

Reportage Jazoon 08



par Alain Sahli ([Tutoriels PHP de Alain Sahli](#)) ([Blog](#))

Date de publication : 21.07.2008

Ce reportage vous présente des résumés des conférences que j'ai suivies lors de Jazoon 08. Ce sont principalement des reportages qui traitent GWT et AJAX.

- I - Google Web Toolkit : Retour d'expérience
- II - Google Web Toolkit : Faire de l'AJAX plus rapidement qu'à la main !
- III - Pensé pour l'utilisateur final - Un cas pour GWT et Gears
- IV - Créer des applications Web RESTful avec AJAX et les Web Forms 2.0
- V - Liens et remerciements
- VI - Téléchargements

I - Google Web Toolkit : Retour d'expérience

M. Dmitry Buzdin a dû, au sein de son entreprise, évaluer GWT et nous a donc transmis ses conclusions concernant ce produit. Comme dans chaque présentation il nous a tout d'abord présenté GWT.

Tout d'abord ce qu'il faut savoir c'est que GWT est un toolkit pour créer des applications portables en AJAX. La couche de présentation est en Java et le même modèle de programmation que Swing. Il y a un compilateur *Java to JavaScript* ce qui permet de programmer en Java et ensuite obtenir du JavaScript. Le côté très pratique de GWT réside dans le fait que le code qu'il génère est compatible sur la plupart des navigateurs récents (cross-browser). GWT intègre également un mode de débogage en local.

Afin d'étudier correctement GWT, l'orateur a mis en place un prototype et a réalisé plusieurs tests pour évaluer l'utilisation de GWT en entreprise ainsi que les performances. Il a tout d'abord essayé de remplacer Struts et JSP par GWT. Les performances de génération du JavaScript ne sont pas très bonnes, la génération est lente. Par contre au niveau de la mémoire c'est très intéressant car GWT consomme jusqu'à huit fois moins de mémoire. C'est beaucoup plus facile de programmer en Java (surtout au niveau du débogage) qu'en JavaScript et les développeurs Java aiment GWT. Les seuls points qui manquent encore pour convaincre la majorité des entreprises sont la validation des entrées ainsi qu'une couche de *databinding* (association de données).

Au niveau du développement avec GWT les dépendances sont minimales, les couches sont propres et les composants sont réutilisables. Il y a un seul point d'entrée et le code peut-être partagé entre le client et le serveur (exemple : validation côté client). Côté client il y a un modèle MVC avec des *listeners*, des *widgets* et des *callbacks*. On peut l'intégrer avec d'autres technologies comme JQuery, JSP, Spring, Hibernate, Portlets, etc. Les tests unitaires ne sont pas faciles à écrire et peuvent poser pas mal de problèmes, quant au *screen clicking* cela demande beaucoup d'effort pour y parvenir ! Aucun bug critique n'a été découvert. Finalement il faut savoir que GWT est plus dur à prendre en main que Struts, qu'il faut bien réfléchir lors des requêtes asynchrones et faire attention à la sérialisation.

 *L'orateur a découvert que IE supporte deux requêtes simultanées !*

Pour conclure, l'orateur estime que GWT est assez mûr pour être utilisé dans de grandes entreprises et pour de gros projets ! De plus GWT 1.5 devrait fonctionner avec Java 6 et des frameworks pour GWT font gentiment leur apparition.

II - Google Web Toolkit : Faire de l'AJAX plus rapidement qu'à la main !

Cette présentation a été faite par Adrian Buerki et se situait juste après celle que j'ai résumé ci-dessus. La majorité des points abordés lors de cette conférence ont déjà été traités dans la première conférence, je ne vais donc que vous résumer les points qui ne sont pas abordés au-dessus.

L'orateur a traité un peu plus en détail la compilation *Java to JavaScript*. Un des grands avantages de cette compilation réside dans le fait que GWT génère du code JavaScript extrêmement optimisé. C'est-à-dire que le code en devient presque illisible car il n'y a aucun commentaire, aucune indentation syntaxique, et les variables ou fonctions sont le plus court possible (a, b, c, etc.). Cette optimisation permet de gagner passablement de place et rend les scripts beaucoup moins lourds.

Un des grands problèmes rencontrés lors de cette compilation est qu'au final votre application sera en JavaScript ; et dans ce langage la réflexion ainsi que le chargement dynamique des classes n'existent pas ! Or ces concepts sont très utiles pour optimiser les scripts car, dans le cas d'une application Web, l'utilisateur qui utilise Firefox n'a pas besoin de charger les parties de codes spécifiques à Opera ou IE. Pour remédier à ce problème, l'équipe de GWT a mis en place le *deferred binding* : ce mécanisme consiste à charger les classes durant la compilation et non pas durant l'exécution comme c'est le cas en Java. Donc si vous avez une application bilingue et qu'elle est compatible avec quatre navigateurs vous aurez 8 fichiers compilés (navigateur1_DE, navigateur1_FR, navigateur2_DE, etc.).

Pour finir l'orateur nous a fait une petite démo que vous pouvez voir à la fin du PDF de la présentation.

III - Pensé pour l'utilisateur final - Un cas pour GWT et Gears

Cette présentation a été faite par Maarten Volders qui nous a présenté GWT d'un point de vue utilisateur. Comme à chaque présentation de GWT il y a une petite partie d'introduction à GWT que je ne vais pas résumer...


Depuis que le Web existe les utilisateurs ont comme désir un chargement plus rapide des pages. De l'autre côté les développeurs font toujours des interfaces plus lourdes avec plus de fonctionnalités mais ne se concentrent que très peu sur le point essentiel : le poids de l'application. GWT essaye de satisfaire ces désirs en générant du code JavaScript le moins lourd possible et en utilisant AJAX pour ne mettre à jour que les parties qui en ont besoin. Il importe peu à l'utilisateur de savoir que le code est propre, bien structuré ou si l'application est lente...

Le but de GWT est :

- Réduire le nombre de requêtes HTTP ;
- Utiliser le plus possible le cache ;
- Rendre l'application multi-navigateur et multi-plateforme sans restrictions ;
- Optimiser le code JavaScript de manière extrême.

Vous remarquerez qu'une grande partie des ces buts touche un même point : rendre le chargement plus rapide ! Et c'est effectivement le cas, GWT est beaucoup plus rapide qu'une application AJAX classique. Ceci est notamment le cas grâce au *deferred binding*.

La présentation s'est terminée sur les *linker*, je dois avouer que ce sujet m'a complètement dépassé et que je ne suis pas capable de vous en résumer les points importants ! Les personnes qui connaissent bien le sujet peuvent lire la présentation à partir de la page 120.

 *Gears n'a pas été abordé durant cette conférence par manque de temps !*

IV - Créer des applications Web RESTful avec AJAX et les Web Forms 2.0

Lars Trierloff nous a tout d'abord présenté ce qu'est REST puis nous a parlé des Web Forms 2.0.

Alors que tout le monde parle d'AJAX comme la solution du Web 2.0, certaines personnes continuent de croire que l'avenir du Web réside dans REST. REST est un style d'architecture d'applications Web qui définit quelques standards simples. Je vais en citer un : HTTP fournit toutes les opérations nécessaires (GET, POST, DELETE et PUT). PUT serait la méthode idéale pour mettre à jour un profil ou poster une nouvelle entrée sur un blog tandis que POST est la pire méthode ! Il y a très peu d'applications qui respectent totalement REST car c'est trop difficile à respecter. Heureusement on peut contourner les difficultés grâce à AJAX.

Les Web Forms 2.0 ont été ma grande surprise de Jazoon 08 je n'en avais jamais entendu parler et je dois dire que c'est très alléchant ! Ils permettent de construire des formulaires avec des fonctions beaucoup plus poussées qu'actuellement. Ils supportent les méthodes PUT et DELETE, ce qui n'est pas le cas des formulaires actuels. Les Web Forms 2.0 sont un **standard** Web défini par le W3C et seul Opera 9.5 supporte ces formulaires... Comme grande amélioration il y a une validation côté client qui est directement intégrée ainsi que des répéteurs. Je vais vous faire un exemple ci-dessous :

Validation intégrée


```
<input type="number" min="10">
<input type="range" min="10" max="100"step="10">
<input type="text" pattern="[A-Z][a-z]+">
<textarea maxlength="140" oninvalid="alert('Make it short, this is twitter');return
false"></textarea>
```

Comme vous pouvez le voir, on peut enfin définir un type pour nos *input* et assurer une validation côté client. Imaginez le temps que l'on gagne par rapport à une validation de formulaire en AJAX...

Répéteurs

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
<head>
<title>Sample Order Form</title>
</head>
<body>
<form>
<table>
<tr>
<th>Product</th>
<th>Quantity</th>
</tr>
<tr id="order" repeat="template" repeat-start="3">
<td><input type="text" name="row[order].product" value=""></td>
<td><input type="text" name="row[order].quantity" value="1"></td>
<td><button type="remove">Remove This Row</button></td>
</tr>
</table>
<p><button type="add" template="order">Add Row</button></p>
<p><button type="submit">Submit</button></p>
</form>
</body>
</html>
```

Grâce aux répéteurs, vous n'aurez plus besoin de faire cent copier-coller si vous avez un tableau à cent lignes... Vous remarquerez également avec quelle simplicité on a ajouté un bouton pour ajouter ou supprimer une ligne du tableau. Et tout ceci est possible sans JavaScript !

 Ces fragments de codes viennent du site W3C vous pouvez les consulter à cette adresse





:  **Web Forms 2.0.**

Jazoon 08 était une très bonne conférence avec des gens compétents et une très bonne organisation ! J'ai particulièrement apprécié la présentation de Maarten Volders qui l'a rendu très dynamique.

V - Liens et remerciements









Je tiens à remercier **Ricky81** pour la relecture de cet article !

Je vous donne également quelques liens en rapport avec le reportage :

-  **Mon blog avec les billets concernant Jazoon 08**
-  **Le site de Jazoon**
-  **Les cours Java**
-  **Le forum Java**

VI - Téléchargements

Ci-dessous vous avez les différentes présentations sous format PDF :

-  **Google Web Toolkit : Retour d'expérience** ou  **ici**
-  **Google Web Toolkit : Faire de l'AJAX plus rapidement qu'à la main !** ou  **ici**
-  **Pensé pour l'utilisateur final - Un cas pour GWT et Gears** ou  **ici**
-  **Créer des applications Web RESTful avec AJAX et les Web Forms 2.0** ou  **ici**

