Spring Boot

Overview

- Easily create standalone, production-grade Spring applications that you can “just run”.
- Opinionated out of the box but gets out quickly as requirements start to diverge from the defaults.
- Provides wide range of non-functional features common to large classes of projects such as embedded servers, security, metrics, health checks, externalized configuration, etc.
- No code generation and no requirements for XML configuration.

Creating a Spring Boot application

```
<project>
  <groupId>com.example</groupId>
  <artifactId>myproject</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.9.RELEASE</version>
  </parent>
</project>
```

Spring Boot starters

- One-stop-shop for including Spring and related technologies in a Spring Boot project.
- Common starers include:
  - **spring-boot-starter**: Core starters, auto-configuration support, logging, etc.
  - **spring-boot-starter-data-jpa**: Includes Spring Data JPA and Hibernate.
  - **spring-boot-starter-web**: For building web, and RESTful applications using Spring MVC. Uses Tomcat as default embedded container.
  - **spring-boot-starter-security**: Starter for Spring Security.
  - **spring-boot-starter-test**: For testing Spring Boot applications.
  - etc.

Including starters in a Spring Boot application

Example:

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

Specifying the Java version

```
<project>
  <properties>
    <java.version>1.8</java.version>
  </properties>
</project>
```

Using the Spring Boot Maven plugin

- Uses to:
  - Start the application via Maven with “mvn spring-boot:run”
  - Package the project as an executable jar with “mvn package”.
- To use it:

```
<project>
  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
</project>
```

“main” application class

- Best to locate it in a root package above other classes.
- **@SpringBootApplication** annotation
  - Includes @EnableAutoConfiguration, @ComponentScan and @Configuration
- Example:

```
@SpringBootApplication
public class MyApplication {
  public static void main(String[] args) {
    SpringApplication.run(MyApplication.class, args);
  }
}
```

“main” application class for console applications

```
@SpringBootApplication
public class MyApplication {
  public static void main(String[] args) {
    ApplicationContext ctx =
        SpringApplication.run(MyApplication.class, args);
    ctx.getBean(InitialClass.class).initialMethod();
  }
}
```

Exercises

Exercise: SpringWaterHeater

- Modify the WaterHeater application to use the Spring framework.
- Control whether the WaterHeater uses the standard or the efficient thermoregulator using a Spring profile.

* Alternative: SpringElectricalSystem.Best

Application

Application: Twitter

Modify the Twitter application to use the Spring framework.