Installation et configuration - Esup Annuaire2

Récupération des sources

Vous pouvez récupérer les sources sur le dépôt Subversion de Sourcesup, à l'adresse suivante: `https://subversion.renater.fr/esup-annuaire2/`

Il est recommandé d'utiliser la dernière version (tags) ou le trunk.

Adaptation à l'établissement

Principes

Un établissement souhaitant adapter l'application à son annuaire Ldap (attributs spécifiques, règles de gestion particulières), devra écrire un nouveau module Maven en lieu et place de esup-annuaire2-ldep-services-lille1.
Nous recommandons d'enrichir le projet initial, en utilisant le SVN, de manière à mutualiser nos efforts.

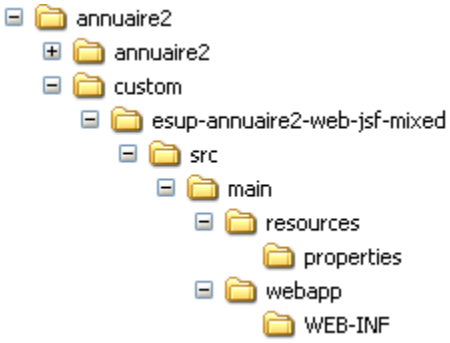
L'écriture d'un tel module **correspond à l'implémentation d'une série d'interfaces pour permettre la gestion des Personnes, des Rôles, des Structures et des Etudiants.**

- Les paramètres de configuration sont centralisés dans les fichiers defaults.properties et config.properties qui se trouvent dans le module esup-annuaire2-web-jsf-mixed (\src\main\resources\properties). Si le config.properties n'existe pas, créez le, il ne doit contenir que les éléments que vous souhaitez utiliser en lieu et place de ceux de defaults.properties.
  Prenez soin de ne jamais commiter config.properties car il peut contenir des éléments propres et sécurisés de votre établissement !
- Pour bien maitriser ces paramètres, reportez vous aux beans Spring où ils sont invoqués... Ces beans se trouvent dans le module esup-annuaire2-web-jsf-mixed (\src\main\resources\properties),**et aussi dans les répertoires \src\main\resources\META-INF\ des différents modules**.

Personnaliser le déploiement

Pour mettre pleinement en œuvre la procédure ci-dessus dans le cadre d'un établissement, tout en gardant vos paramètres de configuration particuliers, vous pouvez suivre ces différentes étapes:

1. Si ce n'est déjà fait, mettre en place un espace de BUILD qui soit différent de la production.
2. Créer dans cet espace un répertoire spécifique pour l'application (ici nommé annuaire2). Et dans celui-ci un répertoire "custom".
3. Y construire une hiérarchie de répertoires contenant vos fichiers personnalisés.



Dans le répertoire WEB-INF, vous aurez le "porlet.xml" personnalisé (**les éléments display-name et keywords**).

```
<?xml version="1.0" encoding="UTF-8"?>
<portlet-app xmlns="http://java.sun.com/xml/ns/portlet/portlet-app_1_0.xsd"
        version="1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/portlet-app_1_0.xsd
                        http://java.sun.com/xml/ns/portlet/portlet-app_1_0.xsd">

        <portlet>
                <portlet-name>${portlet.name}</portlet-name>
                <display-name xml:lang="fr">esup-annuaire2</display-name>
                <portlet-class>javax.portlet.faces.GenericFacesPortlet</portlet-class>
        <!-- if you don'nt need mobile, you can change this first page to
              /stylesheets/desktop/welcome.xhtml -->
                <init-param>
                        <name>javax.portlet.faces.defaultViewId.view</name>
                        <value>/stylesheets/home.xhtml</value>
                </init-param>
                <expiration-cache>0</expiration-cache>
                <supports>
                        <mime-type>text/html</mime-type>
                        <portlet-mode>VIEW</portlet-mode>
                </supports>
                <supported-locale>fr</supported-locale>
                <supported-locale>en</supported-locale>
                <portlet-info>
                        <title>${portlet.name}</title>
                        <short-title>${portlet.name}</short-title>
                        <keywords>esup-annuaire2</keywords>
                </portlet-info>
        </portlet>

</portlet-app>
```

Dans le répertoire properties, vous aurez le fichier config.properties de l'application personnalisé(**les éléments ici laissés vides**) .

Précisions:

- ldap.alternateUrl correspond à un annuaire de secours. Si vous n'en possédez pas, reprenez l'Url de l'annuaire.
- annuaire2.ldap.schema correspond au préfixe des classes implémentant une série d'interfaces pour permettre la gestion des Personnes, des Rôles, des Structures et des Etudiants (voir plus loin applicationContext.xml).
- tomcat.port.string correspond par exemeple à ":8080".

```
################################################################################
# AUTHENTICATION
#

cas.url=

################################################################################
# CACHE
#

cache.path=

################################################################################
# DOMAIN
#

directoryMaintainerMail=

################################################################################
# EXCEPTIONS
#

exceptionHandling.email=

################################################################################
# LOGGING
#

log.level=debug
log.output=file
log.file=

################################################################################
# LDAP
#

ldap.url=
ldap.alternateUrl=
ldap.userDn=
ldap.password=
ldap.base=dc=
ldap.testUserFilter=uid=
annuaire2.ldap.schema=

################################################################################
# SERVER
#
tomcat.host=
tomcat.port.string=

################################################################################
# SMTP
#

smtp.host=
smtp.fromEmail=
smtp.interceptAll=
smtp.interceptEmail=
```

Toujours dans le répertoire properties, vous aurez aussi le fichier applicationContext.properties (**ici la variable annuaire2.ldap.schema utilisée est "Lille1"**):

```
<?xml version="1.0" encoding="UTF-8" ?>
<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="http://www.springframework.org
/schema/context"
        xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema
/beans/spring-beans-3.0.xsd
            http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-
context-3.0.xsd
                http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-
3.0.xsd">

        <!-- /properties/applicationContext.xml The main Spring configuration file.
                This file should not be modified but by the developers. This file is referenced
                by /webapp/WEB-INF/web.xml. -->

        <context:property-placeholder
                location="classpath:/properties/defaults.properties,classpath:/properties/config.properties" />

        <bean id="app_context"
                class="org.esupportail.commons.context.ApplicationContextHolder" />

        <import resource="classpath*:META-INF/esup-annuaire2-domain-services-auth.xml" />
        <import resource="classpath*:META-INF/esup-annuaire2-domain-services-domain.xml" />

        <import
                resource="classpath*:META-INF/esup-annuaire2-ldap-services-ldapCommons.xml" />

        <!-- Attention, dans un environnement de production, la substitution de
                variable ne s'applique pas aux imports Il est donc indispensable d'effectuer
                le remplacement de la variable annuaire2.ldap.schema manuellement -->
        <import
                resource="classpath*:META-INF/esup-annuaire2-ldap-services-ldapLille1.xml" />
        <import
                resource="classpath*:META-INF/esup-annuaire2-ldap-services-locationUtilLille1.xml" />

        <import resource="classpath*:META-INF/esup-annuaire2-utils-agentUtil.xml" />

        <import resource="cache/cache.xml" />
        <import resource="exceptionHandling/exceptionHandling.xml" />
        <import resource="i18n/i18n.xml" />
        <import resource="misc/abstractBeans.xml" />
        <import resource="misc/application.xml" />
        <import resource="smtp/smtp.xml" />
        <import resource="tags/tags.xml" />
        <import resource="web/controllers.xml" />
        <import resource="ws/cxf.xml" />

</beans>
```

Déployer en mode portlet / servlet

1. Depuis ce répertoire (annuaire2, à la racine de l'image des répertoires ci-dessus) , lancer un script avec les commandes suivantes (ici avec le trunk):

   ```
   svn co https://subversion.renater.fr/esup-annuaire2/trunk/esup-annuaire2
   zip -r packages/esup-annuaire2.zip esup-annuaire2/
   rm -rf esup-annuaire2/
   ```

2. puis lancer un autre script avec les commandes suivantes:

```
ant init
cd=`pwd`
cd annuaire2
mvn clean install
cd esup-annuaire2-web-jsf-mixed/
mvn -DProduction=true clean package
cd /home/tomcat/portail/BUILD/esup-package
ant portlet.deploy -DportletApp=/home/tomcat/portail/BUILD/applis/annuaire2/annuaire2/esup-annuaire2-web-
jsf-mixed/target/esup-annuaire2-0.1.0.war
cd $cd
```

Commentaires:
La cible "init" va partir du .zip construit par le script précédent pour y incorporer les fichiers que vous avec personnalisés, elle est constituée ainsi:

```
<target name="init" depends="unzip">
    <copy overwrite="true" todir="${appli.base}/">
        <fileset dir="custom" includes="**/*" />
    </copy>
</target>

<target name="unzip">
    <delete dir="${appli.base}"/>
    <delete dir="${appli.tmp}"/>
    <mkdir dir="${appli.tmp}"/>
    <unzip src="${appli.packages}/${appli.package-name}" dest="${appli.tmp}"   />
    <move todir="${appli.base}/">
        <fileset dir="${appli.tmp}/${appli.name-version}"/>
    </move>
</target>
```

Les propriétés utilisées sont celles-ci:

```
appli.name-version=esup-annuaire2
appli.package-name=${appli.name-version}.zip
appli.name=annuaire2
appli.root=/home/tomcat/portail/BUILD/applis/${appli.name}
appli.base=/home/tomcat/portail/BUILD/applis/${appli.name}/${appli.name}
appli.tmp=${appli.root}/temp
appli.packages=${appli.root}/packages
```

La commande Maven "clean install" va créer un répertoire "target", compiler les différents modules de l'application
On se place ensuite dans le module web de l'application.
La commande Maven -DProduction=true clean package réalise la construction du ficher WAR.
La suite, c'est l'assemblage dans un nouveau web.xml du web.xml livré dans le WAR avec le fichier portlet.xml.