

MODULE 8

SESSION ATTRIBUTE

TABLE OF CONTENT

- Bean Scope
- Session Attribute
- Session Attribute Injection

Bean Scope - *Application*

- ▶ Persists as long as the web application lives
 - Objects with this scope
 - Are visible in all request/response cycles for all clients as long as the application is active
 - Are created upon the first HTTP request when the web application starts up
 - Are destroyed when the web application shuts down

Bean Scope - Session

- ▶ Persists across multiple HTTP requests belonging to the user session
 - Objects with this scope
 - Are visible for any request/response cycle that belongs to a **session**
 - Have their state persisted between requests until the end of the session

Bean Scope - *Request*

- ▶ Persists during a single HTTP request in a web application
 - Objects with this scope
 - Are visible from the start of the request
 - Until the end of the request, i.e, when the response has been sent to the client

Session Attribute

- ▶ To pass model data from controller to view (jsp page)
 - Through Model dictionary
 - By **default**, scope of the dictionary attributes objects = **request scope**
 - One instance of each model object created for each request
- ▶ **>< Session** scope
 - One instance of the model object created for an entire session
 - To **maintain state between several requests**

Session Attribute

- ▶ Declare session attributes using **@SessionAttributes**
 - *import org.springframework.web.bind.annotation.SessionAttributes;*
 - Give them a unique name (id)
- ▶ Declare them in get/post method as input
- ▶ Use **@ModelAttribute** with the same unique name (id)
- ▶ Use them in jsp pages thanks to the same unique name (id)
 - Syntax : **\${uniqueNameID}**

Session Attribute – *Model Class*

- ▶ E.g, a session attribute to store the color chosen by the user
- ▶ Model class

```
public class Color {  
    private String name;  
    public Color() { }  
    public String getName() { return name; }  
    public void setName(String name) { this.name = name; }  
}
```


Session Attribute - Controller

```
@Controller
@RequestMapping(value="/color")
@SessionAttributes({ColorController.COLOR})
public class ColorController {

    protected static final String COLOR = "mainColor";

    @ModelAttribute(COLOR)
    public Color chosenColor()
    { return new Color(); }

    @RequestMapping(method=RequestMethod.GET)
    public String home(Model model, @ModelAttribute(value=COLOR) Color mainColor)
    { return "colorPage"; // E.g, to choose the color }

    @RequestMapping(value="/changeColor", method=RequestMethod.POST)
    public String getFormData(Model model, @ModelAttribute(value=COLOR) Color mainColor)
    { return "showColorPage"; // E.g, to display the chosen color }
}
```

Annotations and their purposes:

- `@SessionAttributes({ColorController.COLOR})` → Apply session scope to mainColor attribute
- `protected static final String COLOR = "mainColor";` → Constant ID for the session attribute
- `@ModelAttribute(COLOR)` → Declare mainColor as a model attribute
- `@ModelAttribute(value=COLOR) Color mainColor` → Bind the session attribute to the method parameter

Session Attribute – Jsp Page

► colorPage.jsp

```
<form:form id="colorChoice" method="POST"  
    action="/first/color/changeColor"  
    modelAttribute="mainColor">  
    <form:label path="name"> Name </form:label>  
    <form:input path="name"/>  
    <form:button>Submit</form:button>  
</form:form>
```

► showColorPage.jsp

```
Your chosen color is ${mainColor.name}
```

Session Attribute

- ▶ If no value for **@ModelAttribute**
 - The value is the name of the attribute

▶ E.g,

```
@ModelAttribute
```

```
public Color chosenColor()  
{ return new Color(); }
```

—→ No value for the annotation

—→ The value is *chosenColor*

Session Attribute

- ▶ Other way to create Session bean
 - Through components

Session Attribute Injection

- ▶ How to access session attributes in other controllers?
- ▶ Through **@SessionAttributes**
 - On Controller class
 - Using the same name (value)

Session Attribute Injection

- E.g, *mainColor* session attribute created in *ColorController*

```
@Controller
@RequestMapping(value="/color")
@SessionAttributes({ColorController.COLOR})
public class ColorController {

    protected static final String COLOR = "mainColor";

    @ModelAttribute(COLOR)
    public Color chosenColor() {
        return new Color();
    }
}
```

➞ Access in *HelloController*

```
@Controller
@RequestMapping(value="/hello")
@SessionAttributes("mainColor")
public class HelloController {
```