

Avant-propos

Pourquoi ce livre ?

Lorsque la plate-forme GWT est sortie en 2006, nous n'étions qu'une petite poignée à parier sur le potentiel de cette technologie. Les composants graphiques étaient austères, l'environnement de développement balbutiant et les performances plutôt médiocres.

C'est un peu par hasard que j'ai découvert la valeur réelle de GWT : en parcourant un jour le code source d'une classe égarée sur mon disque dur chargée de traduire le code Java en JavaScript (`JavaToJavaScriptCompiler`).

Dès lors, GWT ne m'a plus quitté. Il était clair pour moi que les jours des frameworks web Java tels que Struts ou JSF étaient comptés. Le concept de la compilation façon GCC adapté à Java et reléguant JavaScript au rang de vulgaire assembleur, la voie était toute tracée.

Les deux années qui ont suivi ont vu GWT gagner en maturité. Malgré les défauts de jeunesse de la version 1.x, les premiers projets ont été couronnés de succès. Jour après jour, les développeurs GWT gagnaient en productivité.

Au même rythme, les premiers ouvrages anglophones sont arrivés. Il en existe aujourd'hui pléthore couvrant la version 1.0. Reproduire en français de tels livres n'aurait eu aucun intérêt. On n'écrit pas un livre pour copier mais pour créer quelque chose. Sans compter qu'il ne faut jamais, c'est bien connu, se précipiter sur les premières versions d'un framework.

GWT ne fait pas exception à cette règle. La version 1.x a prouvé la viabilité du concept mais a également mis en avant quelques limites. Lorsque j'ai découvert les premiers travaux autour de la version 2.0, j'ai tout de suite compris que GWT 2 ne ressemblerait plus jamais à GWT 1.x.

Il aura fallu quasiment deux ans à l'équipe des contributeurs GWT pour finaliser la version 2. Si les concepts des premiers jours restent d'actualité, de nouvelles fondations ont été bâties pour préparer les outils de demain. Avec GWT 2, les littératures francophone et anglophones actuelles devenaient de facto obsolètes. Une case se libérait dans l'amoncellement de livres dédiés à GWT. Plus de doute possible, il en fallait un sur le sujet !

À l'heure de la publication de cet ouvrage, il n'existera aucun équivalent anglais ou français.

Dans quelles conditions a été écrit ce livre ?

Pendant plus de six mois, j'ai vécu (quelques soirs et week-ends) en immersion totale avec l'équipe GWT. Un peu comme quelqu'un qui regarderait au-dessus de l'épaule d'un développeur pour formaliser les moindres lignes de code qu'il écrit. Au fur et à mesure que les choix de conception s'opéraient, j'écrivais. Et comme un développeur rature souvent, altère ou supprime son code, des chapitres entiers de ce livre ont été modifiés ou supprimés après leur finalisation. J'aurais sûrement pu écrire un second ouvrage avec toutes les pages supprimées ou modifiées.

Pendant six mois, je n'ai pu m'appuyer sur aucune documentation technique. Il n'y en avait simplement pas. Les quelques pages de wiki qui ornaient le site du projet étaient à moitié bâclées et souvent publiées trop en avance.

Et puis j'ai appris à décrypter les cas de tests publiés au jour le jour sur le tronc SVN. J'ai appris à solliciter les développeurs, Joel, Bob, Bruce, John et Fred. Ils m'ont toujours aidé avec l'humilité et la modestie qui les caractérise tant.

J'ai aussi appris à critiquer le code au fur et à mesure qu'il se construisait, je me suis fait une opinion sur les développeurs et sur la qualité du code. Dans ce chemin semé d'embûches, j'ai dû faire face à d'innombrables bogues de jeunesse qui me handicapaient toujours plus dans la quête d'une finalisation qui s'éternisait au fil des évolutions du code source. Qu'importe, j'avais également appris à modifier le code source de GWT. Dans un premier temps je le faisais pour les besoins du livre... puis, je l'ai fait pour la communauté, en soumettant des correctifs.

J'ai bravé tous ces obstacles pour que ce livre soit le premier au monde à traiter de GWT 2. Des livres en anglais sortiront sûrement dans un an ou deux, mais celui-ci restera le premier livre sur GWT 2 avec ses qualités, mais aussi ses défauts.

J'espère que vous prendrez autant de plaisir à le lire que j'en ai eu à l'écrire.

À qui s'adresse ce livre ?

Ce livre s'adresse à tous les développeurs, architectes, chefs de projet ou testeurs souhaitant comprendre les bénéfices de GWT.

Nul besoin d'être un expert en programmation ou un gourou des langages HTML ou JavaScript. Il suffit de connaître un minimum la syntaxe Java et la structure d'une page HTML. Si vous avez un profil plutôt débutant en Java, il faudra tout de même vous armer d'un peu de courage. Ce livre n'est pas à prendre comme un tutoriel dans lequel on explique pas à pas l'utilisation des concepts. C'est un ouvrage qui se veut le plus exhaustif possible et qui occulte volontairement certains éléments de mise en place.

Le chapitre 8 se focalise sur l'intégration avec les serveurs d'application existants, notamment Spring et les EJB. Si vous ne connaissez pas du tout cette partie de Java, n'hésitez pas à vous aider des nombreux tutoriels disponibles sur le Web.

Structure de l'ouvrage

Le point essentiel de ce livre est qu'il n'y a aucun passage obligé. Excepté les trois premiers chapitres qui posent les bases du framework et présentent les outils, tous les autres chapitres peuvent être lus dans le désordre.

Les **chapitres 1, 2 et 3** abordent la partie visible de GWT, les composants, les outils, l'environnement de développement et le nouveau modèle de placement CSS.

Le **chapitre 4** est un tour d'horizon des différentes bibliothèques du marché.

L'intégration JavaScript (**chapitre 5**) est traitée comme préalable à l'implémentation de composants personnalisés (**chapitre 6**).

Les deux chapitres sur l'architecture des services RPC (**chapitre 7**) vous montrent comment exploiter efficacement la communication avec le serveur et intégrer des services existants Spring, EJB3 (**chapitre 8**)...

Le chargement à la demande (**chapitre 9**) est une des grosses nouveautés de GWT 2. Ce chapitre en illustre le principe à travers un exemple concret et vous présente un design pattern permettant de se prémunir contre ses pièges.

Le **chapitre 10** sur la liaison différée est complexe à comprendre mais important pour bien saisir les fondations de GWT et la manière dont les spécificités des différents navigateurs sont couvertes. Les débutants pourront faire l'impasse sur ce chapitre et y revenir dès qu'ils seront plus aguerris.

Les chapitres suivants, que ce soit la gestion des ressources (**chapitre 11**), l'internationalisation (**chapitre 13**) ou les tests (**chapitre 14**) peuvent être parcourus dans n'importe quel ordre.

En revanche, le **chapitre 15** sur les design patterns traite de sécurité, de gestion des sessions et surtout d'architecture avec les patterns MVC et MVP. Quant au **chapitre 12**, il met en lumière les innombrables vertus du compilateur et des optimisations GWT. Ces deux chapitres sont une source d'informations précieuses, même pour ceux ayant déjà une première expérience GWT.

Le livre se poursuit sur le **chapitre 16** qui décrit UIBinder, l'une des grandes nouveautés de GWT 2 et qui s'appuie intensivement sur le mécanisme des ressources, l'internationalisation et la liaison différée.

Curieusement, c'est le dernier **chapitre 17** qui est dédié au plugin Eclipse de GWT. Cela est essentiellement lié au fait que nous souhaitons vous présenter en priorité... tout ce qu'il est censé vous masquer.

Formation GWT proposée par DNG Consulting

Une très grande partie du contenu technique de ce livre s'appuie sur la formation GWT 2.0 proposée par DNG Consulting. Si vous souhaitez creuser le sujet de manière plus interactive nous vous encourageons vivement à faire partie des centaines de stagiaires formés depuis nos débuts avec GWT.

Remerciements

Il est coutume de dire que l'écriture est un exercice solitaire, c'est vrai. Mais ce livre n'aurait jamais pu voir le jour sans l'aide précieuse des quelques personnes que je souhaite remercier ici. Tout d'abord l'équipe éditoriale Eyrolles pour son professionnalisme et pour avoir accepté de me publier dans des conditions parfois difficiles : Muriel, Sophie, Anne-Lise, Pascale, Gaël et Éric.

Mes remerciements vont aussi à mes deux amis et précieux relecteurs, Romain Hochedez, consultant pour DNG Consulting et animateur de la formation GWT, et Jan Vorwerk, développeur GWT au sein de JDSU. Grâce à son œil de lynx, Romain m'a aidé à améliorer la qualité de certains chapitres, à éviter certains pièges. Jan m'a apporté ses précieux conseils techniques et pédagogiques. Il faut être plus qu'un vrai passionné pour accepter de passer soirées et week-ends à relire des manuscrits parfois indigestes comme ce fut le cas avec mes brouillons.

Difficile de ne pas remercier ceux qui ont créé GWT alors qu'ils n'étaient pas encore employés chez Google, Bruce Johnson et son équipe. La qualité principale de Bruce est d'avoir su instaurer autour de ce projet une convivialité quasi familiale. Que ce soit Joel, Ray Ryan, John, Fred ou Bob, tous m'ont aidé à corriger les bogues qui m'empêchaient d'avancer correctement dans ce livre.

Je remercie également les clients de DNG Consulting qui nous font confiance depuis trois ans pour les accompagner dans leurs projets GWT. Je dois l'expérience de ce livre à toutes les missions enrichissantes que DNG a pu assurer durant cette période.

Le dernier mot, enfin, est pour ma famille. Je ne souhaite à personne d'avoir à rédiger un livre de plus de 450 pages en trois mois tout en assurant la direction quotidienne d'un cabinet de conseil. Anatole France disait : « Il n'est pas d'amour qui résiste à l'absence ». Stéphanie, Yanis et Alicia m'ont prouvé le contraire. Sans eux, vous n'auriez jamais eu ce livre entre les mains.

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>	362
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>	390
Modèle vue présenteur (MVP)	392
<i>Le pattern Action orienté RPC</i>	399
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