Open Source Software

Appendix

anthemopensource
Feb 23, 2018

Section A - Examples

Showcases common code examples using the framework.

A.1 Conditional Operations

The following samples showcase how to apply common conditional scenarios using the framework’s *@Conditional* annotations.

A1.1. Make Textbox "Read-Only" by Default

```
@TextBox
@ParamContext(enabled = false, visible = true)
private String textbox;
```

A1.2. Make Textbox "Read-Only" When a Question is Answered as "Yes"

```
@TextBox(postEventOnChange = true)
@EnableConditional(when = "state == 'Yes'", targetPath = { "/../textbox" })
private String condition;

@TextBox
private String textbox;
```

A1.3. Make Textbox Mandatory by Default

```
@TextBox
@NotNull
private String textbox;
```

A1.4. Make Textbox Mandatory When a Question is Answered as "Yes"

```
@TextBox(postEventOnChange = true)
```

```
@ValidateConditional(when = "state == 'Yes'", targetGroup = ValidationGroup.GROU
private String condition;

@TextBox
@NotNull(groups = { ValidationGroup.GROUP_1.class })
private String textbox;
```

A1.5. Make Textbox Validate by Pattern and be Mandatory When a Question is Answered as "Yes"

```
@TextBox(postEventOnChange = true)
@ValidateConditional(when = "state == 'Yes'", targetGroup = ValidationGroup.GROU
private String condition;

@TextBox
@NotNull(groups = { ValidationGroup.GROUP_1.class })
@Pattern(regexp = "Hello (.*)!", groups = { ValidationGroup.GROUP_1.class })
private String textbox;
```

A1.6. Make Textbox Editable by Default

```
@TextBox
private String textbox;
```

A1.7. Make Combobox Values Change and be Mandatory Based on Another Field

```
@TextBox(postEventOnChange = true)
@ValidateConditional(when = "state == 'Yes'", targetGroup = ValidationGroup.GROU
private String condition;

@ComboBox(postEventOnChange = true)
@NotNull(groups = { ValidationGroup.GROUP_1.class })
@ValuesConditional(target = "../condition", condition = {
    @Condition(when = "state=='Yes'", then = @Values(YES_VALUES.class)),
    @Condition(when = "state=='No'", then = @Values(NO_VALUES.class))
})
private String combobox;
```

A1.8. Make TextBox Editable or Not-Editable by Default When a Form Loads

```
// Editable by default
@TextBox
```

```
private String textbox1;

// Not Editable by default - Method 1 (Preferred)
@TextBox
@ParamContext(enabled = false, visible = true)
private String textbox2;

// Not Editable by default - Method 2
@TextBox
@EnableConditional(when = "false", targetPath = "../")
private String textbox2;
```

A1.9. Make Textbox Editable and Apply Validations Based on the Value of Another Field

```
@TextBox(postEventOnChange = true)
@EnableConditional(when = "state == 'Yes'", targetPath = { "/../textbox" })
@ValidateConditional(when = "state == 'Yes'", targetGroup = ValidationGroup.GROU
private String condition;

@TextBox
@NotNull(groups = { ValidationGroup.GROUP_1.class })
private String textbox;
```

A1.10. Make Textbox Already Having Validations have Additional Validations Based on the Value of Another Field

```
@TextBox(postEventOnChange = true)
@EnableConditional(when = "state == 'Yes'", targetPath = { "/../textbox" })
@ValidateConditional(when = "state == 'Yes'", targetGroup = ValidationGroup.GROU
private String condition;

@TextBox
@NotNull(groups = { ValidationGroup.GROUP_1.class })
@Min(10)
private String textbox;
```

A1.11. Make Form/Section Disabled Based on Some Condition

```
@TextBox(postEventOnChange = true)
@EnableConditional(when = "state == 'Yes'", targetPath = { "/../form" })
private String condition;

@Form
```

```
private MyForm form;

@Model @Getter @Setter
public static class MyForm {
    // Form implementation here...
}
```

A1.12. Make Textbox Hidden on Some Condition and On Another Condition, Make it Visible and Mandatory.

```
@TextBox(postEventOnChange=true)
@ActivateConditional(when = "state == 'No'", targetPath = { "../textbox" })
@ValidateConditional(when = "state == 'Yes'", targetGroup = GROUP_1.class)
private String condition;

@TextBox
@NotNull(groups = { GROUP_1.class })
private LocalDateTime textbox;
```

A1.13 Make Textbox Mandatory When State is Anything but a Particular Value

```
@TextBox(postEventOnChange=true)
@ValidateConditional(when = "state != null &amp;&amp; state != 'Yes'", targetGro
private String condition;

@TextBox
@NotNull(groups = { GROUP_1.class })
private String textbox;
```

A1.14 Make Textbox "Read-Only" When State is Anything but a Particular Value

```
@TextBox(postEventOnChange=true)
@EnableConditional(when = "state != null &amp;&amp; state != 'Yes'", targetGroup
private String condition;

@TextBox
@NotNull(groups = { GROUP_1.class })
private String textbox;
```

A1.15 Make Textbox Hidden When State is All but a Particular Value

```
@TextBox(postEventOnChange=true)
```

```
@ActivateConditional(when = "state!= null &amp;&amp; state != 'Yes'", targetGrou
private String condition;

@TextBox
@NotNull(groups = { GROUP_1.class })
private String textbox;
```

- Release versioning - Naming conventions

No labels