

3.7.1 Installeur IzPack



A compléter

???

Sommaire :

- [Objectifs](#)
- [Processus](#)
- [Éléments nécessaires](#)
- [Éléments à modifier](#)
 - [Pom.xml](#)
 - [Installeur](#)
- [Les modifications dans Tomcat](#)
 - [Dans le apache](#)

Objectifs

L'objectif principal est de pouvoir construire, sur la base du fichier war généré :

- Un fichier zip permettant une installation sur un serveur
- Un fichier jar contenant le fichier War et un serveur d'application (Tomcat) permettant son installation sur un poste Linux ou Mac
- Un fichier exécutable .exe qui encapsule le fichier jar pour une installation sur un poste Windows.

L'utilisateur pourra indiquer qu'il accepte la licence, préciser l'emplacement de l'installation et créer les raccourcis dans son menu de démarrage via l'interface d'installation. Il est possible d'ajouter d'autres éléments à ce programme d'installation (demander les paramètres d'accès à la base de données par exemple)

Processus

Le processus est lancé par la tâche Maven install (mvn install).

Les opérations à effectuer sont décrites dans le pom.xml, elles sont définies comme suit :

- Suppression des répertoires et fichiers de la dernière install
- Copie du répertoire datarelease en data dans le répertoire de préparation de l'install serveur
- Copie du war dans le répertoire de l'install serveur
- Copie d'un fichier lisezmoi dans le répertoire de préparation de l'installateur
- Copie d'un fichier war dans le répertoire de préparation de l'installateur
- Copie du serveur Tomcat dans le répertoire de préparation de l'installateur (et dézip)
- Changement des paramètres du serveur dézippé
- Changement dans le fichier du nom du service Tomcat
- Création du jar
- Création de l'exécutable

Éléments nécessaires

Pour fonctionner l'installateur utilise:

- les tâches décrites dans le pom.xml
- le Tomcat déposé dans /src/util/utills/tomcat
- 7zip déposé dans /src/util/utills/7zip

- /datarelease qui contient le fichier config.properties pré-paramétré pour un fonctionnement de l'application en production. Ce répertoire peut contenir d'autre fichier de paramétrage en fonction des besoins applicatifs. Lors de la construction de l'installateur, il sera renommé en /data, le paramétrage du Tomcat embarqué dans l'installateur contient une variable application.config.location qui pointera vers le fichier de configuration du répertoire /data.

- un répertoire installeur utilisé pour faire le package izpack. Ce répertoire contient une page d'accueil lisezmoi, la description de l'installation (licence, chemin par défaut, description, ..) et les raccourcis des menus. La documentation complète de l'outil utilisé est disponible [ici](#).

Éléments à modifier

Pom.xml

Variables

```
<your.standalone-compiler.version>4.3.2</your.standalone-compiler.version>

<zip.location>${basedir}/src/main/util/utils/7zip/7za.exe</zip.location>

<izpack.application.name>Example</izpack.application.name>
```

- **your.standalone-compiler.version** permet de spécifier la version du plugin izpack

- **zip.location** permet de préciser le chemin d'accès à l'outil 7zip. Le chemin est à changer en fonction de votre installation (chemin et système Linux ou Windows).

- **izpack.application.name** permet de préciser le nom qui sera utiliser par izpack pour l'appel au service Tomcat sous Windows. Le nom ne doit pas contenir de caractères spéciaux ni point ".", ni tiret "-".

Tâches Ant

L'ensemble des tâches Ant sont dans le plugin maven-antrun-plugin du pom.xml. Elles suivent le processus décrit plus haut.

Il est possible d'ajouter ici ses propres tâches.

Par exemple l'ajout d'une documentation d'installation dans la livraison serveur :

```
<copy file="${basedir}/target/${project.build.finalName}.war" tofile="${basedir}/target/serverzip/chainedit.war" />

<copy file="${basedir}/src/main/doc/docs/installation.pdf" tofile="${basedir}/target/serverzip/installation.pdf" />
```

Installeur

L'installeur izpack se trouve dans /src/main/izpack.

Packages d'install

Le fichier ./config/install.xml permet de préciser :

- Les différents panneaux qui vont s'afficher lors de l'installation.
- Les pack de programmes à installer (Tomcat et Example)

Shortcut

Il y a 2 fichier qui vont permettre de préciser les raccourcis qui seront insérés dans les menus :

- shortcutSpec.xml pour Windows.
- Unix_sortcutSpec.sml pour Linux.
- Il n'y a actuellement pas de gestion de raccourcis pour Mac.

Logo

Si votre application dispose de son propre logo, vous pouvez déposer une image dans le répertoire ./config/jar (par défaut logo_commons.jpg).

Vous pouvez alors modifier le fichier install.xml pour pointer sur votre image.

Licence

Le fichier licence.html permet de préciser la licence d'utilisation de l'application, à l'installation l'utilisateur devra confirmer son acceptation de cette licence pour poursuivre l'installation.

Attention, le Html doit rester très simple, le programme d'installation n'est pas un navigateur.

Texte de présentation

Le fichier Info.html permet de créer un texte de présentation de l'application qui sera présenté à l'utilisateur lors de l'installation. Là encore le fichier doit rester très simple.

Page d'accueil

Dans ./config/custom se trouve un fichier index.html, c'est la page d'accueil qui sera affichée à l'utilisateur lors de son accès à Tomcat.

Elle est également accessible via les raccourcis présent dans le menu. Il est conseillé d'y mettre l'URL d'accès à l'application (<http://localhost:8080/Example-web-jsf-servlet/> par exemple).

Il est également conseillé d'y indiquer le login/password d'accès à l'application par défaut s'il existe.

Fichier de configuration pour l'exécutable

Le fichier ./exe/config.txt permet de préciser le message qui demandera à l'utilisateur de confirmer le lancement de l'installation ainsi que le titre de l'application.

Pour tout autre ajout, voir la documentation de l'utilitaire izpack disponible [ici](#).

Les modifications dans Tomcat

A priori, il n'est pas nécessaire de toucher à ces éléments, mais l'explication est néanmoins utile pour les administrateurs qui voudraient installer l'application à parti du fichier zip sur un serveur Linux ou Windows.

Dans le apache

Dézipper le apache-tomcat-x.x.x.zip Modifier les fichiers comme indiqué ci-dessous en ajoutant les lignes précisées au début de chaque fichier. Attention à la sauvegarde des .sh sous Windows à faire sous Eclipse pour éviter des problèmes de caractères sous Linux. Rezipper le apache-tomcat-x.x.x.zip et le mettre dans src/main/util/utls/tomcat

Le service.bat

1. Au début : set CATALINA_HOME=%INSTALL_PATH%\apache-tomcat-6.0.29\
2. A la fin remplacer : "%EXECUTABLE%" US%SERVICE_NAME% ++JvmOptions "-Djava.io.tmpdir=%CATALINA_BASE%\temp;-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager;-Djava.util.logging.config.file=%CATALINA_BASE%\conf\logging.properties;-Dapplication.config.location=%INSTALL_PATH%\data\config.properties" --JvmMs 256 --JvmMx 512

Ceci est valable pour une installation serveur si l'on souhaite utiliser le fichier de configuration du répertoire /data.

Le catalina.bat

```
set CATALINA_OPTS = "-Dapplication.config.location=%INSTALL_PATH%\data\config.properties"
```

Ceci est valable pour une installation serveur si l'on souhaite utiliser le fichier de configuration du répertoire /data.

Le catalina.sh

```
export CATALINA_OPTS = "-Dapplication.config.location=%INSTALL_PATH%\data\config.properties"
```

Ceci est valable pour une installation serveur si l'on souhaite utiliser le fichier de configuration du répertoire /data.

Explication

Au lancement l'application regarde si application.config.location existe. Si oui, il prend ce fichier de config. Sinon il prend celui qui est dans properties. Cette information est définie dans le applicationContext.xml :

```
<bean id="propertyConfigurer">
<property name = "locations">
  <list>
    <value>classpath:/properties/config.properties</value>
    <value>file:${ application.config.location }</value>
  </list>
</property>
<property name = "ignoreResourceNotFound" value="true" />
</bean>
```