



Opentheso v20.12

Par Miled Rousset miled[dot]rousset[[@](mailto:miled@mom.fr)][mom.fr](mailto:miled@mom.fr)

Installation sous Debian

téléchargez la dernière version de Debian sur le site (<http://www.debian.org/>)



Pour pouvoir installer les packages ci-dessous, il faut être root ou avoir le droit (sudo).

- installation effectuée avec (Debian 10.6)
- java 11 et plus
- tomcat 9 et plus



La version 20.12 d'Opentheso exige la version 11 de Java minimum

Installation de Java

1. Installez la dernière version Openjdk

- `apt install default-jre`
- `java -version` (devrait afficher):
openjdk version "11.0.9" 2020-10-20
OpenJDK Runtime Environment (build 11.0.9+11-post-Debian-1deb10u1)
OpenJDK 64-Bit Server VM (build 11.0.9+11-post-Debian-1deb10u1, mixed mode, sharing)

Installation de Postgresql (9.6 minimum)

2. Installez le serveur Postgresql

- `apt-get install postgresql`

pour contrôler la version : `psql --version`

psql (PostgreSQL) 11.9 (Debian 11.9-0+deb10u1)

3. ensuite il faut changer le mot de passe de Postgres

- `passwd postgres`
- Saisissez le nouveau mot de passe :
- Saisissez-le à nouveau :

Installation de Tomcat (tomcat 9 minimum)

1. pour installer tomcat, il faut repasser en root et procéder comme suit :
 - `aptitude install tomcat9`

l'installation de tomcat installe en même temps java, si ce n'est pas le cas, il faut installer la version 11 de java.

(Si java n'a pas été installé comme décrit ci-dessus, voici comment le faire autrement)

Il faut tester d'abord si java est installé :

- `java -version`
`openjdk version "11.0.9.1" 2020-11-04`

sinon,

- `nano /etc/apt/sources.list`

Ajouter les lignes suivantes :

```
# jessie backports
deb http://http.debian.net/debian jessie-backports main
```

ensuite,

- `apt-get update`
- `apt-get install -t jessie-backports openjdk-8-jdk`

pour passer de la version 8 à 11 de java :

- `update-alternatives --config java`

Il existe 2 choix pour l'alternative java (qui fournit `/usr/bin/java`).

Sélection	Chemin	Priorité	État

* 0	/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java	1061	mode automatique
1	/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java	1061	mode manuel
2	/usr/lib/jvm/java-11-openjdk-amd64/jre/bin/java	1051	mode manuel

Appuyez sur <Entrée> pour conserver la valeur par défaut[*] ou choisissez le numéro sélectionné :2

Apache2

Si Apache2 n'est pas installé, il faut l'installer avec la commande :

- `apt install apache2`

Installation du module AJP (pont de connexion entre apache et tomcat)

- `apt install libapache2-mod-jk`

aller sur `/etc/apache2/mods-enabled` et vérifier la présence de (`jk.load` et `jk.conf`)

ouvrir le fichier `/etc/libapache2-mod-jk/workers.properties`

vérifier si les liens vers tomcat et java sont justes.

Exemple :

```
workers.tomcat_home=/usr/share/tomcat9
```

```
workers.java_home=/usr/lib/jvm/default-java
```

dans `/etc/libapache2-mod-jk/` vous devriez trouver les fichiers suivants :

```
lrwxrwxrwx 1 root root 33 juin 4 21:18 httpd-jk.conf -> ../apache2/mods-available/jk.conf
```

```
-rw-r--r-- 1 root root 2944 déc. 2 15:08 workers.properties
```

Pour activer le Site sécurisé,

- `a2ensite default-ssl`
- `cd /etc/apache2/sites-enabled`
- `ls`
`default-ssl.conf`

il faut chercher le texte suivant dans `/etc/tomcat9/server.xml`:

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
<!--
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
-->
```

puis enlever le commentaire autour du connector et ajouter l'encodage UTF-8 :

```
<Connector port="8009" URIEncoding="UTF-8" protocol="AJP/1.3"
redirectPort="8443"/>
```

Modifier en

```
<Connector protocol="AJP/1.3"
  URIEncoding="UTF-8"
  address=":::1"
  port="8009"
  redirectPort="8443" secretRequired="false" />
```

- `/etc/init.d/tomcat9 restart`

Modifier les chemins dans le fichier de conf :

`/etc/libapache2-mod-jk/workers.properties`

Il faut donner les bons chemins et le nom à utiliser avec apache (exp: `ajp13_worker`)

Server.xml

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector protocol="AJP/1.3"
  port="8009"
  address="127.0.0.1"
  redirectPort="8443"
  secretRequired="true"
  secret="{mot de passe identique au worker apache}"
  allowedRequestAttributesPattern=".*" />
```



Attention, pour que la communication fonctionne entre apache et tomcat, il faut vérifier les actions ci-dessous :

- Créer un lien symbolique vers le fichier de conf dans apache : `/etc/libapache2-mod-jk/httpd-jk.conf`

```
ln -s /etc/apache2/mods-available/httpd-jk.conf /etc/libapache2-
mod-jk/httpd-jk.conf
```

```
root@cchum-kvm-frantiq-pactols:/etc/libapache2-mod-jk# ls -l
total 4
lrwxrwxrwx 1 root root 41 mars 21 10:51 httpd-jk.conf -> /etc/apache2/mods-
available/httpd-jk.conf
```

```
-rw-r--r-- 1 root root 3023 mars 19 13:41 workers.properties
```

- puis faire un lien des mods-enabled vers mods-available

```
ln -s /etc/apache2/mods-available/httpd-jk.conf /etc/apache2/mods-
enabled/httpd-jk.conf
```

- Supprimer tout fichier de type `jk.conf`

Parametrage de Apache (Virtual Host)

(ceci est un exemple de paramétrage dans `/etc/apache2/sites-enabled/`)

il faut créer un fichier de conf propre à Opentheso, voici un exemple suivant l'installation décrite ici :

- `cd /etc/apache2/sites-enabled/`
- `vi opentheso.conf`

```
<VirtualHost *:80>
    ServerAdmin toto@mondomaine.fr

    JkMount /opentheso/* ajp13_worker
    JKMount /opentheso ajp13_worker

    ServerName opentheso.mondomaine.fr

    DocumentRoot /var/lib/tomcat9/webapps/opentheso
    ErrorLog /var/lib/tomcat9/logs/error.log
    CustomLog /var/lib/tomcat9/logs/access.log common
    <Directory /var/lib/tomcat9/webapps/opentheso>
        Options -Indexes +FollowSymLinks +MultiViews
        AllowOverride All
        Order allow,deny
        Allow from all
    </Directory>

    <Location "/META-INF/">
        deny from all
    </Location>
    <Location "/WEB-INF/">
        deny from all
    </Location>

    <IfModule mod_rewrite.c>
        RewriteEngine On
        RewriteRule ^/$ /opentheso/$1 [R]
    </IfModule>

</VirtualHost>
```

Il faut maintenant activer le module « rewrite » dans apache2 :

- `a2enmod rewrite`

puis relancer apache

- `/etc/init.d/apache2 restart`

paramétrage de Tomcat 9

créer le fichier /usr/share/tomcat9/bin/setenv.sh
Ajouter les lignes suivantes (suivant la charge à faire supporter par l'instance)
export CATALINA_OPTS="\$CATALINA_OPTS -Xms2048m"
export CATALINA_OPTS="\$CATALINA_OPTS -Xmx8192m"
export CATALINA_OPTS="\$CATALINA_OPTS -XX:MaxPermSize=1024m"

Installation d'Opentheso :

Version 20.12 (c'est la version actuelle à la rédaction de la doc)

Déploiement du war Opentheso :

pour déployer le « war » sur le serveur que vous venez d'installer, il faut déposer le fichier « opentheso.war » dans le dossier /var/lib/tomcat9/webapps/



Attention, il faut renommer le war qui est fourni sur GitHub pour avoir la même configuration que la doc exp : (opentheso.war au lieu de opentheso-20.12.war).

*à partir de ce moment là, on peut modifier les fichiers de paramètres décrits ci-dessous :
les fichiers de conf se trouvent dans :
/var/lib/tomcat9/webapps/opentheso/WEB-INF/classes*

Modification du fichier « preferences.properties » :

(avant tout, il faut renommer le fichier, il est fourni par défaut sous le nom : preferences_dist.properties)

c'est la langue de base du site dans laquelle les thésaurus seront générés par défaut.

```
#Preferred language  
workLanguage=fr
```

choix de time-out
timeout_nbr_minute=29

```
#Serveur de mail pour les alertes du module candidats  
protocolMail=smtp  
hostMail=smtp.doamine.fr  
portMail=25  
authMail=false  
mailFrom=opentheso@mom.fr  
transportMail=smtp
```

Modification du fichier de connexion à la base de données Postgresql

« **hikari.properties** » :

(avant tout, il faut renommer le fichier, il est fournit par défaut sous le nom : hikari_dist.properties)

```
minimumIdle=1
autoCommit=true
setMaximumPoolSize=1000
idleTimeout=30000
connectionTimeout=30000
connectionTestQuery=SELECT 1
dataSourceClassName=org.postgresql.ds.PGSimpleDataSource
```

Ces paramètres sont à adapter suivant l'installation de la base de données d'Opentheso sous Postgresql.

#mon thesaurus

```
dataSource.serverName=localhost
dataSource.user=opentheso
dataSource.password=opentheso
dataSource.databaseName=opentheso
```

Installation de la base de données pour Opentheso :

Pour pouvoir installer la base de données sous postgresql, il faut récupérer le script SQL fourni, voici les étapes :

pour réaliser cette partie, il faut connaître le fonctionnement d'un serveur postgresql (création d'utilisateurs, de bases de données, lancement de scripts etc) il y a pas mal de documentations sur internet à ce sujet.

2. création du rôle et de la base de données *(ne pas saisir le texte en vert)*

- su postgres
- psql
- postgres=# CREATE USER opentheso PASSWORD 'opentheso';
- postgres=# create Database opentheso OWNER opentheso;
- postgres=# ALTER USER opentheso WITH SUPERUSER;
- \q
- Ctrl + d

3. Aller dans /var/lib/tomcat9/webapps/opentheso/WEB-INF/classes/install

4. psql -U opentheso -h localhost -p 5432 opentheso < ./opentheso_current.sql

```
5. psql -U opentheso -h localhost -p 5432 opentheso <
   ./maj_bdd_current2.sql
```

maintenant, la base de données est installée.

si tout va bien, alors on doit pouvoir relancer tomcat et apache2 :

- `/etc/init.d/tomcat8 restart`
- `/etc/init.d/apache2 restart`

Maintenant, dernière étape, il faut aller sur le domaine de votre installation pour tester si tout fonctionne :

<http://opentheso.votredomaine.fr/>

c'est le paramètre que vous avez défini dans le virtualHost avec la commande « `ServerName opentheso.votredomaine.fr` »

Première connexion à Opentheso :

Pour se connecter la première fois, il y a un utilisateur super-admin qui est défini par défaut :

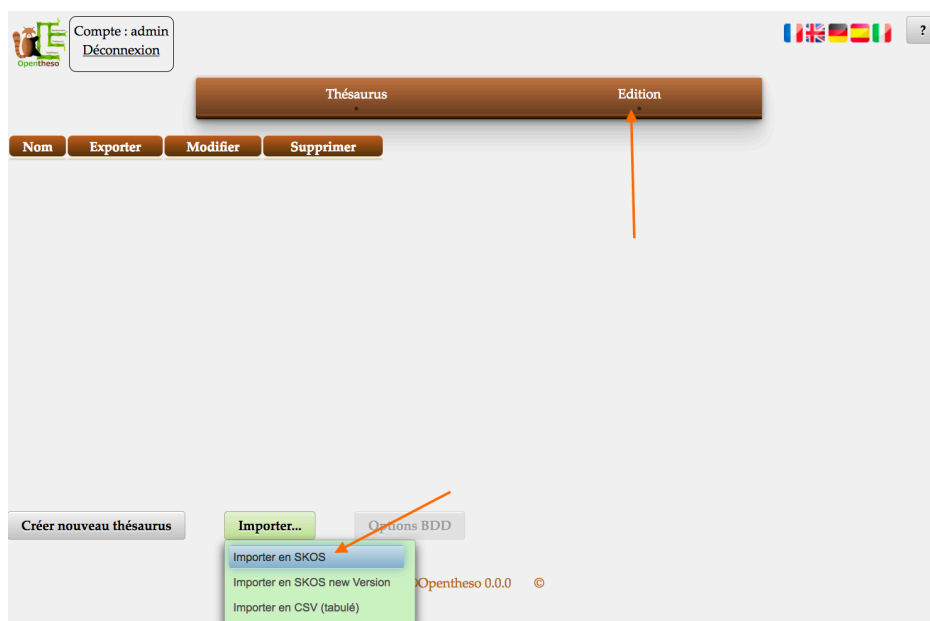
user : admin

pass : admin

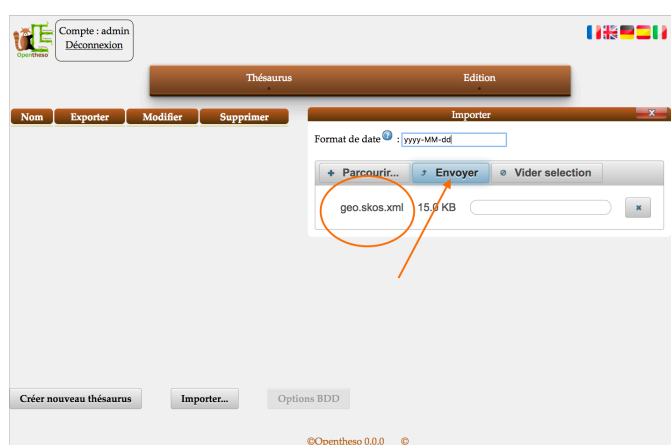
ensuite, il faut importer le thesaurus de test fourni dans le package : « geo.skos.xml »

pour
dans
puis

SKOS)

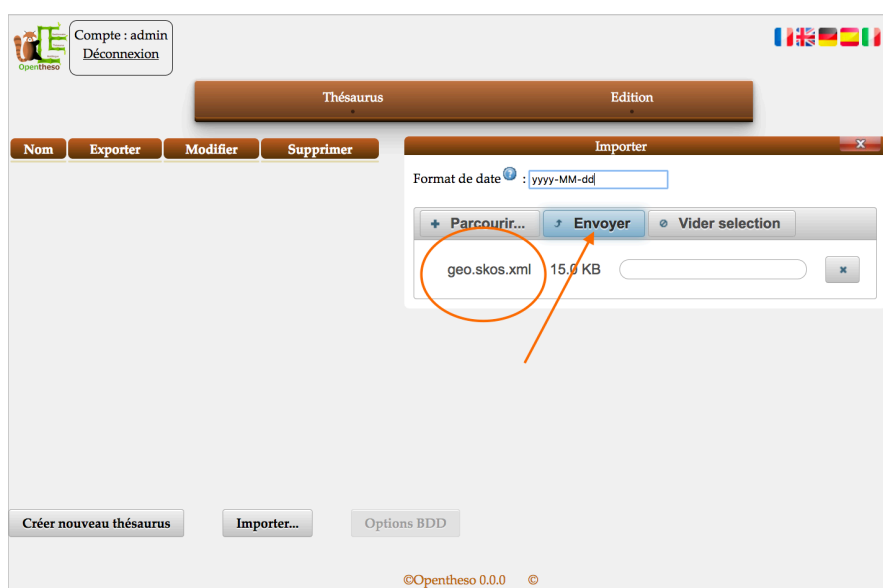


cela, allez
(Edition) ->
en bas, il y a
(Importer en



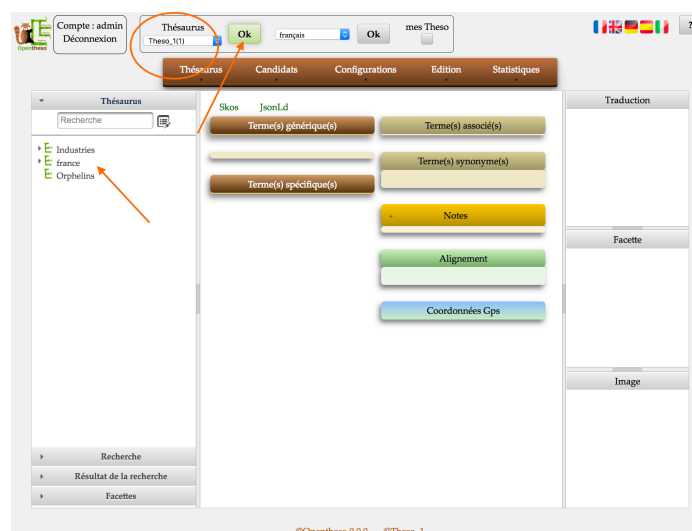
pour le choix de la date : il faut taper : (yyyy-MM-dd) ce qui correspond au format de la date dans le fichier Skos (2014-07-21).
 puis cliquez (Envoyer)

sur



ensuite Il faut
vous devriez avoir
test

recharger la page et
votre thésaurus de



Pour accéder au WebServices, il faut aller à cette adresse :

<http://votredomaine.fr/opentheso/copyright.xhtml>

vous avez des exemples prédéfinis concernant le fonctionnement de ce service.

💡 **Attention, pour pouvoir tester votre webservices, il faut d'abord naviguer dans votre thesaurus et se positionner sur un concept en particulier, ensuite, allez sur la page ci-dessus et testez.**

exemple :

http://votredomaine.fr/opentheso/webresources/rest/skos/concept/id=C_26984&th=1
devrait produire :

```
- <rdf:RDF>
- <skos:Concept rdf:about="http://193.48.137.89/opentheso/?id=C_26984&idt=1">
  <skos:prefLabel xml:lang="fr">productions</skos:prefLabel>
  <skos:inScheme rdf:resource="http://193.48.137.89/opentheso/?id=MT_18&idt=1"/>
  <skos:broader rdf:resource="http://193.48.137.89/opentheso/?id=C_26975&idt=1"/>
  <skos:narrower rdf:resource="http://193.48.137.89/opentheso/?id=C_26985&idt=1"/>
  <skos:narrower rdf:resource="http://193.48.137.89/opentheso/?id=C_26986&idt=1"/>
  <skos:narrower rdf:resource="http://193.48.137.89/opentheso/?id=C_26987&idt=1"/>
  <dcterms:created>2014-07-21</dcterms:created>
  <dcterms:modified>2014-07-21</dcterms:modified>
  <dcterms:identifier>C_26984</dcterms:identifier>
</skos:Concept>
</rdf:RDF>
```