Spring Framework - Web
Web Application Development

Zsolt Tóth
University of Miskolc

2018
Table of Contents

1. Spring Web
   - Architecture
   - Dispatcher Servlet

2. Controllers
   - Request
   - Response
   - Exception Handling
Spring Web MVC

- J2EE Technology
  - based on it
  - replace it
- De Facto Standard
- Single Servlet (dispatcher)
- Facilitates Development
  - Marshalling
  - Annotations
  - Exception Handling
- MVC Architecture
Dispatchers Servlet

- J2EE Servlet
- Registered in `web.xml`
- Capture all HTTP Requests
- Find Controllers
- Forward Requests
- Process Response
  - Forward Result
  - Resolve View
Dispatcher Servlet

HTTP Request

DispatcherServlet

HTTP Response

1. Handler Mapping
2. Controller
3. View Resolver
4. View

Zsolt Tóth (University of Miskolc)
Table of Contents

1. Spring Web
   - Architecture
   - Dispatcher Servlet

2. Controllers
   - Request
   - Response
   - Exception Handling
@Controller

End–Point
- URL Mapping
  - HTTP Requests
- Openness
  - Standardized
  - Data Formats
  - API Documentation
- Services
  - Use–Case
  - service wrapper
- MVC

HTTP ↔ Method Invocation
- Processing Requests
- Unmarshalling
- Sanitize Input
- Invoke Services
- Marshalling
- Exception Handling
@Controller

- Java Class
- End–Point
- @Controller annotation
- Spring Component
- Contains web services
- Dependencies
  - Services
  - Injection
    - Mandatory
    - Optional

```java
@Controller
public class MyController{
    private MyService service;
    public MyController(
        MyService service) {
        this.service = service;
    }
}
```
@RequestMapping

- Mapping method to URL
- Class
  - URL prefix
  - Path only
- Method
  - In @Controller
  - Specify Function
  - Parameters?
  - Return Type?
  - Exceptions?

Parameters

- consumes media types
- produces media types
- method RequestMethod
- name assigned name
- path "/path/to/service"
- headers narrowing mapping
@PathVariable

- Part of URL
- Dynamic paths
- Name matching

Query by id
- /products/id

@PathVariable

```java
@RequestMapping("/products/{id}\")
public ProductDTO getProductById(
    @PathVariable String id
){...}
```

```java
@PathVariable

@RequestMapping("/orders/fromDate/toDate")
public Collection<Order> listOrdersBetween(
    @PathVariable Date fromDate,
    @PathVariable(required=false) Date toDate
){...}
```
@RequestParam

- GET Request
- Query Part of URL
- Automatic Conversion
  - Simple Types
  - Dates
  - Formatters

```
@RequestMapping("...")
public void method(
    @RequestParam("param1")
    String name,
    @RequestParam("param2")
    int age
){...}
```

```
http://host:port/webapp/path/to/service?
param1=a&param2=b
```
<table>
<thead>
<tr>
<th>POST or PUT Request</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forms</td>
<td>Unmarshalling / Parsing</td>
</tr>
<tr>
<td>Complex Data</td>
<td>Schema Validation</td>
</tr>
<tr>
<td>Data structures</td>
<td>IllegalArgumentException</td>
</tr>
<tr>
<td>JSON, XML</td>
<td>Sanitize</td>
</tr>
<tr>
<td>Conversion to Java Objects</td>
<td>Data is valid</td>
</tr>
<tr>
<td>Bean</td>
<td>Value checking</td>
</tr>
<tr>
<td>Default Constructor</td>
<td>Input modification</td>
</tr>
<tr>
<td>public getter/setter Methods</td>
<td>checksum, regex, etc.</td>
</tr>
</tbody>
</table>
**ModelAndView**

- **View**
  - HTML
  - JavaServerPages
- **Model**
  - Contains Data
  - Map<String, Object>
  - Inserted into View
  - Multiple Model Attributes
- **ModelAndView**
  - ViewResolver
  - Redirection
- **MVC–View**

**JSP**

- Servlet
- HTML Extension
- Java Tags
- `<% ... %>`
- Custom tag libraries
- **Viability**
  - Outdated
  - Still exists
  - Does not fit Single Page Applications
@ResponseBody

- JSON, XML
- Generated by Server
- Fetch Services
- Error Messages
- Data Transfer Object
  - Not Domain Model
  - Information Hiding
  - Simplify Response
  - Minimize Network Traffic
- Marshalling

- Next Lecture
- XSD
- Check "Data Handling in XML" course.
@ExceptionHandler

- Controller Scope
  - Same Source File
  - Redundancy
- Catch Exceptions
  - @RequestMapping
  - throws definition
- Parameters
  - Throwable[]

Method
- Parameter
  - Exception
  - Optional
- Return Type
  - ModelAndView
  - String
  - Custom Object
- Logging
@ControllerAdvice

- Spring Component
  - Contains
    - @ExceptionHandler
    - @InitBinder
    - @ModelAttribute
  - Has to be scanned
  - Global
  - For All Controllers
  - Generic Exceptions
    - Database is Unavailable
@ResponseStatus

- **Code**
  - `HttpStatus`
  - `Enumeration`

- **Reason**
  - `String`

```java
@ResponseStatus(
    value=HttpStatus.CONFLICT,
    reason="Username already exists!"
)
@ExceptionHandler(UserAlreadyEx.class)
public void usrExistsExHandler()
{
    //Logging
}
```