Session 26

Spring
Model / View

1

Reading & Reference

- Reading

  https://www.baeldung.com/spring-mvc-model-model-map-model-view

  https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html

- 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

    https://docs.spring.io/spring-framework/docs/4.3.4.RELEASE/javadoc-api/index.html?org/springframework/web/bind/annotation/RequestMapping.html

2

© Robert Kelly, 2018

Lecture Objectives

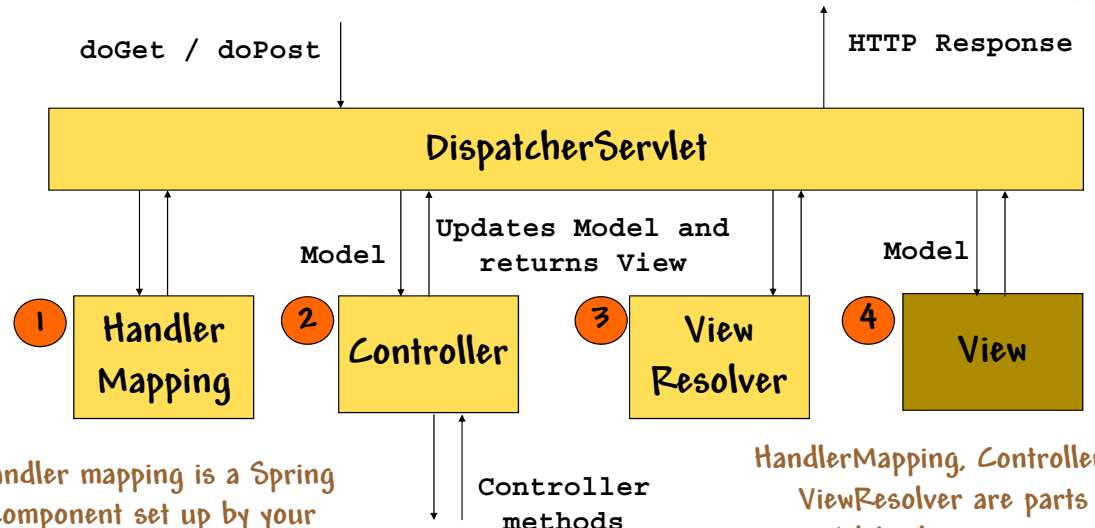
- Understand the structure of the Spring Model and View
- Become familiar with the use of an IDE to develop a simple Spring/Thymeleaf application

Spring is far too complex to cover all its features in a few class sessions. We just cover enough for you to understand its architecture

© Robert Kelly, 2018

3

Spring MVC Control Flow



Handler mapping is a Spring component set up by your controller annotation

HandlerMapping, Controller, and ViewResolver are parts of WebApplicationContext

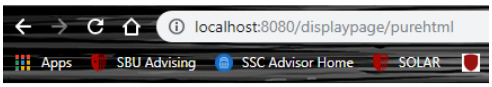
© Robert Kelly, 2018

Diagram from tutorialspoint.com

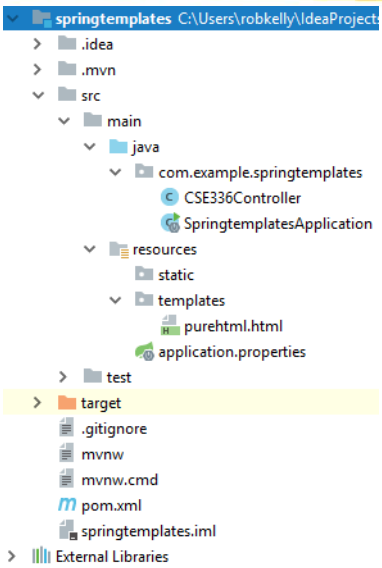
4

Review Spring

- Simple Spring project
  - Controller class
  - Main class
  - Html file
- Entering localhost:8080/displaypage/purehtml displays the page below



Hello CSE336



© Robert Kelly, 2018

5

Example

- URI request (localhost:8080/displaypage/purehtml) is routed to Controller method
- Method returns the name of the resource template (e.g., purehtml)
- Spring
  - looks for purehtml.html in templates directory
  - Returns html to browser

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;


@Controller
@RequestMapping(value="/displaypage")
public class CSE336Controller {

    @RequestMapping(value="/purehtml", method=RequestMethod.GET)
    String displayPage() {
        return "purehtml";
    }
}
```

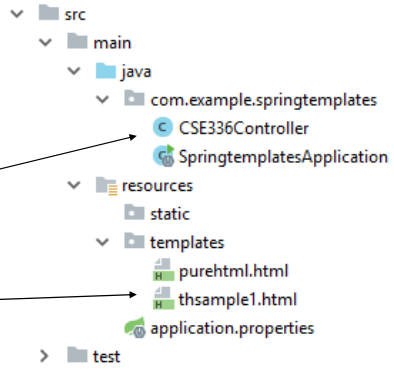
Convention in Spring MVC is to return the string name of the template file, less the file extension

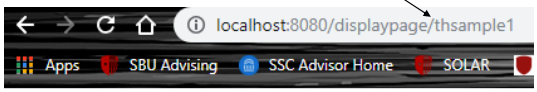
© Robert Kelly, 2018

6

Simple Thymeleaf Example

- We make just a few simple changes to the html example above to produce our first Thymeleaf template
- We
  - Change URI path in CSE336Controller
  - Add thsample1





Hello CSE336

© Robert Kelly, 2018

7

Thymeleaf Template

- Template is almost the same as original html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org" >
<head lang="en">
  <meta http-equiv="Content-Type"
     content="text/html; charset=UTF-8">
  <title>Thymeleaf</title>
</head>
<body>
<h1>Hello CSE336</h1>
</body>
</html>
```

© Robert Kelly, 2018

8

Compare Thymeleaf Template with Browser HTML

Server html

```
<!DOCTYPE html>
<html
xmlns:th=http://www.thymeleaf.org >
<head lang="en">
    <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">
    <title>Thymeleaf</title>
</head>
<body>
<h1>Hello CSE336</h1>
</body>
</html>
```

Browser html

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">
    <title>Thymeleaf</title>
</head>
<body>
<h1>Hello CSE336</h1>
</body>
</html>
```

Thymeleaf processes the template to generate html to send to the client (and display in the browser)

© Robert Kelly, 2018

9

Model

- The Model object is a generalized version of the shared scopes you have used (Session, ServletContext, etc.)
- Your Controller method can take a Model as a parameter
- Contains name/value pairs
- Usually accessed as a ModelMap

© Robert Kelly, 2018

10

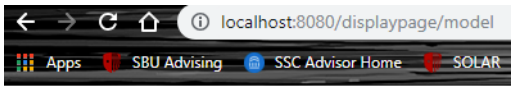
Example

- Let's pass a message from the controller to the Thymeleaf template

```
@RequestMapping(value="/model", method=RequestMethod.GET)
  String getMessage(Model m) {
    m.addAttribute("message", "Model Message to CSE336");
    return "thsample2";
  }
```

thsample2.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
...
<body>
<h1 th:text=${message}></h1>
</body>
</html>
```



Model Message to CSE336

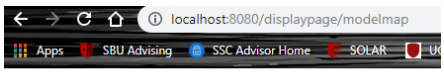
© Robert Kelly, 2018

11

ModelMap

- Also used to pass values to a view
- Pass a collection of values and treat those as within a Map

```
@RequestMapping(value="/modelmap", method=RequestMethod.GET)
   String getMessage(ModelMap m) {
       m.addAttribute("message1", "ModelMap Message to CSE336");
       m.addAttribute("message2", "Welcome to Thymeleaf");
       return "thsample3";
   }
```



ModelMap Message to CSE336

Welcome to Thymeleaf

thsample3.html

```
</head>
<body>
<h1 th:text="${message1}"></h1>
<h1 th:text="${message2}"></h1>
</body>
</html>
```

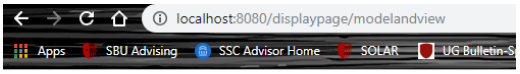
© Robert Kelly, 2018

12

ModelAndView

- Used to pass both values and a view name

```
@RequestMapping(value="/modelandview", method=RequestMethod.GET)
    public ModelAndView passParameters() {
        ModelAndView m = new ModelAndView("thsample4");
        m.addObject("message1", "ModelAndView Message to CSE336");
        m.addObject("message2", "Welcome to  ModelAndView");
        return m;
    }
```



ModelAndView Message to CSE336

Welcome to ModelAndView

thsample4.html

```
</head>
<body>
<h1 th:text="${message1}"></h1>
<h1 th:text="${message2}"></h1>
</body>
</html>
```

© Robert Kelly, 2018

13

@RequestParam

- Annotation that binds a method parameter with a Web request parameter
- If method parameter is a Map<String, String> and a parameter name is not specified, the map parameter is populated with all request parameter names and values

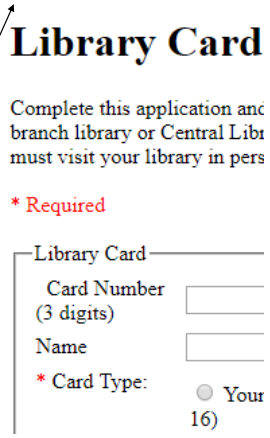
```
passParameters(@RequestParam("nickname") String nickname)
```

© Robert Kelly, 2018

14

Are We On Track?

- Download an html file from the class Web site
- https://www3.cs.stonybrook.edu/~cse336/CSE336-Project-SpringThymeleafTrack.html
- Set up a Spring project with the html as a Thymeleaf template and a controller that will receive requests from the form
- Inject the nickname parameter into the controller method parameter
- Pass the nickname parameter + message (e.g., Welcome back, Allonzo) to the view
- Display the message at the top of the page



Use the original html in your browser to send the request to your server

© Robert Kelly, 2018

15

Were We On Track?

```
@RequestMapping(value="/requestparam", method=RequestMethod.GET)
   public ModelAndView passParameters(
            @RequestParam("nickname") String nickname) {
       ModelAndView m = new ModelAndView("thtrack1");
       String message;
       if (nickname.equals("")) {
           message = "Please enter your name";
       }
       else {
           message="Welcome back, " + nickname;
       }
       m.addObject("error", message);
       return m;
   }
```

Controller

thtrack1.html

```
<body>

<h3 th:text="${error}">This is a test</h3>

<form method="put"
action="http://localhost:8080/displaypage/springtrack" id="form1">
```

© Robert Kelly, 2018

16