Western Norway University of Applied Sciences

Web Development: Spring MVC (Action-based)

Created by Lasse Jenssen

Home

Agenda: Using Spring Web MVC

- Action Based Web Framework.
- Introduction to Spring Web MVC.
- Introduction to Thymeleaf.
- Rewrite our Shopping List Application to use:
    - Spring Boot
    - Spring Web MVC
    - Thymeleaf

Sources/ Syllabus:

- https://www.baeldung.com/spring-controllers#Overview
  - Section 2, 3, 4 (only part regarding web.xml),

    5 (only part regarding DispatcherServlet XML file) and 6
  - See "demo-spring-webmvc.zip" for reference.
- https://www.baeldung.com/thymeleaf-in-spring-mvc
  - Section 1, 3, 5 and 6
  - See "demo-05-spring-web.zip" for reference.
- https://www.baeldung.com/spring-boot-internationalization
  - All sections

Action-based Web Framework

Spring Web MVC

- Spring Web MVC is the original web framework **built on the Servlet API** (included in the Spring Framework from the very beginning).
- Spring Web MVC **Documentation** (you do not need to read this) (https://docs.spring.io/spring-framework/docs/current/reference/html/web.html#mvc)
- New: **Spring WebFlux**: Reactive Web programming (not a part of this course)
- Combines all the advantages of the MVC pattern with the convenience of Spring.

Action-based Web Framework

Spring Web MVC

- Spring implements MVC with the front controller pattern using its **DispatcherServlet**.

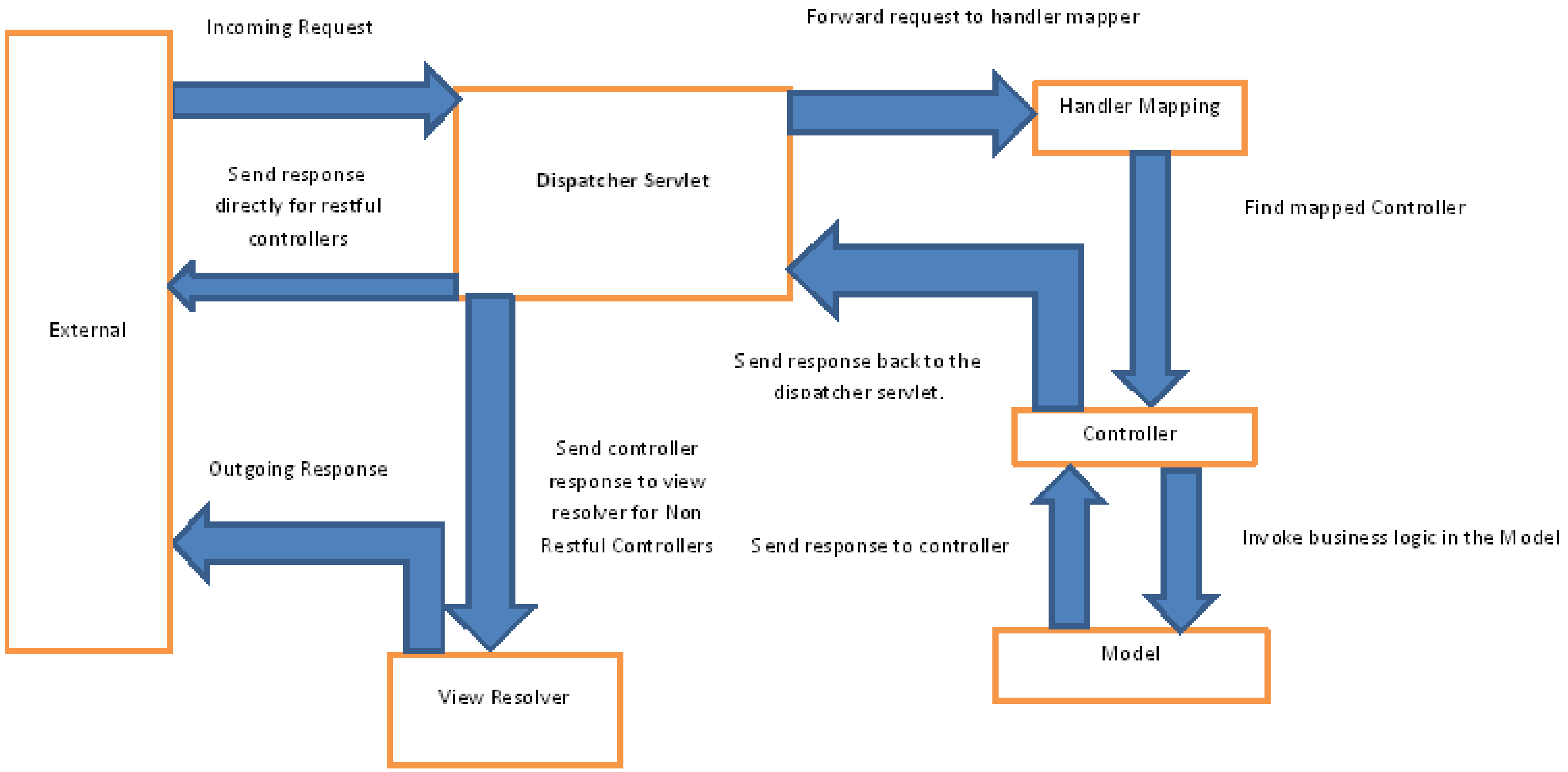

Incoming Request
Forward request to handler mapper
Handler Mapping
Dispatcher Servlet
Send response directly for restful controllers
Find mapped Controller
External
Send response back to the dispatcher servlet.
Controller
Outgoing Response
Send controller response to view resolver for Non Restful Controllers
Send response to controller
Invoke business logic in the Model
Model
View Resolver

Fig 1 MVC Architecture flow

Maven Dependencies

Web Frameworks: Spring Web MVC (Action based)

```
<dependency>
   <groupid>org.springframework</groupid>
   <artifactid>spring-web</artifactid>
   <version>${org.springframework.version}</version>
</dependency>

<dependency>
   <groupid>org.springframework</groupid>
   <artifactid>spring-webmvc</artifactid>
   <version>${org.springframework.version}</version>
</dependency>
```

- In "demo-05-spring-web" we'll use Spring Boot (see next slide)
- I have uploaded a small sample application where I use Core Spring.

  See: demo-spring-webmvc.zip (not important for any exam)

Spring Boot: Maven Dependencies

Web Frameworks: Spring Web MVC (Action based)

```xml
<parent>
   <groupid>org.springframework.boot</groupid>
   <artifactid>spring-boot-starter-parent</artifactid>
   <version>2.7.1</version>
   <relativepath></relativepath> <!-- lookup parent from repository -->
</parent>
...
<dependencies>
   ...
   <dependency>
      <groupid>org.springframework.boot</groupid>
      <artifactid>spring-boot-starter-web</artifactid>
   </dependency>
   ...
</dependencies>
```

Demo: demo-05-spring-web

- Smal demo applications keeping track of Inventory (Items).
- Based on Spring Boot.
- Using **Thymeleaf** library: an XML/XHTML/HTML5 template engine.
- Run by (either):
  - ***mvn clean package spring-boot:run***
  - ***java -jar target/demo-05-spring-web-0.0.1-SNAPSHOT.jar***
- *Code: demo-05-spring-web.zip (see course overview)*

pom.xml

Project: demo-05-spring-web

```xml
< ?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.or
   <modelversion>4.0.0</modelversion>

   <parent>
      <groupid>org.springframework.boot</groupid>
      <artifactid>spring-boot-starter-parent</artifactid>
      <version>2.7.4</version>
      <relativepath></relativepath> <!-- lookup parent from repository -->
   </parent>

   <groupid>no.hvl.dat152</groupid>
   <artifactid>demo-05-spring-web</artifactid>
   <version>0.0.1-SNAPSHOT</version>
   <name>demo-05-spring-web</name>
   <description>Demo application with ShoppingList (Items example)</descriptio

   <properties>
      <java.version>11</java.version>
   </properties>
   ...
</project>
```

pom.xml

Project: demo-05-spring-web

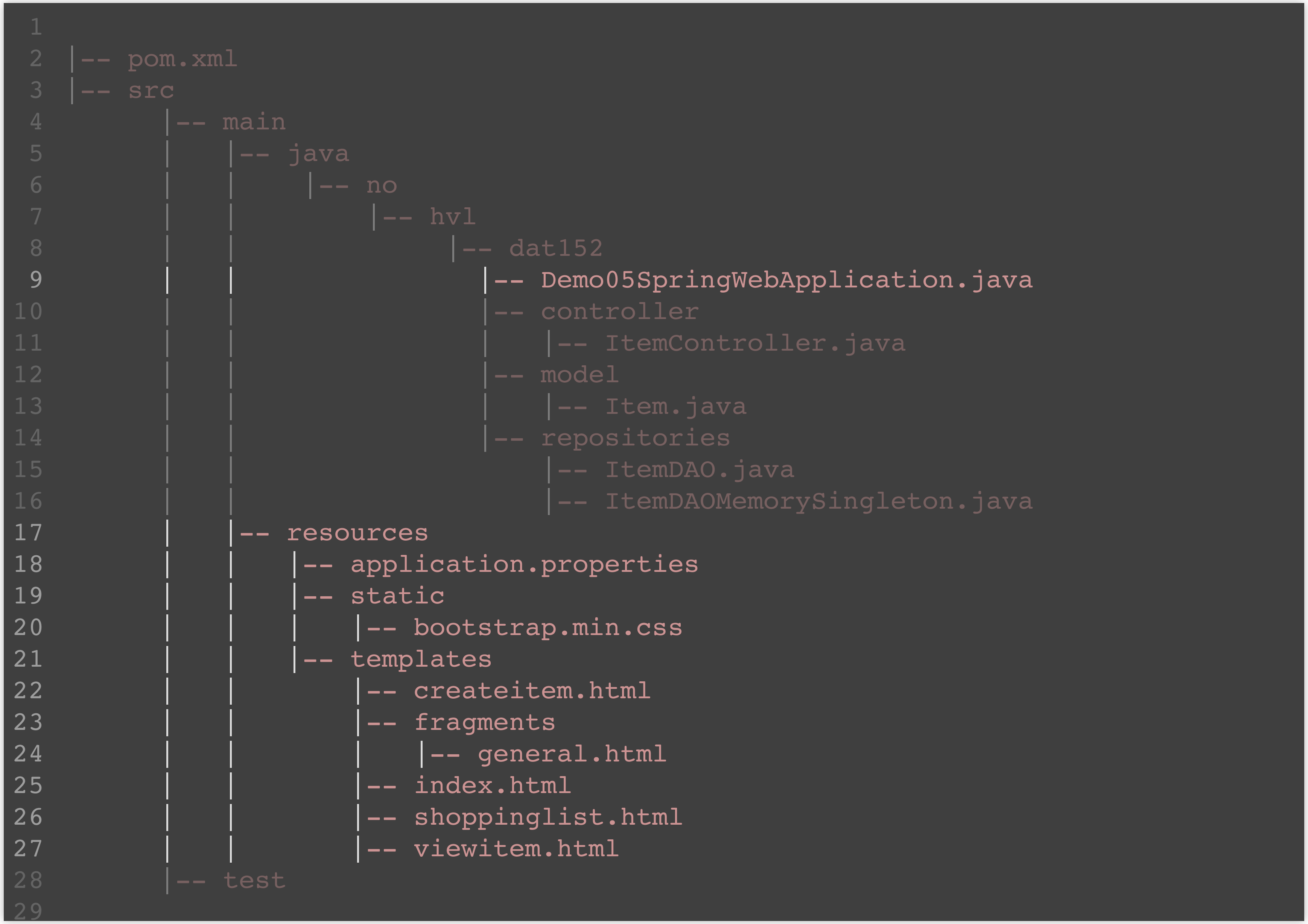
```
...
<dependencies>
    <dependency>
        <groupid>org.springframework.boot</groupid>
        <artifactid>spring-boot-starter-web</artifactid>
    </dependency>

    <dependency>
        <groupid>org.springframework.boot</groupid>
        <artifactid>spring-boot-starter-thymeleaf</artifactid>
    </dependency>

    <dependency>
        <groupid>org.springframework.boot</groupid>
        <artifactid>spring-boot-starter-test</artifactid>
        <scope>test</scope>
    </dependency>
</dependencies>
...
```

pom.xml

Project: demo-05-spring-web

```
...
<build>
   <plugins>
      <plugin>
         <groupid>org.springframework.boot</groupid>
         <artifactid>spring-boot-maven-plugin</artifactid>
      </plugin>
   </plugins>
</build>
...
```

Project: demo-05-spring-web



```

|-- pom.xml
|-- src
          |-- main
          |      |-- java
          |      |        |-- no
          |      |              |-- hvl
          |      |                      |-- dat152
          |      |                         |-- Demo05SpringWebApplication.java
          |      |                         |-- controller
          |      |                         |      |-- ItemController.java
          |      |                         |-- model
          |      |                         |      |-- Item.java
          |      |                         |-- repositories
          |      |                                |-- ItemDAO.java
          |      |                                |-- ItemDAOMemorySingleton.java
          |      |-- resources
          |      |      |-- application.properties
          |      |      |-- static
          |      |      |      |-- bootstrap.min.css
          |      |      |-- templates
          |      |             |-- createitem.html
          |      |             |-- fragments
          |      |             |      |-- general.html
          |      |             |-- index.html
          |      |             |-- shoppinglist.html
          |      |             |-- viewitem.html
          |-- test

```

Project: demo-05-spring-web

src/main/java/no/hvl/dat152/Demo05SpringWebApplication.java

```
@SpringBootApplication
public class Demo05SpringWebApplication {

   public static void main(String[] args) {
      SpringApplication.run(Demo05SpringWebApplication.class, args);
   }

}
```

Project: demo-05-spring-web

src/main/java/no/hvl/dat152/controller/ItemController.java

```
@Controller
public class ItemController {

   @RequestMapping(value = "/", method = RequestMethod.GET)
   public String viewShoppingDefault() {
      return "index";
   }

   ...
}
```

Project: demo-05-spring-web

src/main/resources/templates/index.html

See project: demo-05-spring-web.zip

```
< !DOCTYPE HTML>
< html xmlns:th="http://www.thymeleaf.org">
< head th:replace="fragments/general.html :: headerfiles (title='Home Page')">
< /head>
< body>
<div class="container">
    <div class="page-header" id="banner">
       <div th:replace="fragments/general.html :: header"></div>
       <div th:replace="fragments/general.html :: menu"></div>
    </div>

    <p>Welcome to the Shopping List at HVL and dat152 lecture.</p>

  <div th:replace="fragments/general.html :: footer"></div>

</div>
< /body>
< /html>
```

Project: demo-05-spring-web

Model: a placeholder for model attributes

```
@Controller
public class ItemController {

   ...

   @RequestMapping(value = "/viewitems", method = RequestMethod.GET)
   public String viewShoppingList(Model model) {

     final List< Item> items = ItemDAOMemorySingleton.getInstance().findAllIte

     model.addAttribute("items", items);

     return "shoppinglist";
   }

   ...
}
```

Project: demo-05-spring-web

@PathVariable

```java
@Controller
public class ItemController {

    ...

    @RequestMapping(value = "/viewitem/{id}", method = RequestMethod.GET)
    protected String viewItem(@PathVariable String id, Model model) {

            final Item item = ItemDAOMemorySingleton.getInstance().findItem(id);

            model.addAttribute("item", item);

            return "viewitem";
    }

    ...
}
```

Project: demo-05-spring-web

POST request and @RequestParameter

```java
@Controller
public class ItemController {
    ...

    @RequestMapping(value = "/createitem", method = RequestMethod.GET)
    protected String createItem(Model model) {
        final String id = ItemDAOMemorySingleton.getInstance().getNextId();
        model.addAttribute("id",id);
        return "createitem";
    }

    @RequestMapping(value = "/createitem", method = RequestMethod.POST)
    protected String createItem(@RequestParam String id,
                                @RequestParam String name,
                                @RequestParam Double price,
                                @RequestParam String description) {

        final Item newItem = new Item(id, name, price, description);
        ItemDAOMemorySingleton.getInstance().createItem(newItem);

        return "redirect:viewitems";
    }
}
```

Introduction to Thymeleaf

- A Java **template** engine for processing and creating HTML, XML, JavaScript, CSS and text.
- A full featured supstitude for JSP.
- Display internationaliazation (i18n) messages from message files (we'll see this later).
- Provides full integration with Spring Framework.

Spring Boot: Maven Dependencies

Introduction to Thymeleaf

```xml
...
   <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>

      <dependency>
         <groupid>org.springframework.boot</groupid>
         <artifactid>spring-boot-starter-test</artifactid>
         <scope>test</scope>
      </dependency>
   </dependencies>
   ...
```

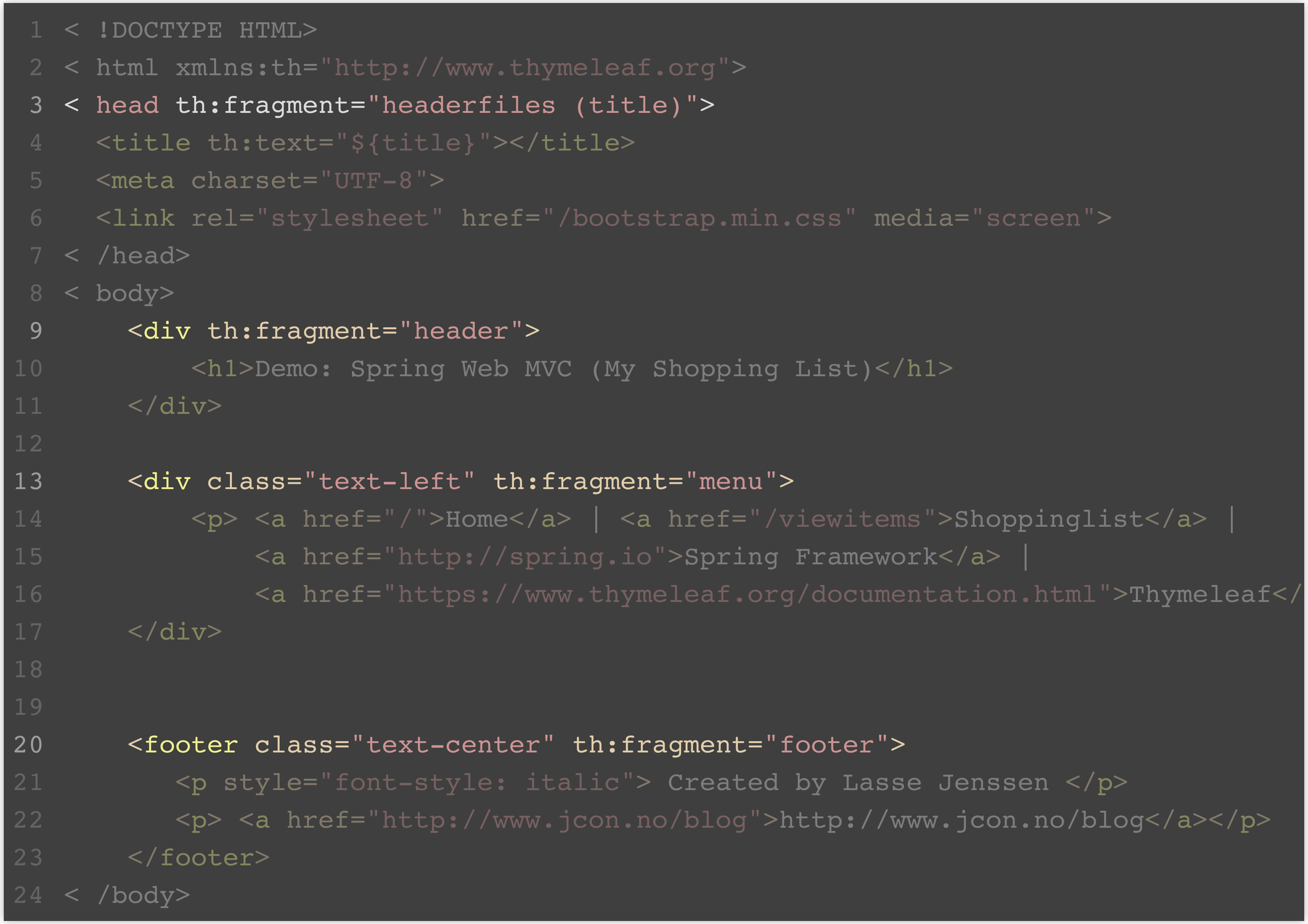
Core Spring: Maven Dependencies

Introduction to Thymeleaf

```xml
<dependency>
   <groupid>org.thymeleaf</groupid>
   <artifactid>thymeleaf</artifactid>
   <version>3.0.11.RELEASE</version>
</dependency>
<dependency>
   <groupid>org.thymeleaf</groupid>
   <artifactid>thymeleaf-spring5</artifactid>
   <version>3.0.11.RELEASE</version>
</dependency>
```

Thymeleaf Template: *src/main/resources/templates/fragments/general.html*

th:fragment="[fragmentname] ([param1][,param2])"



```
< !DOCTYPE HTML>
< html xmlns:th="http://www.thymeleaf.org">
< head th:fragment="headerfiles (title)">
  <title th:text="${title}"></title>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="/bootstrap.min.css" media="screen">
< /head>
< body>
    <div th:fragment="header">
        <h1>Demo: Spring Web MVC (My Shopping List)</h1>
    </div>

    <div class="text-left" th:fragment="menu">
        <p> <a href="/">Home</a> | <a href="/viewitems">Shoppinglist</a> |
            <a href="http://spring.io">Spring Framework</a> |
            <a href="https://www.thymeleaf.org/documentation.html">Thymeleaf</
    </div>


    <footer class="text-center" th:fragment="footer">
       <p style="font-style: italic"> Created by Lasse Jenssen </p>
       <p> <a href="http://www.jcon.no/blog">http://www.jcon.no/blog</a></p>
    </footer>
< /body>
```

Simple Expressions

Introduction to Thymeleaf

- Variables: ${...}
- Messages: #{...}
- Link URLs: @{...}

Thymeleaf Template: *src/main/resources/templates/viewitem.html*

Introduction to Thymeleaf

```
< html>
   ....
< body>
   <div class="container">
      <div class="page-header" id="banner">
         <div th:replace="fragments/general.html :: header"></div>
         <div th:replace="fragments/general.html :: menu"></div>
      </div>

      < table cellpadding="10">
         ...
      < /table>

      <div th:replace="fragments/general.html :: footer"></div>
   </div>
< /body>
< /html>
```

Thymeleaf Template: *src/main/resources/templates/viewitem.html*

Introduction to Thymeleaf

```html
<table cellpadding="10">
   <tbody><tr>
      <td width="100px">Id:</td>
      <td><b margin-left="15px" th:text="${item.id}"></b></td>
   </tr>
   <tr>
      <td>Name:</td>
      <td><b th:text="${item.name}"></b></td>
   </tr>
   <tr>
      <td>Price:</td>
      <td><b th:text="${item.price}"></b></td>
   </tr>
   <tr>
      <td>Description:</td>
      <td><b th:text="${item.description}"></b></td>
   </tr>
</tbody></table>
```

th:replace vs th:insert vs th:include

Introduction to Thymeleaf

- **Replace**: substitute the host tag by the fragment's
- **Insert**: insert the specified fragment as the body of its host tag including the fragment tag
- **Include**: insert the specified fragment as the body of its host tag but excluding the fragment tag.

Demo: demo-05-spring-web

- Same functionallity as "demo-01".
- FrontController: Controlled by Spring (DispatcherServlet).
- *Code: demo-05-spring-web.zip (see course overview).*
- *Let's look at the code.*

Introduction to ErrorHandling in Spring Boot

- General error handling do not prevent you from writing robust code.
- Disable Whitelabel error page and set error path (application.properties):

```
# Error handling
server.error.whitelabel.enabled=false
server.error.path=/error
```

Make your code robust

Introduction to ErrorHandling in Spring Boot

```java
@RequestMapping(value = "/viewitem/{id}", method = RequestMethod.GET)
protected String viewItem(@PathVariable String id, Model model,
                           RedirectAttributes redirectAttrs) {

      final Item item = ItemDAOMemorySingleton.getInstance().findItem(id);

      if (item == null) {
         redirectAttrs.addFlashAttribute("errormsg", "Item with id " +id+ " no
         return "redirect:/viewitems";
      }

      model.addAttribute("item", item);

      return "viewitem";
}
```

Why do we need a **FlashAttribute**?

- **RequestAttributes** won't survive a redirection across different controllers.
- **SessionAttributes** will last for the entire session even after the form submission is over.
- **FlashAttributes** remain available for the subsequent request after redirect, and then they're gone.

Added to src/main/resources/shoppinglist.html

Introduction to ErrorHandling in Spring Boot

```
<div th:if="${errormsg}">
   <p th:text="${errormsg}"></p>
</div>
```

Adding general error handling

Introduction to ErrorHandling in Spring Boot

```java
@Controller
public class ItemErrorController implements ErrorController {

    @RequestMapping(path = "/error")
    public String handleError(RedirectAttributes redirectAttrs) {

        String msg = "An error occured. Please try again. If the error remains,

        redirectAttrs.addFlashAttribute("errormsg", msg);

        return "redirect:/viewitems";
    }
}
```

src/main/resources/messages.properties

Internationalization with Spring and Thymeleaf

```properties
label.heading=My Shopping List
label.main-text=Welcome to the Shopping List at HVL and dat152 lecture

label.name=Name
label.price=Price
label.description=Description
label.button-new-item=New Item
label.create-item=Create Item

lang.change=Change Language
lang.uk=English(UK)
lang.no=Norwegian(NO)
```

src/main/resources/messages_no.properties

Internationalization with Spring and Thymeleaf

```
label.heading=Min Shopping liste
label.main-text=Velkommen to HVL og DAT152 sin Shopping liste

label.name=Navn
label.price=Pris
label.description=Beskrivelse
label.button-new-item=Ny gjenstand
label.create-item=Lag ny gjenstand

lang.change=Endre spraak
lang.uk=Engelsk(UK)
lang.no=Norsk(NO)
```

Refactored: fragments/general.html: Menu: with option box to choose lang

Internationalization with Spring and Thymeleaf

```
<div class="text-left" th:fragment="menu">
   <p> <a href="/">Home</a> | <a href="/viewitems">Shoppinglist</a> |
       <a href="spring.io">Spring Framework</a> |
       <a href="https://www.thymeleaf.org/documentation.html">Thymeleaf</a> |

       <span th:text="#{lang.change}"></span>:
       <select id="locales">
        <option value=""></option>
        <option value="uk" th:text="#{lang.uk}"></option>
        <option value="no" th:text="#{lang.no}"></option>
       </select>
   </p>
</div>
```

Refactored: fragments/general.html: HEAD
Added jQuery script to set locale after selected

Internationalization with Spring and Thymeleaf

```
< head th:fragment="headerfiles (title)">
<title th:text="${title}"></title>
<meta charset="UTF-8">
<link rel="stylesheet" href="/bootstrap.min.css" media="screen">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"
</script>
<script type="text/javascript">
      $(document).ready(function () {
         $("#locales").change(function () {
            var selectedOption = $('#locales').val();
            if (selectedOption != '') {
                  window.location.replace('?lang=' + selectedOption);
            }
         });
      });
</script>
< /head>
```

Refactored: src/main/resources/templates/viewitem.html

Internationalization with Spring and Thymeleaf

```html
<table cellpadding="10">
   <tbody><tr>
      <td width="100px">Id:</td>
      <td><b margin-left="15px" th:text="${item.id}"></b></td>
   </tr><tr>
   </tr><tr>
      <td><b th:text="#{label.name}">:</b></td>
      <td><b th:text="${item.name}"></b></td>
   </tr><tr>
   </tr><tr>
      <td><b th:text="#{label.price}">:</b></td>
      <td><b th:text="${item.price}"></b></td>
   </tr><tr>
   </tr><tr>
      <td><b th:text="#{label.description}">:</b></td>
      <td><b th:text="${item.description}"></b></td>
   </tr><tr>
</tr></tbody></table>
```

Refactored: FORM in src/main/resources/templates/createitem.html

Internationalization with Spring and Thymeleaf

```html
<form th:action="@{/createitem}" method="post" enctype="multipart/form-data">
   <div><input type="hidden" th:value="${id}" name="id"></div>
   <table>
      <tbody><tr>
         <td><span th:text="#{label.name}">:</span></td>
         <td><input name="name"></td>
      </tr>
      <tr>
         <td><span th:text="#{label.price}">:</span></td>
         <td><input name="price"></td>
      </tr>
      <tr>
         <td><span th:text="#{label.description}">:</span></td>
         <td><input name="description"></td>
      </tr>
   </tbody></table>
   <br>
   <div>
      <input type="submit" th:value="#{label.button-new-item}" name="button">
   </div>
</form>
```

New class: InternationalizationConfig implements WebMvcConfigurer

Internationalization with Spring and Thymeleaf

```java
@Configuration
public class InternationalizationConfig implements WebMvcConfigurer {
    @Bean
    public LocaleResolver localeResolver() {
        SessionLocaleResolver localeResolver = new SessionLocaleResolver();
        localeResolver.setDefaultLocale(Locale.UK);
        return localeResolver;
    }

    @Bean
    public LocaleChangeInterceptor localeChangeInterceptor() {
        LocaleChangeInterceptor localeChangeInterceptor =
                                                    new LocaleChangeInterceptor();
        localeChangeInterceptor.setParamName("lang");
        return localeChangeInterceptor;
    }

    @Override
    public void addInterceptors(InterceptorRegistry registry) {
        registry.addInterceptor(localeChangeInterceptor());
    }
}
```

Summary: Web Development: Frameworks

Where are we now?

- Framework: Spring and Spring Boot
- Embedded Server
- Library: Thymeleaf
- Internationalization

Next

Web Services: SOAP vs REST

Home