Maven Exercise

Maven

Maven is a project management tool that helps you build applications and handles the dependencies of the application. For any non-trivial application, you will need a project management tool, such as Maven or Gradle. We will use Maven in all our exercises and projects. Spring Boot uses Maven or Gradle and we will use Maven in all our Spring Boot projects.

Read about Maven here:

https://maven.apache.org/what-is-maven.html

https://maven.apache.org/maven-features.html

Exercise 1 - Create the project

Create a Spring Boot project as described in the exercise for the Spring Boot Crash Course, but do **NOT** select any dependencies! Do **not** select Web – Web or Template Engings – Thymeleaf, just skip this stage and create the project.

Just as in the Spring Boot Crash Course exercise, put a **@RestController** annotation over the **@SpringBootApplication** annotation and create the same simple method that returns a string of “hello” with a **@GetMapping(“/”)** annotation over it.

Notice that the annotations become red in IntelliJ IDEA. This is because IntelliJ doesn’t know about these annotations because we haven’t added the dependencies to the jar-files that contain these classes yet. That is what the Spring Initializer does for us automatically by checking these dependencies when we create the project. This time, we will add the dependencies manually with Maven instead.

Check the file **pom.xml** in the root of the project. This is the configuration file of Maven.

Dependencies are added to Maven in the **<dependencies>** tag. Notice there is already a dependency for Spring Boot, it looks like this:

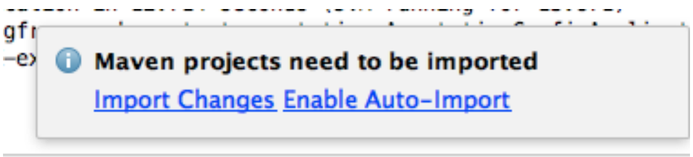
```
<dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter</artifactId>
</dependency>
```

Now, under this dependency declaration (inside the dependencies tag) add a new dependency for the Spring Boot Web support. It should look like this:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

(This is exactly what the Spring Boot Initializer does when you select **Web –** Web in dependencies)

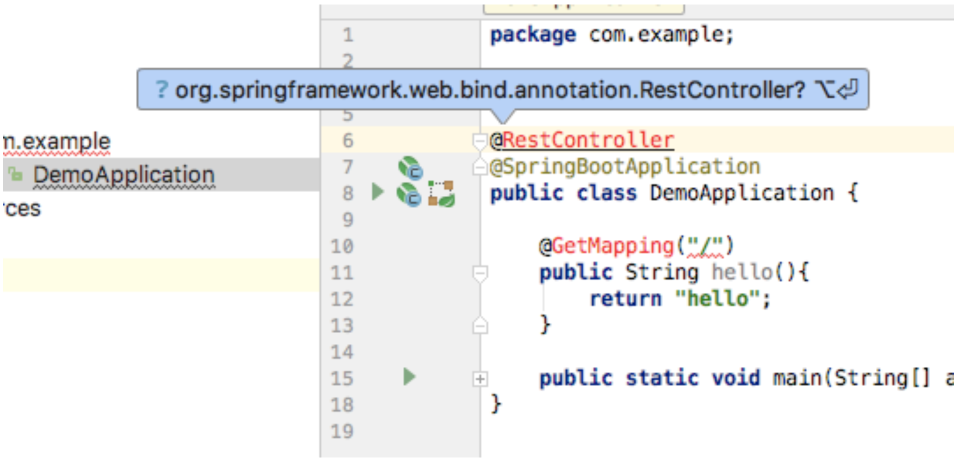
When you add this dependency declaration, you should get this information from IntelliJ IDEA in the lower right corner:



Maven projects need to be imported

Click “Enable Auto-Import”.

If you now go back to the **DemoApplication** class, you should see a suggestion from IntelliJ to import the annotations. If you don’t, place the marker on the red annotation and you should see something like this:



? org.springframework.web.bind.annotation.RestController? ⌥⏎

```
package com.example;

@RestController
@SpringBootApplication
public class DemoApplication {

    @GetMapping("/")
    public String hello(){
        return "hello";
    }

    public static void main(String[] a
}
```

This is a sign that Maven has now downloaded the dependency to the Spring Boot Web support and is now asking you if you want to import this new annotation that it found in this dependency.

Select to import the suggested annotations, and they should not be red anymore.

Congratulations! You have now used Maven directly to specify a dependency in the application, this made Maven go and download the dependency from its repository, and you are now using the dependency in the application.

You can look at the Maven repository here:

https://mvnrepository.com

Exercise 2 - Build the project with Maven

Maven is a Build System, and could be used for much more than dependency management.

Maven can be downloaded, installed, and run on the command line. You can read more about this here:

https://maven.apache.org/install.html

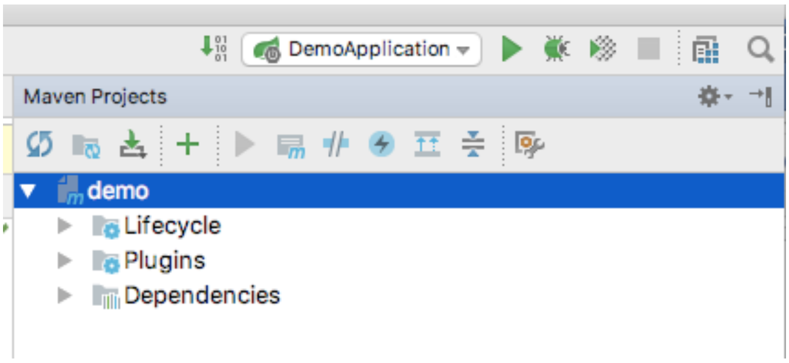
Maven is also integrated in IntelliJ IDEA so you can use it from IntelliJ. Read more about the integration here:

https://www.jetbrains.com/help/idea/2016.3/maven.html

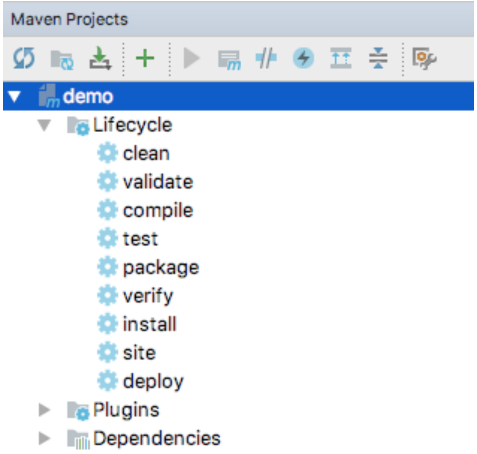
To easily control the lifecycle of the application, IntelliJ has a Maven Tool Window. Open the Maven Tool Window like this:

Select in the IntelliJ menu **View – Tool Windows – Maven Projects**

You should see something like this on the right side of IntelliJ IDEA:



Expand the Lifecycle alternative and you should see something like this:



Here, you can run different Maven commands to compile the project. Try to compile the project by double clicking on compile.

You should get a **BUILD SUCCESS** message in the console.

Stretch Tasks (if you have time)

If you have time, read the IntelliJ IDEA getting started guide with maven:

https://www.jetbrains.com/help/idea/2016.3/getting-started-with-maven.html