



UNISTRA - 2013

Documentation exploitation

**Dispositif de diffusion de cours en ligne AudioVideoCast
(anciennement AudioVideoCours)**

<http://audiovideocast.unistra.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 AudioVideoCast :.....	7
8. Configuration d'apache.....	7
9. Activation du streaming	9
10. Séparer l'encodage des médias.....	10
11. Suppression des tests.....	10
12. Configuration du live	11
13. Page d'administration	11
14. Annexe.....	11

Note: Un script d'installation d'AudioVideoCast appelé « AVC_install_ubuntu1204.sh » est disponible sur le svn du projet, dans le répertoire « documentation ».

Dans le cas d'une installation avec lighttpd pour la diffusion des flux, il vous faudra également récupérer le script « AVC_light » (celui-ci sera automatiquement appelé par le script précédent).

Attention: ces scripts sont prévus pour une Ubuntu fraîchement installée et mise à jour.

1. Installation de la distribution

Distribution recommandée: Ubuntu 12.04 (Linux)

Installation des paquets suivants :

#packages nécessaires au fonctionnement du site :

```
apt-get install postgresql pgadmin3 tomcat6 apache2 vsftpd zip unzip mailutils  
libpg-java libcommons-fileupload-java libcommons-io-java libcommons-lang-java  
libcommons-beanutils-java libcommons-collections3-java libcommons-logging-java ant  
subversion liblog4j1.2-java openjdk-6-jdk
```

#paquets pour l'encodage :

```
apt-get install vorbis-tools lame ffmpeg libavcodec-extra-53 eyeD3 ffmpeg2theora  
yamdi zip unzip python2.6 python-reportlab python-psycopg2 atomicparsley
```

#Modification des presets ipod

```
sed -i 's/^profile/vprofile/' /usr/share/avconv/libx264-ipod320.avpreset  
sed -i 's/^profile/vprofile/' /usr/share/avconv/libx264-ipod640.avpreset
```

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 AudioVideoCast.

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 dernières sources stables du projet via la commande :

```
svn co http://subversion.cru.fr/audiovideocours/tags/server/release-X.XX;mv release-X.XX univ-r_av
```

« X.XX » correspond à la version du dernier tag disponible. Le répertoire univ-r_av contient alors l'ensemble des sources du projet.

Il vous faut également récupérer le player flash JW player, en acceptant la licence « Creative Commons : Attribution-Noncommercial-Share Alike 3.0 Unported » (voir <http://creativecommons.org/licenses/by-nc-sa/3.0/>) :

```
wget http://audiovideocast.unistra.fr/releases/player.swf
mv player.swf /root/src/server/univ-r_av/WebContent/files/jwflvplayer/
```

4. Création de la base de données

Redémarrez le serveur pgsql : `/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, slide, tags, amphi, user, selection, discipline, level, job

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'AudioVideoCast.

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">
    <Environment name="volume" value="1" type="java.lang.Short" override="false"/>
    <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"
    />
    <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.

Il contient un répertoire (par défaut 1) appelé « volume de stockage ». Il est possible de modifier la valeur du volume dans le contexte de tomcat, si l'on veut monter un nouveau filesystem (pour raison de place par exemple).

Chaque cours se trouve dans une arborescence de sous-répertoires basée sur l'id du cours (formaté sur 8 chiffres).

Par exemple, pour le cours n°4209, l'arborescence sera :
`/audiovideocours/cours/1/00/00/42/09`

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

7. Déploiement de l'application AudioVideoCast :

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` (voir Annexe).
- 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.cert
    SSLCertificateKeyFile /repertoire-certs/mon-site.key
    Include /etc/apache2/sites-available/mon-site.include
    RewriteEngine On
    RewriteCond %{HTTPS} on
    RewriteRule ^(.*)/avc/courseaccess(.*)$ http://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
    RewriteRule ^(.*)/avc/liveaccess(.*)$ http://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
</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 les commandes :

```
a2dissite default
a2ensite mon-site.conf
```

Redémarrer Apache via :

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

9. Activation du streaming

Via Apache

Dans le cas classique où c'est apache qui s'occupe de diffuser les flux des différents médias des enregistrements, il vous est possible d'activer la diffusion de ces flux en streaming.

On récupère les sources des modules de streaming (flv et mp4) :

```
wget http://github.com/osantana/mod_flvx/raw/master/mod_flvx.c
wget http://h264.code-shop.com/download/apache_mod_h264_streaming-2.2.7.tar.gz
tar -zxvf apache_mod_h264_streaming-2.2.7.tar.gz
```

On installe les outils de développement d'apache :

```
apt-get install apache2-threaded-dev
```

Compilation et installation du module flvx :

```
apxs2 -i -a -c mod_flvx.c
a2enmod flvx
echo "AddHandler flv-stream .flv" >> /etc/apache2/apache2.conf
```

Compilation et installation du module h264 :

```
cd mod_h264_streaming-2.2.7
./configure --with-apxs=`which apxs2`
make
make install
echo "LoadModule h264_streaming_module
/usr/lib/apache2/modules/mod_h264_streaming.so" > /etc/apache2/mods-
available/h264.load
a2enmod h264
echo "AddHandler h264-streaming.extensions .mp4" >> /etc/apache2/apache2.conf
```

Redémarrer Apache via :

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

Via lighttpd

Dans le cas particulier où vous souhaitez séparer les serveurs diffusant le site web et les serveurs diffusant les flux en streaming avec lighttpd (pour des raisons de performance), il vous faudra compiler une version de lighttpd avec les modules de streaming activés.

Pour se faire, nous vous invitons à consulter l'annexe et le script « AVC_light.sh » présent sur subversion (partie documentation).

10. Séparer l'encodage des médias

Si vous souhaitez effectuer l'encodage des médias sur des serveurs à part, il vous faudra :

Configurer le fichier de conf univrav.properties du projet :

```
# To separate medias encodage
sepEnc=true
```

Copier les scripts d'encodage à l'endroit voulu, par ex :

```
cp -R /root/src/server/univ-r_av/WebContent/scripts/jobs_encodage
/audiovideocours/cours/
```

Puis, il faut modifier la chaîne de connexion du script « accessbase.py » pour la récupération des jobs (cf « dbapi2.connect »).

Enfin, il faut configurer la tâche CRON de la manière suivante :

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

# m h dom mon dow user  command
1-59/2 * * * * root    nice -n 19 bash
/audiovideocours/cours/jobs_encodage/JobEnc.sh /audiovideocours/cours/jobs_encodage
http://audiovideocast.unistra.fr /audiovideocours/cours &> /dev/null
```

11. Suppression des tests

- 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.

```
# /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
```

12. Configuration du live

Pour faire fonctionner le live avec AudioVideoCast, il vous faut installer un serveur Flash Media Serveur.

Pour ce faire, veuillez-vous référer à la documentation officielle :

<http://www.adobe.com/products/flashmediaserver/>

Une fois le serveur FMS installé, il suffit de configurer le paramètre « flashServerIp » avec l'url du serveur FMS, dans le fichier « univrav.properties ». N'oubliez pas de configurer le client également.

13. 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, commande findTracks et findStats, version des clients)
7. Page dédié aux tests (code d'accès spécifiques modifiables dans le fichier « univrav.properties » via « testKeyWord ») : <http://mon-site/avc/tests>
8. Gestion des sélections et des collections de cours
9. Test des versions des clients des amphithéâtres

NB: Pour les statistiques, il y a la possibilité d'utiliser google analytics (voir fichier « WEB-INF/views/include/google_analytics.jsp » pour y coller votre script ga)

14. Annexe

Détail des fichiers de configuration :

Détail du fichier « univrav.properties » :

```
# The Url of the server
serverUrl = http://localhost/univ-r_av
Il s'agit de l'url du serveur. Elle doit être changée.

# The Url to access to the course on internet Prod : http://[HOSTNAME].u-strasbg.fr/
coursesUrl = http://localhost/univ-r_av/coursv2/
Il s'agit de l'url permettant d'accéder aux enregistrements. Elle doit être changée.
Il est possible d'utiliser une fonction RAND[1-10] pour faire du load balancing sur plusieurs serveurs.
Ex: streamRAND[1-10].example.fr va load balancer sur stream1,stream2,...,stream10
Il est également possible d'utiliser une fonction [HOSTNAME] pour récupérer le nom du serveur.
```

```
# Folders on the file system Prod : /audiovideocours/cours/ /audiovideocours/ftp/
/audiovideocours/ftp/live/
coursesFolder = /audiovideocours/cours/
Dossier contenant les cours de l'application.
ftpFolder = /audiovideocours/ftp/
Dossier contenant les cours zippés envoyés par le client.
liveFolder = /audiovideocours/ftp/live/
Dossier servant à recevoir la capture d'écran envoyé par le client pendant le Live.

# Default media filenames in the archive sent by the client
defaultMp3File = enregistrement-micro.mp3
defaultFlashFile = enregistrement-video.flv
Ces deux paramètres ne doivent pas être changés pour le bon fonctionnement avec le client actuel.

# Copyright comment
comment = Owned by the author
Commentaire pour les tags des médias.

# IP address of the Flash Server for the video and audio live
flashServerIp = serveurflash.example.fr
Adresse du serveur flash pour la diffusion en live. Doit être changée.

# The settings of the RSS files, of the permalien (interface flash) and of emails
rssTitle = Audiovideocast
rssName = Audiovideocast
rssDescription = Université de Strasbourg
rssImageUrl = http://localhost/univ-r_av/files/img/univr-av-logo-rss.png
rssCategory = Enseignement
language = fr
Paramètres pour les flux RSS. Peut être changé.
recordedInterfaceUrl = http://localhost/univ-r_av/avc/courseaccess
Lien utilisé pour les accès directs aux enregistrements. Doit être changé.

# The setting of the RSS files for iTunes
itunesAuthor = Université de Strasbourg
itunesSubtitle = Enregistrement culturel
itunesSummary = Retrouvez les supports synchronisés sur audiovideocast.unistra.fr
itunesImage = http://localhost/univ-r_av/releases/Illustration_itunes_AVC.jpg
itunesCategory = Enseignement
itunesKeywords = science,culture,société,technologies,éducation
Paramètres pour les flux RSS sous iTunes. Peut être changé.

# University parameters
univName = Université de Strasbourg
univAcronym = UDS
univLink = http://www.unistra.fr/
Informations concernant l'université sur la page d'accueil

# Publication free
pubFree = true
pubTest = true
Autoriser ou non la publication libre de contenu

# The numbers of courses to display at the same time
lastCourseNumber = 10
selectionCourseNumber = 10
collectionCourseNumber = 10
recordedCourseNumber = 10
Permet de définir le nombre de cours à afficher dans la page d'accueil et dans la page des
enregistrements.
# The default style
defaultStyle = orange-theme
Thème du site par défaut.

# The keyword to identify the tests to delete (genre is equal to this keyword)
testKeyWord1 = Suppression

# The keyword to identify the tests to hide (title begins with this keyword)
testKeyWord2 = Testulpmmm
# testKeyWord3 = Essai
Mots clé pour le code d'accès d'un enregistrement permettant de le masquer.
#Admin email for notification
#AdminEmail1 is used for "Project leader contact"
adminEmail1=admin1@fai.fr
adminEmail2=admin2@fai.fr
```

```
adminEmail3=admin3@fai.fr
Adresse email des administrateurs. Doit être changé.

#CAS Logout
casLogoutUrl=https://cas.example.fr:443/cas/logout
Adresse de déconnexion du serveur cas. Doit être changé.

#Additional document formats
addDocFormats=pdf html swf ppt pptx odp docx doc odt xlsxlsx ods rtf txt jpg jpeg png gif bmp zip bz bz2
ark rar 7z ac3 avi divx flv m3u m4a mov movie mp2 mp3 mp4 mpg mpeg ogg ra rm rv wav wma wmv aac
Formats de fichiers autorisés pour l'upload de documents additionnels.

#Upload media formats
uploadFormats=mp3 ogg wav wma avi divx mp4 mpg mpeg mov wmv mkv flv ogv webm m4v
Formats de fichiers autorisés pour l'upload de fichiers audio et video.

#Link for support (help page)
supportLink=http://support.unistra.fr
helpLink=https://sites.google.com/site/wikiaudiovideocast/
clientLink=http://sites.google.com/site/audiovideocours/client-logiciel-fr
tracLink=http://sourcesup.cru.fr/projects/audiovideocours
docLink=http://documentationlogicielle.u-strasbg.fr/DUN/IPM/AudiovideoCours
Lien d'aide pour l'application (page help). Peut être changé.

#LDAP search properties
ldapBaseDn=ou=uds,ou=people,o=annuaire
ldapSearchFilter=uid
ldapMail=mail
ldapFirstname=givenName
ldapLastname=sn
ldapProfile=eduPersonPrimaryAffiliation
ldapAffectation=supannetablissement
ldapEtpPrimaryCode=udsPrimaryEtpCode
Propriétés de recherche du ldap. A modifier en fonction du ldap.

# To separate medias encodage
sepEnc=false
Si false, l'application web encode les médias (par défaut). Si true, l'encodage des medias doit se faire
séparément (via tâche cron qui appelle les scripts).
```

Détail du filtre CAS du fichier « web.xml » :

```
<filter>
  <filter-name>CAS filter</filter-name>
  <filter-class>edu.yale.its.tp.cas.client.filter.CASFilter</filter-class>
  <init-param>
    <param-name>edu.yale.its.tp.cas.client.filter.loginUrl</param-name>
    <param-value>https://cas.example.fr:443/cas/login</param-value>
    Url de login du serveur CAS. A changer.
  </init-param>
  <init-param>
    <param-name>edu.yale.its.tp.cas.client.filter.validateUrl</param-name>
    <param-value>https://cas.example.fr:443/cas/serviceValidate</param-value>
    Url de validation du serveur CAS. A changer
  </init-param>
  <init-param>
    <param-name>edu.yale.its.tp.cas.client.filter.serverName</param-name>
    <param-value>localhost</param-value>
    Url de retour après connexion au serveur CAS. Il s'agit de l'adresse de votre site. A
    changer
  </init-param>
</filter>
```

Problèmes de cookies avec les iframes :

Note: le problème n'a pas été rencontré sous Firefox, Opera et Chrome

1. Sous Internet Explorer

Les iframes peuvent poser des problèmes lors de la création de cookies (d'authentification notamment) sous Internet Explorer (Erreur de connexion) lorsque le site est appelé de puis une iframe. Ceci est lié à un problème de compatibilité entre IE et la norme W3C "Platform for Privacy Preferences".

Ajouter l'entête P3P :

Pour résoudre ce problème, il suffit de positionner l'entête p3p dans Apache :

Activer le mode headers pour Apache :

```
a2enmod headers
```

Puis, ajouter la ligne suivante dans "/etc/apache2/mods-enabled/headers.load" :

```
Header append P3P "CP=\"NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM\""
```

Enfin, redémarrer Apache

```
/etc/init.d/apache/restart
```

2. Sous Safari

Idem que la procédure précédente. Sauf que dans le cas de Safari, l'header p3p peut ne pas suffire car il possède une sécurité plus renforcée.

Mettre un lien direct :

Safari ne veut pas créer de cookie si la page qui crée le cookie n'a pas été appelé via un lien direct par l'utilisateur :

```
<a href="http://pagequicreelecookie.fr">...</a>
```

Donc, si la page qui crée le cookie est appelé par une méthode javascript par exemple (via onclick), ca ne fonctionne pas.

Problèmes de cross-domain avec flash

(Ne concerne que le cas de la diffusion séparée des flux via lighttpd)

Il faut créer le fichier "crossdomain.xml" suivant dans "/audiovideocours/cours" :

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy
  SYSTEM "http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
  <allow-access-from domain="*" />
</cross-domain-policy>
```

Packaging du module MP4 pour Lighttpd

Le but est de patcher Lighttpd afin d'y inclure le module MP4 fournit par code-shop.

Environnement de compilation :

Créer un répertoire de travail :

```
mkdir lightty
cd lightty
```

Installer les paquets nécessaires à la recompilation du paquet fournit par Ubuntu :

```
aptitude install build-essential fakeroot dpkg-dev
```

Récupérer les sources :

```
apt-get source lighttpd
sudo aptitude build-dep lighttpd
wget http://h264.code-shop.com/download/lighttpd-1.4.18_mod_h264_streaming-
2.2.9.tar.gz
tar xvzf lighttpd-1.4.18_mod_h264_streaming-2.2.9.tar.gz
dpkg-source -x lighttpd_1.4.26-1ubuntu4.dsc
```

Modification des sources :

Créer le script contenant le code suivant à la racine du répertoire courant :

```
cd lighttpd-1.4.18/src
cp -v mod_h264_streaming.c \
  mod_streaming_export.h \
  moov.c moov.h \
  mp4_io.c mp4_io.h \
  mp4_reader.c mp4_reader.h \
  mp4_writer.c mp4_writer.h \
  mp4_process.c mp4_process.h \
  output_bucket.c output_bucket.h \
  output_mp4.c output_mp4.h \
  ../../lighttpd-1.4.26/src/
```

Exécuter le script fraîchement créé :

```
chmod +x script.sh
./script.sh
```

Ajouter les ordres de compilation du module MP4 au fichier lighttpd-1.4.26/src/Makefile.am (avant la ligne "lib_LTLIBRARIES += mod_flv_streaming.la"):

```
lib_LTLIBRARIES += mod_h264_streaming.la
mod_h264_streaming_la_SOURCES = mod_h264_streaming.c \
                                mod_streaming_export.h \
                                moov.c moov.h \
                                mp4_io.c mp4_io.h \
                                mp4_reader.c mp4_reader.h \
                                mp4_writer.c mp4_writer.h \
                                mp4_process.c mp4_process.h \
                                output_bucket.c output_bucket.h \
                                output_mp4.c output_mp4.h
mod_h264_streaming_la_CFLAGS = $(AM_CFLAGS) -DBUILDING_H264_STREAMING
mod_h264_streaming_la_LDFLAGS = -module -export-dynamic -avoid-version -no-undefined
mod_h264_streaming_la_LIBADD = $(common_libadd)
```

Indiquer au paquet Ubuntu l'emplacement du module MP4 :

```
echo debian/tmp/usr/lib/lighttpd/mod_h264_streaming.so >> lighttpd-1.4.26/debian/lighttpd.install
```

Compilation du paquet :

Lancer ces commandes afin de créer le paquet :

```
cd lighttpd-1.4.26
dpkg-buildpackage -b -rfakeroot
```

Installation du paquet :

Petite subtilité ici, la commande "hold" d'aptitude qui permet d'indiquer que l'on souhaite conserver le paquet lighttpd à sa version actuelle même si une mise à jour est disponible :

```
cd ..
sudo dpkg -i lighttpd_1.4.26-1ubuntu4_amd64.deb
sudo aptitude hold lighttpd
```

Configuration de Lighttpd :

Ajouter ces lignes au fichier /etc/lighttpd/lighttpd.conf

```
server.modules = (
    ...,
    "mod_h264_streaming",
    "mod_flv_streaming"
)
h264-streaming.extensions = ( ".mp4" )
flv-streaming.extensions = ( ".flv" )
```


Référence et fichiers :

- <http://h264.code-shop.com/trac/wiki/Mod-H264-Streaming-Lighttpd-Version2>
- [lighttpd-1.4.18_mod_h264_streaming-2.2.9.tar.gz](#)
- <http://blog.lighttpd.net/articles/2006/03/09/flv-streaming-with-lighttpd>

Serveurs d'encodage séparés

Note : n'oubliez pas de définir le paramètre « sepEnc=true » du fichier de configuration « univrav.properties » d'Audiovideocast pour que le serveur tomcat ne lance pas l'encodage des enregistrements.

Installation des paquets :

Les paquets suivants sont nécessaires pour l'encodage des medias :

```
apt-get install vorbis-tools lame ffmpeg libavcodec-extra-53 mp3info yamdi zip unzip  
python2.6 python-reportlab python-psycpg2 atomicparsley bc
```

Configuration de la tâche CRON pour l'encodage :

On copie tout d'abord le répertoire "jobs_encodage"(cf src Audiovideocast) contenant les différents scripts d'encodage à l'endroit voulu. Puis, on configure la tâche cron de la manière suivante :

Dans /etc/crontab, on ajoute par exemple :

```
1-59/2 * * * * root nice -n 19 bash /var/lib/tomcat6/webapps/univ-  
r_av/scripts/jobs_encodage/JobEnc.sh /var/lib/tomcat6/webapps/univ-  
r_av/scripts/jobs_encodage http://localhost/univ-r_av /audiovideocours/cours &>  
/dev/null
```

JobEnc.sh est le script principal à lancer. Le premier argument est le chemin du dossier contenant l'ensemble des scripts d'encodage. Le deuxième argument est l'url du serveur Audiovideocast. Le troisième argument est le répertoire contenant les cours.

Sous Ubuntu 12.04, la redirection des erreurs "&> /dev/null" dans la tâche CRON est indispensable pour le bon fonctionnement des scripts d'encodage.