

Managing Forms

Forms in HTML

- ▶ HTML forms are a way for users to input data and send it to the server
- ▶ To create a form in HTML you use the `<form>` tag with various attributes
- ▶ One other important attribute is the method attribute that defines if the form should be sent with a GET or a POST request
- ▶ Inside the `<form>` tag typically various input fields are added that can be used for text input

Demo 1 - HTML form

- ▶ What a HTML form looks like
- ▶ Attributes in the form tag
- ▶ Inputs in the form

Forms used by Spring MVC

- ▶ A form used to submit input to a Spring MVC application needs some attributes:
- ▶ The action attribute defines the URL of the form (to where it will be sent)
- ▶ The method attribute defines the method of the form (whether it should be sent as a HTTP GET or a HTTP POST request)
- ▶ The data in the input fields will be sent as request params and to be able to fetch these they need to have names, this is done using the name attribute

Demo 2 - HTML form for Spring MVC

- ▶ Attributes in the form tag: action and method
- ▶ Attributes in the input: name

Displaying the form

- ▶ To be able to add input to the form a user must first get an html page with the form
- ▶ You could use a @GetMapping method that returns the name of the Thymeleaf template with the form to generate html with the form

Demo 3 - Displaying the form

- ▶ Use a @GetMapping method to display the form

Receiving the form

- ▶ Typically the form used the HTTP POST method that hides the form data in the body of the request
- ▶ Then you need a `@PostMapping` method to receive the request
- ▶ The URL of the `@PostMapping` method should be the same as the value of the action attribute in the `<form>` tag
- ▶ To receive the input fields from the form, `@RequestParam` input arguments are used in the method

Demo 4 - Receiving the form

- ▶ Using a @PostMapping method
- ▶ Using @RequestParam input arguments to receive input fields

Default values in the form

- ▶ If default data is to be displayed in the form, the html value attribute can be used in the input tag
- ▶ If the default data comes from the Controller and is added as an attribute in the Model, then the th:value attribute must be used

Demo 5 - Default data in the form

- ▶ Using th:value

Form-backing object

- ▶ Imagine if you have like 25 input fields in a form. Then you would have 25 @RequestParam input arguments in the @PostMapping method
- ▶ A more convenient way to handle this is with a form-backing object
- ▶ It is used to bind the form input fields to instance variables in an object in the Model
- ▶ A th:object attribute in the form tag is used to bind an object in the Model to the form, then th:field attributes are used to bind each input field in the form to an instance variable in the object

Demo 6 - Form-backing object

- ▶ Binding the form to the object
- ▶ Binding the input fields to instance variables
- ▶ Adding the object as an attribute in the Model
- ▶ Receiving the object in the @PostMapping method with @ModelAttribute

Exercise 1 - Managing Forms

- ▶ You could use the HandlingFormsDemo project for these exercises
- ▶ Create a template with a login form with input fields for username and password (you could use type=password in the input for the password)
- ▶ Add a button to submit the form with an input with type=submit
- ▶ Create a @GetMapping method that displays the page with the form and a @PostMapping method that receives the input from the form

Exercise 2 - Managing Forms

- ▶ Create a new version of the login form with a form-backing object
- ▶ There is already a User class in the ManagingFormsDemo project, use this as the form-backing object
- ▶ Add the object as an attribute in the Model in the @GetMapping method
- ▶ Bind the form to the object and the fields to the instance variables
- ▶ Create a @PostMapping method that receives the object as a @ModelAttribute

Exercise 3 - Managing Forms

- ▶ Experiment with other inputs than input of type=text, for example textarea, checkbox, select
- ▶ Look here for information about creating forms with Thymeleaf: <https://www.thymeleaf.org/doc/tutorials/3.0/thymeleafspring.html#creating-a-form>