

Session 24

Spring Framework Introduction

1

Reading & Reference

■ Reading

dev.to/lechatthecat/how-to-use-spring-boot-java-web-framework-with-intellij-idea-202p

<http://engineering.pivotal.io/post/must-know-spring-boot-annotations-controllers/>

<http://engineering.pivotal.io/post/spring-for-normal-people/>

■ Reference

■ Spring home page

spring.io/

■ Spring 4.3 Reference Documentation

docs.spring.io/autorepo/docs/spring/4.3.0.RELEASE/spring-framework-reference/pdf/spring-framework-reference.pdf

© Robert Kelly, 2018

2

Lecture Objectives

- Understand the structure of the Spring controller
- Become familiar with Spring controller annotation

© Robert Kelly, 2018

3

IDE Support

- Thymeleaf integrates well with Spring
 - We defer Thymeleaf until after we cover Spring Controller
 - Thymeleaf does not integrate well with NetBeans, so examples use IntelliJ
- If you do not use IntelliJ, you can download from

www.jetbrains.com/idea/download/

© Robert Kelly, 2018

4

Interesting Quote

- Like many of my friends and fellow engineers, I have recently started working on some Spring projects, specifically using Spring Boot. It has been several years since I have worked with Spring, and some things were immediately apparent:
- I had no idea how to do things The Spring Way™.
- There is a *ton* of documentation.
- Almost all the documentation assumes you already know everything about Spring in order to understand it.

Source: Ian Fisher: Pivotal Engineering Post

© Robert Kelly, 2018

5

What is Tedious About Using Servlets?

- One servlet for each form to be processed
- Servlet controller code that either updates a bean or forwards to a view (or both)
- Bean actions
 - Storing data
 - Setting up visibility
 - Etc.

A good framework can take some of the tedium out of development

© Robert Kelly, 2018

6

Spring Overview

- Open source, easy to use
- MVC Framework (tightly coupled to Servlet API)
- Integrates well with other tools
- Enables development of Java EE systems
- Initially developed in 2003
- Currently well supported and expanding rapidly

© Robert Kelly, 2018

7

Spring Versions

- Version 3.x - mappings are provided in xml files (similar to mappings found in web.xml and tld files)
- Version 4.x (4.3 is final version in this series) - mappings use annotations
- Version 5.x

© Robert Kelly, 2018

8

Inversion of Control

- Design principle
- Custom written portions of a computer program receive the flow of control from a generic framework
- "Inversion" highlights the contrast with use of reusable libraries (e.g., APIs) that are called from a custom written control module
- Similar to event-driven programming

© Robert Kelly, 2018

9

Spring Architecture

- Modular
- You can select which modules to use
- Modules
 - Data access (e.g., JDBC, object-relational mapping, JPA, etc.)
 - Web
 - Core
 - Messaging
 - Test
 - More

© Robert Kelly, 2018

10

Spring Boot

- Pre-configured solution
- Requires you to provide exceptions rather than a complete configuration
- Integrates with Maven
- Features
 - Embedded application server
 - No code generation
 - No requirement for XML configuration files
 - Spring initializer that will configure, download, and install your project files

<https://start.spring.io/>

© Robert Kelly, 2018

11

Maven

- Build automation tool
- Collects all jars on classpath and builds a single runnable jar
- Apache Software Foundation
- XML file describes
 - How software is built
 - Dependencies on external modules
 - Build order
 - Required plug-ins
 - Directories
- Projects configured using a Project Object Model (pom.xml)

Sample pom.xml file

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <version>1.0</version>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies> </project>
```

© Robert Kelly, 2018

12

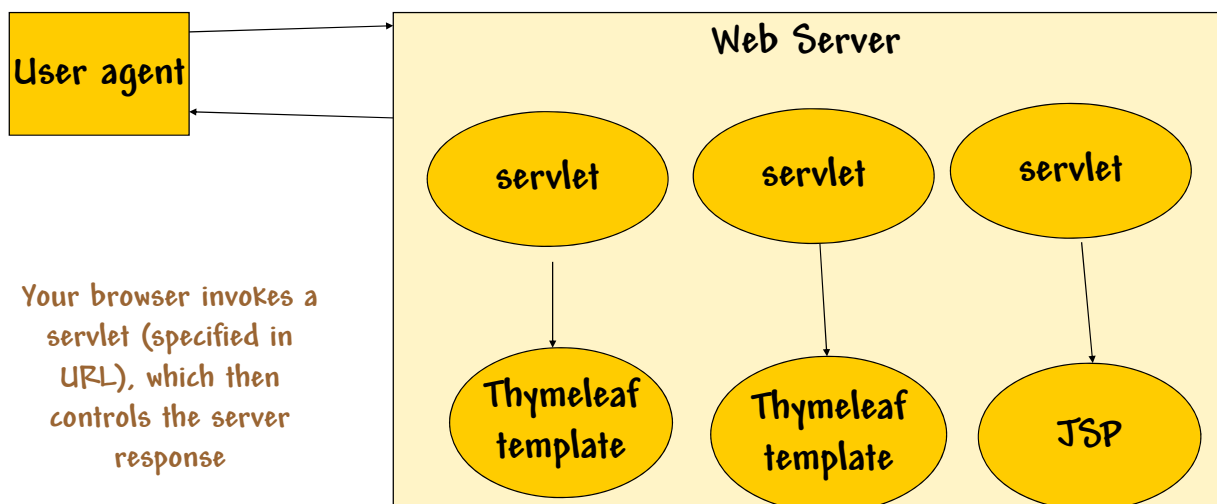
Mapping Approach

- Relationship between forms, servlets, views, and beans are specified in mapping files, not in the code
- Recall mapping done in web.xml and tld files
- Spring versions
 - Version 3.x - mappings are provided in xml files (similar to mappings found in web.xml and tld files)
 - Version 4.x - mappings use annotations

© Robert Kelly, 2018

13

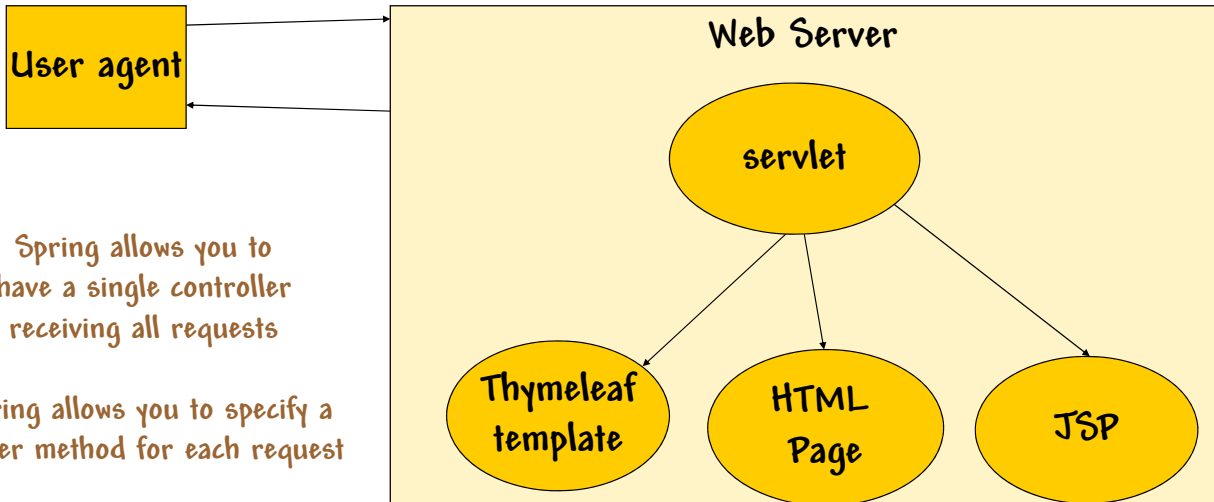
Standard Servlet Control Flow



© Robert Kelly, 2018

14

Servlet Mapping



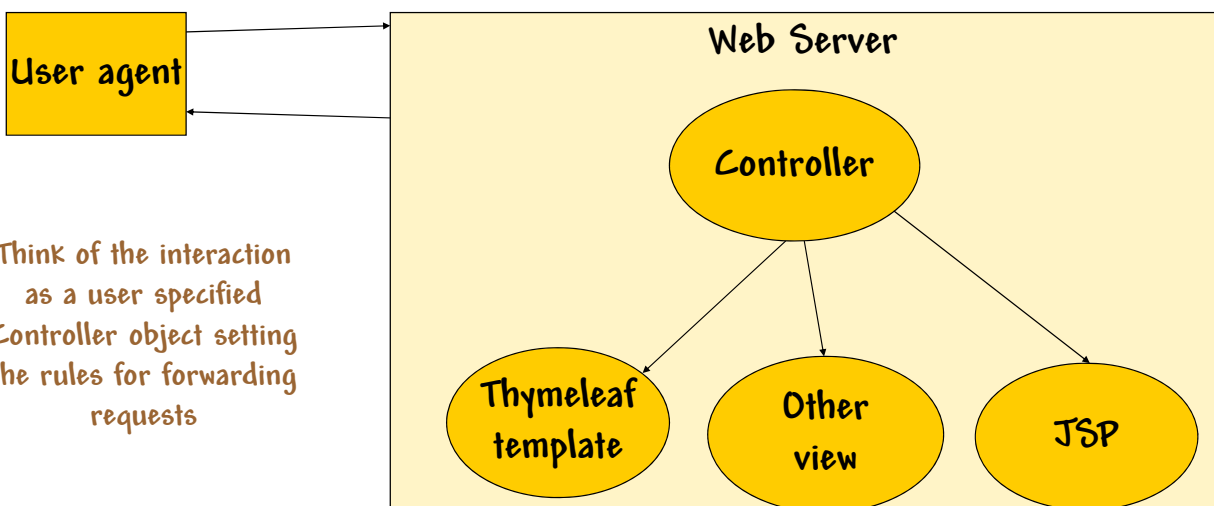
Spring allows you to have a single controller receiving all requests

Spring allows you to specify a server method for each request

© Robert Kelly, 2018

15

Spring MVC



Think of the interaction as a user specified Controller object setting the rules for forwarding requests

© Robert Kelly, 2018

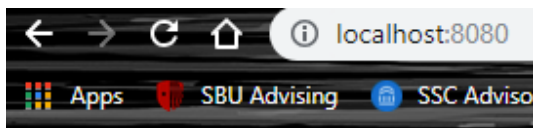
16

Your First Spring Boot

- Step-by-step instructions in

dev.to/lechatthecat/how-to-use-spring-boot-java-web-framework-with-intellij-idea-202p

- We show the code, then look at how it works



After we build the application, and enter localhost:8080 in the browser, this page appears

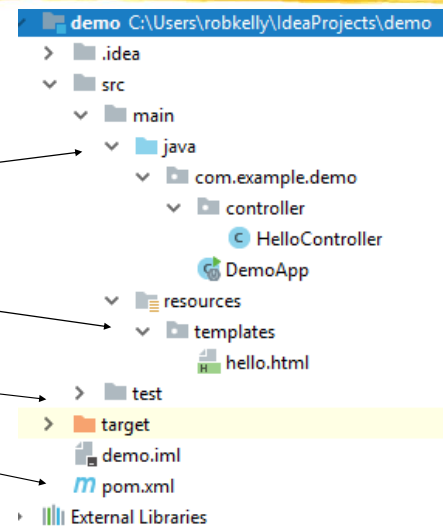
Hello CSE336!

© Robert Kelly, 2018

17

Example

- IntelliJ project
- Standard structure generated by Maven
- Contains
 - Java code
 - Templates (e.g., Thymeleaf)
 - Java tests
 - Project Object Model



© Robert Kelly, 2018

18

Example - Project Object Model ...

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>
  <artifactId>demo</artifactId>
  <version>1.0-SNAPSHOT</version>

  ...
```

An artifact is a file (e.g., JAR) that gets deployed to a Maven repository

© Robert Kelly, 2018

19

... Example - Project Object Model ...

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.0.2.RELEASE</version>
</parent>

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

pom.xml inherits from a super-pom

Maven automatically downloads dependencies

© Robert Kelly, 2018

20

... Example - Project Object Model

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

</project>
```

© Robert Kelly, 2018

21

Annotation Recap

- Annotations can be defined to have elements
- Elements appear in parentheses following the annotation
- Elements are a collection of comma-separated, name-value pairs
- If the element name is missing, value is the default element name

```
@RequestMapping(method = RequestMethod.GET)
```

```
@RequestMapping("/welcome")
```

Is the same as

```
@RequestMapping(value = "/welcome")
```

© Robert Kelly, 2018

22

Spring Boot Application

- Runs a SpringApplication
- Specifies the main class
- @SpringBootApplication equivalent to
 - @Configuration,
 - @EnableAutoConfiguration, and
 - @ComponentScan

```
import
org.springframework.boot.SpringApplication;

import
org.springframework.boot.autoconfigure.Spr
ingBootApplication;

@SpringBootApplication

public class DemoApp {

    public static void main(String[] args) {

        SpringApplication.run(DemoApp.class, args);

    }

}
```

© Robert Kelly, 2018

23

What Happens When Application Runs?

- Configuration resolves
- Application server (e.g., Tomcat) runs
- Ready to accept requests to the server port (according to controller rules)

```
main] s.w.s.m.a.RequestMappingHandlerMapping : mapped "[{}],methods={GET}" onto public java.lang.S
main] s.w.s.m.a.RequestMappingHandlerMapping : Mapped "[{/error}]" onto public org.springframework
main] s.w.s.m.a.RequestMappingHandlerMapping : Mapped "[{/error},produces={text/html}]" onto publi
main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/webjars/**] onto handler of type
main] o.s.w.s.handler.SimpleUrlHandlerMapping : Mapped URL path [/**] onto handler of type [class o
main] o.s.j.e.a.AnnotationMBeanExporter      : Registering beans for JMX exposure on startup
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context
main] com.example.demo.DemoApp                : Started DemoApp in 1.278 seconds (JVM running for 2
exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]     : Initializing Spring FrameworkServlet 'dispatcherSer
exec-1] o.s.web.servlet.DispatcherServlet       : FrameworkServlet 'dispatcherServlet': initializatio
exec-1] o.s.web.servlet.DispatcherServlet       : FrameworkServlet 'dispatcherServlet': initializatio
```

© Robert Kelly, 2018

24

Example - Controller

- Controller annotation (`@Controller`) indicates that the annotated class is a Web controller
- Controller class is auto-detected
- Typically use in combination with annotated handler methods based on RequestMapping annotation

```
package com.example.demo.controller;  
import org.springframework.stereotype.Controller;  
import  
org.springframework.web.bind.annotation.GetMapping;  
@Controller  
public class HelloController {  
    @GetMapping  
    public String getHello() {  
        return "hello";    } } }
```

Shortcut for `@RequestMapping` with a GET parameter

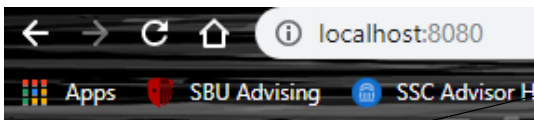
© Robert Kelly, 2018

25

What Happens if We Change the Controller

- Text (not html) is sent to browser

```
package com.example.demo.controller;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.*;  
@RestController  
public class HelloController {  
    @GetMapping  
    public String getHello() {  
        return "hello";    } } }
```



hello

© Robert Kelly, 2018

26

Controller Types

- Each controller contains annotation describing what type of controller it is
- `@Controller`
 - often used to serve Web pages
 - Controller methods will return a String that indicates which template to render or which route to redirect to
- `@RestController`
 - Serves text (JSON, XML, etc.)
 - Controller methods return an object that will be serialized to the specified format

© Robert Kelly, 2018

27

Track Initializer

- To build a Spring project in IntelliJ, you can use Spring Initializr
- Set Web dependencies

The screenshot shows the IntelliJ IDEA 'New Project' dialog. On the left, a list of project types includes 'Spring Initializr', which is highlighted. The 'Project Metadata' section is active, showing fields for Group (com.example), Artifact (track), Type (Maven POM), Language (Java), Packaging (Jar), Java Version (8), Version (0.0.1-SNAPSHOT), Name (track), Description (Demo project for Spring Boot), and Package (com.example.track). On the right, the 'Dependencies' section is visible, with 'Web' selected and checked, and other options like 'Reactive W', 'Rest Repos', and 'SQL' unchecked.

© Robert Kelly, 2018

28

Are We on Track?

- Build your first Spring application by following the steps in

dev.to/lechatthecat/how-to-use-spring-boot-java-web-framework-with-intellij-idea-202p

- Or by using Spring Initializr
- Run the project with a
- @RestController and @Controller

