



## **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

I. Fonctionnalités du site web.....	3
II. Technologies.....	3
III. Autres programmes requis.....	4
IV. Formats multimédia utilisés.....	4
V. Démonstration.....	5
1. Processus de création et d'ajout de cours en différé.....	5
2. Page « mon espace ».....	5
3. Lancement d'un enregistrement en direct.....	6
VI. Fonctionnement de l'Interfaçage Univ-R.....	6
VII. Page d'administration.....	7
VIII. Base de données.....	7
IX. Stockage des cours.....	8
X. Scripts.....	8
XI. Tâches CRON.....	8
XII. Mise en place de la solution technique.....	9
1. Récupération des sources et le développement.....	9
2. Déploiement.....	9
3. Configuration de Tomcat.....	11
4. Configuration d'Apache et utilisation du mod_jk.....	12
5. Configuration de la base de données.....	14

## I. Fonctionnalités du site web

- Recevoir les données audio et vidéo envoyées depuis les clients
- Diffuser les contenus en direct et en différé sur la plupart des systèmes d'exploitation du marché
- Proposer des contenus publics accessibles à tout le monde
- Proposer des contenus privés protégés par un code d'accès
- Proposer une fonction de recherche et de filtrage des cours
- Proposer une fonction de recherche des cours par tags
- Disposer d'un dispositif de téléchargement des cours audio et vidéo
- Présenter les derniers ajouts de cours sous forme de flux rss, podcasting
- Etre disponible en plusieurs langues, en donnant la possibilité de rajouter facilement une nouvelle langue
- Proposer plusieurs thèmes visuels, en donnant la possibilité d'en rajouter de nouveaux facilement
- Contenir des outils d'administration permettant de gérer la base de données
- Etre interopérable avec l'application Univ-R développée par l'ULP Multimédia
- Disposer d'une documentation technique et d'une aide utilisateur
- Disposer d'un fichier de configuration pour définir les paramètres de l'application tels que le dépôt des ressources

## II. Technologies

JAVA 1.5

POSTGRES 8.3

TOMCAT 6.0.18, APACHE 2.2.11,

SERVEUR LINUX (Ubuntu 9.04)

*Serveur fms 3 sous linux, licence illimité*

### III. Autres programmes requis

Pour pouvoir mettre en place le site AudioVideoCours, il vous faut installer un certain nombre de logiciels, programmes, et librairies :

- Le serveur web *Apache 2.2* (avec le connecteur tomcat mod\_proxy\_ajp)
- Le *Java Runtime Environment (JRE)* version 1.5
- Le SGBD PostgreSQL
- Un serveur FTP (vsftpd 2.0.7)
- Pour encoder les fichiers podcast, il faut installer les paquets suivants:
  - o oggenc (paquet vorbis-tools 1.2.0)
  - o lame 3.98
  - o mp32ogg 0.11
  - o mplayer 2:1.0, mencoder 2:1.0
  - o ltag 0.14
  - o ffmpeg 3:0 (avec libavcodec-unstripped-52, libavdevice52, libavfilter0, libavformat52, libavutil-unstripped-49, libpostproc51, libswscale0)
  - o libfaad0 2.6.1
  - o mp3info 0.8.5
- Pour créer et décompacter des archives, les programmes *zip 2.32* et *unzip 5.52*
- Python 2.6 avec le paquet suivant :
  - o python-reportlab 2.3 pour lancer le script de création des PDF des diapos de cours.

### IV. Formats multimédia utilisés

En diffusion: Flash (par défaut, FLV), SMIL/REAL, MP3, OGG, PDF, ZIP (contenant diapo+media), MP4(HD)

En envoi client: ZIP (contenant flv, mp3, diapositives, timecodes)

En upload: vidéo: Avi, Mov, Mp4, Rm, Rv, Mkv, Divx, Mpg, Wmv, Flv.

audio: Mp3, Ogg.

## V. Démonstration

### 1. Processus de création et d'ajout de cours en différé

1. L'enseignant enregistre son cours grâce au client Audiovideocours. Quand celui-ci est fini, il remplit les informations relatives au cours telles que le titre, l'auteur, la formation, la description, et le code d'accès éventuel servant à rendre un cours privé. Il valide ensuite l'envoi vers le serveur.
2. Le client crée une archive du cours contenant l'enregistrement audio ou vidéo, les captures d'écran du support de cours, et un fichier .csv contenant la liste des « timecodes » de synchronisation des diapositives de cours avec la bande son ou la vidéo.
3. Le client envoie l'archive de cours au serveur via FTP et appelle le contrôleur en lui passant les informations sur le cours et le nom du fichier envoyé par FTP.
4. L'archive de cours est extraite dans le répertoire stockant les cours sur le serveur.
5. Le type de cours est détecté en fonction de la présence d'un fichier .mp3, .rm ou .flv dans le répertoire du cours.
6. Le script Python de vérification des miniatures des diapositives de cours est lancé. Celui-ci permet de créer les miniatures manquantes des diapositives de cours.
7. Si le cours est un cours audio, le fichier .mp3 est renommé avec l'id du cours et retraité grâce à un programme C qui permet d'éviter les problèmes de lecture dans RealPlayer. Si le cours est un cours vidéo, le fichier .flv est renommé avec l'id du cours et un fichier .mp3 de ce cours est créé grâce à un script shell.
8. Les métadonnées avec les informations sur le cours sont ajoutées au fichier .mp3 du cours.
9. La durée du cours est récupérée à partir du fichier .mp3 du cours.
10. Un fichier .pdf contenant toutes les diapositives de cours est créé par un script Python.
11. Le SMIL du cours est créé. Il utilise le fichier .csv pour réaliser la synchronisation entre les médias audio ou vidéo et l'affichage des diapositives de cours.
12. Le fichier .mp3 est converti en .ogg (oggEnc).
13. Un fichier .zip est créé contenant le média du cours (.mp3 ou .flv), le fichier .smil, et le fichier .pdf contenant l'ensemble des diapositives de cours.
14. Le cours est ajouté dans la base de données.
15. Les flux RSS sont régénérés.

### 2. Page « mon espace »

Permet de déposer un fichier audio ou video sur le site.

1. Se logger (mon espace) : <http://mon-site/avc/myspace>
2. Permet d'éditer ses cours envoyés avec le login ENT
3. Possibilité de « déposer un fichier » (voir 4. pour les types de fichiers supportés) et de modifier son e-mail

### **3. Lancement d'un enregistrement en direct**

1. Un enregistrement est lancé sur un client configuré pour la diffusion en direct.
2. Le client contacte le serveur en lui fournissant l'adresse IP de l'amphi et lui indiquant qu'il s'est mis à émettre
3. Le serveur modifie l'état de l'amphi dans sa base de données pour indiquer qu'un cours en direct est diffusé et donner l'accès à la page de visualisation de cours en direct pour cet amphi.
4. Le client envoie les diapositives de cours par FTP en écrasant successivement un fichier image dont le nom est formé à partir de l'adresse IP de l'amphi.
5. Lorsqu'un utilisateur désire visualiser un cours : si celui-ci est diffusé dans une salle permettant la diffusion en vidéo et en audio, un test est lancé pour voir si le client diffuse en audio ou en vidéo. Un cadre affichant les diapositives de cours dans la page web se recharge toutes les 10 secondes pour afficher la diapositive successivement modifiée par FTP. Le flux audio ou vidéo est lu dans la page via un player flash.
6. Lorsque le cours est terminé, Le client contacte le contrôleur en lui fournissant l'adresse IP de l'amphi et lui indiquant qu'il a cessé d'émettre.
7. Le serveur modifie l'état de l'amphi dans sa base de données pour indiquer qu'aucun cours n'est en diffusion et empêcher l'accès à la page de visualisation de cours en direct pour cet amphi.

## **VI. Fonctionnement de l'Interfaçage Univ-R**

1. Un enseignant se connecte sur la plate-forme WebUniv-R.
2. Il clique sur un lien pour enregistrer un cours. Une fenêtre lui demande d'entrer les informations relatives au cours telles que le titre, la description ...
3. Lorsque l'enseignant valide sa saisie, WebUniv-R contacte le contrôleur de l'application AudioVideoCours et lui donne les informations sur le cours ainsi que l'Id de l'enseignant, le code du groupe, et l'identifiant de session Tomcat de WebUniv-R et le nom de l'établissement
4. Le contrôleur vérifie que l'utilisateur est bien connecté à WebUniv-R, et qu'il a le droit de diffuser des cours pour le groupe grâce à l'API Univ-R.
5. Le contrôleur ajoute alors les informations qui sont pour le moment disponibles sur le cours dans la base de données, mais masque ce cours car les fichiers de cours n'ont pas encore été envoyés.
6. Le contrôleur contacte le client (présent sur la machine sur laquelle l'enseignant s'est connecté à WebUniv-R) par socket et lui envoie l'Id du cours qu'il a généré.
7. Le client débute alors l'enregistrement du cours.
8. Lorsque l'enregistrement est terminé, le client renvoie l'Id du cours et le nom du fichier envoyé par FTP au serveur.
9. Le serveur lance alors le processus d'ajout de cours en différé.
10. Ensuite, il complète les informations manquantes dans la base de données et rend le cours visible.
11. Enfin, il fait un ajout dans la base de données d'Univ-R grâce à l'utilisation de l'API Univ-R.

Ensuite, pour la visualisation d'un cours créé depuis Univ-R, le contrôleur dispose d'une méthode permettant l'affichage de l'interface de visualisation depuis Univ-R en contrôlant si l'utilisateur est bien connecté et qu'il a accès au cours.

## VII. Page d'administration

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

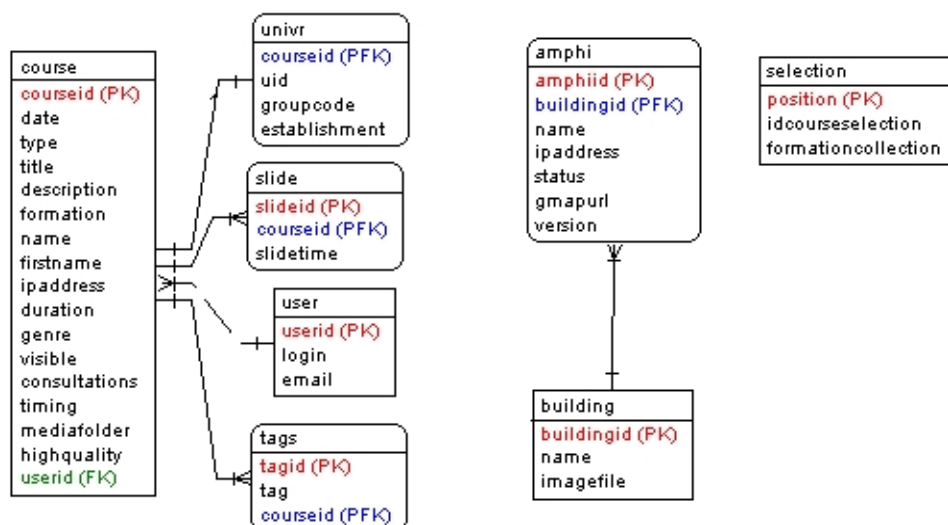
<http://mon-site/avc/admin>

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 « testulpmm »: <http://mon-site/avc/tests>)
8. Gestion des sélections et des collections de cours
9. Test des versions des clients des amphithéâtres

## VIII. Base de données

La connexion à la base de données se fait sous la forme d'un pool de connexions géré par Tomcat. Celui-ci a alors besoin du pilote JDBC de connexion au SGBD utilisé. Il doit être placé dans le dossier *common/lib* de l'application web et une ressource de données doit être configurée dans le fichier *server.xml*. Le script de création de la table se trouve dans le fichier *WebContent/scripts/script\_creation\_database* de l'application AudioVideoCours.

[1,1]



## IX. Stockage des cours

- FTP pour la réception des cours

Un serveur FTP doit être installé et un utilisateur FTP doit être créé avec un login et un mot de passe qui devront ensuite être configurés dans le client d'envoi des cours qui transmet ceux-ci par FTP.

Le répertoire de réception des archives de cours peut être configuré dans le fichier de configuration d'Audiovideocours (par exemple: /mon-ftp)

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

- o *canceled* sert à stocker les cours enregistrés par le client mais annulés avant la publication.
- o *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.
- o *live* est le répertoire de stockage des diapositives d'un cours en direct envoyées par FTP.
- o *releases* stocke les programmes d'installation du client Audiovideocours.

- Stockage des répertoires de cours

Le répertoire de stockage des cours peut également être modifié dans le fichier de configuration d'Audiovideocours (par exemple /mes-cours).

Chaque cours se trouve dans un répertoire horodaté+id du cours.



## X. Scripts

Différents scripts sont utilisés lors de l'ajout de cours. Ceux-ci sont de plusieurs langages différents : Shellscript, C, Python.

## XI. 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.
- Vérification toutes les heures de l'état du site (wget). On redémarre tomcat si on n'obtient pas de réponses.
- Génération de statistiques sur les médias téléchargés tous les jours.

```
# /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  
30 * * * * root bash repertoire-tomcat5.5/webapps/univ-  
r_av/scripts/AVC_test/AVC  
15 15 * * * root bash repertoire-tomcat5.5/webapps/univ-  
r_av/scripts/AVC_Stat_scpt/script_log > /dev/null
```

## XII. Mise en place de la solution technique

### 1. Récupération des sources et le développement

Pour récupérer et modifier les sources, il faut installer un IDE, par exemple : Eclipse IDE for Java EE Developers (3.3 Europa)

De plus, le JDK Java 1.5 doit être installé et configuré dans Eclipse, de même que le serveur d'application Tomcat5.5.

Le plugin *Subclipse* d'Eclipse peut être utilisé pour télécharger directement dans l'IDE les sources de l'application enregistrées sur un serveur subversion. L'adresse de ce serveur peut être trouvée sur le site web <http://audiovideocours.u-strasbg.fr>.

Pour créer un fichier .war de l'application depuis Eclipse, il suffit d'utiliser le fichier build.xml, qui est à la racine du projet, en l'ajoutant en tant que tâche ANT pour l'exécuter.

## 2. Déploiement

- Configurer le site par le fichier univrav.properties dans le répertoire conf des sources (voir page suivante)
- Génération du .war de l'application via les sources présentes sur le SVN, à l'aide du fichier build.xml qui est à la racine du projet (ANT).
- Déploiement du .war sur Tomcat (actuellement via l'URL: <http://mon-serveur-tomcat:8180/manager/html>)

univrav.properties :

```
# The Url to access to the course on internet
coursesUrl = http://mon-site/coursv2/

# The Url to access to the statistics generated by Webalizer
statsUrl = http://mon-site/webalizer/
oldStatsUrl = http://mon-site/webalizer/avc/

# Folders on the file system
coursesFolder = /mes-cours
ftpFolder = /mon-ftp
liveFolder = /mon-ftp/live/

# Default media filenames in the archive sent by the client
defaultMp3File = enregistrement-micro.mp3
defaultRmFile = enregistrement-video.rm
defaultSmilFile = cours.smil
defaultFlashFile = enregistrement-video.flv
defaultScreenshotsFolder = screenshots

# Copyright comment
comment = Owned by the author

# IP address of the Flash Server for the video and audio live
flashServerIp = xx.xxx.xxx.xxx

# The settings of the RSS files and for the permalien (interface flash) and email
rssTitle = Univ-R AV
rssName = univrav
rssDescription = Audiovidéocours
serverUrl = http://mon-site
rssImageUrl = http://mon-site/files/img/univr-av-logo-rss.png
recordedInterfaceUrl = http://mon-site/avc/courseaccess
language = fr

# The numbers of courses to display at the same time
```

```
homeCourseNumber = 4
recordedCourseNumber = 10

# The default style
defaultStyle = orange-theme

# 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 = Testulpmm
# testKeyWord3 = Essai

# The client port for the Univ-R integration
clientSocketPort = 3737

#Admin email for notification
#AdminEmail1 is used for "Contact"
adminEmail1=monmail1@mail.mail
adminEmail2=monmail2@mail.mail
adminEmail3=monmail3@mail.mail

#CAS Logout
casLogoutUrl=https://mon-serveur-cas:8443/cas/logout
```

### 3. Configuration de Tomcat

- Le fichier `.jar` du pilote jdbc de connexion à la base de données *postgresql-\*.jdbc.jar* doit être placé dans le répertoire *common/lib* de Tomcat, ainsi que les bibliothèques *commons-fileupload-1.1.1.jar* et *commons-io-1.2.jar*
- Un utilisateur et un rôle doivent être créés dans le fichier *conf/tomcat-users.xml* pour les pages nécessitant une authentification (pages de l'interface d'administration).  
Voir: repertoire-tomcat5.5/conf/tomcat-users.xml
- Pour la connexion SSL, il faut modifier les ports de redirection SSL de tomcat 8443 en 443, qui est le port SSL d'Apache, dans le fichier *server.xml*. En effet, la connexion SSL est géré par Apache.

Ainsi un utilisateur *ulp* de rôle *ulpmm* est créé par exemple.

- Un hôte virtuel doit être configuré dans le fichier *conf/server.xml* ainsi que la source de données permettant l'accès au pool de connexion à la base de données PostgreSQL.  
Voici un exemple de configuration pour créer l'hôte virtuel *mon-site* pointant vers l'url [http://mon-site/univ-r\\_av](http://mon-site/univ-r_av) et contenant les informations de connexion à la base de données *ma-base* :  
Voir: repertoire-tomcat5.5/conf/server.xml

```
<Host name="mon-site" appBase="webapps" unpackWARs="true"
  autoDeploy="true" xmlValidation="false" xmlNamespaceAware="false">

  <Context path="" docBase="univ-r_av">
```

```
<Resource name="jdbc/postgres" auth="Container"
type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
url="jdbc:postgresql://databaseserver.u-strasbg.fr:5432/ma-base"
username="databaseuser" password="secret" maxActive="20" maxIdle="10"
maxWait="-1" />

</Context>
</Host>
```

- 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. Dans le fichier */etc/init.d/tomcat5.5*, section *start*, ajouter la ligne suivante :

```
rm -rf $CATALINA_HOME/work/Catalina/* || echo "Impossible de supprimer le
cache";
```

## 4. Configuration d'Apache et utilisation du mod\_jk

mod\_jk est le module qui permet la communication entre Apache et Tomcat. Ainsi, Apache recevra toutes les requêtes sur son port par défaut (80) et transmettra automatiquement les requêtes des pages JSP vers le serveur Tomcat.

Il faut créer un fichier *mon-site.conf* dans le répertoire *sites\_available* d'Apache 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 *coursv2* pour pouvoir accéder aux cours depuis l'extérieur, les dossiers *live* et *releases* mettant à disposition des fichiers via FTP, et le dossier *manager* pour donner l'accès à l'interface *manager* de Tomcat.

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>
```

```
ServerAdmin webmaster@localhost
```

```
JkMount /*.jsp ajp13_worker
JkMount /admin/*.jsp ajp13_worker
JkMount /avc/* ajp13_worker # URL of the controller
JkMount /audiocours_v2/servlet/* ajp13_worker # compatibility with the old clients
JkMount /manager/* ajp13_worker
```

```
DocumentRoot /repertoire-tomcat5.5/webapps/univ-rav/
```

```
<Directory />
Options FollowSymLinks
AllowOverride None
</Directory>
```

```
Alias /manager "/repertoire-tomcat5.5/server/webapps/manager"
Alias /mes-cours "/mes-cours"
Alias /live "/mon-ftp/live"
Alias /releases "/mon-ftp/releases"
```

```
<Directory "/repertoire-tomcat5.5/server/webapps/manager">
Options -Indexes FollowSymLinks MultiViews
DirectoryIndex index.jsp
AllowOverride None
Order allow,deny
allow from all
</Directory>
```

```
<Directory "/repertoire-tomcat5.5/webapps/univ-r_av/">
Options -Indexes FollowSymLinks MultiViews
DirectoryIndex index.jsp
AllowOverride None
Order allow,deny
allow from all
</Directory>
```

```
<Directory "/mes-cours">
Options -Indexes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
allow from all
</Directory>

<Directory "/live">
Options -Indexes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
allow from all
</Directory>

<Directory "/releases">
Options -Indexes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny
allow from all
</Directory>
```

## 5. Configuration de la base de données

Afin de pouvoir utiliser le site Audiovideocours, il faut avoir le système de gestion de base de données PostgreSQL d'installé. Vous pouvez utiliser pgAdmin pour vous aider à gérer PostgreSQL. Voici ce qu'il faut configurer :

- Utilisateurs et rôles  
Un rôle ayant des droits d'accès restreints doit être créé, il faut donc laisser les « droits du rôle » décochés comme par défaut. Un utilisateur doit être créé pour ce rôle, il ne faut pas oublier de lui définir un mot de passe. Le nom d'utilisateur et le mot de passe doivent être configurés comme *sous tomcat*
- postgresql.conf  
Le paramètre *listen-addresses* doit être avoir pour valeur '\*' (écoute sur toutes les adresses IP) afin d'accepter les connexions à la base de données depuis l'extérieur si l'on désire gérer la base de données avec pgAdmin.
- pg\_hba.conf  
Il faut modifier ce fichier pour autoriser les accès à distance. Ceux-ci doivent être configurés à *md5* et l'utilisateur créé précédemment doit avoir accès sur l'interface accessible depuis l'extérieur.
- Base de données  
Une base de données *univrav* doit être créée par le superutilisateur de PostgreSQL grâce au script de création de la base. Pour chacune des tables, il faut donner les droits *INSERT*, *SELECT*, *UPDATE*, *DELETE* (*arwd*) au rôle de l'utilisateur créé. L'URL d'accès à la base de données doit ensuite être configurée comme sous tomcat.