Créer votre propre système de mise à jour en Java

par Baptiste Wicht (home)

Date de publication : 24 Juillet 2006

Dernière mise à jour :

Avec ce tutoriel, vous allez apprendre à créer votre propre système de mise à jour en Java. Ainsi, vous pourrez ajouter une simple option dans votre application pour permettre à l'utilisateur de mettre à jour son application.

Copyright © 2006 Baptiste Wicht . Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts

I - Introduction

Nous allons réaliser ce système en Java, mais il est tout à fait possible d'adapter mon protocole dans la quasi-totalité des langages de programmation.

Tout d'abord, avant de commencer à coder le moindre code, il nous faut définir un "protocole" de mise à jour. Ce "protocole devra définir les différentes étapes dans la mise à jour de notre programme".

Voilà le protocole que je propose pour ce tutoriel :

1. On récupère les versions disponibles sur internet
2. On les compare avec la version actuelle
3. Si la version actuelle est plus récente, on affiche un message disant à l'utilisateur qu'il n'y a pas de mise à jour possible
4. Sinon, on informe l'utilisateur des versions possibles et on lui donne le choix quand à laquelle télécharger.
5. S'il veut en télécharger une, on télécharge celle qu'il a choisie
6. Ensuite, on redémarre le programme
7. Le programme au démarrage doit voir s'il y a une nouvelle version, et le cas échéant, la prendre ==> utilisation d'un lanceur.

Je n'ai pas la prétention de dire que ce "protocole" est le meilleur, mais c'est celui que l'on va employer pour ce tutoriel, rien ne vous empêche d'ailleurs de le modifier.

Avant de mettre en oeuvre notre protocole, on va d'abord apprendre à réaliser les opérations compliquées de ce tutoriel et ensuite mettre ensemble le tout.

Copyright © 2006 Baptiste Wicht . Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts

II - Récupération de la version

Pour qu'on puisse récupérer la dernière version, il faut que celle-ci soit stockée quelque part et accessible. Quoi de plus accessible qu'internet ? Et quoi de plus pratique qu'un fichier XML ? On va donc employer un fichier XML hébergé sur internet. Ce fichier devra contenir en tout cas un champ contenant la version :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<versions>
 <version>
  <nom>Numéro de version</nom>
  <files>
   <file>
    <url>Chemin vers le fichier</url>
    <destination>Destination relative</destination>
   </file>
  </files>
 </version>
</versions>
```

Ainsi, avec un fichier XML pareil, vous pouvez avoir plusieurs versions disponibles sur internet, qui peuvent demander plusieurs fichiers à télécharger.

Voilà, vous avez maintenant une page avec les dernières versions à jour de votre application, mais maintenant, il s'agit de la récupérer ces version.

On va donc récupérer le flux de ce fichier XML et le parser avec l'Api JDOM (si vous voulez vous familiariser avec cet api, je ne peut que vous conseiller **ce tutoriel**). On va donc récupérer toutes les versions disponibles.

```
//Chemin vers le fichier XML
private String xmlPath = "Lien vers fichier XML";

//Document xml
private Document xmlDocument = null;

private ArrayList<String> getVersions(){
 ArrayList<String> versions = new ArrayList<String>();

 try {
  URL xmlUrl = new URL(xmlPath);

  //On ouvre une connections sur la page
  URLConnection urlConnection = xmlUrl.openConnection();
  urlConnection.setUseCaches(false);

  //On se connecte sur cette page
  urlConnection.connect();

  //On récupère le fichier XML sous forme de flux
  InputStream stream = urlConnection.getInputStream();

  SAXBuilder sxb = new SAXBuilder();

  //On crée le document xml avec son flux
  try {xmlDocument = sxb.build(stream);
  } catch (JDOMException e) {e.printStackTrace();
  } catch (IOException e) {e.printStackTrace();}

  //On récupère la racine
  Element racine = xmlDocument.getRootElement();
```


Copyright © 2006 Baptiste Wicht . Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts

```
    //On liste toutes les versions
    List listVersions = racine.getChildren("version");
    Iterator iteratorVersions = listVersions.iterator();

    //On parcourt toutes les versions
    while(iteratorVersions.hasNext()){
     Element version = (Element)iteratorVersions.next();

     Element elementNom = version.getChild("nom");

     versions.add(elementNom.getText());
    }

    //On trie la liste
    Collections.sort(versions);

   } catch (MalformedURLException e) {
    e.printStackTrace();
   } catch (IOException e) {
    e.printStackTrace();
   }

   return versions;
  }
```

Voilà, vous arrivez maintenant à récupérer la dernière version disponible. Vous avez remarqué qu'on ne fait pas de test dans le cas ou la version n'a pas pu être récupérée, mais on fera ces tests, quand on récupéra la dernière version.

Copyright © 2006 Baptiste Wicht . Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts

III - Télécharger la dernière version

Maintenant que l'on a récupéré la dernière version disponible, il faut aussi bien entendu la télécharger.

Donc il faut aussi que votre fichier .jar soit disponible sur le serveur, vous devrez donc aussi l'héberger quelque part.

Pour récupérer ce fichier sur internet, on va donc de nouveau établir une connexion par URL et cette fois on va lire l'entier du fichier. Et ensuite écrire tous ces bits dans le fichier en local, dont le chemin est passé en paramètre. Il faudra juste nommer d'une façon différente le fichier qu'on vient de téléchargé et le fichier sur lequel on travaille actuellement.

```
private void downloadFile(String filePath, String destination) {
 URLConnection connection = null;
 InputStream is = null;
 FileOutputStream destinationFile = null;

 try {
  //On crée l'URL
        URL url = new URL(filePath);

  //On crée une connection vers cet URL
  connection = url.openConnection( );

  //On récupère la taille du fichier
  int length = connection.getContentLength();

  //Si le fichier est inexistant, on lance une exception
  if(length == -1){
   throw new IOException("Fichier vide");
        }

  //On récupère le stream du fichier
  is = new BufferedInputStream(connection.getInputStream());

  //On prépare le tableau de bits pour les données du fichier
  byte[] data = new byte[length];

  //On déclare les variables pour se retrouver dans la lecture du fichier
  int currentBit = 0;
  int deplacement = 0;

  //Tant que l'on n'est pas à la fin du fichier, on récupère des données
  while(deplacement < length){
   currentBit = is.read(data, deplacement, data.length-deplacement);
   if(currentBit == -1)break;
   deplacement += currentBit;
  }

  //Si on n'est pas arrivé à la fin du fichier, on lance une exception
  if(deplacement != length){
   throw new IOException("Le fichier n'a pas été lu en entier (seulement "
    + deplacement + " sur " + length + ")");
  }

  //On crée un stream sortant vers la destination
  destinationFile = new FileOutputStream(destination);

  //On écrit les données du fichier dans ce stream
  destinationFile.write(data);

  //On vide le tampon et on ferme le stream
  destinationFile.flush();
```


Copyright © 2006 Baptiste Wicht . Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts

```
        } catch (MalformedURLException e) {
         System.err.println("Problème avec l'URL : " + filePath);
        } catch (IOException e) {
          e.printStackTrace();
        } finally{
         try {
          is.close();
       destinationFile.close();
          } catch (IOException e) {
           e.printStackTrace();
          }
        }
  }
```

Le code est suffisamment commenté pour que tout le monde comprenne chaque étape de l'opération de l'opération. Je me passerai donc de commentaires supplémentaires sur cette méthode.

Copyright © 2006 Baptiste Wicht . Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts

IV - Création d'un lanceur

Maintenant que l'on a récupéré le numéro de version, le nouveau fichier, il va falloir créer notre lanceur.

Concrètement, à quoi va-t-il servir ? Eh bien tout simplement, il va vérifier si l'on a téléchargé une nouvelle version ou non. Dans le cas ou il y a une nouvelle version, il va la renommer, supprimer l'ancienne et lancer la nouvelle et dans le cas ou il n'y en aurait pas, il va tout simplement lancer l'actuelle.

Pour lancer notre programme, j'ai choisi d'employer JDIC. Pourquoi ? Tout simplement, parce qu'avec JDIC, on conserve une portabilité parfaite et ainsi votre code marchera sous Windows, Linux et Mac. Vous pouvez aller le télécharger **ici**. Vous pouvez aussi lire **cet article** pour en savoir plus.

Notre lanceur sera un tout petit programme, composé d'une seule méthode main et de plusieurs champs.

```
import java.io.File;
import javax.swing.JOptionPane;
import org.jdesktop.jdic.desktop.Desktop;
import org.jdesktop.jdic.desktop.DesktopException;

public class Lanceur {
 //Variables contenant les noms des fichiers à charger
 private static final String pathCurrent = File.separator + "Actual.jar";
 private static final String pathNew = File.separator + "New.jar";
 private static final String pathOld = File.separator + "Old.jar";

 //Variable contenant le nom du répértoire courant
 private static final String currentFolder = System.getProperty("user.dir");

 public static void main(String[] args) {
  File current = new File(currentFolder + pathCurrent);
  File newVersion = new File(currentFolder + pathNew);
  File old = new File(currentFolder + pathOld);

  //Si une nouvelle version a été téléchargée
  if(newVersion.exists()){
   //On renomme la version actuelle (donc la vielle)
   current.renameTo(old);

   //On renomme la nouvelle avec le nom de l'ancienne
   newVersion.renameTo(current);

   //On supprimme l'ancienne
   old.delete();

   try {
    //On lance le nouveau fichier .jar
    Desktop.open(current);
   } catch (DesktopException e) {
    e.printStackTrace();
   }
  //S'il n'y a qu'une version courante et pas de nouvelles
  }else if(current.exists()){
   try {
    //On lance le jar actuel
    Desktop.open(current);
   } catch (DesktopException e) {
    e.printStackTrace();
   }
  //Si aucun fichier n'existe
  }else{
   //On avertit d'un problème
```


Copyright © 2006 Baptiste Wicht . Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts

```
      JOptionPane.showMessageDialog(null,"Aucun fichier jar à lancer...");
     }
    }
   }
```

Comme vous le voyez, c'est extrêmement simple comme classe. Il faut juste renseigner les champs pathCurrent, pathOld et pathNew et le programme s'occupe de tout, c'est transparent pour l'utilisateur.

La seule chose qui va changer avec l'emploi d'un lanceur, c'est le fait que l'utilisateur ne devra plus lancer directement votre programme mais le lanceur, s'il veut nécessiter des mises à jour.

Copyright © 2006 Baptiste Wicht . Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts

V - Application du protocole

Maintenant que l'on sait comment faire les parties compliquées, on va appliquer notre "protocole". On va donc créer une classe Updater avec les 2 méthodes créées précédemment et une nouvelle que l'on va faire maintenant.

```
private String lanceurPath = "Chemin vers lanceur";

//Version actuelle
private String version = "XXX";

//Variable contenant le nom du répertoire courant
private String currentFolder = System.getProperty("user.dir");

public void update(){
 ArrayList<String> versions = getVersions();

 //Si la version est nulle
 if(versions.size() == 0){
  JOptionPane.showMessageDialog(null,"Impossible de se connecter au service, vérifiez votre " +
   "connection internet");
 }else{
  //Si la dernière version n'est pas la même que l'actuelle
  if(!versions.get(versions.size() - 1).equals(version)){

   String versionChoisie = (String)JOptionPane.showInputDialog(null,"Choississez la version à " +
    "installer", "Versions disponibles",JOptionPane.QUESTION_MESSAGE,
    null,versions.toArray(),versions.get(versions.size() - 1));

   //S'il veut la télécharger
   if(versionChoisie != ""){
    Element racine = xmlDocument.getRootElement();

    //On liste toutes les versions
    List listVersions = racine.getChildren("version");
    Iterator iteratorVersions = listVersions.iterator();

    //On parcourt toutes les versions
    while(iteratorVersions.hasNext()){
     Element version = (Element)iteratorVersions.next();

     Element elementNom = version.getChild("nom");

     //Si c'est la bonne version, on télécharge tous ses fichiers
     if(elementNom.getText().equals((String)versionChoisie)){
      Element elementFiles = version.getChild("files");

      //On liste tous les fichiers d'une version
      List listFiles = elementFiles.getChildren("file");
      Iterator iteratorFiles = listFiles.iterator();

      //On parcours chaque fichier de la version
      while(iteratorFiles.hasNext()){
       Element file = (Element)iteratorFiles.next();

       //On télécharge le fichier
       downloadFile(file.getChildText("url"),currentFolder +
        File.separator + file.getChildText("destination"));
      }

      break;
     }
    }

    JOptionPane.showMessageDialog(null,"La nouvelle version a été téléchargée, "  +
```


Copyright © 2006 Baptiste Wicht . Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts

```
        "le programme va être relancé");

      File lanceur = new File(lanceurPath);

      try {
       //On lance le lanceur
       Desktop.open(lanceur);

       //On quitte le programme
       System.exit(0);
      } catch (DesktopException e) {
       JOptionPane.showMessageDialog(null,"Impossible de relancer le programme");
      }
     }
    }
    else{
     JOptionPane.showMessageDialog(null,"Pas de nouvelles version disponible pour le moment");
    }
   }
  }
```


Copyright © 2006 Baptiste Wicht . Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts

VI - Conclusion

VI-A - Conclusion

Voilà, vous savez maintenant comment créer un système de mise à jour pour vos programmes. Vous pouvez dès à présent employer votre nouveau savoir dans un de vos programmes, par exemple lors du clic sur un bouton ou un menu.

VI-B - Les sources

Voici les 2 classes que l'on a employées dans ce tutoriel :

- **Lanceur.java**
- **Updater.java**

Et voici ou vous pouvez télécharger les apis que l'on a utilisées :

- **JDIC**
- **JDOM**

Copyright © 2006 Baptiste Wicht . Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts

Copyright © 2006 Baptiste Wicht . Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts