

Installation

- [Pré-requis](#)
 - [Configuration Tomcat](#)
 - [Configuration Apache Shibboleth](#)
 - [Paramétrage mémoire JVM :](#)
- [Sources](#) : <https://github.com/EsupPortail/esup-carte-culture>
- [Configurations](#) (src/main/resources/META-INF/spring)
 - [database.properties](#)
 - [applicationContext-security.xml](#)
 - [applicationContext.xml](#)
- [Compilation / Déploiement](#)
 - [Obtention du war pour déploiement sur tomcat ou autre :](#)
 - [Déploiement](#)
 - [Lancement de la mise à jour de la base de données \(si besoin\)](#)

Pré-requis

- Java (OpenJDK 8) : le mieux est de l'installer via le système de paquets de votre linux.
- Maven : le mieux est de l'installer via le système de paquets de votre linux.
- PostgreSQL 9 : le mieux est de l'installer via le système de paquets de votre linux.
- Tomcat (Tomcat 8) <https://tomcat.apache.org/download-80.cgi>
- Apache + libapache2-mod-shib2 : <https://services.renater.fr/federation/docs/installation/sp>
- Git : le mieux est de l'installer via le système de paquets de votre linux.

Configuration Tomcat

Décompresser un serveur tomcat dans /opt/tomcat-carte-culture

Vérifier le connecteur AJP conf/server.xml

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" asyncTimeout="1200000" tomcatAuthentication="false" URIEncoding="UTF-8" />
```

Configuration Apache Shibboleth

L'authentification repose sur Shibboleth. Apache doit être configuré pour faire du mod_shib.

Une fois le SP Shibboleth et Apache configurés usuellement (voir : <https://services.renater.fr/federation/docs/installation/sp>), il faut sécuriser certains répertoires en ajoutant ceci à la conf apache (à adapter cependant en fonction des versions d'Apache et mod_shib) :

```

ProxyPass /Shibboleth.sso !
ProxyPass /secure !
ProxyPass / ajp://localhost:8009/ ttl=10 timeout=3600 loadfactor=100 retry=1

<Location /secure>
    AuthType shibboleth
    ShibRequestSetting requireSession 1
    require shib-session
    ShibUseHeaders On
</Location>
<Location /login>
    AuthType shibboleth
    ShibRequestSetting requireSession 1
    require shib-session
    ShibUseHeaders On
</Location>
<Location /partenaire>
    AuthType shibboleth
    ShibRequestSetting requireSession 1
    require shib-session
    ShibUseHeaders On
</Location>
<Location /admin>
    AuthType shibboleth
    ShibRequestSetting requireSession 1
    require shib-session
    ShibUseHeaders On
</Location>
<Location /etudiant>
    AuthType shibboleth
    ShibRequestSetting requireSession 1
    require shib-session
    ShibUseHeaders On
</Location>
<Location /manager>
    AuthType shibboleth
    ShibRequestSetting requireSession 1
    require shib-session
    ShibUseHeaders On
</Location>
<Location /ent>
    Deny from All
    Allow from 'ip autorisées'
</Location>
<Location /ws>
    Deny from All
    Allow from 'ip autorisées'
</Location>

```



Attention, il ne faut pas sécuriser le dossier racine "/" pour laisser l'accès libre à certains web service.

Configuration PostgreSQL

- pg_hba.conf : ajout de

```
host all all 127.0.0.1/32 password
```

- redémarrage de postgresql
- psql

```
create database carteculture;  
create USER carteculture with password 'carteculture';  
grant ALL ON DATABASE carteculture to carteculture;
```

Paramétrage mémoire JVM :

Pensez à paramétrer les espaces mémoire JVM :

```
export JAVA_OPTS="-Xms1024m -Xmx1024m -XX:MaxPermSize=256m"
```

Pour maven :

```
export MAVEN_OPTS="-Xms1024m -Xmx1024m -XX:MaxPermSize=256m"
```

Sources : <https://github.com/EsupPortail/esup-carte-culture>

```
cd /opt  
git clone https://github.com/EsupPortail/esup-carte-culture
```

Configurations (src/main/resources/META-INF/spring)

database.properties

```
database.driverClassName=org.postgresql.Driver  
database.url=jdbc\:postgresql\://localhost\:5432/carteculture  
database.username=carteculture  
database.password=carteculture
```

applicationContext-security.xml

ici vous paramétrez l'adresse IP du serveur Esup-Nfc-Tag, l'attribution des roles en fonction des groupes et les différents ldap auxquels on doit accéder.

```

<http disable-url-rewriting="true" use-expressions="true" entry-point-ref="http403EntryPoint">
...
    <intercept-url pattern="/nfc-ws/**" access="hasIpAddress('127.0.0.1') or hasIpAddress('Adresse IP du
serveur Esup-Nfc-Tag')" />
...
</http>

<beans:bean id="authUserDetailsService"
    class="org.esupportail.esupnfccarteculture.security.ShibAuthenticatedUserDetailsService">
    <beans:property name="mappingGroupesRoles">
        <beans:map>
            <beans:entry key="cn=for.carte-culture.admin,ou=groups,dc=univ-ville,dc=fr"
value="ROLE_ADMIN" />
            <beans:entry key="cn=for.carte-culture.manager,ou=groups,dc=univ-ville,dc=fr"
value="ROLE_MANAGER" />
            <beans:entry key="cn=for.carte-culture.partenaire,ou=groups,dc=univ-ville,
dc=fr" value="ROLE_PARTENAIRE" />
        </beans:map>
    </beans:property>
</beans:bean>

<beans:bean id="ldapContextSourceUnivVille" class="org.springframework.ldap.core.support.LdapContextSource">
    <beans:property name="url" value="ldap://ldap.univ-ville.fr" />
    <beans:property name="base" value="dc=univ-ville,dc=fr" />
    <beans:property name="userDn" value="cn=test,dc=univ-ville,dc=fr" />
    <beans:property name="password" value="secret" />
</beans:bean>

<beans:bean id="ldapContextSourceAutre" class="org.springframework.ldap.core.support.LdapContextSource">
...
</beans:bean>

```

applicationContext.xml

Dans ce fichier il faut déclarer les beans LdapTemplate et PersonLdapDao correspondants aux différentx Ldap auxquels on accède.

Le bean TagService permet de paramétrer :

- Le nombre maximum de recharges
- Les différents types de salle et le nombre de coupons qui y sont attribuées
- Le type de salle TypeSalleJoker est spécial, ce type de coupon est utilisable dans tous les autres types de salle

Le bean EtudiantService permet de régler les paramètres suivants:

- affichageDetailCoupons : L'affichage du détail des coupons dans la GUI
- preInscription : Activer la pré-inscription (inscription via la GUI sans utilisation de la carte)
- preInscriptionNomSalle : Le nom de la salle de pré-inscription
- authorizedStudentLdapFilter : le filtre ldap des personnes autorisées à s'inscrire à la carte culture
- eppnFilterRegex : une expression régulière pour filtrer les eppn des personnes autorisées à s'inscrire à la carte culture
- ldapCsnSearchAttribut : le nom de l'attribut ldap dans lequel se trouve le CSN des cartes étudiant
- ldapCsnMultiValueTag : préciser le nom du tag dans le cas où le ldapCsnSearchAttribut est multivalué
- ldapCsnMultiValueTagExtractRegex : expression régulière permettant d'extraire le CSN provenant d'un champ multivalué

```

<bean id="ldapTemplateUnivVille" class="org.springframework.ldap.core.LdapTemplate">
    <constructor-arg ref="ldapContextSourceUnivVille" />
</bean>

<bean id="ldapTemplateAutre" class="org.springframework.ldap.core.LdapTemplate">
    <constructor-arg ref="ldapContextSourceAutre" />
</bean>

<bean id="personDaoUnivVille" class="org.esupportail.esupnfccarteculture.ldap.PersonLdapDao">
    <property name="ldapTemplate" ref="ldapTemplateUnivVille" />
</bean>

<bean id="personDaoAutre" class="org.esupportail.esupnfccarteculture.ldap.PersonLdapDao">
    <property name="ldapTemplate" ref="ldapTemplateAutre" />
</bean>

<bean id="tagService" class="org.esupportail.esupnfccarteculture.service.TagService">
    <property name="nbRechargeMax" value="3" />
    <property name="typeSalles">
        <list>
            <bean id="salleJoker" class="org.esupportail.esupnfccarteculture.domain.
TypeSalleJoker">
                <property name="maxCoupon" value="1" />
                <property name="description" value="Coupon utilisable dans tous les types
de salle" />
            </bean>
            <bean id="salleCinema" class="org.esupportail.esupnfccarteculture.domain.
TypeSalle">
                <property name="nom" value="cinema" />
                <property name="maxCoupon" value="0" />
                <property name="description" value="Coupon cinéma" />
            </bean>
            <bean id="salleTheatre" class="org.esupportail.esupnfccarteculture.domain.
TypeSalle">
                <property name="nom" value="theatre" />
                <property name="maxCoupon" value="1" />
                <property name="description" value="Coupon théâtre" />
            </bean>
            <bean id="salleAutre" class="org.esupportail.esupnfccarteculture.domain.
TypeSalle">
                <property name="nom" value="autre" />
                <property name="maxCoupon" value="2" />
                <property name="description" value="Coupon autre" />
            </bean>
        </list>
    </property>
</bean>

<bean id="etudiantService" class="org.esupportail.esupnfccarteculture.service.EtudiantService">
    <property name="affichageDetailCoupons" value="true" />
    <property name="preInscription" value="true" />
    <property name="preInscriptionNomSalle" value="Pré-inscription" />
    <property name="authorizedStudentLdapFilter" value="(|(eduPersonAffiliation=student)
(eduPersonAffiliation=affiliate))" />
    <property name="eppnFilterRegex" value=".*@univ-ville.fr|.*@univ-autre.fr"/>
    <property name="ldapCsnSearchAttribut" value="supannRefId" />
    <property name="ldapCsnMultiValueTag" value="{ISO15693}" />
    <property name="ldapCsnMultiValueTagExtractRegex" value="^.*\\{ISO15693\\}([^;]*).*$" />
</bean>

```

Compilation / Déploiement

Obtention du war pour déploiement sur tomcat ou autre :

```
cd /opt/esup-carte-culture
mvn clean package
```

Déploiement

On copie/colle le répertoire webapp packagé ainsi dans le tomcat :

```
rm -rf /opt/tomcat-carte-culture/webapps/ROOT && cp -rf /opt/esup-carte-culture/target/esupCarteCulture-X.X.X-SNAPSHOT /opt/tomcat-esup-carte-culture/webapps/ROOT
```

On arrête le tomcat avant et on le redémarre ensuite.

Lancement de la mise à jour de la base de données (si besoin)

```
mvn exec:java -Dexec.args="dbupgrade"
```