

# JAVA SERVER FACES IN ACTION

Kito D. Mann

Foreword by Ed Burns

 MANNING



# contents

---

*foreword* xxi  
*preface* xxiii  
*acknowledgments* xxv  
*about this book* xxvii  
*about the title and cover* xxxiii

---

## PART 1 EXPLORING JAVASERVER FACES ..... 1

---

### **1** *Introducing JavaServer Faces* 3

#### 1.1 It's a RAD-ical world 4

*So, what is JavaServer Faces?* 5 ■ *Industry support* 10

#### 1.2 The technology under the hood 10

*Hypertext Transfer Protocol (HTTP)* 11 ■ *Servlets* 12  
*Portlets* 13 ■ *JavaBeans* 14 ■ *JSP and other display technologies* 15

#### 1.3 Frameworks, frameworks, frameworks 16

*Why do we need frameworks?* 16 ■ *She's a Model 2* 17  
*JSF, Struts, and other frameworks* 18

#### 1.4 Components everywhere 19

- 1.5 Hello, world! 22
  - Dissecting hello.jsp* 24
  - *Dissecting goodbye.jsp* 31
  - Examining the HelloBean class* 32
  - *Configuration with faces-config.xml* 34
  - *Configuration with web.xml* 36
- 1.6 Summary 37

## 2 **JSF fundamentals** 38

- 2.1 The key pieces of the pie 39
  - User interface components* 41
  - *Renderers* 43
  - Validators* 44
  - *Backing beans* 45
  - *Converters* 48
  - Events and listeners* 49
  - *Messages* 55
  - *Navigation* 56
- 2.2 The Request Processing Lifecycle 57
  - Phase 1: Restore View* 61
  - *Phase 2: Apply Request Values* 63
  - Phase 3: Process Validations* 65
  - *Phase 4: Update Model Values* 66
  - *Phase 5: Invoke Application* 66
  - *Phase 6: Render Response* 68
- 2.3 Understanding component and client identifiers 69
  - Naming containers* 72
  - *Referencing identifiers* 73
- 2.4 Exploring the JSF expression language 76
  - Understanding scoped variables* 80
  - *Using implicit variables* 81
  - *Using the EL with components* 83
- 2.5 Summary 86

## 3 **Warming up: getting around JSF** 88

- 3.1 Setting up your JSF environment 89
  - Basic requirements* 89
  - *Choosing a JSF implementation* 89
  - Directory structure* 90
  - *Configuration* 92
- 3.2 The role of JSP 102
  - Using JSP includes* 103
  - *Using JSF with JSTL and other JSP custom tags* 104
- 3.3 Creating and initializing beans 110
  - Declaring managed beans* 113
  - *Declaring Lists and Maps as managed beans* 123
  - *Setting values with value-binding expressions* 125
- 3.4 Navigating the sea of pages 129
- 3.5 Summary 136

- ## 4 **Getting started with the standard components 137**
- 4.1 It's all in the components 138
    - Using HTML attributes 142* ▪ *Understanding facets 143*
    - The power of tools 145* ▪ *The render kit behind the scenes 148*
  - 4.2 Common component properties 148
  - 4.3 Controlling the page with UIViewRoot 149
  - 4.4 Setting component parameters with UIParameter 151
  - 4.5 Displaying data with the Output components 153
    - Displaying ordinary text with HtmlOutputText 153* ▪ *Using UIOutput with the <f:verbatim> tag 155* ▪ *Creating input labels with HtmlOutputLabel 158* ▪ *Using HtmlOutputFormat for parameterized text 160* ▪ *Displaying hyperlinks with HtmlOutputLink 165*
  - 4.6 Displaying images with HtmlGraphicImage 167
  - 4.7 Displaying component messages with HtmlMessage 169
  - 4.8 Displaying application messages with HtmlMessages 172
  - 4.9 Grouping and layout with the Panel components 176
    - Grouping components with HtmlPanelGroup 176*
    - Creating tables with HtmlPanelGrid 178*
  - 4.10 Summary 184
- ## 5 **Using the input and data table components 185**
- 5.1 Registering event listeners 186
    - Declaring value-change listeners 187*
    - Declaring action listeners 187*
  - 5.2 Common component properties 189
  - 5.3 Handling forms with HtmlForm 190
  - 5.4 Handling basic user input 192
    - Declaring basic text fields with HtmlInputText 193* ▪ *Using HtmlInputTextarea for memo fields 194* ▪ *Displaying password fields with HtmlInputSecret 195* ▪ *Declaring hidden fields with HtmlInputHidden 197*

- 5.5 Using `HtmlSelectBooleanCheckbox` for checkboxes 198
  - 5.6 Defining item lists 199
    - Using `UISelectedItem` for single items* 200
    - *Using `UISelectItems` for multiple items* 203
  - 5.7 Handling multiple-item selections 205
    - Using `HtmlSelectManyCheckbox` for checkbox groups* 205
    - Displaying listboxes with `HtmlSelectManyListbox`* 208
    - Using `HtmlSelectManyMenu` for single-item listboxes* 210
  - 5.8 Handling single-item selections 212
    - Using `HtmlSelectOneRadio` for radio button groups* 212
    - Using single-select listboxes with `HtmlSelectOneListbox`* 215
    - Declaring combo boxes with `HtmlSelectOneMenu`* 217
  - 5.9 Executing application commands 219
    - Declaring buttons with `HtmlCommandButton`* 219
    - Creating an action link with `HtmlCommandLink`* 221
  - 5.10 Displaying data sets with `HtmlDataTable` 223
  - 5.11 Summary 233
- 6 *Internationalization, validators, and converters* 234**
- 6.1 Internationalization and localization 235
    - Looking into locales* 236
    - *Creating resource bundles* 238
    - Using resource bundles with components* 241
    - Internationalizing text from back-end code* 244
  - 6.2 Input validation 245
    - Using validator methods* 245
    - *Using validators* 246
    - Using the standard validators* 247
    - *Combining different validators* 251
  - 6.3 Type conversion and formatting 251
    - Using converters* 254
    - *Working with the standard converters* 255
  - 6.4 Customizing application messages 269
  - 6.5 Summary 273

---

**PART 2 BUILDING USER INTERFACES ..... 275**


---

**7 Introducing ProjectTrack 277**

- 7.1 Requirements 278
- 7.2 The conceptual model 281
- 7.3 User interface 283
- 7.4 Development team 284
- 7.5 Summary 286

**8 Developing a user interface without Java code:  
the Login page 287**

- 8.1 Getting started 289
  - Setting up web.xml* 289 ▪ *Setting up faces-config.xml* 290
- 8.2 Creating the Login page 291
  - Starting with HtmlGraphicImage and HtmlOutputText components* 292 ▪ *Adding a form* 295
- 8.3 Sprucing things up 300
  - Using an image for the button* 301 ▪ *Integrating with JavaScript* 301 ▪ *Adding Cascading Style Sheets* 303
- 8.4 Adding validators 304
  - Customizing validation messages* 307
- 8.5 Improving layout with HtmlPanelGrid 308
- 8.6 Summary 314

**9 Developing a user interface without Java code:  
the other pages 316**

- 9.1 Building the header with a custom component 317
  - Using a custom toolbar component* 321 ▪ *Configuring the navigation rule* 323
- 9.2 Prototyping data tables with panels 324
  - The Inbox page* 325 ▪ *Configuring the navigation rule* 329
  - The Show All page* 330 ▪ *Configuring the navigation rule* 330

- 9.3 Creating input forms 331
    - The Approve a Project page* 331
    - *Configuring the navigation rule* 337
    - *The Reject a Project page* 338
    - *Configuring the navigation rule* 341
    - *The Create a Project page* 341
    - Configuring the navigation rule* 347
  - 9.4 The Project Details page 347
    - Configuring the navigation rule* 351
  - 9.5 Summary 353
- 10 Integrating application functionality 354**
- 10.1 Understanding JSF development approaches 355
  - 10.2 Exploring the application environment 357
  - 10.3 Reorganizing pages for security 360
  - 10.4 The Login page 360
    - Updating the navigation rule* 364
  - 10.5 The header 365
    - Updating the navigation rule* 369
  - 10.6 Integrating data grids 370
    - The Inbox page* 370
    - *The Show All page* 378
  - 10.7 Integrating input forms 379
    - Updating the includes* 379
    - *The Approve a Project page* 382
    - The Reject a Project page* 385
    - *The Create a Project page* 386
  - 10.8 The Project Details page 390
    - Updating the navigation rule* 395
  - 10.9 Adding an error page 396
    - Updating web.xml* 397
    - *Updating the navigation rule* 397
  - 10.10 Internationalizing and localizing the UI 398
    - Externalizing text into the resource bundle* 398
    - Internationalizing the header* 400
    - Localizing for Russian* 402
  - 10.11 Summary 404

---

**PART 3** DEVELOPING APPLICATION LOGIC ..... 407

---

- 11**    ***The JSF environment***    **409**
- 11.1 From servlets to JSF    410
  - 11.2 The application foundation    413
    - Application*    413    ▪ *Evaluation expressions*    417
  - 11.3 It's all in the context    419
    - FacesContext*    420    ▪ *FacesMessage*    422
    - ExternalContext*    424
  - 11.4 Event handling    428
    - FacesEvent*    430    ▪ *Handling action events*    432    ▪ *Handling value-change events*    434    ▪ *Handling phase events*    435
  - 11.5 Components revisited    438
    - UIComponent*    442    ▪ *UIViewRoot*    446
    - ValueHolder*    449    ▪ *EditableValueHolder*    451
    - SelectItem* and *SelectItemGroup* model beans    453
  - 11.6 Summary    455
- 12**    ***Building an application: design issues and foundation classes***    **456**
- 12.1 Layers of the pie    457
  - 12.2 Roasting the beans    460
    - The importance of toString*    461    ▪ *Serialization for breakfast*    462    ▪ *It's all in the properties*    462
    - Exposing beans*    472
  - 12.3 Exploring the business layer and data layers    473
  - 12.4 Developing the application layer    476
    - Handling constants*    478    ▪ *Organizing utility methods*    480
    - Initializing singletons*    482    ▪ *Adapting business objects*    484
  - 12.5 Writing a visit object for session state    491
  - 12.6 Developing a base backing bean class    494
  - 12.7 Summary    498



## 13 ***Building an application: backing beans, security, and internationalization*** 499

- 13.1 Writing backing beans 500
  - Thread safety* 501
  - *Handling errors* 501
  - *Performing authentication* 505
  - *Listing projects with UIData and parameterizing listeners* 511
  - *Updating projects* 522
  - *Creating new projects* 528
  - *Paging through the project history with UIData* 534
  - *Working with JDBC ResultSets and UIData* 540
- 13.2 Adding security 545
  - Container-based vs. custom security* 546
  - Using custom security* 547
- 13.3 Supporting internationalization in code 551
  - Internationalizing text with resource bundles* 552
  - Internationalizing messages* 557
- 13.4 Design consequences and alternatives 562
  - Accessing the business layer* 562
  - *Organizing beans by function* 563
  - *Action methods implemented by backing beans* 564
  - *Initializing backing bean properties with the Managed Bean Creation facility* 565
- 13.5 Summary 566

## 14 ***Integrating JSF with Struts and existing applications*** 568

- 14.1 What integration means 569
- 14.2 When to use JSF with other frameworks 569
- 14.3 The many faces of requests and responses 571
- 14.4 Integrating JSF with Struts applications 572
  - First steps* 575
  - *Migrating Struts JSP tags* 577
  - Using JSF action methods and managed beans* 597
  - Who's controlling whom?* 599
- 14.5 Integrating JSF with non-Struts applications 600
- 14.6 Summary 601

**PART 4** WRITING CUSTOM COMPONENTS, RENDERERS,  
VALIDATORS, AND CONVERTERS..... 603

---

**15** *The JSF environment: a component  
developer's perspective 605*

- 15.1 Three steps to UI extension nirvana 606
  - 15.2 Developing UI components 607
    - Deciding when to write a UI component 608* ▪ *Classes and interfaces 610* ▪ *Event handling with method bindings 623*
    - Registration 624* ▪ *JSP integration 627*
  - 15.3 Developing renderers 636
    - Deciding when to write a renderer 640* ▪ *Renderer 641*
    - RenderKit 643* ▪ *Registration 644* ▪ *JSP integration 647*
  - 15.4 Developing validators 648
    - Validator 649* ▪ *Registration 650* ▪ *JSP integration 652*
  - 15.5 Developing converters 654
    - Converter 654* ▪ *Registration 657* ▪ *JSP integration 658*
  - 15.6 Handling internationalization 660
  - 15.7 Packaging UI extensions 660
  - 15.8 Summary 661
- appendix A: Using JSF without JSP 665*
- references 675*
  - index 679*

*online extension:* See page xviii for contents

# *online extension*

---

The online extension consists of five chapters in part 5 as well as four appendixes that are not included in the print edition. They are available for free download from the book's web site at [www.manning.com/mann](http://www.manning.com/mann). This is the table of contents for the online extension.

---

<b>PART 5</b>	<b>WRITING CUSTOM COMPONENTS, RENDERERS, VALIDATORS, AND CONVERTERS: EXAMPLES .....</b>	<b>703</b>
---------------	---	------------

---

<b>16</b>	<b><i>UIInputDate: a simple input component</i></b>	<b>705</b>
16.1	Writing the <code>UIInputDate</code> class	708
	<i>Encoding</i> 709 ▪ <i>Decoding</i> 715 ▪ <i>Implementing StateHolder methods</i> 717	
16.2	Registering the component	718
16.3	JSP integration	718
	<i>Writing the JSP custom tag</i> 718 ▪ <i>Validating the tag</i> 721	
	<i>Adding the tag to the tag library</i> 722	
16.4	Using the component	724
16.5	Summary	726

- 17**    ***RolloverButton renderer: a renderer with JavaScript support***    **727**
- 17.1    Writing the RolloverButtonRenderer class    729
    - Encoding*    731    ▪    *Decoding*    735
    - Registering the renderer*    736
  - 17.2    JSP Integration    737
    - Writing the HtmlBaseTag class*    738    ▪    *Writing the JSP custom tag*    741    ▪    *Validating the tag*    744    ▪    *Adding the tag to the tag library*    745
  - 17.3    Using the renderer    748
  - 17.4    Wrapping an existing renderer    750
    - Developing the RolloverButtonDecoratorRenderer class*    750
  - 17.5    Summary    754
- 18**    ***UIHeadlineViewer: a composite, data-aware component***    **756**
- 18.1    RSS and the Informa API    758
  - 18.2    Using UIData with Informa    763
  - 18.3    Subclassing DataModel    765
  - 18.4    Writing the UIHeadlineViewer class    768
  - 18.5    Registering the component    780
  - 18.6    JSP integration    781
    - Writing the JSP custom tag*    781
    - Adding the tag to the tag library*    787
  - 18.7    Using the component    789
  - 18.8    Summary    793
- 19**    ***UINavigator: a model-driven toolbar component***    **794**
- 19.1    Writing the model classes    796
  - 19.2    Writing the UINavigator class    801
    - Implementing ActionSource methods*    803    ▪    *Overriding UIComponentBase methods*    806    ▪    *Implementing StateHolder methods*    807    ▪    *Developing NavigatorActionListener: a custom ActionListener*    809

- 19.3 Registering the component 810
- 19.4 Writing the `ToolbarRenderer` class 811
  - Encoding* 811 ▪ *Decoding* 820
- 19.5 Registering the renderer 821
- 19.6 JSP integration 822
  - Writing the `Navigator_ToolbarTag` component tag* 822
  - Writing the `NavigatorItemTag` tag handler* 826
  - Adding the tags to the tag library* 831
- 19.7 Using the component 834
- 19.8 Summary 838

## 20 **Validator and converter examples** 839

- 20.1 Validator methods vs. validator classes 840
- 20.2 Developing a validator 840
  - Writing the `RegularExpressionValidator` class* 842
  - Registering the validator* 847 ▪ *Integrating with JSP* 847
  - Using the validator* 852
- 20.3 When custom converters are necessary 854
- 20.4 Developing a converter 854
  - Writing the `UserConverter` class* 856 ▪ *Registering the converter* 865 ▪ *JSP integration* 866
  - Using the converter* 870
- 20.5 Summary 872

*appendix B: A survey of JSF IDEs and implementations* 873

*appendix C: Extending the core JSF classes* 935

*appendix D: JSF configuration* 958

*appendix E: Time zone, country, language, and currency codes* 976