

Installation

Installation
Installation Georchestra
Prérequis
Téléchargement de l'archive (non disponible pour le moment)
Construction du projet depuis les Sources
Prérequis
 Récupération des sources.
 Construction du projet
Mise en place de l'application sous tomcat
 déploiement de l'archive
Conteneur de Servlets ou Environnement spécifiques
 Installation du jdk java de sun version 6
 Installation du conteneur de servlets tomcat
 Modification de la variable d'environnement JAVA_HOME
 Désactivation du security manager tomcat:
 Copie de l'archive war dans le répertoire webapps
Paramétrage spécifique de l'application
 Configuration de l'application pour utiliser un proxy http

Installation Georchestra

Prérequis

- java 6 ou supérieur. La commande java -version doit indiquer un numéro de version commençant par 1.6. GNU GCJ n'est pas supporté et ne fonctionne pas avec georchestra.
- Tomcat 5.5 installé
- la distribution mapfishapp (mapfishapp.war)

Téléchargement de l'archive (non disponible pour le moment)

Téléchargez mapfishapp.war [TODO: définir l'url de téléchargement]

Mapfishapp peut fonctionner dans n'importe quel conteneur de servlet. Dans la présente documentation, nous utiliserons tomcat version 5.5.

Construction du projet depuis les Sources

Prérequis

Pour compiler les sources les outils suivants doivent être installés sur la machine:

- jdk 6 ou supérieur. La commande javac -version doit indiquer un numéro de version commençant par 1.6. GNU GCJ n'est pas supporté et ne fonctionne pas avec georchestra.
- maven 2.0 ou supérieur. La commande mvn -version doit afficher une version supérieur à 2.0.

Récupération des sources.

Récupérez les sources depuis le repository subversion

```
% svn co http://repository.georchestra.org/svn-georchestra/mapfishapp/
```

Construction du projet

Allez dans le répertoire mapfishapp

```
% cd mapfishapp
```

Lancer la construction à l'aide de la commande suivante:

```
% mvn -Dmaven.test.skip=true exec:exec package
```

L'application se trouve alors dans le répertoire target/mapfishapp.war

```
% ls target/mapfishapp.war
```

Pour éviter les erreurs liées aux tests unitaires, le projet doit être construit avec les tests invalidés (-Dmaven.test.skip=true)
 L'option de maven exec:exe est nécessaire pour la construction des fichiers javascripts.

Mise en place de l'application sous tomcat

* Stoppez le conteneur de servlets tomcat

```
% /etc/init.d/tomcat stop
```

déploiement de l'archive

* Copiez le fichier war téléchargé ou construit, en le renommant mapfishapp.war, dans le repertoire webapps/ de l'installation tomcat (\$CATALINA_HOME/webapps)

```
% cp mapfishapp-xxx.war $CATALINA_HOME/webapps/mapfishapp.war
```

* Demarrez le conteneur de servlets tomcat

```
% /etc/init.d/tomcat start
```

* Allez à l'url <http://localhost:8080/mapfishapp/>

Le conteneur de servlets peut être configuré pour écouter sur un port différent de 8080. Vérifiez la documentation spécifique du conteneur de servlets ainsi que sa configuration.

Si le conteneur de servlets est installé sur un hôte différent de la machine locale, l'url est alors: http://<nom_ou_ip_du_serveur>:8080/mapfishapp.

Conteneur de Servlets ou Environnement spécifiques

Installation sous debian (lenny)

Installation du jdk java de sun version 6

```
% apt-get install sun-java6-jdk
```

Installation du conteneur de servlets tomcat

```
% apt-get install tomcat5.5
```

Modification de la variable d'environnement JAVA_HOME

L'application nécessite que le conteneur de servlets tomcat utilise le jdk sun en version 6 ou supérieur.

Editez le fichier /etc/default/tomcat5.5

```
# The home directory of the Java development kit (JDK). You need at least
# JDK version 1.4. If JAVA_HOME is not set, some common directories for
# the Sun JDK, various J2SE 1.4 versions, and the free runtimes
# java-gcj-compat-dev and kaffe are tried.
JAVA_HOME=/usr/lib/jvm/java-6-sun
```

Désactivation du security manager tomcat:

Afin d'éviter tout problème il est préférable de désactiver le security manager de tomcat sous debian. Pour cela, editez le fichier /etc/default/tomcat5.5

```
# Use the Java security manager? (yes/no, default: yes)
# WARNING: Do not disable the security manager unless you understand
# the consequences!
# NOTE: java-gcj-compat-dev currently doesn't support a security
# manager.
#TOMCAT5_SECURITY=yes
TOMCAT5_SECURITY=no
```

Copie de l'archive war dans le répertoire webapps

Copiez l'archive war de mapfishapp téléchargée dans le répertoire /var/lib/tomcat5.5/webapps/

Paramétrage spécifique de l'application

Configuration de l'application pour utiliser un proxy http

Configuration du conteneur de servlet pour utiliser un proxy pour les requêtes HTTP vers l'extérieur

L'application réalisant des requêtes HTTP vers des serveurs WMS externes au domaine sur laquelle elle est installée, il peut être nécessaire de déclarer un proxy HTTP afin que la servlet OGCPProxy puisse réaliser les requêtes vers l'extérieur.

Le serveur tomcat doit être lancé avec les propriétés system suivantes configurées:

http.proxyHost
http.proxyPort

```
JAVA_OPTS="-Dhttp.proxyHost=<monproxy> -Dhttp.proxyPort=<mon port de proxy>"
```