Creating a Simple Spring Boot App

1. Creating a Spring Boot app in IntelliJ
2. Understanding the application

1. Creating a Spring Boot App in IntelliJ

- Overview of IntelliJ
- Creating a Spring Boot project
- Specifying project dependencies
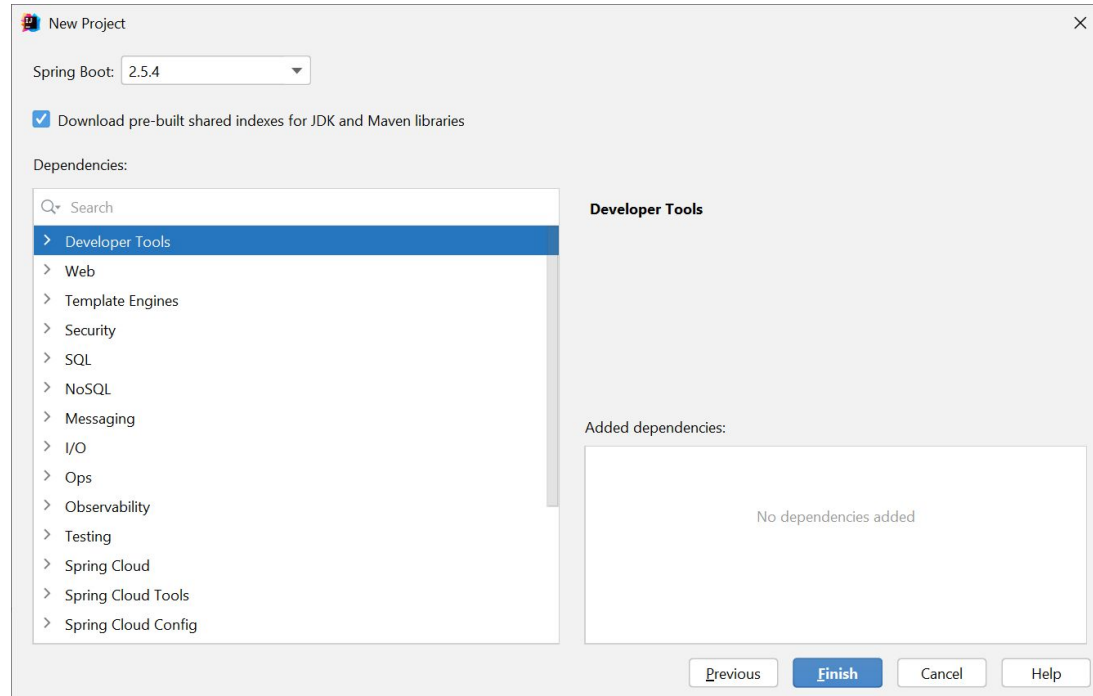
Overview of IntelliJ

- IntelliJ IDEA Ultimate has excellent support for Spring

  - Spring Boot and Spring Framework

- IntelliJ Java dependencies:

  - JDK (e.g. JDK 11)

  - Set `JAVA_HOME` to the JDK folder

  - Set `PATH` to include the JDK binary folder

Creating a Spring Boot Project

- Start IntelliJ, click New Project, and select Spring Initializr

- Specify project info as follows:
  - Enter a suitable project name and location
  - Language - Java
  - Enter a suitable group ID, artifact ID, and package name
  - Project SDK - Java 11
  - Java version - 11
  - Packaging - JAR
  - Then click Next

Specifying Project Dependencies

- In the next window you can add dependencies to your project
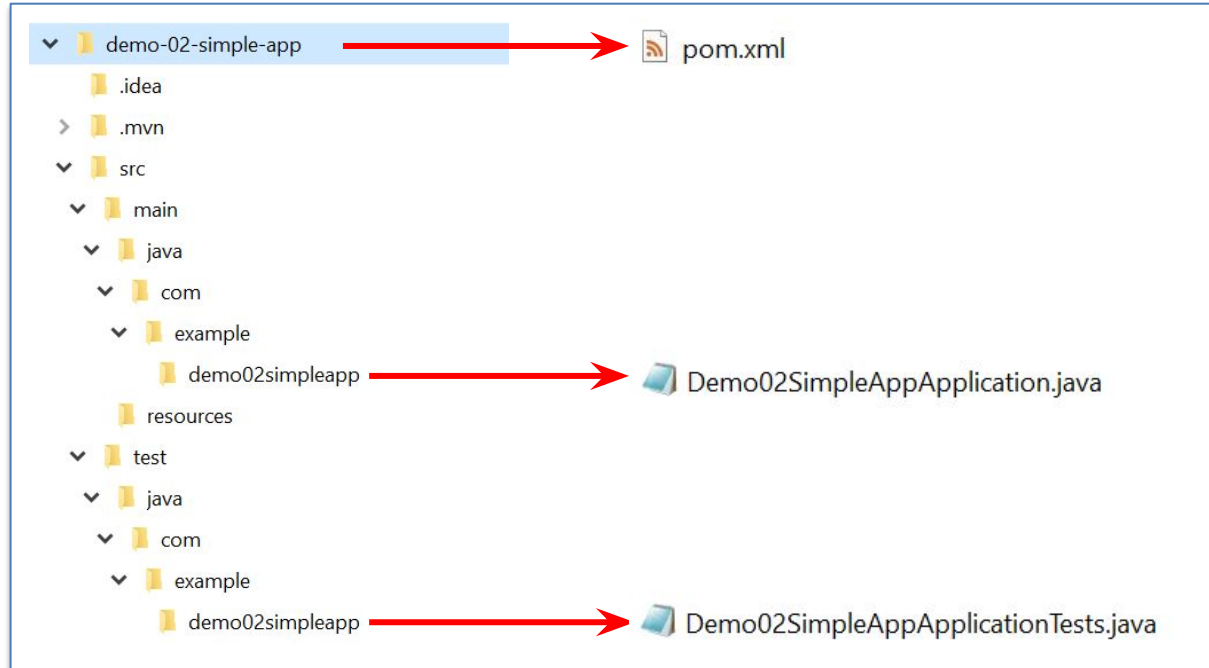 - We don't need any dependencies yet, so just click Finish

2. Understanding the Application

- Understanding the application structure
- Understanding the Maven POM file
- Understanding the application code
- Running the application
- Viewing the application output

Understanding the Application Structure

- The generated application is a regular Maven project:

Understanding the Maven POM File

- Here are the relevant sections in the Maven POM file:

```xml
<project … >
    …
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.5.4</version>
        <relativePath/>
    </parent>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>
    …
```

Parent POM

Spring Boot dependency

Spring Boot test dependency

pom.xml

Pearson

Understanding the Application Code

- Here's the generated application code:

```java
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(Demo02SimpleAppApplication.class, args);
    }
}                                         Demo02SimpleAppApplication.java
```

- `@SpringBootApplication` is equivalent to:
  - `@Configuration`
  - `@EnableAutoConfiguration`
  - `@ComponentScan`

Pearson

Running the Application

- You can build/run the app via the `mvn` command in the project root directory as follows:

```
mvn spring-boot:run
```

- If you don't have Maven installed separately on your machine, you can run the following command instead:

```
mvnw spring-boot:run
```

- It's also possible to run the application directly in IntelliJ
  - Right-click in the main class, and click Run

Viewing the Application Output

- This is the application output
  - Displays a "Spring Boot" banner
  - The app terminates immediately, because it's so simple 😊

Summary

- Creating a Spring Boot app in IntelliJ
- Understanding the application