

# Créer une extension

*Par Yan Morin*

## Sommaire

Le but de ce tutoriel est de créer une extension pour Firefox 0.9 qui nous dis "Allo" à chaque démarrage du navigateur et qui permet de modifier son nom à l'aide d'un item dans la barre de menu.

### *Sommaire*

1. Créer la structure (répertoire)
2. Générer le fichier d'installation
3. Générer le menu
4. Écrire le code javascript
5. Archiver le .jar
6. Archiver le .xpi
7. Distribuer l'extension
8. Mot final

Pour exécuter le tout, nous avons besoin de:

- 1 gestionnaire de fichier pour créer les répertoires.
- 1 éditeur texte
- 1 programme d'archivage PKZIP (Winzip, zip, ark, file-roller).
- 1 site web pour distribuer notre extension.

## 1. Créer la structure (répertoire)

Premièrement, il faut trouver un nom pour merveilleuse petite extension. Appelons la "allotoi".

Voici la structure que je vous propose de créer:

```
allotoi/ // Racine de tout les fichiers,
nécessaire au paquetage .xpi // Installation de l'extension
allotoi/fichier.xpi // Configuration pour
allotoi/install.rdf // Extension en tant que telle
l'installation // Archive qui contient notre
allotoi/chrome/ // Formulaires, les menus et les
allotoi/chrome/allotoi.jar // Spécifique à notre extension
extension, présent chez les clients // Information sur les fichiers de
allotoi/chrome/content/ // Notre !!!Extension!!!
fichiers scripts // Notre code javascript
allotoi/chrome/content/allotoi/ // Ensemble des étiquettes texte
allotoi/chrome/content/allotoi/contents.rdf // Les contents.rdf sont
notre extension obligatoires
allotoi/chrome/content/allotoi/allotoi.xul // Étiquette texte anglaise pour
allotoi/chrome/content/allotoi/allotoi.js // notre extension
allotoi/chrome/locale/en-US/allotoi/ // Thème et fichier CSS
anglaise // Lui aussi est obligatoire
allotoi/chrome/locale/en-US/allotoi/contents.rdf // Fichier css particulier à notre
obligatoires // extension
allotoi/chrome/locale/en-US/allotoi/allotoi.dtd //
notre extension //
allotoi/chrome/skin/classic/allotoi/ //
allotoi/chrome/skin/classic/allotoi/contents.rdf //
allotoi/chrome/skin/classic/allotoi/allotoi.css //
```

C'est beaucoup de fichiers vous ne trouvez pas? Et bien, dans une extension normale, il y en a encore plus... Des fichiers css pour nos fichiers, des images, des formulaires apropos, etc.

Voici un script bash pour tout créer:

```
#!/bin/bash
[ ! -e $1 ] || exit 1
packagename=$1
lang="en-US"

# creation des répertoires
mkdir $packagename/
mkdir $packagename/chrome/
mkdir $packagename/chrome/content/
mkdir $packagename/chrome/content/$packagename/
mkdir $packagename/chrome/locale/
mkdir $packagename/chrome/locale/$lang/
mkdir $packagename/chrome/locale/$lang/$packagename/
mkdir $packagename/chrome/skin/
mkdir $packagename/chrome/skin/classic/
mkdir $packagename/chrome/skin/classic/$packagename/
```

```
# creation des fichiers vides
touch $packagename/install.rdf
touch $packagename/chrome/content/$packagename/contents.rdf
touch $packagename/chrome/content/$packagename/$packagename.xul
touch $packagename/chrome/content/$packagename/$packagename.js
touch $packagename/chrome/locale/$lang/$packagename/contents.rdf
touch $packagename/chrome/locale/$lang/$packagename/$packagename.dtd
touch $packagename/chrome/skin/classic/$packagename/contents.rdf
touch $packagename/chrome/skin/classic/$packagename/$packagename.css
```

Je nomme habituellement ce script structure.sh et je l'exécute avec la commande bash structure.sh alltoi

Une fois les répertoires et les fichiers créés on peut commencer notre extension.

## 2. Générer le fichier d'installation

Dans le nouveau Firefox 0.9 et dans le nouveau Thunderbird 0.7, il faut créer un fichier "install.rdf" pour installer notre extension. Une fois créé, le gestionnaire d'extension de Firefox ou de Thunderbird l'utilisera pour installer notre extension.

C'est un fichier RDF (xml) avec tout sorte de section étrange. Mais ne vous en faites pas, j'ai un générateur de fichier "install.rdf" pour vous aider. Entrer "Allo toi" pour le nom de l'extension, "allotoi" pour le nom du paquetage, générer un nouveau GUID aléatoire et entrer le numéro de version 0.0.1. Choisissez Firefox 0.9 à 1.0 et laissez I18N à en-US. Entrer votre nom d'auteur et une description si cela vous chante. Enfin, cliquer sur le bouton "Générer install.rdf". Une fois le code du fichier généré, copier le résultat dans le fichier "allotoi/install.rdf" déjà créé à l'étape 1. Ne fermer la page. Nous en aurons de besoin pour le fichier contents.rdf.

## 3. Générer le menu

### 3.1 Les fichiers contents.rdf

Pour insérer, exécuter ou voir notre extension, le navigateur à besoin des fichiers contents.rdf.

On peut facilement le générer à partir encore d'une fois de mon générateur. J'espère que

vous avez laisser les informations.

### ***Répertoire "content/"***

Le nom du paquetage (allotoi) est obligatoire. De plus, si l'on veut insérer notre extension dans le navigateur, il faut cocher la case "Insérer dans browser.xul" et l'option "Répertoire content/". Une fois renseigné, cliquer sur le bouton "Générer contents.rdf".

Enfin, copier-coller le code généré dans le fichier allotoi/chrome/content/allotoi/contents.rdf

### ***Répertoire "locales/"***

Ensuite, il faut générer le contents.rdf du répertoire "locale/". On utilise la même technique qu'en 3.1, mais on choisit l'option "Répertoire locale/". On enregistre dans le fichier allotoi/chrome/locale/en-US/allotoi/contents.rdf.

### ***Répertoire "skin/" (le dernier et non le moindre)***

On choisit "Répertoire skin/" avec l'option "Insérer dans browser.xul", on copie et on insère dans le fichier allotoi/chrome/skin/classic/allotoi/contents.rdf

## ***3.4 Écrire notre extension: allotoi.xul***

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin" type="text/css"?>
<?xml-stylesheet href="chrome://allotoi/skin/allotoi.css" type="text/css"?>

<!DOCTYPE overlay SYSTEM "chrome://allotoi/locale/allotoi.dtd">

<overlay id="webdeveloper"
  xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">

  <script type="application/x-javascript"
src="chrome://allotoi/content/allotoi.js"/>
  <!--
    on insère par dessus (donc à la fin) de la barre de menu
    ça m'a pris un gros 2 heures avant de le comprendre...
  -->
  <menubar id="main-menubar">
```

```
        <menuitem label="Allo Toi Options..." accesskey="A"
oncommand="allotoi_options();" />
    </menubar>
</overlay>
```

Voici quelques explications:

```
xmlstylesheet href="chrome://global/skin"
    Permet d'utiliser les feuilles de styles globales
```

```
xmlstylesheet href="chrome://allotoi/skin/allotoi.css"
    Permet d'utiliser notre propre feuilles de styles. Remarquer que le chemin est raccourci. En effet, sur le disque dur le fichier est "allotoi/chrome/skin/classic/allotoi/allotoi.css", mais dans la hiérarchie chrome, le répertoire devient: "chrome://allotoi/skin/allotoi.css".
```

```
chrome://allotoi/locale/allotoi.dtd
    Ce fichier peut contenir des ENTITY accessibles par la syntaxe &Nom.Element;. Dans le même style que les entités &lt; ou &gt; ou &nbsp;
```

```
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul"
    L'espace de nom de tous les fichiers XUL.
```

```
<overlay id="webdeveloper"></overlay>
```

Un overlay permet de surcharger ou ajouter des fonctionnalité à une interface contrairement à une balise <window> qui crée une fenêtre d'application.

```
<script type="application/x-javascript" src="chrome://allotoi/content/allotoi.js">
    Cette ligne permet de charger du code javascript (donc comme en HTML ou XHTML).
```

```
menubar id="main-menubar"
```

main-menubar est la barre de menu principal de Firefox (celle avec File, Edit, View, ...). L'overlay permet de la surcharger. Les autres options ne seront donc pas supprimées.

```
menuitem
```

menuitem est notre simple item qui nous permettra de changer notre nom.

## 4. Écrire le code javascript

Après avoir générer notre menu et insérer notre code javascript sur l'item du bouton avec la

commande `allotoi_options()`, il est temps d'écrire notre petit `allo`.

Dans le fichier `allotoi/chrome/content/allotoi/allotoi.js`, on va écrire:

```
allotoi_init(); /* à chaque chargement de l'extension (du navigateur) */

/**
 * Regarde pour le nom dans les préférences et dit allo lors du chargement de
 l'extension.
 */
function allotoi_init() {
    const preferencesService = Components.classes["@mozilla.org/preferences-
service;1"].getService(
Components.interfaces.nsIPrefService ).getBranch("");
    var nom = '';

    if(preferencesService.prefHasUserValue("allotoi.nom"))
    {
        nom = preferencesService.getCharPref("allotoi.nom");
    } else {
        nom = allotoi_options();
    }

    if ( nom != "" ) {
        alert('Allo ' + nom );
    }
}

function allotoi_options() {
    const preferencesService = Components.classes["@mozilla.org/preferences-
service;1"].getService(
Components.interfaces.nsIPrefService ).getBranch("");
    var nom = '';

    if(preferencesService.prefHasUserValue("allotoi.nom"))
    {
        nom = preferencesService.getCharPref("allotoi.nom");
    }
    nom = window.prompt('Entrez votre nom',nom,'Allo toi: Options');
    preferencesService.setCharPref("allotoi.nom",nom);
    return nom;
}
```

## 5. Archiver le .jar

Une fois que le tout est terminé, il ne reste qu'à archiver le tout dans un fichier `.jar`.

Il faut archiver les dossiers dans le répertoire `allotoi/chrome/` sans le `chrome/`. Le fichier `allotoi/chrome/allotoi.jar` doit contenir:

```
/content/  
/content/allotoi/  
/content/allotoi/contents.rdf  
/content/allotoi/allotoi.xul  
/content/allotoi/allotoi.js  
/locale/en-US/allotoi/  
/locale/en-US/allotoi/allotoi.dtd  
/locale/en-US/allotoi/contents.rdf  
/skin/classic/allotoi/  
/skin/classic/allotoi/allotoi.css  
/locale/en-US/allotoi/contents.rdf
```

En ligne de commande j'ai utilisé (dans le répertoire allotoi/chrome/):

```
zip -r allotoi.jar *
```

## 6. Archiver le .xpi

Après avoir compressé nos fichiers, il faut créer notre fichier d'installation compressé qui contient notre fichier de configuration install.rdf et notre fichier allotoi.jar.

Donc dans le répertoire allotoi/

```
zip allotoi.xpi install.rdf chrome/allotoi.jar
```

## 7. Distribuer l'extension

Finalement, après toutes ces étapes, on peut essayer notre extension. On peut la mettre sur un serveur web et utiliser ce code javascript pour l'installer:

```
<a href="allotoi.xpi"  
    onclick="xpi={'Allo toi':this.href};InstallTrigger.install(xpi);return  
false;">  
Installer mon extension allotoi.xpi </a>
```

## 8. Mot final

Étrangement, j'ai eu quelques erreurs pour la création de la préférence allotoi.moi au

démarrage la première fois. Cependant, si on la crée après avoir chargé le navigateur (ou avant de redémarrer le navigateur lorsqu'on installe l'extension), tout est correct. J'ai beaucoup appris sur XUL, les principes d'overlay et Firefox a beaucoup mais beaucoup de potentiel! J'espère que quelqu'un pourra écrire un script encore plus efficace que le mien où un formulaire et un seul clic pourra générer le tout. La modification de l'interface de Firefox est sans limite, mais il manque sérieusement un bon IDE pour faire des tâches aussi simple qu'ajouter une option.

