



Réduire la taille de votre code

Réduire la taille des exemples de test

Pour aider les développeurs à résoudre un problème, ils doivent d'abord le comprendre. Pour ce faire, ils ont besoin de comprendre tout ce qui se passe dans une situation où le problème est reproductible. Souvent, le problème est seulement reproductible dans un certain type de situation, ou dans une situation peu utilisée qui n'est pas exploitée par le code couramment utilisé. (En outre, certains problèmes sont le résultat d'une erreur de l'utilisateur).

Dans le but d'aider les développeurs à vous aider, la meilleure chose à faire est de circonscrire les erreurs dans *la plus petite portion de code qui peut les provoquer*. De plus, lors des tentatives de reproduction un développeur devra définir le code afin qu'il puisse être lancé dans l'environnement de test du développeur. Cela signifie qu'il est idéal de supprimer les références externes aux autres fichiers JavaScript, et fichiers externes lorsque c'est possible (clairement cela n'est pas toujours possible : les bugs de serveurs WFS ne peuvent pas toujours être démontrés dans une seule page, par exemple – mais vous devez réduire les dépendances externes autant que possible).

Une fois que cela a été fait, vous devez *supprimer toutes les lignes de code inutiles* de votre exemple. Est ce que le problème nécessite le contrôleur ScaleBar pour se manifester ? Si non enlevez-le. En clair, toute ligne de code qui n'est pas lié directement à la reproduction du problème doit être supprimée, puisque chaque ligne devra être lue par le développeur – et dans le cas où il y a plusieurs développeurs travaillant sur le problème, lue par *chaque* développeur – dans le but de déterminer si le problème est lié à celles-ci.

Cette étape de simplification doit inclure la suppression de toutes les parties inutiles en JavaScript, fichiers CSS, HTML, etc. jusqu'à ce que le code résultant soit aussi petit que possible.

Souvent, en faisant cela, vous allez tomber sur une étape de simplification qui va supprimer le problème. C'est bon signe car cela signifie que vous avez mis le doigt sur l'aspect du code qui provoque le problème. Remettez-le et continuez la simplification.

De plus, en agissant ainsi, vous trouverez une construction particulière qui vous aide à comprendre comment contourner le problème.

Si ce n'est pas le cas, continuez à la section suivante.

Références à la bibliothèque d'OpenLayers

Il y a plusieurs versions hébergées de la bibliothèque d'OpenLayers.

<http://openlayers.org/api/OpenLayers.js>

Ceci représentera toujours la version 'stable' la plus récente de l'API d'OpenLayers.

<http://openlayers.org/dev/OpenLayers.js>

C'est toujours une compilation du trunk d'OpenLayers 10 minutes plus tôt.

Pour aider les développeurs à configurer leur propre environnement de test, il est souvent bénéfique de pointer directement vers une de ces URL. De plus, cela indiquera que le problème n'est pas spécifique à votre compilation d'OpenLayers.

Publier votre problème

Une fois que vous avez simplifié votre cas d'étude, vous devez le publier. En général, cela est plus facile si vous publiez une page HTML avec une URL accessible par le Web. Même si votre projet n'est pas encore public, vous pouvez probablement placer une page sur un autre serveur qui montre le problème. En faisant cela, il est plus probable qu'un développeur suive le lien et regarde votre problème. Cela est *spécialement* vrai pour les problèmes liés aux WFS qui demandent la configuration d'un proxy ; télécharger la page, mettre en place un proxy et tester localement représente beaucoup de travail pour un développeur pour simplement confirmer qu'un problème existe.

Si vous n'avez *aucun* endroit pour publier des pages web, vous pouvez tenter de copier votre code dans un site public comme 'nopaste.com'. Cependant, soyez assuré qu'en faisant cela un développeur doit réaliser plus d'étapes pour reproduire votre problème – et chaque étape supplémentaire éloigne probablement d'une résolution rapide et aisée du problème.

Communiquer sur votre problème

La meilleure façon de communiquer votre problème est d'envoyer un courriel à la liste des utilisateurs expliquant le problème. Parfois les autres utilisateurs seront capables de pointer un endroit particulier dans votre code qui cause l'erreur, ou expliqueront que le comportement est un manque de fonctionnalité connu dans OpenLayers.

Soyez clair sur les étapes de la reproduction. Les utilisateurs qui ne savent pas ce qu'ils sont supposés faire pour provoquer le bug ne seront pas capables de le voir, et s'ils ne le voient pas, ils ne peuvent pas vous aider.

Soyez clair sur le comportement attendu. Souvent, la raison qui vous amène à penser que cela ne fonctionne pas n'est pas claire, parce que le comportement que vous voyez est ce à quoi il faut s'attendre ou le problème n'est pas évident. Expliquez ce que vous vous attendez à voir.

Si vous avez déterminé la modification particulière dans le code source d'OpenLayers qui est nécessaire pour modifier le comportement, alors il est plus que probable que la liste Développeurs soit le meilleur endroit où aller. Toute discussion qui implique le code d'OpenLayers lui-même est probablement plus pertinente sur la liste dev.

Enfin,

En suivant ces étapes :

- Simplifier/Réduire
- Publier
- Communiquer

(si vous préférez, vous pouvez rajouter un "???", Profiter !" à la fin de ceci.)

Vous pouvez ainsi vous assurer que le développeur détermine, aussi facilement que possible, si le problème que vous avez provient de la bibliothèque. Vous pouvez également permettre plus facilement, pour les développeurs et les utilisateurs, la découverte des problèmes potentiels dans votre utilisation de la bibliothèque et la proposition de solutions. Enfin, vous pouvez trouver vous-même le bug lors ce processus, ce qui permet d'économiser votre temps et celui des autres.

Le résultat final est un système plus fonctionnel pour tous. Plus il est facile de comprendre le

problème que vous avez, plus cela sera rapide et facile d'obtenir de l'aide.