



Déploiement

OpenLayers est fournie avec des exemples tout prêts et pré-configurés : téléchargez simplement une version d'OpenLayers et vous avez un ensemble complet d'exemples simples à utiliser. Cependant, ces exemples sont conçus pour être utilisés en mode développement. Lorsque vous êtes prêt à déployer votre application, vous désirez une distribution d'OpenLayers fortement optimisée pour limiter la bande passante et le temps de chargement.

Bibliothèque en un seul fichier

OpenLayers a deux différents types d'usage : en fichier unique, où tout le code Javascript est compilé en un seul fichier, `OpenLayers.js`, et la version de développement, où les fichiers Javascript sont tous chargés au moment de démarrer l'application. Le fichier seul compilé prend un ensemble de fichiers Javascript d'OpenLayers, les ordonne en fonction des dépendances décrites dans les dossiers, et compresse le fichier obtenu, en utilisant la librairie de compression `jsmin`.

Construire une version fichier unique de la bibliothèque OpenLayers modifie légèrement le comportement de la bibliothèque : par défaut, la version de développement d'OpenLayers s'attend à être placée dans un répertoire appelé "lib", et espère que les images et CSS soient placées dans le répertoire au-dessus du fichier `OpenLayers.js`

```
img/pan-hand.png
theme/default/style.css
lib/OpenLayers.js
lib/OpenLayers/Map.js
...
```

Toutefois, lors du déploiement d'un fichier unique compilé d'OpenLayers, il est prévu que la bibliothèque soit plutôt au même niveau que les répertoires des thèmes et `img`

```
OpenLayers.js
theme/default/style.css
```

img/pan-hand.png

...

Compiler la bibliothèque en un seul fichier

Les outils de compilation en fichier unique sont déployés avec une version d'OpenLayers dans le répertoire "build". Ils nécessitent Python pour compiler.

Sous Linux et autres systèmes d'exploitation similaires, étant donné que OpenLayers est stockée dans le répertoire 'OpenLayers', un seul fichier compilé pourrait être créé avec les commandes suivantes:

```
cd openlayers/build
./build.py
```

Cela créera un fichier appelé "OpenLayers.js" dans le répertoire build qui contient tout le code de la bibliothèque pour votre fichier unique compilé d'OpenLayers.

Sous Windows, à partir du menu Démarrer, sélectionnez Exécuter. Copiez le chemin d'accès vers build.py de la barre d'adresse de la fenêtre d'explorateur dans la zone de texte, puis ajoutez le nom du fichier de configuration (ou un blanc par défaut) :

C:\Downloads\OpenLayers-2.6\build\build.py lite

Profils de compilation personnalisés

Afin d'optimiser l'expérience des utilisateurs finaux, la distribution d'OpenLayers comprend des outils qui vous permettent de construire votre propre version du code en un fichier unique. Ce code utilise un fichier de configuration pour choisir quels fichiers doivent être inclus dans la compilation : de cette manière, pour un usage en production, vous pouvez supprimer des classes de votre fichier JavaScript de la bibliothèque d'OpenLayers qui ne sont pas utilisées dans votre application.

OpenLayers est distribuée avec deux configurations standards pour créer une version en fichier unique :

full :

C'est la compilation complète avec tous les fichiers.

lite:

Ce fichier inclus un petit ensemble du code d'OpenLayers, conçu pour être intégré dans une autre application. Il inclus seulement les types Layers nécessaire pour créer des WMS tuilés ou non tuilés, et n'inclut aucun contrôleur. C'est le résultat de ce qui était appelé "Webmap.js" au temps du BOF Web Mapping lors du FOSS4G 2006.

Les profils sont simples à créer. Vous pouvez commencer par copier library.cfg ou lite.cfg en autre chose, par exemple myversion.cfg dans le répertoire de compilation.

Le début des profils de compilation doit inclure la même section [first] utilisée dans le fichier lite.cfg:

```
[first]
OpenLayers/SingleFile.js
OpenLayers.js
OpenLayers/BaseTypes.js
OpenLayers/BaseTypes/Class.js
OpenLayers/Util.js
```

Ces fichiers sont nécessaires pour que la compilation d'OpenLayers fonctionne.

Une fois que vous avez inclu ces fichiers, vous devez ajouter d'autres fichiers dans la section '[include]' du fichier. Les fichiers répertoriés ici devraient être la liste des fichiers contenant toute classe que vous utiliserez dans votre application. Vous pouvez généralement trouver ces classes en regardant, dans votre code, tous les cas où "new OpenLayers.ClassName()" est utilisée.

En prenant l'exemple 'lite.html', nous constatons qu'il existe deux 'nouveaux' états dans le fichier : un pour la classe OpenLayers.Map, et un pour la classe OpenLayers.Layer.WMS. Nous ajoutons ensuite les fichiers correspondant à notre section include

```
[include]
OpenLayers/Map.js
OpenLayers/Layer/WMS.js
```

Une fois que nous l'avons fait, nous pouvons construire notre profil en ajoutant le nom du profil à la fin de notre commande de compilation précédente

```
./build.py myversion
```

Cela va créer une version d'OpenLayers beaucoup plus petite, adaptée pour des applications limitées.

Presque toutes les applications peuvent bénéficier d'un profil de compilation personnalisé. OpenLayers gère de nombreux types de couches différents, mais la plupart des applications en utilisera une ou deux seulement, et de nombreuses applications n'ont pas besoin de la gestion complète de beaucoup de fonctionnalités d'OpenLayers. Afin de limiter votre temps de téléchargement et la taille de la bibliothèque, la compilation d'un profil personnalisé est fortement recommandée avant de déployer une application OpenLayers : il peut aider à réduire la taille de votre bibliothèque d'un facteur cinq par rapport à une utilisation de la bibliothèque complète.

Fichiers de déploiement

Dans le but de déployer OpenLayers, il y a plusieurs choses différentes que vous devez déployer :

OpenLayers.js

La bibliothèque. Elle fournit le code JavaScript que votre application utilise.

le répertoire theme

Le répertoire theme contient les fichiers images et CSS pour les contrôleurs les plus récents, dont le style et le positionnement est contrôlé entièrement par CSS.

le répertoire img

Ce répertoire fournit les images qui sont utilisées par certains contrôleurs, comme le contrôleur PanZoom, qui n'utilise pas les CSS pour son style.

Comme décrit ci-dessus, lors du déploiement de ces fichiers avec un fichier unique compilé d'OpenLayers, ils doivent tous être dans le même répertoire : ceci permet à OpenLayers de les trouver correctement et de les inclure.