

# Thymeleaf & Validation

Thymeleaf Helpers, Validators



**SoftUni Team**  
Technical Trainers



**SoftUni**



**Software University**

<https://softuni.bg>

## 1. Thymeleaf Helpers

- Dates
- Strings
- Numbers
- Aggregates
- Validations

sli.do

**#java-web**



**Thymeleaf Helpers**

- Objects that provide built-in functionalities that helps you enhance your view



Thymeleaf



Dates



Strings



List



Numbers



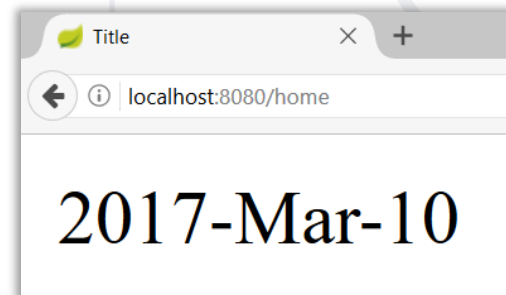
WhiskeyController.java

```
@GetMapping("/home")  
public String getHomePage(Model model){  
    model.addAttribute("myDate", new Date());  
    return "whiskey-home";  
}
```

Format Date

whiskey-home.html

```
<div th:text="${#dates.format(myDate, 'yyyy-MMM-dd')}"></div>
```



# Date – Week Name of Day

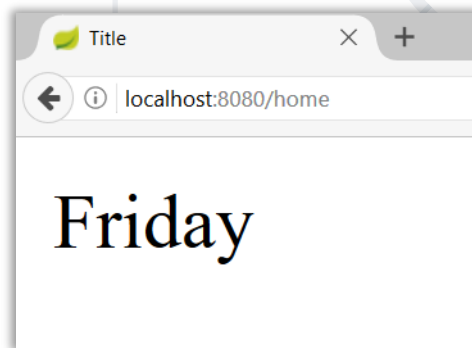
WhiskeyController.java

```
@GetMapping("/home")  
public String getHomePage(Model model){  
    model.addAttribute("myDate", new Date());  
    return "whiskey-home";  
}
```

Day Name

whiskey-home.html

```
<div th:text="${#dates.dayOfWeekName(myDate)}"></div>
```



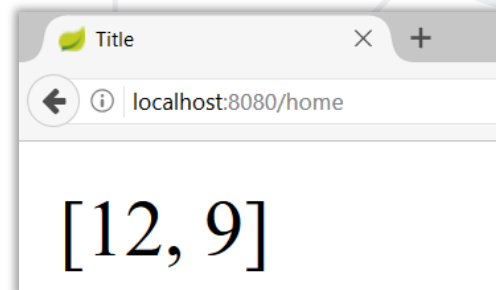
## WhiskeyController.java

```
@GetMapping("/home")
public String getHomePage(Model model){
    // List of dates -> 2016-12-12, 2017-04-09 -> yyyy-MM-dd
    model.addAttribute("myDates", myDates);
    return "whiskey-home";
}
```

## List Days

## whiskey-home.html

```
<div th:text="${#dates.listDay(myDates)}"></div>
```





# Date – Get Current Date

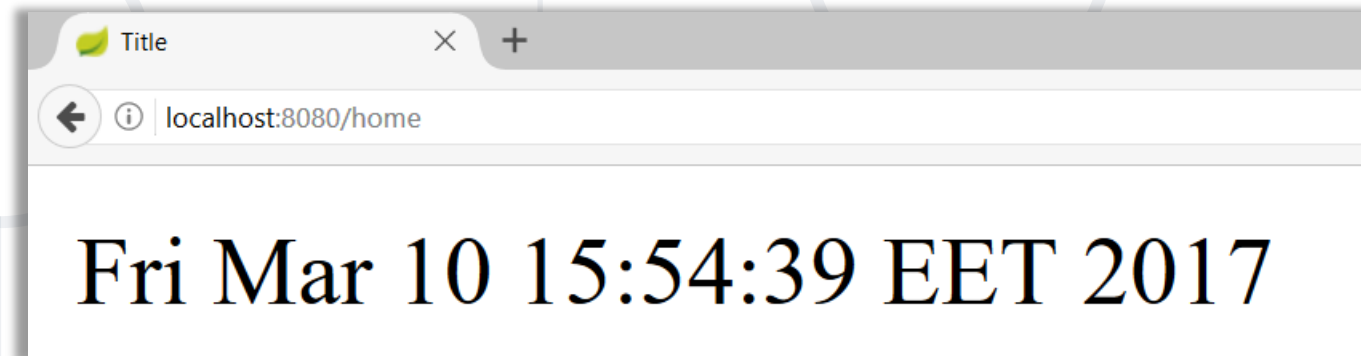
WhiskeyController.java

```
@GetMapping("/home")  
public String getHomePage() {  
    return "whiskey-home";  
}
```

Today's Date

whiskey-home.html

```
<div th:text="${#dates.createNow()}"></div>
```



WhiskeyController.java

```
@GetMapping("/home")
public String getHomePage(Model model){
    model.addAttribute("myDate", LocalDate.now());
    return "whiskey-home";
}
```

Format Date

whiskey-home.html

```
${#temporals.format(myDate, 'dd-MMM-yyyy')}|
```

- To use **LocalDate** we need to add new **dependency**

```
<dependency>
    <groupId>org.thymeleaf.extras</groupId>
    <artifactId>thymeleaf-extras-java8time</artifactId>
    <version>3.0.4.RELEASE</version>
</dependency>
```

# Strings – is Empty

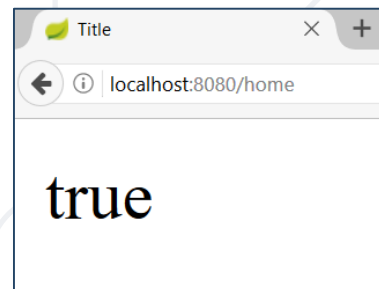
WhiskeyController.java

```
@GetMapping("/home")
public String getHomePage(Model model) {
    String whiskeyNull = null;
    model.addAttribute("whiskey", whiskeyNull);
    return "whiskey-home";
}
```

Null/Empty Check

whiskey-home.html

```
<div th:text="${#strings.isEmpty(whiskey)}"></div>
```

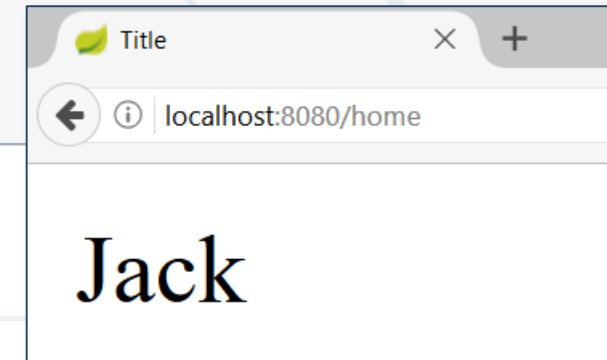


# Strings – Substring

WhiskeyController.java

```
@GetMapping("/home")
public String getHomePage(Model model) {
    String whiskey = "Jack Daniels";
    model.addAttribute("whiskey", whiskey);
    return "whiskey-home";
}
```

Substring



whiskey-home.html

```
<div th:text="${#strings.substring(whiskey,0,4)}"></div>
```

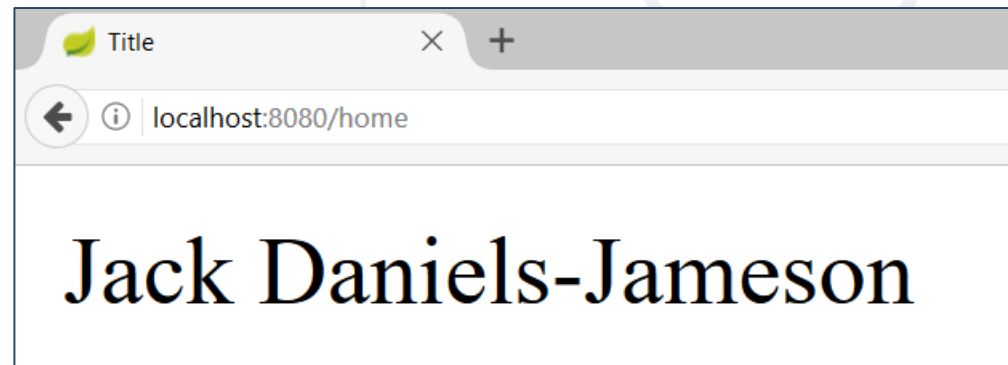
## WhiskeyController.java

```
@GetMapping("/home")
public String getHomePage(Model model) {
    model.addAttribute("whiskeys", whiskeys);
    // Jack Daniels, Jameson
    return "whiskey-home";
}
```

Join

## whiskey-home.html

```
<div th:text="${#strings.listJoin(whiskeys, '- ')}"></div>
```



# Strings – Capitalize

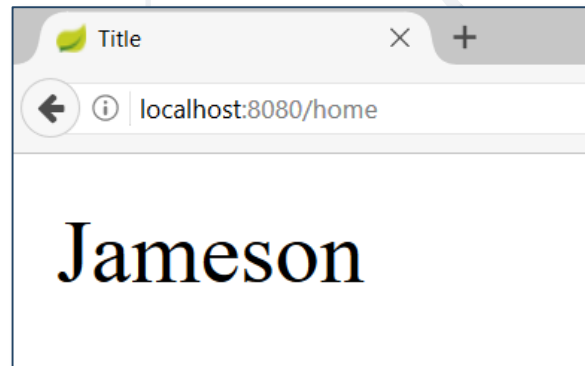
WhiskeyController.java

```
@GetMapping("/home")
public String getHomePage(Model model) {
    String whiskey = "jameson";
    model.addAttribute("whiskey", whiskey);
    return "whiskey-home";
}
```

Capitalize

whiskey-home.html

```
<div th:text="${#strings.capitalize(whiskey)}"></div>
```



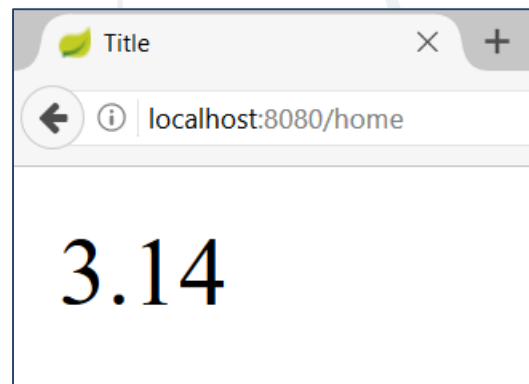
## MathController.java

```
@GetMapping("/home")
public String getHomePage(Model model) {
    double num = 3.14159;
    model.addAttribute("num", num);
    return "home";
}
```

Format

## home.html

```
<div th:text="${#numbers.formatDecimal(num,1,2)}"></div>
```



# Numbers – Sequence

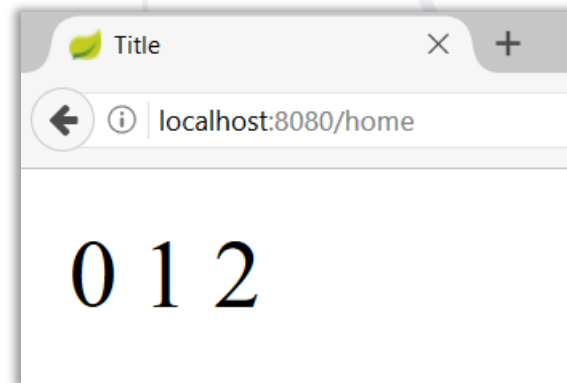
MathController.java

```
@GetMapping("/home")  
public String getHomePage(Model model) {  
    return "home";  
}
```

Sequence

home.html

```
<span th:each="number: ${#numbers.sequence(0,2)}">  
    <span th:text="${number}"></span>  
</span>
```





# Aggregates – Sum

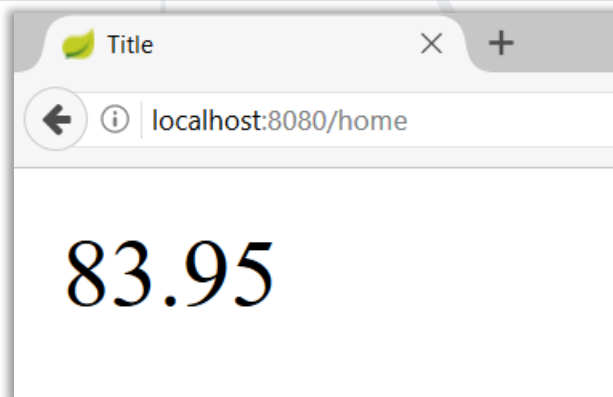
## WhiskeyController.java

```
@GetMapping("/home")
public String getHomePage(Model model) {
    double[] whiskeyPrices
        = new double[]{29.23, 21.22, 33.50};
    model.addAttribute("whiskeyPrices", whiskeyPrices);
    return "whiskey-home";
}
```

Sum

## whiskey-home.html

```
<div th:text="${#aggregates.sum(whiskeyPrices)}">
```



# Thymeleaf in JavaScript

## JSController.java

```
@GetMapping("/js")
public String getMapPage(Model model){
    String message = "Hi JS!";
    model.addAttribute("message", message);
    return "page";
}
```

## script.js

```
<script th:inline="javascript">
    let message = [[${message}]];
</script>
```



**How to Validate?**

- Making a simple **Model validation** and **Error rendering**

## SomeModel.java

```
public class SomeModel {  
  
    @NotNull  
    @Size(min = 3, max = 10,  
        message = "Invalid name")  
    private String name;  
  
}
```

## SomeController.java

```
@Controller  
public class SomeController {  
  
    @GetMapping("/add")  
    public String getPage(Model model) {  
  
        if(!model.containsKey("bindingModel"){  
            model.addAttribute("bindingModel",  
                new BindingModel());  
        }  
  
        return "add";  
    }  
}
```

Adding a  
model to  
the view

- Making a simple **Model validation** and **Error rendering**

SomeController.java

```
@PostMapping("/add")
public String add (@Valid @ModelAttribute("bindingModel") SomeModel bindingModel,
BindingResult bindingResult, RedirectAttributes rAtt) {
    if(bindingResult.hasErrors()){
        rAtt.addFlashAttribute("bindingModel", bindingModel);
        rAtt.addFlashAttribute(
            "org.springframework.validation.BindingResult.SomeModel", bindingResult);
        return "redirect:/add";
    }

    this.someService.save(bindingModel);

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

Validate the  
model

Validation  
Result

- Making a simple **Model validation** and **Error rendering**

Set Binding Model

add.html

```
<div th:object="${productBindingModel}" >
  <div class="justify-content-center">
    <label for="name" class="h4 mb-2 text-white">Name</label>
  </div>
  <input th:field="*{name}" th:errorclass="bg-danger" type="text"
        class="form-control" id="name" name="name"/>
  <small th:if="${#fields.hasErrors('name')}"
        th:errors="*{name}" class="text-danger"> Name error</small>
</div>
```

Access  
Field

Render Error

Name

Invalid name

# List All Errors

add.html

```
<ul th:if="${#fields.hasErrors('*')}">
  <li th:each="err : ${#fields.errors('*')}" th:text="${err}">
    Input is incorrect
  </li>
</ul>
```

- Invalid creator
- Invalid name
- Mutation cannot be null
- Invalid description
- Invalid hours
- You must select capitals

add.html

```
<ul th:if="${#fields.hasErrors('${someModel.*}')}">
  <li th:each="err : ${#fields.errors('${someModel.*}')}>
    Input is incorrect
  </li>
</ul>
```

- You can also implement **custom validation** annotations
  - Sometimes it is necessary due to complex validation functionality

## PresentOrFuture.java

```
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.FIELD)
@Constraint(validatedBy = PresentOrFutureValidator.class)
public @interface PresentOrFuture {

    String message() default "Invalid Date";

    Class<?>[] groups() default {};

    Class<? extends Payload>[] payload() default {};

}
```



# Custom Annotations (2)

- You can also use the **@PresentOrFuture** validation annotations.
- You will have to implement a **custom validator** too

SomeModel.java

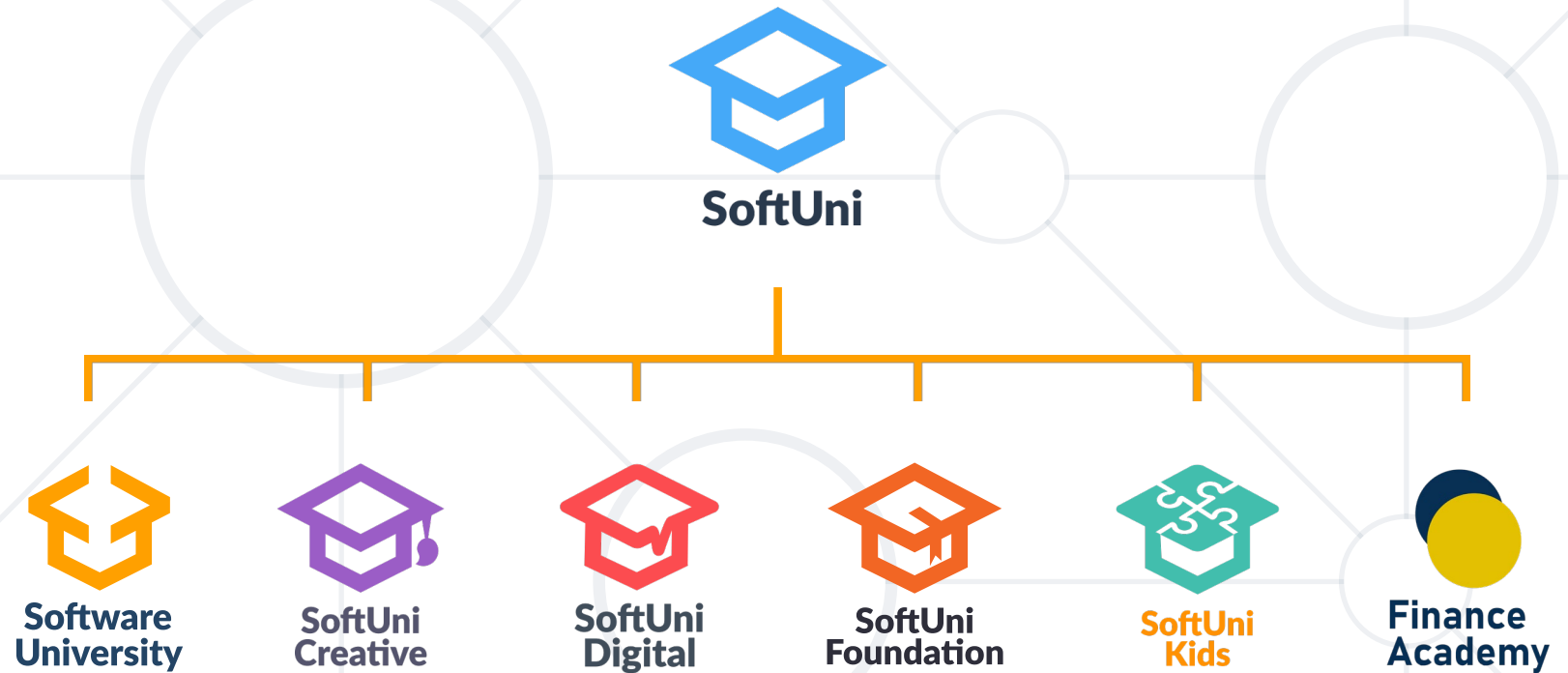
```
public class SomeModel {  
    @NotNull  
    @PresentOrFuture  
    @DateTimeFormat(pattern = "dd/MM/yyyy")  
    private Date startDate;  
}
```

Annotation

- Objects that provide built-in functionalities that helps you enhance your view
- **Thymeleaf** provides helpers and validations
- Making a simple **Model validation** and **Error rendering**



# Questions?



# SoftUni Diamond Partners

**SUPER  
HOSTING  
.BG**



**Coca-Cola HBC  
Bulgaria**

 **Flutter**<sup>TM</sup>  
International

**INDEAVR**  
Serving the high achievers

 **AMBITIONED**

 **DRAFT  
KINGS**

 **SOFTWARE  
GROUP**

 **BOSCH**

 **Postbank**  
*Решения за твоето утре*

 **PHAR  
VISION**

 **SmartIT**

**DXC**  
TECHNOLOGY

**createX**  


- Software University – High-Quality Education, Profession and Job for Software Developers

- [softuni.bg](http://softuni.bg), [about.softuni.bg](http://about.softuni.bg)

- Software University Foundation

- [softuni.foundation](http://softuni.foundation)

- Software University @ Facebook

- [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)

- Software University Forums

- [forum.softuni.bg](http://forum.softuni.bg)



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg/>
- © Software University – <https://softuni.bg>

