

Comment déployer une application J2EE

Information serveur :

- Serveur : ellsworth.iut2.upmf-grenoble.fr (accessible uniquement par VPN)
- Java version "1.8.0_121"
- Payara version "5.184"
- MySQL version "5.5.53"

VPN IUT2 : Pour déployer, besoin du VPN IUT2 !

Les documents pour utiliser le VPN se trouvent sur l'intranet de l'IUT2 (<http://www.iut2grenoble.com/index.php?action=lire&id=18063>) dans :

PÔLES & SERVICES > Pôle Informatique > Comment faire? > Spécifiques étudiants > Accéder aux ressources informatiques de l'IUT2 à distance > Installer le client

Etape 0. Prendre en main son environnement glassfish et phpmyadmin

On va vous attribuer un numéro qui déterminera votre login : 50 -> etudiant50

Login : etudiantXX (XX étant votre numéro)

Mot de passe : (jalmmi4)

- **Glassfish 5**

Accès à l'administration de votre domaine : <https://ellsworth.iut2.upmf-grenoble.fr:XX48/>

Modifier le mot de passe : Domain > Administrator Password

- **Phpmyadmin**

Accès : <http://ellsworth.iut2.upmf-grenoble.fr/phpmyadmin>

Modifier le mot de passe : Page d'accueil > "modifier le mot de passe"

Etape 1. Créer un WAR (Web ARchive) avec intellij

- **Créer une persistence-unit** avec les informations du serveur de production dans le fichier dans le persistence.xml

Modifier dans "Projet JPA MYSQL DIST" l'url, le user et le password :

```
<property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/etudiantXX"/>;  
<property name="javax.persistence.jdbc.user" value="etudiantXX"/>;  
<property name="javax.persistence.jdbc.password" value="NEW_PASSWORD"/>;
```

- **Changer d'appel au persistence-unit** en fonction de votre déploiement dans le GestionFactory.java

Par exemple, pour le déploiement sur le serveur ellsworth

```
//private static final String PERSISTENCE_UNIT_NAME = "Projet_JPA_SQLITE";
//private static final String PERSISTENCE_UNIT_NAME = "Projet_JPA_MYSQL";
private static final String PERSISTENCE_UNIT_NAME = "Projet_JPA_MYSQL_DIST";
```

- **Ajouter la librairie MySQL** permettant la connection au SGBD sur le serveur

Dans votre pom.xml, ajouter :

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.23</version>
</dependency>
```

Puis penser à recharger et installer cette dépendance :

Dans IntelliJ, dans l'onglet Maven (à droite), *reload All Maven Projects* puis dans lifecycle lancer la commande *install*.

- **Changer le "level" de Java** pour correspondre au attente du serveur (java 1.8) :

File > Project structure > Project > Project Language Level > Choisir 8 - Lambdas, type annotations etc.

- (Optionnel au cas où vous n'avez pas de .war) **Créer l'artifact WAR :**

Dans IntelliJ, *File > Project structure > Artifacts > + Web Archive: Archive > For "project:war:expoded"*

- **Compiler le WAR (Web ARchive) :**

Dans IntelliJ, *Build > Build Artifact > All Artifacts > Clean*

Build > Build Artifact > All Artifacts > Buuild

Etape 2. Déployer le WAR sur le serveur en passant par l'administration web

Accès à l'administration de votre domaine : <https://ellsworth.iut2.upmf-grenoble.fr:XX48/>

Déployer : *Application > Deploy ...*

- *Choose File >* prendre le fichier .war dans votre projet target
- Ajouter un context root (path relative to server's base URL) : votre LOGIN IUT (Par exemple, brunetfr)

Etape 3. Vérifier le résultat

Votre projet sera accessible à l'adresse :

<http://ellsworth.iut2.upmf-grenoble.fr:XX80/LOGIN>

Par exemple :

<http://ellsworth.iut2.upmf-grenoble.fr:8180/brunetfr>