



GWT IN ACTION

Easy Ajax with the
Google Web Toolkit

Robert Hanson
Adam Tacy

 MANNING

contents

preface xix
acknowledgments xxi
about this book xxiii
about the title xxx
about the cover illustration xxxi

PART 1 GETTING STARTED1

1 *Introducing GWT* 3

1.1 A walk through GWT 5

Explaining GWT's Java-to-JavaScript compiler 6 ■ *Using JSNI to execute JavaScript from Java* 8 ■ *Accessing the JRE emulation library* 10 ■ *Understanding GWT's widget and panel library* 12 ■ *Examining GWT's internationalization and configuration tools* 14 ■ *Calling remote procedures with GWT* 16 ■ *Investigating GWT's XML parser* 20 ■ *Managing the browser history* 21 ■ *Introducing GWT's JUnit integration* 22

1.2 GWT vs. other solutions 23

GWT vs. Swing 25 ■ *GWT vs. Echo2* 26 ■ *GWT vs. JavaServer Faces* 27 ■ *GWT vs. Ruby on Rails* 29

- 1.3 Building your first GWT application 30
 - Building and running an example application* 30
 - Building Tic-Tac-Toe with GWT* 32
- 1.4 Summary 37
- 2 *Creating the default application* 38**
 - 2.1 The GWT application development lifecycle 39
 - 2.2 Stage 1: Creating a GWT application 44
 - Creating the project* 47 ■ *Creating an application* 50
 - Setting up internationalization* 54 ■ *Implementing internationalization* 56 ■ *Creating unit test cases* 57 ■ *Importing into your IDE* 59
 - 2.3 Summary 63
- 3 *Advancing to your own application* 64**
 - 3.1 Describing the application example 65
 - 3.2 Stage 2: Developing your application 67
 - Implementing internationalization* 68 ■ *Constructing the application* 72 ■ *Applying styling* 83
 - 3.3 Stage 3: Testing and debugging in hosted mode 86
 - Preparing for hosted mode* 86 ■ *Running the Dashboard in hosted mode* 90 ■ *Debugging the Dashboard in hosted mode through Eclipse* 92
 - 3.4 Stage 4: Compiling the code 94
 - Compiling the code/preparing for web mode* 95
 - Viewing the compilation results* 97
 - 3.5 Stage 5: Deploying the code 98
 - Deploying to a web server* 98 ■ *Deploying to a filesystem* 99
 - 3.6 Stage 6: Running in web mode 99
 - 3.7 Implementing application logging 100
 - Logging information on the client-side* 100
 - Logging information on the server-side* 103
 - 3.8 Summary 104

PART 2 BUILDING USER INTERFACES 107

4 Working with widgets 109

- 4.1 What is a widget? 111
 - Using widgets as Java objects 112*
 - Considering widgets as DOM elements 113*
- 4.2 The standard GWT widgets 115
 - Interacting with the basic widgets 118 ■ Displaying text on the application 131 ■ Grabbing the user's interaction using focus widgets 133 ■ Getting user input through text input 139*
- 4.3 Creating new widgets 141
 - Creating new widgets by manipulating the DOM 142*
 - Creating new widgets by extending existing widgets 146*
- 4.4 Developing the Dashboard's ToggleMenuItem widget 150
 - Building the TwoComponentMenuItem 151*
 - Building the ToggleMenuItem 154*
- 4.5 Summary 155

5 Working with panels 157

- 5.1 What is a panel? 158
 - Using panels as Java Objects 159*
 - Considering panels as DOM elements 160*
- 5.2 The standard GWT panels 162
 - Interacting with simple panels 164 ■ Considering more complex panels 170 ■ Considering HTML table-based panels 178 ■ Considering composite panels 181*
 - Splitting panels 182*
- 5.3 Creating new panels 182
 - Creating a new panel from scratch 183*
 - Creating a new panel by extending an existing panel 186*
- 5.4 Creating the Dashboard panel 187
- 5.5 Summary 191

6 Handling events 192

6.1 Exploring events 193

Identifying event model browser differences 195
Understanding the GWT event model 196

6.2 Listening to events 200

Sinking events 200 ■ *Managing sunk events with the onBrowserEvent() method 203* ■ *Linking sunk events to the onBrowserEvent() method 207* ■ *Previewing events 208* ■ *Handling events by extending the listener classes 211* ■ *Moving events further into your GWT code 212* ■ *Preventing default browser event handling 216*

6.3 Handling standard browser events 216

Reacting to change 218 ■ *Clicking around 221* ■ *Gaining/Losing focus 222* ■ *Capturing keyboard inputs 223*
Loading images 223 ■ *Managing mouse inputs 224* ■ *Scrolling 224* ■ *Window resize events 226* ■ *Window close events 227*

6.4 Handling other event types 231

Handling forms 231 ■ *Reacting to closing pop-ups 231* ■ *Tab events 232* ■ *Tabling events 234* ■ *Tree events 235*

6.5 Implementing drag-and-drop 236

Implementing the drag part 236
Implementing the drop part 241

6.6 Summary 244

7 Creating composite widgets 246

7.1 What is a composite widget? 247

7.2 Composite widget development steps 248

7.3 Building the editable label 250

Step 1: Identifying the components 251 ■ *Step 2: Choosing the panel layout and structure 253* ■ *Step 3: Implementing the right GWT Java interfaces 255* ■ *Step 4: Building the composite widget 258* ■ *Step 5: Styling the composite widget 267*

7.4 Creating a composite widget from other composite widgets 269

Creating a slider 270 ■ *Constructing the ColourPicker composite 273*

- 7.5 Creating the Dashboard composite 274
- 7.6 Summary 276

8 **Building JSNI components 277**

- 8.1 Introducing JavaScript Native Interface (JSNI) 279
 - Understanding JSNI 281*
- 8.2 Communicating using JSNI 290
 - Chatting to the browser via GWT Java 291* ■ *Chatting to the browser via JavaScript 292* ■ *Talking to a GWT application via a JavaScript API 294* ■ *Talking between GWT applications 296*
- 8.3 Loading a JavaScript library 299
 - Using HTML to load a JavaScript library 299*
 - Using the module XML to load a JavaScript library 299*
- 8.4 Wrapping a simple JavaScript library 300
 - Accessing the loaded JavaScript library 302*
 - Using the widget in an application 307*
- 8.5 Wrapping a complex JavaScript library 309
 - Generating the classes 309* ■ *Keeping JavaScript objects as Java objects 310* ■ *Calling user-defined code from a library 312* ■ *Using a complex wrapped widget in an application 314*
- 8.6 Summary 315

9 **Modularizing an application 317**

- 9.1 Creating a modularization structure 318
 - Modularization in GWT 319* ■ *Including other modules in an application 321* ■ *Setting source and other resource paths 325* ■ *Defining an application's server resources 326* ■ *Managing an application's GWT properties 327* ■ *Replacing classes based on property values 332* ■ *Registering generators in the XML module file 333* ■ *Injecting resources into an application at runtime 334* ■ *Setting an application's entry point 337* ■ *The Dashboard's module XML file 337*
- 9.2 Including third-party modules 339

- 9.3 Packaging your own modules 341
- 9.4 Creating the Java package structure 343
- 9.5 Summary 344

PART 3 ADVANCED TECHNIQUES345

10 *Communicating with GWT-RPC* 347

- 10.1 Underlying RPC concepts 348
 - Understanding asynchronous communication* 350
 - *Restrictions for communicating with remote servers* 351
 - *Creating the Server Status project* 353
- 10.2 Implementing GWT-RPC 356
 - Understanding serializable data objects* 356
 - *Defining the GWT-RPC service* 360
 - *Preparing the client side of a GWT-RPC call* 366
 - *Calling the remote server service* 368
- 10.3 Project summary 370
 - Project overview* 370
 - *Server-side service implementation* 371
 - Calling the service from the client* 372
- 10.4 Summary 374

11 *Examining client-side RPC architecture* 375

- 11.1 Structuring the client code 376
 - Encapsulating the Server Status component* 376
 - *Encapsulating remote calls in a façade* 382
 - *Callback routine using the Command pattern* 384
- 11.2 Examining different polling techniques 392
 - Understanding polling issues* 392
 - *Implementing a continuously updating component* 393
 - *Emulating server-push by blocking server threads* 398
- 11.3 Writing custom field serializers 402
 - Creating a custom field serializer class* 403
 - Implementing custom field serialization* 404
- 11.4 Summary 407

12 *Classic Ajax and HTML forms* 409

- 12.1 Classic Ajax with RequestBuilder 410
 - Examining HTTP methods* 411 ■ *Simple RPC with RequestBuilder* 414 ■ *Using RequestBuilder to load XML data* 417
- 12.2 Examining FormPanel basics 424
 - Introducing the FormPanel* 425 ■ *Listening to FormPanel events* 428 ■ *Altering the FormPanel target* 429 ■ *Using the various form controls* 432
- 12.3 Summary 439

13 *Achieving interoperability with JSON* 442

- 13.1 Introducing JavaScript Object Notation (JSON) 443
 - Understanding the JSON data format* 443
 - Using JSONParser to parse JSON messages* 445
- 13.2 Examining GWT's JSON data objects 445
 - Introducing the JSONValue object* 445 ■ *Examining basic JSON types* 446 ■ *Storing JSONValue objects in a JSONArray* 448
 - Collecting JSONValue objects in a JSONObject* 449
- 13.3 Creating a search component using JSON 450
 - Examining the Yahoo Search API* 451 ■ *Implementing the Yahoo search component* 453 ■ *Sending JSON data to the server* 455
 - Parsing and validating a JSON server response* 457
- 13.4 Implementing a Yahoo Search proxy service 461
 - Using JSON with Java on the server* 461 ■ *Using JSON with Perl on the server* 465 ■ *Using JSON with Ruby on the server* 467
- 13.5 Summary 469

14 *Automatically generating new code* 471

- 14.1 Generating new types 472
- 14.2 Investigating GWT generators 473
 - Basic generator code* 476

- 14.3 Creating a generator for the Dashboard 477
 - Accessing the input class* 478 ■ *Accessing properties of the context* 479 ■ *Adding logging to a generator* 480 ■ *Generating the new type structure* 482 ■ *Creating the new class* 484
 - Using the classes that have been generated* 491
- 14.4 Summary 492

15 **Changing applications based on GWT properties** 494

- 15.1 Quick summary of properties 495
- 15.2 Managing browser differences 496
 - How GWT manages browser differences* 497 ■ *Building the Flash widget* 498 ■ *Setting up the property replacement* 501
- 15.3 Supporting internationalization in full 502
 - Using static-string internationalization* 503
 - Using dynamic string internationalization* 516
- 15.4 Altering the application for the locale 518
 - Implementing the default component* 518
 - Locale-specific classes* 519
- 15.5 Implementing user-defined properties 520
 - Defining user-specified properties* 521 ■ *Defining a user-specified property provider* 521 ■ *Checking the provided property value* 522 ■ *Building the code* 522
- 15.6 Summary 523

PART 4 COMPLETING THE UNDERSTANDING525

16 **Testing and deploying GWT applications** 527

- 16.1 Testing GWT code using JUnit 528
 - Overview of JUnit for GWT developers* 529 ■ *Creating a new test case* 534 ■ *Testing asynchronous code* 537
- 16.2 Deploying GWT applications 539
 - Organizing your project* 540 ■ *Installing RPC servlets* 548
- 16.3 Summary 554

17	<i>Peeking into how GWT works</i>	555	
17.1	Examining the compilation process and output	556	
	<i>Investigating compilation</i>	556 ■ <i>Examining the output</i>	559
17.2	The GWT application-loading mechanism	564	
	<i>Legacy applications</i>	565 ■ <i>Standard applications</i>	566
	<i>Bootstrapping the standard application</i>	567	
	<i>Cross-script applications</i>	572	
17.3	Compiling Java to JavaScript	573	
	<i>Exploring the produced JavaScript</i>	573 ■ <i>Reviewing</i>	
	<i>standard Java objects: the vector object</i>	574 ■ <i>Exploring</i>	
	<i>program code as JavaScript</i>	576 ■ <i>Understanding the</i>	
	<i>initialization code segment</i>	578	
17.4	Summary	579	
	<i>index</i>	581	