

Programmation GWT 2

Développer des applications RIA
et Ajax avec Google Web Toolkit

Sami Jaber



Table des matières

Introduction à GWT	1
Pourquoi GWT va-t-il changer la face du Web ?	1
Dix lignes de code GWT pour convaincre	3
Masquer la complexité du Web	4
Coder en Java	5
<i>Typage statique</i>	5
<i>Debugage</i>	5
<i>Refactoring</i>	6
<i>Tests unitaires</i>	7
<i>Compétences largement disponibles</i>	7
Support multinavigateur	7
Performances	9
Mais qu'est-ce qu'Ajax ?	10
La navigation en mode SPI	13
L'architecture RPC	13
Modèle 1.0 versus modèle 2.0	14
GWT face aux autres frameworks Ajax	15
Qu'en est-il de Flash, Silverlight et JavaFX ?	17
Un projet collaboratif	18
Une communauté active	19
Des navigateurs de plus en plus performants	20
L'évolution et les nouveautés de GWT 2.0	20
CHAPITRE 1	
L'environnement de développement	23
Télécharger et installer GWT	23
Contenu du répertoire d'installation	23
Le packaging GWT	24
Création du premier projet GWT	24
Exécuter l'application	26
Notion de module	27

Structure d'un projet GWT	28
Le package client	29
Le package serveur	30
Les fichiers de configuration	30
La structure du répertoire war	31
La page HTML hôte	31
Le mode développement	33
Le shell	36
Le conteneur de servlets Jetty	37
Le mode production	38
La structure d'un site compilé	39
Les types Java émülés par GWT	40
Le déploiement	42
Fichier Ant	42
Plug-in Maven	43

CHAPITRE 2

Les contrôles	45
Panorama des composants	45
Les classes UIObject et Widget	46
Les feuilles de style CSS	48
La syntaxe	48
Les styles dépendants	50
Les styles prédéfinis	51
La gestion des événements	53
Tour d'horizon des widgets	54
Les composants de formulaires	55
SuggestBox	57
Les bundles d'images	58
Les hyperliens	61
Les conteneurs et gestionnaires de placement	62
Les conteneurs simples (Panels)	62
<i>FormPanel</i>	63
<i>LazyPanel</i>	64
Les conteneurs complexes	65
<i>Exemple d'utilisation</i>	66
<i>HTMLPanel</i>	68
Synthèse des conteneurs GWT	70

CHAPITRE 3

Le modèle de placement CSS..... 73

Pourquoi un modèle de placement ?	73
Les limites du modèle GWT 1.X	74
Une solution basée sur le standard CSS	77
Les API	80
Les nouveaux conteneurs de GWT 2	82
Composant StackLayoutPanel	83
Le widget TabLayoutPanel	84
Sous le capot	84

CHAPITRE 4

Les bibliothèques tierces..... 87

L'écosystème GWT	87
Les bibliothèques de composants graphiques	88
L'incubateur GWT	88
<i>Les grilles paginées (ScrollTable)</i>	88
<i>Les arbres asynchrones (FastTree)</i>	89
<i>Les barres de progression</i>	90
<i>Les autres composants du projet d'incubation</i>	91
Le projet Ext-GWT (GXT)	92
SmartGWT	97
Glisser-déplacer avec GWT-DnD	100
Les bibliothèques de data binding	103
<i>GWT-Beans Binding</i>	104
<i>Le projet Gwittir</i>	105
À quand la liaison de données dans GWT ?	106
Les courbes et graphiques	107
<i>GChart</i>	107
<i>Open Flash Chart</i>	108
<i>Chronoscope</i>	109
Les frameworks complémentaires	110
GWT-WL (Widget Library)	110
GWT-Log	111
Manipuler les services Google avec GWT	113
Gwt-google-apis	113
GWT-GData	117
Conclusion	121

CHAPITRE 5

L'intégration de code JavaScript..... 123

Comprendre JSNI	123
Mise en pratique	124
Intégration d'un fichier JavaScript externe	126
Invoquer une méthode Java en JavaScript	127
Accéder à des attributs Java en JavaScript	129
Correspondance des types entre Java et JavaScript	130
Le type JavaScriptObject (JSO)	131
Undefined vs null	133
Gestion des exceptions JSNI	134
Les types Overlay	135
Un peu d'histoire	135
Mise en pratique des Overlay	137
Intégration Overlay et JSON	140
Sous le capot	141
L'implémentation unique du type JSO	145
Les contraintes associées à un JSO	146
Impact de JSNI sur le framework GWT	147
La magie interne de JSNI	147

CHAPITRE 6

La création de composants personnalisés..... 151

Quelques mots sur le DOM	151
La mécanique des événements	154
Pourquoi JavaScript fuit ?	154
Propagation par bouillonnement et capture	155
Expando et fuite mémoire	157
Créer un composant dérivé de Widget	160
Aller plus loin avec l'API événementielle	165
Dériver de la classe Composite	166
Dériver de la classe UIObject	169
Attachement dans un conteneur	169

CHAPITRE 7

Les services RPC..... 171

L'architecture RPC	172
Différence entre deRPC et RPC 1.0	173
Les étapes de la construction d'un service	174
Créer les deux interfaces de service	175

Créer les objets d'échange	176
Coder l'implémentation	178
Coder et configurer le client	179
La sérialisation	181
La gestion des exceptions	182
Exceptions non vérifiées	184
Accès à la requête HTTP	185
Mode hybride compatible RPC 1.X et RPC 2.X	187
Bonnes pratiques et mode asynchrone	187
Déploiement	188
Déploiement des classes	188
Configuration du fichier web.xml	189
Configuration dans un réseau sécurisé avec un frontal web	189
Classe RequestBuilder et services REST	191
Appel d'URL basique	191
Architecture REST	196
CHAPITRE 8	
L'intégration J2EE.....	197
Le modèle par délégation	197
Le modèle d'extensibilité	199
L'option <code>-noserver</code>	203
Intégration avec les EJB 3 et JPA	203
Le problème des classes instrumentées	212
Intégration des protocoles RMI, Corba et Soap	217
CHAPITRE 9	
Le chargement à la demande	219
Principe général	219
Les types de fragments	223
Positionner efficacement les points de rupture	228
Le design pattern Async package	233
Forcer le chargement des fragments	235
Sous le capot	237
Conclusion	239
CHAPITRE 10	
La liaison différée	241
Principe général	241
Mise en pratique	244

Le script de sélection	247
Propriétés, règles et conditions	247
Propriétés	248
Propriétés de configuration	250
Les propriétés conditionnelles	251
Règles de liaison	252
Générateurs de code	254
Dans la pratique	255
Déboguer	259
Conditions de liaison	260
Conclusion	260

CHAPITRE 11

La gestion des ressources 261

La problématique des ressources	261
Installation et configuration	263
Les différents types de ressources	264
Les ressources textuelles (TextResource)	265
Les ressources textuelles asynchrones	267
Les ressources binaires externes	271
Les ressources images	272
Les options de la liaison différée	275
L'injection dynamique CSS	275
L'injection différée	277
Les constantes	278
La substitution à l'exécution	279
Les fonctions de valeur	280
Les directives conditionnelles	282
Les préfixes de style	284
Les sprites d'image	285

CHAPITRE 12

Sous le capot de GWT 287

Introduction au compilateur	288
Vive les fonctions JavaScript !	288
Les étapes du compilateur	291
Lecture des informations de configuration	292
Création de l'arbre syntaxique GWT	292
La génération de code JavaScript et les optimisations	293
<i>La réduction de code (pruning)</i>	296

<i>La finalisation de méthodes et de classes</i>	298
<i>La substitution par appels statiques</i>	298
<i>La réduction de type</i>	299
<i>L'élimination de code mort</i>	300
<i>L'inlining</i>	300
Tracer les optimisations	301
Les options du compilateur	304
Accélérer les temps de compilation	307
Les linkers	308
La pile d'erreurs en production	313
Table des symboles	317

CHAPITRE 13

L'internationalisation	319
La problématique	319
Paramétrer et définir la locale courante	320
L'API i18n	321
Les dictionnaires à constantes statiques	322
Dictionnaire par recherche dynamique de constantes	324
Les messages	325
Notion de langue par défaut	326
Signification, exemple et description	327
Les formes plurielles	328
Conversion des types	330
Formats monétaires	330
Date et formats horaires	331
Génération automatique de dictionnaires	333
Bénéfices de l'internationalisation statique	335
Externalisation dynamique	335
L'outillage	336
i18nCreator	336
I18nSync	337

CHAPITRE 14

L'environnement de tests	339
GWT et la problématique des tests	339
La mixité des tests	340
Créer un test unitaire	340
Les suites de tests	343
Une architecture modulaire et extensible	344

Le style HtmlUnit – moteur de test par défaut	346
Le style manuel ou interactif	348
Le style Selenium	348
Le style distant	350
Le style externe	351
Synthèse des différentes options et annotations	351
Tests de charge avec la classe Benchmark	352
Les compteurs intégrés de performance	354
Tests fonctionnels robotisés : scénarios joués	357
Selenium IDE	357
<i>Le module WebDriver</i>	<i>362</i>
Les stratégies de test par bouchon (mocking)	365
Quel est l'atelier de tests idéal ?	367

CHAPITRE 15

Les design patterns GWT..... 369

Pourquoi des bonnes pratiques ?	369
Gestion de la session (cliente et serveur)	370
Limiter les besoins mémoire de la session cliente	371
La gestion côté serveur	372
Session et onglets des navigateurs	373
Gestion de l'historique	375
Que signifie le contexte précédent avec Ajax ?	378
Les traitements longs	380
La classe Timer	381
La classe DeferredCommand	382
La classe IncrementalCommand	383
La classe Scheduler	385
Séparer présentation et traitement	386
Le pattern commande	386
L'approche modèle vue contrôleur	389
<i>MVC par l'exemple avec le framework PureMVC</i>	<i>390</i>
Modèle vue présenteur (MVP)	392
<i>Le pattern Action orienté RPC</i>	<i>399</i>
Les failles de sécurité	401
Injection SQL	401
Cross-site Scripting (XSS)	402
CSRF (Cross-Site Request Forgery)	405
Les autres attaques	407
L'authentification	407

Authentification Basic et Digest	407
Authentification par formulaire	408
Les limites de la session HTTP par cookies	410

CHAPITRE 16

La création d'interfaces avec UIBinder 413

Présentation	414
Styles et ressources	418
Incorporation d'images	424
Intégration des ressources de type données	425
Gestionnaires d'événements	426
Intégration d'un flux HTML standard	429
Internationalisation	430
Les emplacements	432
<i>Cas des balises imbriquées.</i>	434
Traduire les attributs	435
Liaison avec des beans externes	436
Modèles composites et constructeurs	440
Parseurs personnalisés	444

CHAPITRE 17

Le plug-in Eclipse pour GWT 445

Le cas AppEngine	445
Le plug-in GWT	446
Création d'un projet GWT	446
Les assistants de création	449
Création d'un point d'entrée	449
Création d'un nouveau module	449
Création d'une page HTML hôte	450
Création d'un squelette ClientBundle	451
Création d'un squelette UIBinder	451
Aide à la saisie de code JSNI	453
Assistants RPC	454

Index 455