Validation

Validation

‣ Validation input from external sources are important!

‣ We can validate manually or use automatic validation by using annotations in the form-backing object

Manual Validation

▸ Manually check the values in variables

▸ If a value is not valid - then add it as an error to the BindingResult object

▸ If errors are found - send the user back to the form

▸ If there are no errors - continue to the next step

Demo 2 - Manual Validation

▸ Manual validation

▸ Using the BindingResult object

▸ Display error messages in the form when validation fails

▸ Add a red border around the input field with the error

Automatic Validation

- ‣ The instance variables in the form-backing object are annotated with annotations that specify the validation rules

- ‣ Change the annotation of the @ModelAttribute input argument in the @PostMapping method to @Valid indicating that it is not only a ModelAttribute but should also be automatically validated before being sent into the @PostMapping method

- ‣ There must be an input argument of type BindingResult and that object already has all the errors from the automatic validation

- ‣ This validation also requires a specific dependency in pom.xml

Annotation for automatic validation

‣ @Size – validates that the value has a size between the attributes min and max

‣ @Email – validates that the value is a valid email address

‣ @NotEmpty – validates that the the value is not null or empty

‣ @Positive and @PositiveOrZero – validate numeric values are positive

‣ @Negative and @NegativeOrZero – validate numeric values are negative

Demo 3 - Automatic Validation

‣ Using annotations in the form-backing object

‣ Using @Valid instead of @ModelAttribute

‣ Adding a validation dependency in pom.xml

Exercise 1 - Manual Validation

▸ Reuse Exercise 2 from Managing Forms (The Login form) and validate manually that the username is between 3 and 12 characters and that the password is not empty

▸ If the validation succeeds then display a new results page, and if the validation fails then display the form again

▸ Try also to add error messages in the form and some styling that shows there is an error in the input field

Exercise 2 - Automatic Validation

‣ Reuse Exercise 3 from Managing Forms (The Register Admin form) and validate automatically that the username is between 3 and 12 characters and that the password is not empty

‣ If the validation succeeds then display a new results page, and if the validation fails then display the form again

‣ Try also to add error messages in the form and some styling that shows there is an error in the input field

‣ Use annotations in the Admin class and the @Valid annotation in the @PostMapping method. Also add the required dependency in pom.xml