

Applet lançant une application

Cette API sert à lancer un jar exécutable à partir d'un Applet.

En fait, l'Applet a le comportement de Java Webstart, c'est à dire : copie des fichiers en local suivi du lancement du jar exécutable. Elle offre en plus la possibilité de copier d'autres fichiers après le lancement de l'application. Et elle offre en plus de la webstart la possibilité de personnaliser sa barre de progression de copie à volonté. La seule chose pour que l'Applet fonctionne, il faudra signer le jar la contenant vous même. En effet vous ne pouvez pas utiliser ma signature, elle n'a qu'une valeur limitée dans le temps, de plus elle n'est pas validée par un organisme. Pour signer l'Applet regardez dans la FAQ General Java du site www.developpez.com par exemple.

Son utilisation est assez simple, vous devez :

- 1) Créer un fichier texte listant tous les fichiers à copier sur le disque local. Chaque ligne du fichier texte étant le chemin relatif de la ressource à copier par rapport où se trouvera le fichier Launch.jar sur votre serveur. Ne pas oublier dans cette liste Launch.jar lui même ainsi des fichiers dont il a besoin. Ni oublier le jar à exécuter ainsi que toutes les ressources dont il a besoin au minimum pour se lancer.
- 2) Créer un fichier texte contenant les ressources supplémentaires. Chaque ligne du fichier texte étant le chemin relatif de la ressource à copier par rapport où se trouvera le fichier Launch.jar sur votre serveur. Ce fichier n'est pas obligatoire.
- 3) Créer un fichier texte nommé « jarList.txt », dont le lanceur se servira pour lancer votre application. Chaque ligne du fichier texte étant le chemin relatif du jar dans le dossier copié
- 4) Créer un fichier texte nommé « mainClass.txt », fichier d'une ligne contenant le nom de la classe principale à lancer dans le jar. Le package complet doit être précisé. Ne mettez pas d'extension (Ni « .class », ni « .java »)
- 5) Créer un fichier texte nommé « nativePath.txt », si vous avez des librairies (DLL). Une ligne précisant le chemin des natives. Bien sûr ce n'est pas obligatoire.

Une fois vos fichiers textes prêts, vous pouvez créer votre html, extrait du html (exemple) :

```
<applet archive="Launcher.jar" code="jackHack.applet.AppletLauncher" codebase="." width=100 height=100>
  <param name="needToCopyListPath" value="toCopy.txt">
  <param name="directoryWhereToCopy" value="JackHack">
  <param name="otherToCopyListPath" value="other.txt">
</applet>
```

Le paramètre `needToCopyListPath` permet de spécifier le fichier décrivant la liste des ressources à copier. Voir l'étape 1. Obligatoire !

Le paramètre `directoryWhereToCopy` permet de spécifier le nom du dossier où se fait la copie. Si vous ne le mettez pas, sera utilisé le dossier par défaut.

Le paramètre `otherToCopyListPath` permet de spécifier le fichier décrivant la liste des autres ressources à copier. Voir l'étape 2. Facultatif.

Pour un exemple regardez le contenu du dossier exemple.

Jack Hack