**70. Spring Boot – Overview**

Spring boot solution:

* Make it easier to get started with Spring development
* Minimize the amount of manual configuration
  + Perform auto-configuration based on props files and JAR classpath
* Help to resolve dependency conflicts (Maven or Gradle)
* Provide an embedded HTTP server
  + Tomcat, Jetty, Undertow, …
  + Jar file includes your application code and includes the server
  + Apps can be run standalone
  + Run the Spring Boot app from the IDE or command-line

Spring Initializr (start.spring.io)

* Quickly create a starter Spring project
* Select your dependencies
* Creates a Maven / Gradle project
* Import the project into your IDE

Deploying Spring Boot Apps:

* Spring Boot apps can also be deployed in the traditional way
* Deploy WAR file to an external server: Tomcat, JBoss, WebSphere etc …

Note 1: Once you do Spring Boot configs then you make use of regular Spring coding

Spring Boot Demo development process:

1. Configure project at Spring initializr
2. Download the zip file
3. Unzip the file
4. Import Maven project into IDE

Maven Wrapper Files. mvnw allows you to run a Maven project. Two files provided:

* mvnw.cmd – for MS Windows
* mvnw.sh – Linux, Mac

If you already have Maven installed, you don’t need these files

pom.xml includes info that you entered in initializer. There is also springframework.boot plugin

Can also just use:

1. mvn package
2. mvn spring-boot:run

@SpringBootApplication annotation enables:

* Auto configuration (@EnableAutoConfiguration)
* Component scanning (@ComponentScan) of current package and subpackages
* Additional configuration (@Configuration) – able to register extra beans with @Bean or import other configuration classes

SpringApplication class is bootstrap your application:

SpringApplication.run(MycoolApplication.class, args)

Place your main application class in the root package above your other packages:

* This implicitly defines a base search package
  + Allows you to leverahe default component scanning
  + No need to reference package manually

You can explicitly list base packages to scan:

scanPackeages={“”,””}

By default, Spring boot will load properties from: application.properties. You can use this file in application using injection:

@Value(“${coach.name}”)

Private String coachName;

By default, Spring Boot will load static resources from “/static” directory.

Do not use source/main/webapp directory if your application is packaged as a JAR. Although this is a standard Maven directory, it works only with WAR packaging. It is silently ignored by most build tools if you generate a JAR.

Spring Boot includes auto-configuration for following template engines:

* FreeMarker
* Thymleaf
* Mustache

Spring Boot Starters a curated list of Maven dependencies. A collection of dependencies grouped together. Tested and verified by the Spring development team. Makes it much easier for the developer to get started with Spring. Spring boot provides: spring-boot-starter-web. Contains:

* Spring-web
* Spring-webmvc
* Hibernate-validator
* Json
* Tomcat and etc.

There are 30+ Spring Boot Starters from the Spring Development Team (luv2code.com/spring-boot-starters)