Struts 2 in Action

Donald Brown
Chad Michael Davis
Scott Stanlick

MANNING

contents

preface xvii
acknowledgments xix
about this book xxii
about the title xxvii
about the cover illustration xxviii

PART 1 STRUTS 2: A BRAND NEW FRAMEWORK 1

1 ***Struts 2: the modern web application framework*** *3*

1.1 Web applications: a quick study 4

Using the Web to build applications 4 ▪ *Examining the technology stack* 4 ▪ *Surveying the domain* 8

1.2 Frameworks for web applications 9

What's a framework? 10 ▪ *Why use a framework?* 10

1.3 The Struts 2 framework 11

A brief history 11 ▪ *Struts 2 from 30,000 feet: the MVC pattern* 12 ▪ *How Struts 2 works* 15

1.4 Summary 18

2 *Saying hello to Struts 2 20*

2.1 Declarative architecture 21

Two kinds of configuration 21 ▪ Two mechanisms for declaring your architecture 22 ▪ Intelligent defaults 25

2.2 A quick hello 25

Deploying the sample application 26 ▪ Exploring the HelloWorld application 30

2.3 HelloWorld using annotations 36

2.4 Summary 38

PART 2 CORE CONCEPTS: ACTIONS, INTERCEPTORS, AND TYPE CONVERSION 41

3 *Working with Struts 2 actions 43*

3.1 Introducing Struts 2 actions 44

What does an action do? 44

3.2 Packaging your actions 46

The Struts 2 Portfolio application 47 ▪ Organizing your packages 47 ▪ Using the components of the struts-default package 50

3.3 Implementing actions 52

The optional Action interface 52 ▪ The ActionSupport class 54

3.4 Transferring data onto objects 62

Object-backed JavaBeans properties 62 ▪ ModelDriven actions 64 Last words on using domain objects for data transfer 67

3.5 File uploading: a case study 67

Getting built-in support via the struts-default package 67 ▪ What does the fileUpload interceptor do? 68 ▪ Looking at the Struts 2 Portfolio example code 69

3.6 Summary 72

4 *Adding workflow with interceptors 74*

4.1 Why intercept requests? 75

Cleaning up the MVC 75 ▪ Reaping the benefits 77 Developing interceptors 78

4.2 Interceptors in action 78

The guy in charge: ActionInvocation 78 ▪ How the interceptors fire 79

4.3 Surveying the built-in Struts 2 interceptors 81

Utility interceptors 82 ▪ Data transfer interceptors 82 Workflow interceptors 84 ▪ Miscellaneous interceptors 88 Built-in stacks 90

4.4 Declaring interceptors 90

Declaring individual interceptors and interceptor stacks 90 Mapping interceptors to actions 93 ▪ Setting and overriding parameters 94

4.5 Building your own interceptor 95

Implementing the Interceptor interface 95 ▪ Building the AuthenticationInterceptor 95

4.6 Summary 99

5 *Data transfer: OGNL and type conversion 101*

5.1 Data transfer and type conversion: common tasks of the web application domain 102

5.2 OGNL and Struts 2 103

What OGNL does 103 ▪ How OGNL fits into the framework 105

5.3 Built-in type converters 108

Out-of-the-box conversions 108 ▪ Mapping form field names to properties with OGNL expressions 109

5.4 Customizing type conversion 122

Implementing a type converter 122 ▪ Converting between Strings and Circles 123 ▪ Configuring the framework to use our converter 124

5.5 Summary 126

PART 3 BUILDING THE VIEW: TAGS AND RESULTS.....129

6 *Building a view: tags 131*

6.1 Getting started 132

The ActionContext and OGNL 132 ▪ The ValueStack: a virtual object 135

6.2 An overview of Struts tags 137

The Struts 2 tag API syntax 138 ▪ Using OGNL to set attributes on tags 139

6.3 Data tags 142

The property tag 142 ▪ The set tag 143 ▪ The push tag 144 ▪ The bean tag 145 ▪ The action tag 148

6.4 Control tags 150

The iterator tag 150 ▪ The if and else tags 151

6.5 Miscellaneous tags 152

The include tag 152 ▪ The URL tag 153 ▪ The i18n and text tags 154 ▪ The param tag 156

6.6 Using JSTL and other native tags 156

6.7 A brief primer for the OGNL expression language 157

What is OGNL? 157 ▪ Expression language features commonly used in Struts 2 157 ▪ Advanced expression language features 163

6.8 Summary 165

7 *UI component tags 167*

7.1 Why we need UI component tags 168

More than just form elements 168

7.2 Tags, templates, and themes 174

Tags 175 ▪ Templates 176 ▪ Themes 176

7.3 UI Component tag reference 178

Common attributes 178 ▪ Simple components 180 ▪ Collection-backed components 190 ▪ Bonus components 198

7.4 Summary 201

8 *Results in detail 202*

8.1 Life after the action 203

Beyond the page: how to use custom results to build Ajax applications with Struts 2 204 ▪ Implementing a JSON result type 205

8.2 Commonly used result types 213

The RequestDispatcher, a.k.a. dispatcher 213 ▪ The ServletRedirectResult, a.k.a. redirect 219 ▪ The ServletActionRedirectResult, a.k.a. redirectAction 222

8.3 JSP alternatives 223

VelocityResult, a.k.a. velocity 224 ▪ FreemarkerResult, a.k.a. freemarker 225

8.4 Global results 227

8.5 Summary 228

PART 4 IMPROVING YOUR APPLICATION 229

9 *Integrating with Spring and Hibernate/JPA 231*

9.1 Why use Spring with Struts 2? 232

What can dependency injection do for me? 232 ▪ How Spring manages objects and injects dependencies 235 ▪ Using interfaces to hide implementations 236

9.2 Adding Spring to Struts 2 238

Letting Spring manage the creation of actions, interceptors, and results 239 ▪ Leveraging autowiring to inject dependencies into actions, interceptors, and results 242

9.3 Why use the Java Persistence API with Struts 2? 244

Setting your project up for JPA with Hibernate 245 ▪ Coding Spring-managed JPA 249

9.4 Summary 253

10 *Exploring the validation framework 255*

10.1 Getting familiar with the validation framework 256

The validation framework architecture 256 ▪ The validation framework in the Struts 2 workflow 258

10.2 Wiring your actions for validation 261

Declaring your validation metadata with ActionClass-validations.xml 262 ▪ Surveying the built-in validators 265

10.3 Writing a custom validator 267

A custom validator to check password strength 267 ▪ Using our custom validator 269

10.4 Validation framework advanced topics 271

Validating at the domain object level 271 ▪ Using validation context to refine your validations 274 ▪ Validation inheritance 277 ▪ Short-circuiting validations 277 ▪ Using annotations to declare your validations 278

10.5 Summary 280

11 *Understanding internationalization 282*

11.1 The Struts 2 framework and Java i18n 283

Retrieving localized text with ResourceBundle and Locale 284 ▪ How Struts 2 can ease the pain of i18n 286

11.2 A Struts 2 i18n demo 287

A quick demo of Struts 2 i18n 287 ▪ A quick look behind the scenes 290

11.3 Struts 2 i18n: the details 291

Struts 2 default TextProvider ResourceBundle location algorithm 291 Retrieving message texts from your bundles 295 ▪ Using the i18n tag to specify a bundle 299 ▪ Parameterizing your localized texts 299 ▪ Formatting dates and numbers 301

11.4 Overriding the framework's default locale determination 302

Letting the user interactively set the locale 302 ▪ Programmatically setting the locale 305

11.5 Summary 305

PART 5 ADVANCED TOPICS AND BEST PRACTICES 307

12 ***Extending Struts 2 with plug-ins 309***

12.1 Plug-in overview 310

How to find plug-ins 311

12.2 Common plug-ins 311

SiteMesh 311 ▪ Tiles 313 ▪ JFreeChart 315

12.3 Internal component system 316

Beans 317 ▪ Constants 318 ▪ Injection 318 ▪ Struts internal extension points 319

12.4 Writing a breadcrumb plug-in 321

12.5 Summary 325

13 ***Best practices 326***

13.1 Setting up your environment 327

Setting up your IDE 327 ▪ Reloading resources 328

13.2 Unit-testing your actions 328

The advantage of IoC for testing 329 ▪ JUnit and the tests 329 ▪ Testing validation.xml files 332

13.3 Maximizing reuse 332

Componentization with the component tag 333 ▪ Leveraging the templated tags 334 ▪ Connecting the UI-to-object dots 335

13.4 Advanced UI tag usage 336

Overriding existing templates 336 ▪ Writing custom templates 337 ▪ Writing custom themes 337

13.5 Summary 338

14 *Migration from Struts Classic 339*

14.1 Translating Struts Classic knowledge 340

Actions 340 ▪ What happened to ActionForms? 341 Switching tag libraries 343 ▪ Breaking up message resources 345

14.2 Converting by piecemeal 346

Eating an elephant a piece at a time 347 ▪ The action mappings 349 ▪ Where the action meets the form 350 Turn the page 352 ▪ No speak English 354 ▪ The data police 355 ▪ Can we just get along? 357

14.3 Summary 359

15 *Advanced topics 360*

15.1 Advanced action usage 361

Alternative method invocation 361

15.2 Dynamic method invocation 362

Wildcard method selection 362 ▪ Dynamic workflows 365

15.3 Using tokens to prevent duplicate form submits 366

Using the <s:token/> form tag 366 ▪ Exceptions to the token interceptor rule 368

15.4 Displaying wait pages automatically 369

When users are impatient 369

15.5 A single action for CRUD operations 371

That CRUD 371 ▪ Interceptors and interfaces 372 ▪ Connecting the parts 377

15.6 Tiles and Struts 2 379

Taking care of the website look and feel 379 Configuring the interplay 380 ▪ Using the declarative architecture 383 ▪ Preparing web page content with a tiles controller 385

15.7 Summary 386

index 387