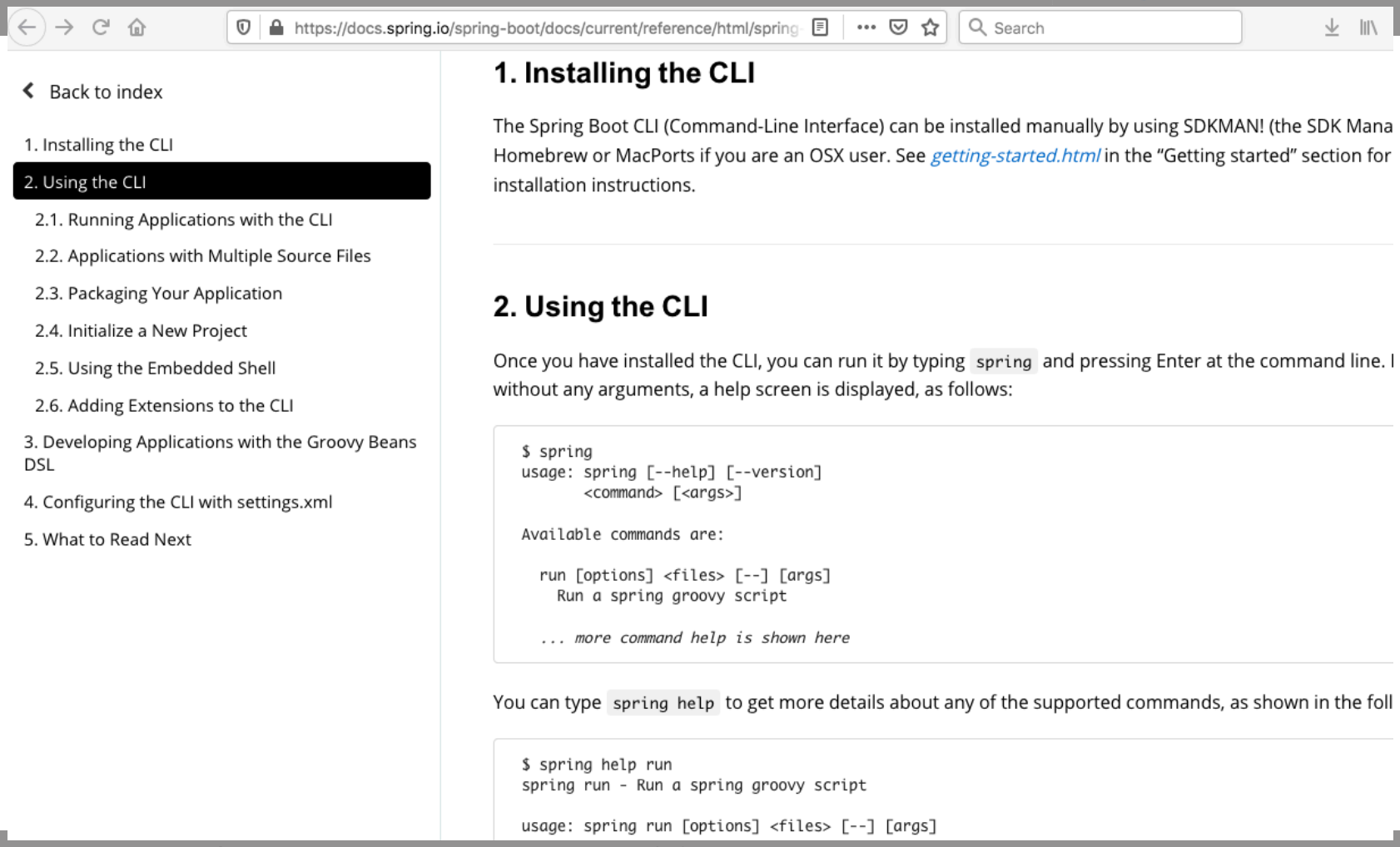
Select needed dependencies



And finish to get the same result



Or use the CLI



https://docs.spring.io/spring-boot/docs/current/reference/html/spring

Search

Back to index

1. Installing the CLI

The Spring Boot CLI (Command-Line Interface) can be installed manually by using SDKMAN! (the SDK Mana Homebrew or MacPorts if you are an OSX user. See *getting-started.html* in the "Getting started" section for installation instructions.

2. Using the CLI

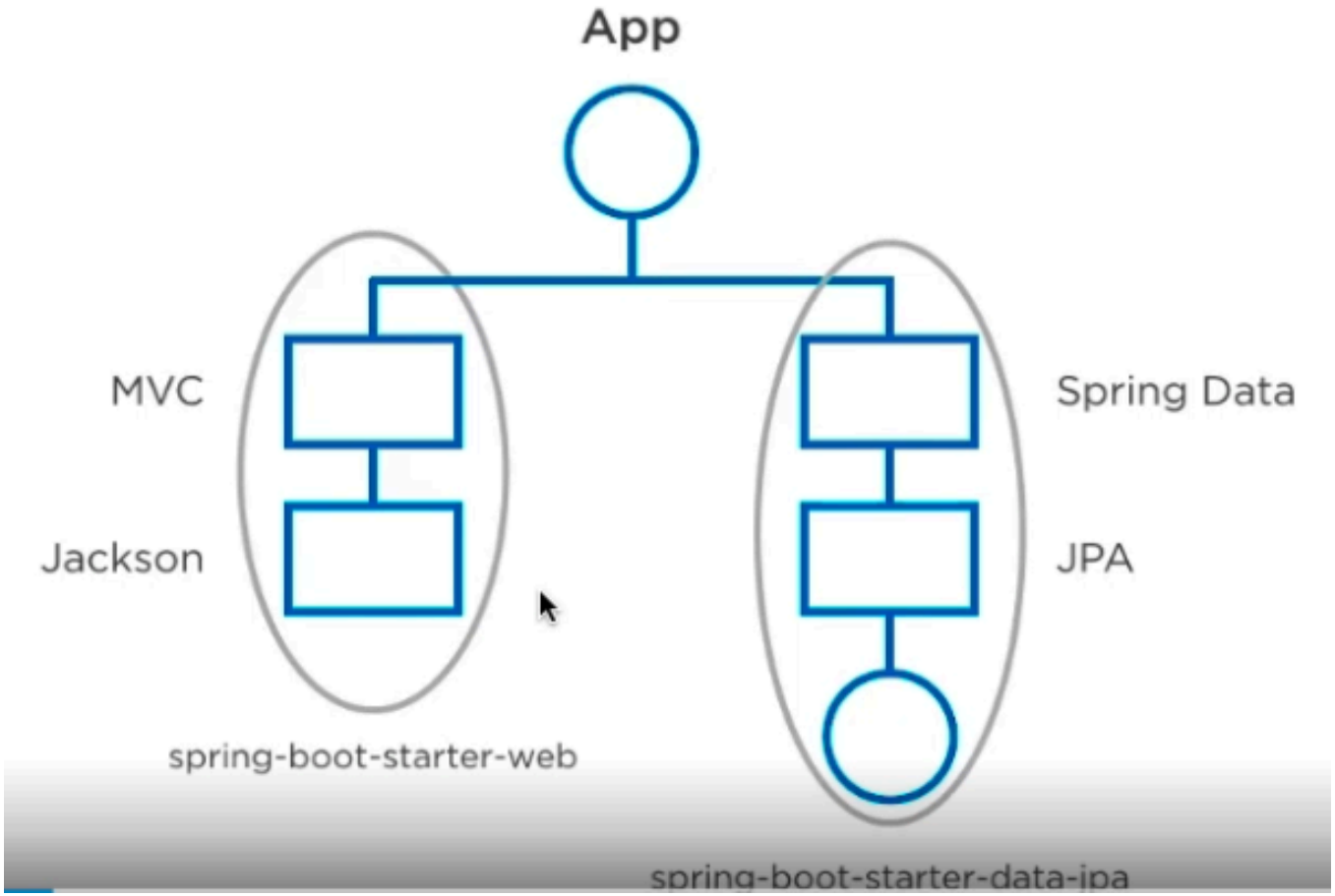
Once you have installed the CLI, you can run it by typing `spring` and pressing Enter at the command line. I without any arguments, a help screen is displayed, as follows:

```
$ spring
usage: spring [--help] [--version]
       <command> [<args>]

Available commands are:

  run [options] <files> [--] [args]
    Run a spring groovy script

  ... more command help is shown here
```

You can type `spring help` to get more details about any of the supported commands, as shown in the foll

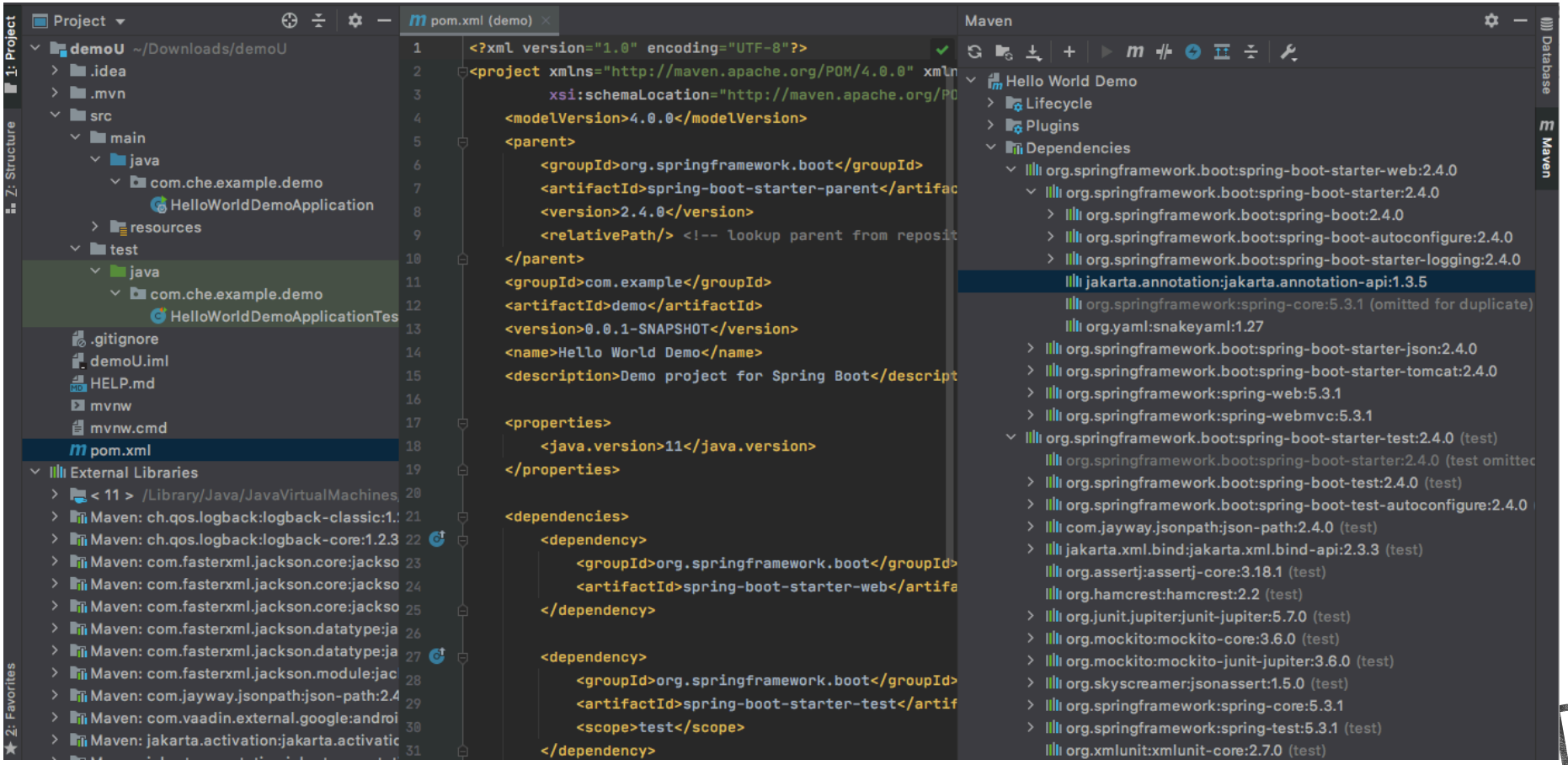
```
$ spring help run
spring run - Run a spring groovy script

usage: spring run [options] <files> [--] [args]
```

Spring Boot Starters



Understanding Spring Boot Starters

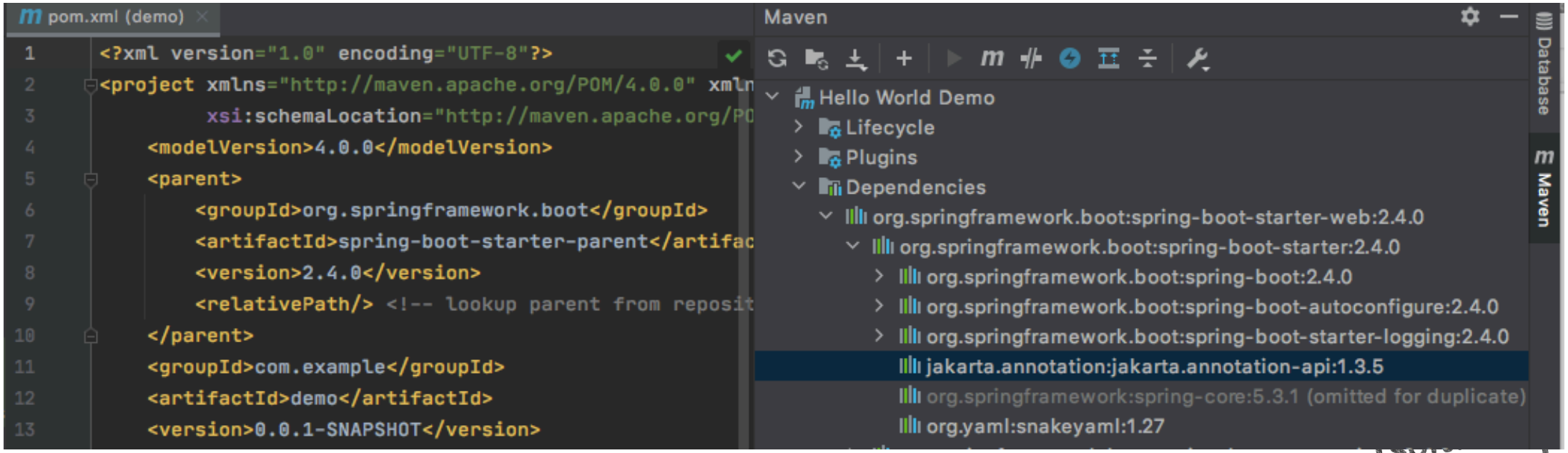
HTL Villach
Future Inside



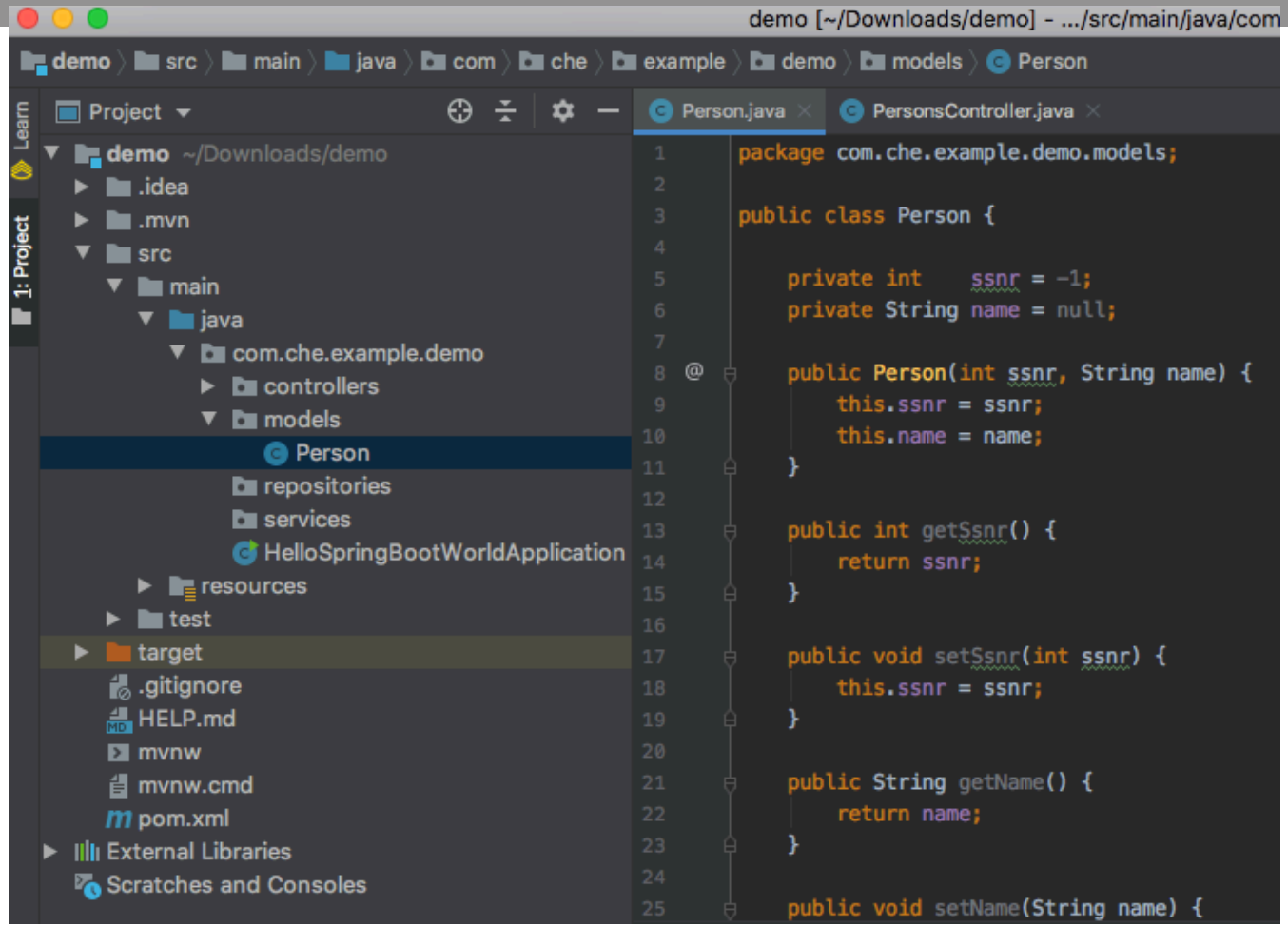
Boot Starters offer

HTL Villach
Future Inside

- Dependency Integration
- Versioning by starter parent

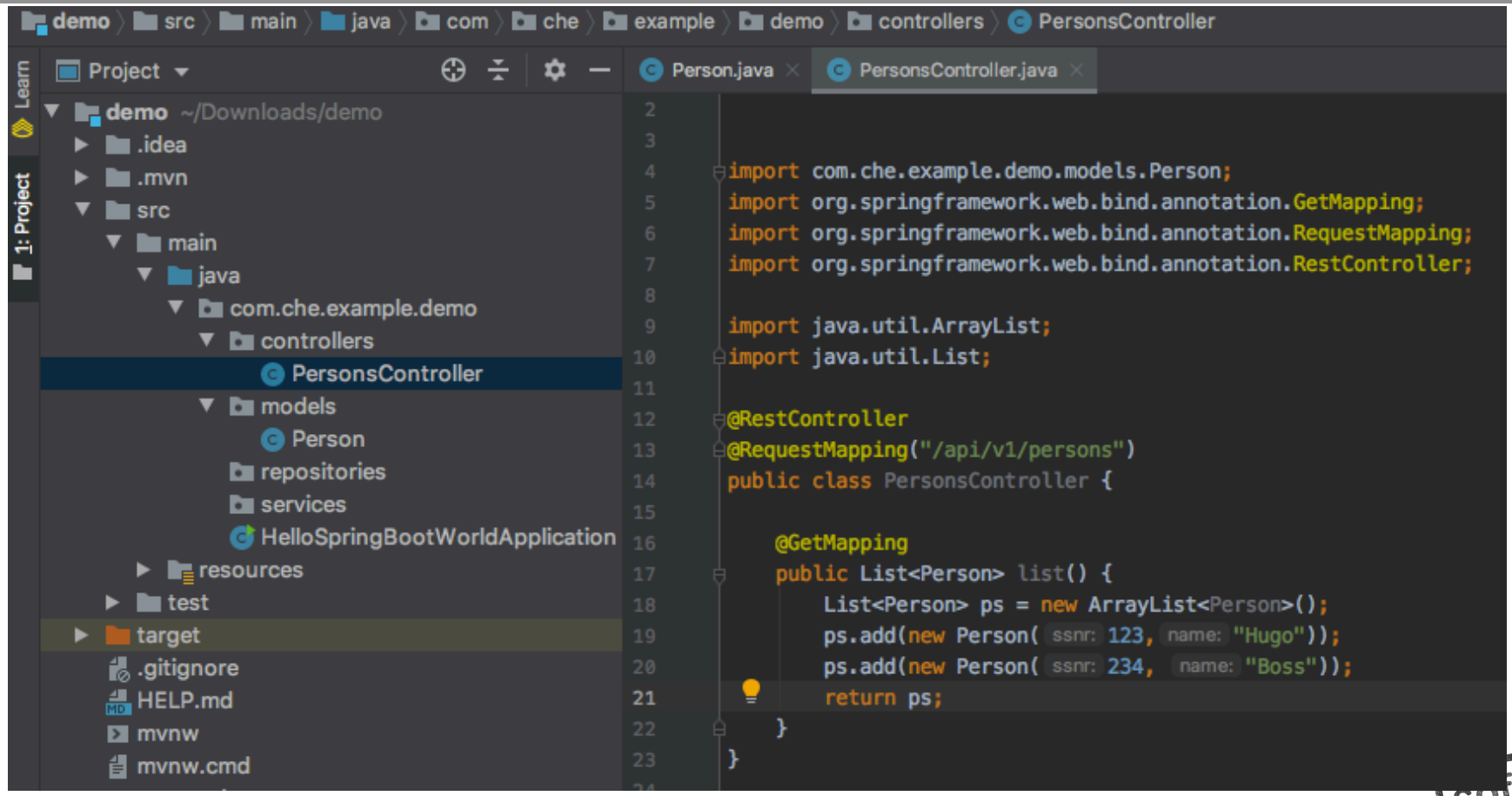


Add POJO models using Cmd/Ctrl+N



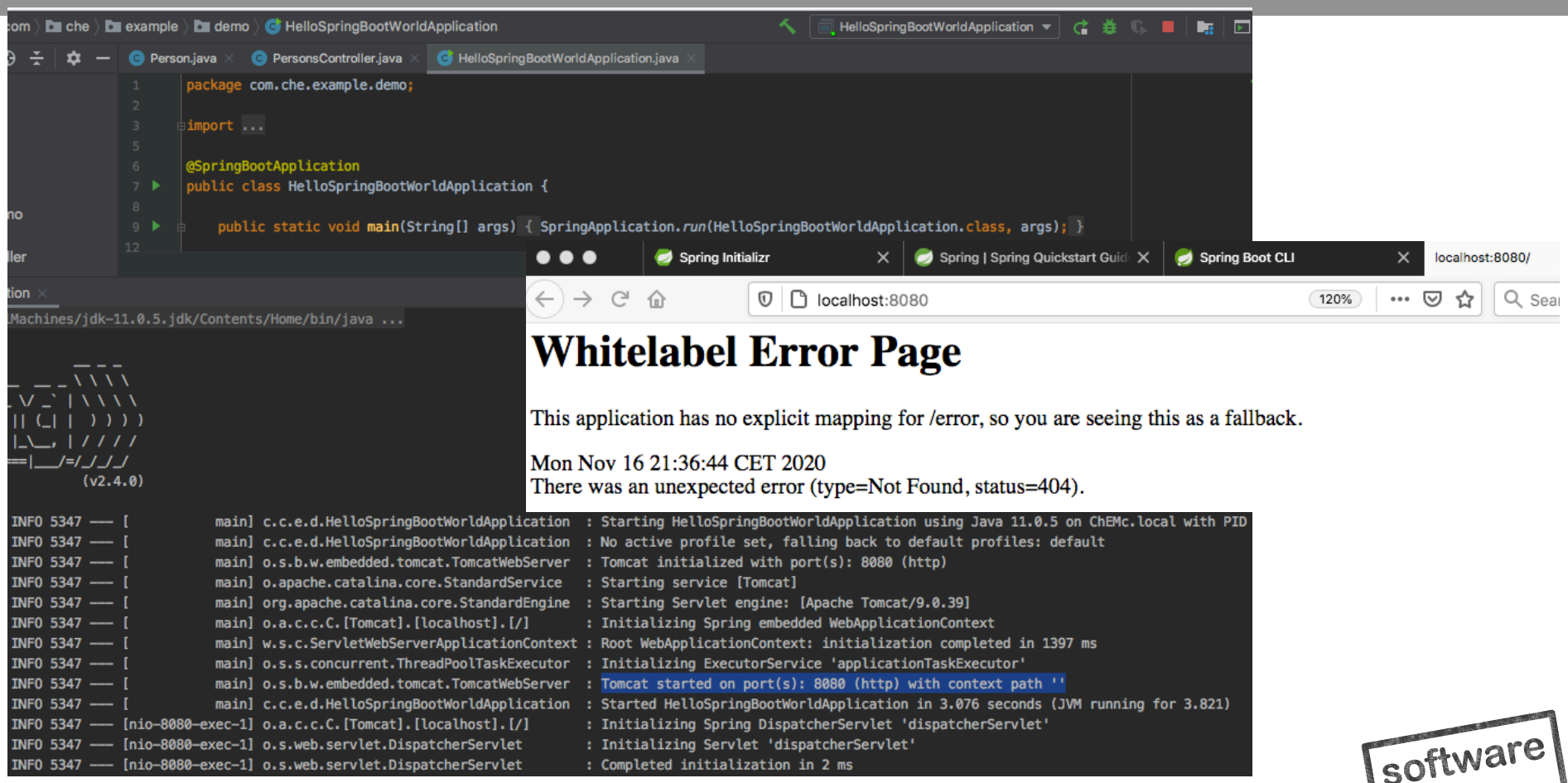
Add controllers

HTL Villach
Future Inside

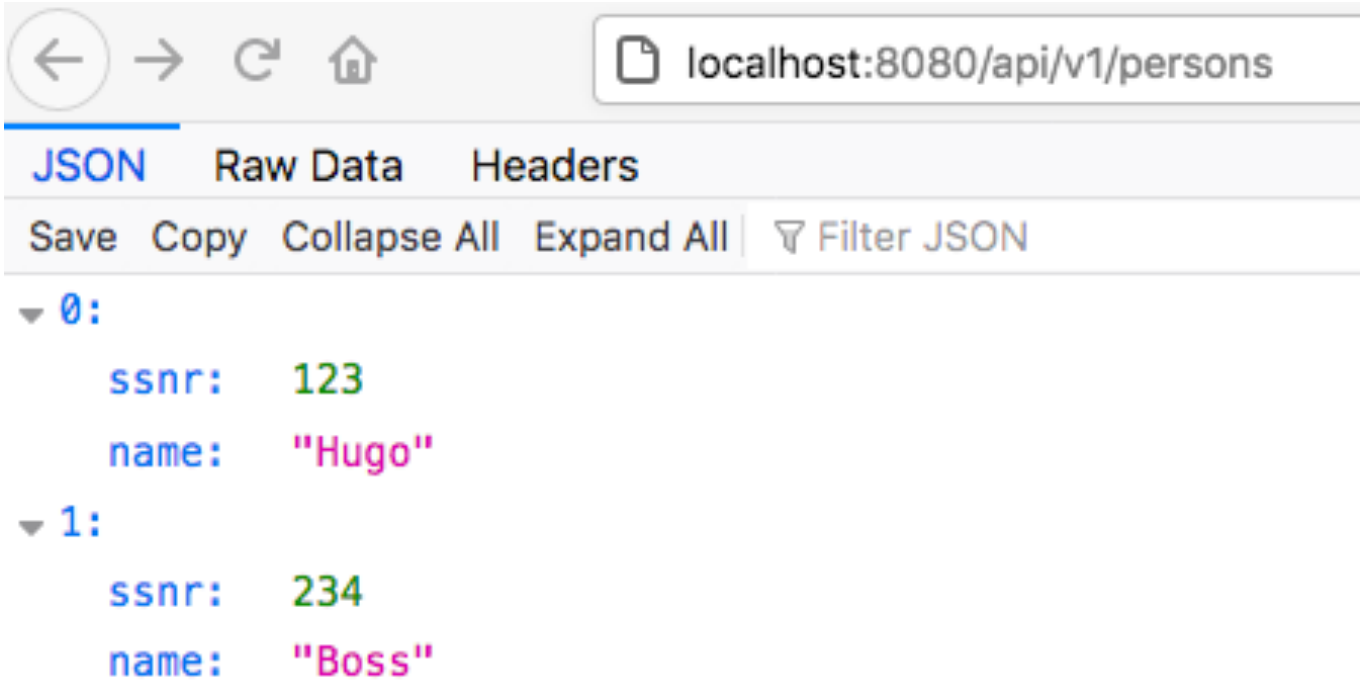


software inside

Build, start, check Tomcat and get “nothing”



or success on supported routes

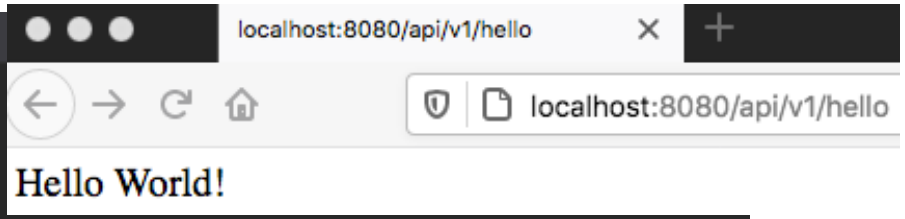


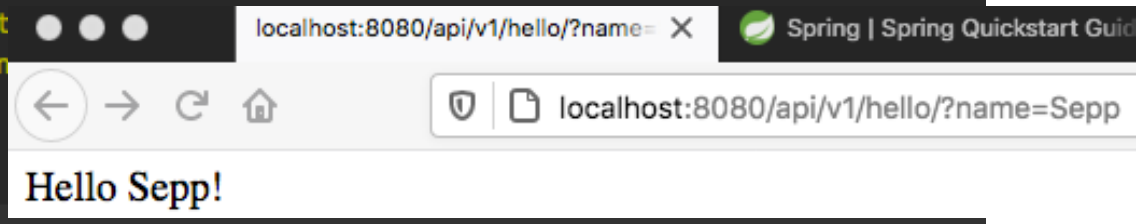
Hello World without POJOs or API controllers

HTL Villach
Software Inside

Person.java | PersonsController.java | HelloSpringBootWorldApplication.java

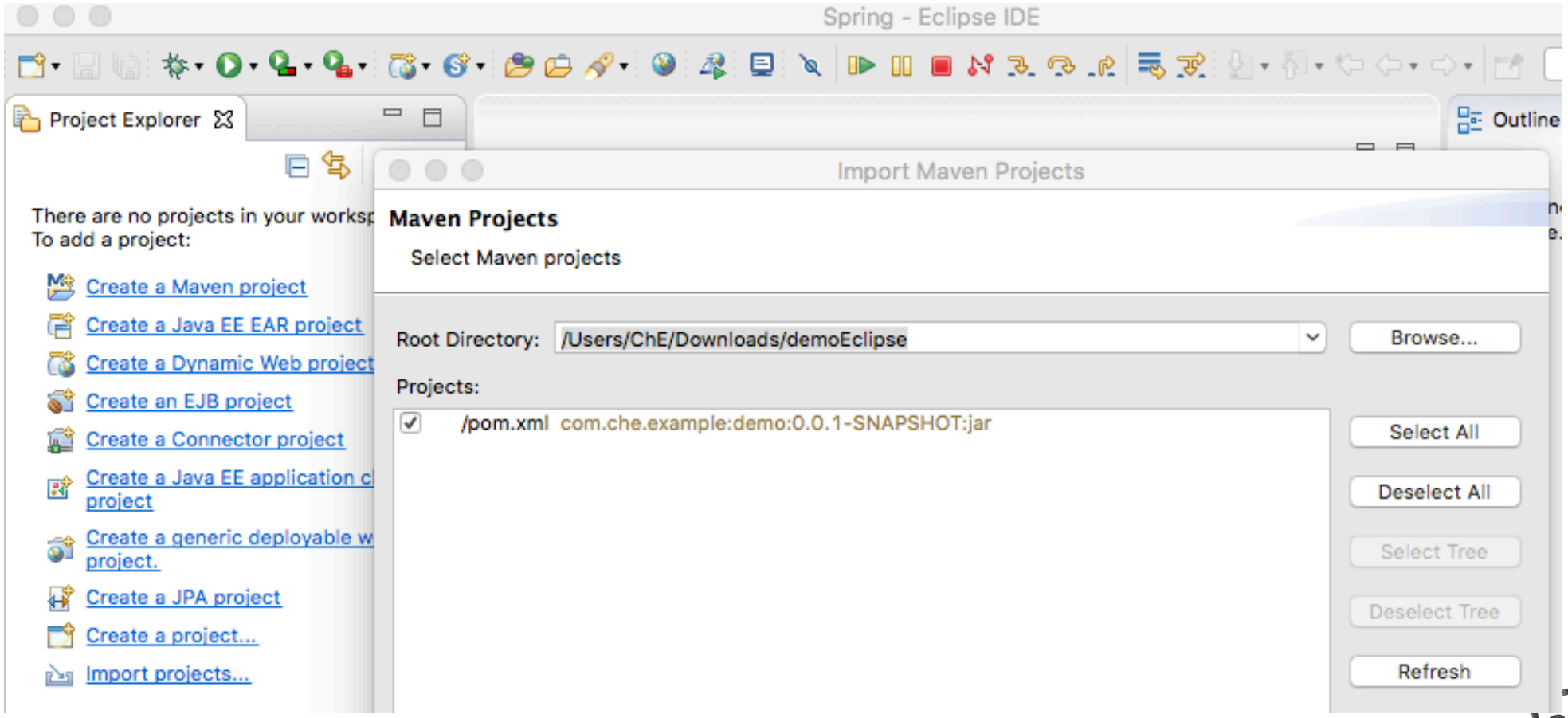
```java
package com.che.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.Request
import org.springframework.web.bind.annotation.RestCon

@SpringBootApplication
@RestController
public class HelloSpringBootWorldApplication {

    public static void main(String[] args) { SpringApplication.run(HelloSpringBootWorldApplication.class, args); }

    @GetMapping("/api/v1//hello")
    public String hello(@RequestParam(value = "name", defaultValue = "World") String name) {
        return String.format("Hello %s!", name);
    }
}
```





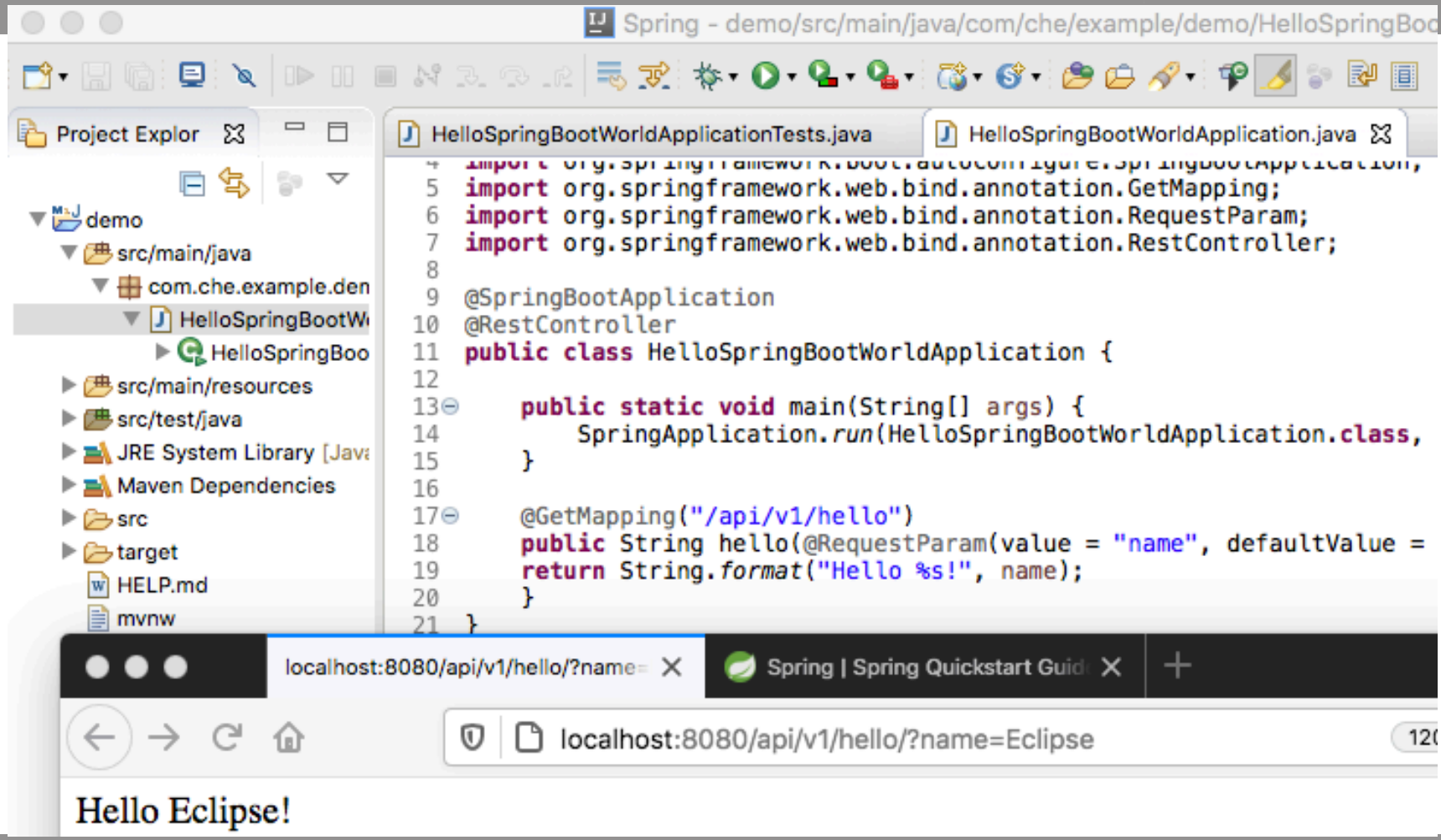
Btw, import extracted zip into Eclipse

HTL Villach
Future Inside



software inside

And start it


Spring - demo/src/main/java/com/che/example/demo/HelloSpringBoo
Project Explor
HelloSpringBootWorldApplicationTests.java
HelloSpringBootWorldApplication.java
demo
src/main/java
com.che.example.den
HelloSpringBootW
HelloSpringBoo
src/main/resources
src/test/java
JRE System Library [Java
Maven Dependencies
src
target
HELP.md
mvnw
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
@SpringBootApplication
@RestController
public class HelloSpringBootWorldApplication {
public static void main(String[] args) {
SpringApplication.run(HelloSpringBootWorldApplication.class,
}
@GetMapping("/api/v1/hello")
public String hello(@RequestParam(value = "name", defaultValue =
return String.format("Hello %s!", name);
}
}
localhost:8080/api/v1/hello/?name=
Spring | Spring Quickstart Guid
localhost:8080/api/v1/hello/?name=Eclipse
Hello Eclipse!

Open Questions / Back to the roots on

HTL Villach
Future Inside

- Annotations
- Dependency Injections
- Builder Patterns
- Build Tools
- ..
- understanding concepts, best practices, techniques, methods, .., tools (in context with teaching top-down or bottum-up)