



UDS - 2009

Documentation exploitation

**Dispositif de diffusion de cours en ligne
AudioVideoCours (Univ-R AV)**

<http://audiovideocours.u-strasbg.fr>
<http://univ-rav.u-strasbg.fr>

Sommaire

1. Installation de la distribution.....	3
2. Paramétrage du ftp.....	3
3. Récupération des sources.....	4
4. Création de la base de données.....	4
5. Configuration de tomcat.....	5
6. Stockage des répertoires de cours.....	6
7. Déploiement de l'application Audiovideocours :.....	7
8. Configuration d'apache.....	7
9. Tâches CRON.....	9
10. Page d'administration	9

Note: Un script d'installation d'Audiovideocours appelé « AVC_install.sh » est disponible sur le svn du projet, dans le répertoire « documentation ».
Attention: ce script est prévu pour une Ubuntu fraîchement installée.

1. Installation de la distribution

Distribution recommandée: Ubuntu 9.04 (Linux)

Installation des paquets suivants (via synaptic):

```
sun-java5-jre (1.5.0.19)
postgresql (8.3.8), pgadmin3 (facultatif)
tomcat6 (6.0.18)
apache2 (2.2.11)
vsftpd (2.0.7)
vorbis-tools (1.2.0), mp32ogg (0.11)
lame (3.98)
mplayer (2.1.0), mencoder (2.1.0)
mp3info (0.8.5)
ffmpeg (3.0), libavcodec-unstripped-52, libfaad0 (2.6.1)
zip (2.32), unzip (5.52)
python2.6 (2.6.2), python-reportlab (2.3)
mailx
ant
subversion
libpg-java, libcommons-io-java, libcommons-fileupload-java, libcommons-lang-java,
libcommons-beanutils-java, libcommons-collections3-java, libcommons-logging-java,
liblog4j1.2-java
yamdi
```

2. Paramétrage du ftp

Créer un répertoire à la racine du serveur :

```
mkdir /audiovideocours
```

Ajouter un nouvel utilisateur du système « ftpuser »:

```
useradd -m -d /audiovideocours/ftp ftpuser
passwd ftpuser
```

Des sous-répertoires doivent également être présents dans le répertoire FTP:

```
mkdir -p /audiovideocours/ftp/canceled /audiovideocours/ftp/client_update
/audiovideocours/ftp/live /audiovideocours/ftp/releases
```

- `canceled` : sert à stocker les cours enregistrés par le client mais annulés avant la publication.
- `client_update` : permet de stocker la dernière version du client d'enregistrement des cours, afin qu'il se mette à jour automatiquement dans les amphis.
- `live` : est le répertoire de stockage des diapositives d'un cours en direct envoyées par FTP.
- `releases` : stocke les programmes d'installation du client Audiovideocours.

Modifier les lignes de `/etc/vsftpd.conf` :

```
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
chroot_local_user=YES
local_root=/audiovideocours/ftp
```

Redémarrer le serveur ftp: `/etc/init.d/vsftpd restart`

3. Récupération des sources

Vous aurez besoin d'utiliser subversion pour récupérer les sources du projet via la commande :

```
svn co http://ulpmm-projets.u-strasbg.fr/svn/videocours/trunk/server
```

Le répertoire `univ-r_av` contient l'ensemble des sources du projet.
Le répertoire `model` contient des fichiers annexes qu'il faudra copier dans le répertoire des cours (voir 6.)

4. Création de la base de données

Redémarrez le serveur postgresql : `/etc/init.d/postgresql-8.3 restart`

Effectuez les modifications ci-dessous via `pgadmin3` ou en ligne de commande :

- Créer un utilisateur « `sqluser` » :
`su - postgres --command="createuser -P sqluser"`
- Créer une base de données « `univrav` » en utf-8 pour cet utilisateur :
`su - postgres --command="createdb -O sqluser -E UTF8 univrav"`
- Lancer le script de création de la base se trouvant dans le répertoire des sources :
`psql -h localhost -U sqluser -d univrav < /root/src/server/univ-r_av/WebContent/scripts/script_creation_database.sql`
- Vérifier que les tables suivantes ont été créées :
`course, building, univr, slide, tags, amphi, user, selection`

5. Configuration de tomcat

Ajouter les librairies suivantes dans le répertoire `/usr/share/tomcat6/lib` :

```
ln -s /usr/share/java/commons-io.jar /usr/share/tomcat6/lib/commons-io.jar
ln -s /usr/share/java/commons-fileupload.jar /usr/share/tomcat6/lib/commons-
fileupload.jar
ln -s /usr/share/java/postgresql-jdbc3.jar /usr/share/tomcat6/lib/postgresql-
jdbc3.jar
ln -s /usr/share/java/commons-lang.jar /usr/share/tomcat6/lib/commons-lang.jar
ln -s /usr/share/java/commons-beanutils.jar /usr/share/tomcat6/lib/commons-
beanutils.jar
ln -s /usr/share/java/commons-collections3.jar /usr/share/tomcat6/lib/commons-
collections3.jar
ln -s /usr/share/java/commons-logging.jar /usr/share/tomcat6/lib/commons-logging.jar
ln -s /usr/share/java/log4j-1.2.jar /usr/share/tomcat6/lib/log4j-1.2.jar
```

Configuration du tomcat-users.xml

Création d'un utilisateur tomcat `ulpmm` de la manière suivante :

```
<tomcat-users>
  <role rolename="tomcat"/>
  <role rolename="manager"/>
  <role rolename="ulpmm"/>
  <user username="ulpmm" password="s3cret" roles="tomcat,manager,ulpmm"/>
</tomcat-users>
```

Note: le rôle `ulpmm` est nécessaire pour accéder aux pages d'administration d'Audiovideocours.

Configuration du server.xml :

```
<Connector port="8080" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="443" />
```

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
  <Connector port="8009" protocol="AJP/1.3" redirectPort="443" />
```

```
<Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true"
  xmlValidation="false" xmlNamespaceAware="false">
```

```
  <Context path="/univ-r_av" docBase="univ-r_av">
    <Resource
      name="jdbc/postgres"
      auth="Container"
      type="javax.sql.DataSource"
      driverClassName="org.postgresql.Driver"
      url="jdbc:postgresql://127.0.0.1:5432/mabase"
      username="databaseuser"
      password="s3cret"
      maxActive="20"
      maxIdle="10"
      maxWait="-1"
    />
  </Context>
```

```
<Resource name="ldap/ox" auth="Container"
    type="com.sun.jndi.ldap.LdapCtx"
    factory="org.ulpmm.univrav.dao.LdapFactory"
    java.naming.factory.initial="com.sun.jndi.ldap.LdapCtxFactory"
    java.naming.provider.url="your_ldap_url"
    java.naming.security.authentication="simple"
    java.naming.security.principal="ldap_user"
    java.naming.security.credentials="ldap_password"
    com.sun.jndi.ldap.connect.pool="true"
    java.naming.security.protocol="ssl"
    com.sun.jndi.ldap.connect.timeout="5000"
    com.sun.jndi.ldap.read.timeout="5000"/>
</Context>
</Host>
```

Configuration des politiques :

Attention: certaines versions de tomcat sont plus stricte que d'autres d'un point de vu sécurité au niveau des politiques.

En cas d'erreur provenant de « java.security », il faut :

- Soit désactiver le « security manager » dans /etc/init.d/tomcat6:
TOMCAT6_SECURITY=no
- Soit configurer les politiques du répertoire /etc/tomcat6/policy.d/

Après avoir effectuées toutes les modifications ci-dessus, redémarrer tomcat via :
/etc/init.d/tomcat6 restart

6. Stockage des répertoires de cours

Créer un répertoire /audiovideocours/cours :
mkdir /audiovideocours/cours

Ce répertoire permet de stocker les cours. Chaque cours se trouve dans un répertoire horodaté+id du cours.

Copier le dossier model du répertoire server des sources du projet dans le répertoire /audiovideocours/cours

On donne les droits d'écriture à l'ensemble du répertoire /audiovideocours :
chmod -R 777 /audiovideocours

7. Déploiement de l'application Audiovideocours :

Pour créer un fichier .war de l'application, il suffit d'utiliser le fichier `build.xml`, qui est à la racine du projet, avec ANT :

- Configurer l'application par le fichier `univrav.properties` dans le répertoire `WebContent/conf` des sources et par le fichier `WebContent/WEB-INF/web.xml`.
- Génération du .war de l'application via les sources récupérées du SVN, en exécutant la commande suivante à la racine du projet : `ant`
- Déploiement du .war sur Tomcat (url : <http://localhost:8180/manager/html>)

Attention :

- Il est préférable de supprimer le cache de Tomcat lors de son redémarrage car il peut y avoir des problèmes de rafraîchissement de pages lorsqu'une nouvelle version est mise en place.

Accéder à l'application via http://localhost/univ-r_av ou via l'url apache que vous avez configurée.

8. Configuration d'apache

Installation des modules

Exécuter la commande :

```
a2enmod proxy proxy_ajp proxy_balancer proxy_http rewrite ssl
```

Configuration d'apache

Il faut créer un fichier `mon-site.conf` dans le répertoire `/etc/apache2/sites_available` qui doit contenir les hôtes virtuels pour le site web (un sur le port 80 et un sur le port SSL 443).

Il faut alors monter le dossier racine afin d'avoir accès à l'arborescence du site, le dossier `/audiovideocours/cours/` pour pouvoir accéder aux cours depuis l'extérieur, les dossiers `/audiovideocours/ftp/live/` et `/audiovideocours/ftp/releases` mettant à disposition des fichiers via FTP.

Pour SSL, il faut créer un certificat et l'activer depuis l'hôte virtuel 443. Puis, depuis l'hôte virtuel 80, on redirige les URLs que l'on désire sécuriser vers le port 443, via le mode `rewrite`.

mon-site.conf :

```
<VirtualHost *:80>
    ServerName mon-site
    ServerAlias mon-site-alias

    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule ^/avc/myspace(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
    RewriteRule ^/admin(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
    RewriteRule ^/avc/admin(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
    RewriteRule ^/avc/publication(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
    RewriteRule ^/avc/authentification(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]

    Include /etc/apache2/sites-available/mon-site.include
</VirtualHost>

<VirtualHost *:443>
    ServerName mon-site
    ServerAlias mon-site

    SSLEngine On
    SSLCertificateFile /repertoire-certs/mon-site.pem
    SSLCertificateKeyFile /repertoire-certs/mon-site.key

    Include /etc/apache2/sites-available/mon-site.include
</VirtualHost>
```

mon-site.include:

```
ServerAdmin webmaster@localhost

DocumentRoot /var/lib/tomcat6/webapps/univ-r_av
Options -Indexes

<Proxy *>
    Allow from all
</Proxy>

<Proxy balancer://tomcat>
    BalancerMember ajp://127.0.0.1:8009 route=localhost connectiontimeout=1000
</Proxy>

ProxyPass /coursv2 !
Alias /coursv2 "/audiovideocours/cours/"

ProxyPass /live !
Alias /live "/audiovideocours/ftp/live/"

ProxyPass /releases !
Alias /releases "/audiovideocours/ftp/releases/"

ProxyPass /balancer-manager !
<Location /balancer-manager>
    SetHandler balancer-manager
    Order Deny,Allow
    Deny from all
    Allow from 127.0.0.1
</Location>

ProxyPass / balancer://tomcat/ stickysession=JSESSIONID|jsessionid
```



```
ProxyPassReverse / balancer://tomcat/ stickysession=JSESSIONID|jsessionid

ErrorLog /var/log/apache2/error-univ-rav.log
LogLevel warn
CustomLog /var/log/apache2/access-univ-rav.log combined
ServerSignature Off
```

Pour activer le site, exécuter la commande :

```
a2ensite mon-site.conf
```

Redémarrer Apache via :

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

9. Tâches CRON

- Supprimer tous les jours les cours dont le code d'accès est « suppression ». Ce code d'accès est utilisé pour identifier les cours servant à faire des tests des clients dans les amphis. De plus, les cours contenant dans leur titre les mots « test » ou « essai » sont masqués.

```
# /etc/crontab: system-wide crontab

# m h dom mon dow user  command
00 0 * * * root    wget http://mon-site/avc/deletetests -O - 2>&1 | grep "200
OK$" > /dev/null || echo Failed
```

10. Page d'administration

La page d'administration permet de gérer différents éléments du site :

<http://mon-site/avc/admin>

1. Gestion des cours (éditer, supprimer)
2. Gestion des cours Univ-r (éditer, supprimer)
3. Gestion des bâtiments et des salles (ajouter, éditer, supprimer)
4. Gestion des utilisateurs (ajouter, éditer, supprimer)
5. Statistiques sur les auteurs
6. Statistiques générales (taille du disque, hits par mois, fichiers téléchargés, etc ...)
7. Page dédié aux tests (mot de passe « suppression » ou titre commençant par « testulpm »: <http://mon-site/avc/tests>)
8. Gestion des sélections et des collections de cours
9. Test des versions des clients des amphithéâtres