



Manuel d'installation : Service Desk

Victor Marfil

Lilian Miesiac

Quentin Monnet

Luca Grunenwald

15.01.2020

TABLE DE MATIERE

TABLE DE MATIERE	1
Pré-requis	2
Récupération des sources	3
Vérification de la configuration de WAMP	4
Compilation des sources	5
Lancement de l'application	9

Pré-requis

Pour pouvoir utiliser notre application, plusieurs prérequis sont nécessaires.

- 1) Puisque l'application a été développée en Java, il faut avoir Java d'installer sur la machine

Il faut donc avec le Java Runtime Environment (JRE) trouvable gratuitement à cette adresse:

<https://www.oracle.com/java/technologies/javase-jre8-downloads.html>

De plus, il faut avoir le Java Development Kit (JDK) trouvable aussi gratuitement a cette adresse:

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

- 2) Un environnement de développement intégré (IDE) supportant Maven.

Pour notre cas, nous utiliserons Netbeans 12.0 trouvable à cette adresse:

<https://netbeans.apache.org/download/nb120/nb120.html>

- 3) Un serveur MySQL

Pour nous simplifier les choses, dans ce manuel d'installation nous utiliserons WAMP (en local) qui est trouvable à cette adresse:

<https://www.wampserver.com/en/>

Attention, WAMP ne fonctionne que sur Windows (d'où le W dans le nom), il existe des alternatives gratuites sur Linux tel que LAMPP.

Récupération des sources

Pour pouvoir installer OSLO, il est nécessaire de récupérer les sources sur github. Pour ce faire, il est conseillé d'avoir installé GIT et ensuite rentrer la ligne suivante dans un invite de commande:

```
D:\Code\Example>git clone https://github.com/Ginkishi/Project_Oslo.git
```

(Ligne de commande: git clone https://github.com/Ginkishi/Project_Oslo.git)

/!\ Cette ligne va cloner le répertoire github dans le répertoire local courant /!\

Si la commande à réussi, vous devriez avoir un résultat proche de celui-ci:

```
D:\Code\Example>git clone https://github.com/Ginkishi/Project_Oslo.git
Cloning into 'Project_Oslo'...
remote: Enumerating objects: 110, done.
remote: Counting objects: 100% (110/110), done.
remote: Compressing objects: 100% (72/72), done.
remote: Total 4212 (delta 49), reused 68 (delta 23), pack-reused 4102
Receiving objects: 100% (4212/4212), 113.64 MiB | 2.45 MiB/s, done.
Resolving deltas: 100% (2387/2387), done.
warning: the following paths have collided (e.g. case-sensitive paths
on a case-insensitive filesystem) and only one from the same
colliding group is in the working tree:

'target/classes/templates/Fonction/edit.html'
'target/classes/templates/fonction/edit.html'
'target/classes/templates/Fonction/list.html'
'target/classes/templates/fonction/list.html'

D:\Code\Example>
```

Vérification de la configuration de WAMP

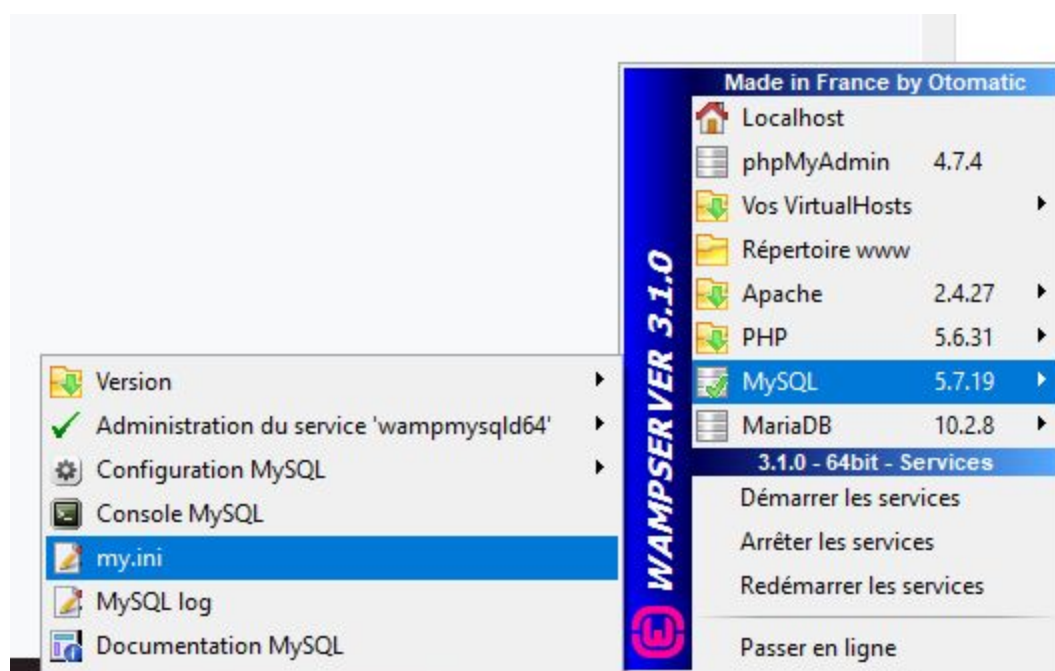
Pour vérifier comment est configuré WAMP, premièrement commencer par ouvrir l'application. Vous devriez voir en bas à droite de la barre des tâches de Windows un nouvel icône:



Si il n'est pas présent, cliquer sur la flèche, il devrait apparaître.

Attendez que cette icône devienne verte!

Cliquer sur cette icône puis allez dans "MySQL" puis cliquer sur "my.ini"



Une fois ouvert, retrouver la ligne "port =" situé sous "[wampmysqld64]" et noté bien sur quel port le serveur MySQL est lancé (dans l'image le port est 3306):

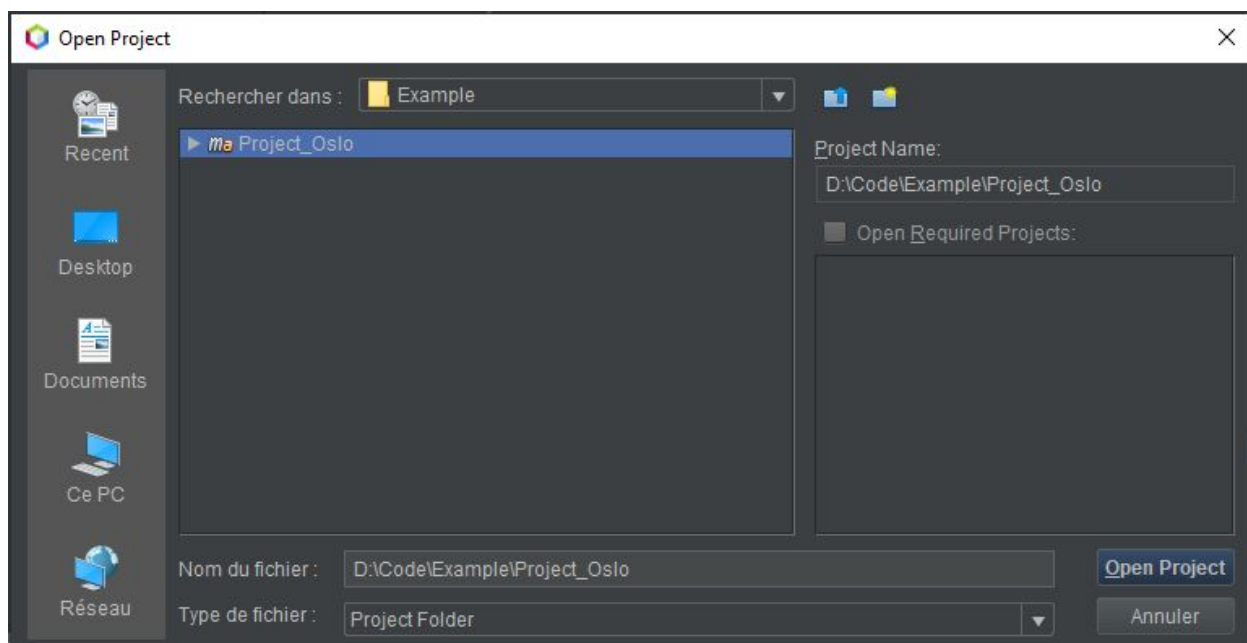
```
; The MySQL server
[wampmysqld64]
;skip-grant-tables
port = 3306
```

Compilation des sources



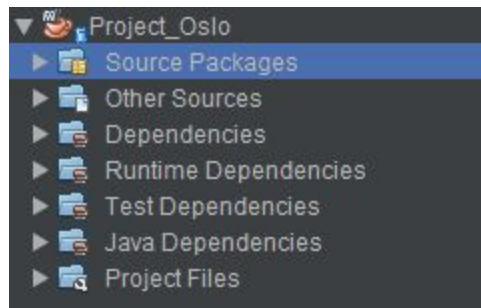
Pour pouvoir compiler les sources, il faut avoir un IDE capable de gérer Maven. Nous recommandons Netbeans et nous utiliserons des images de cet IDE (dans sa version 12.0) pour la suite.

Premièrement, il faut ouvrir le projet dans Netbeans:



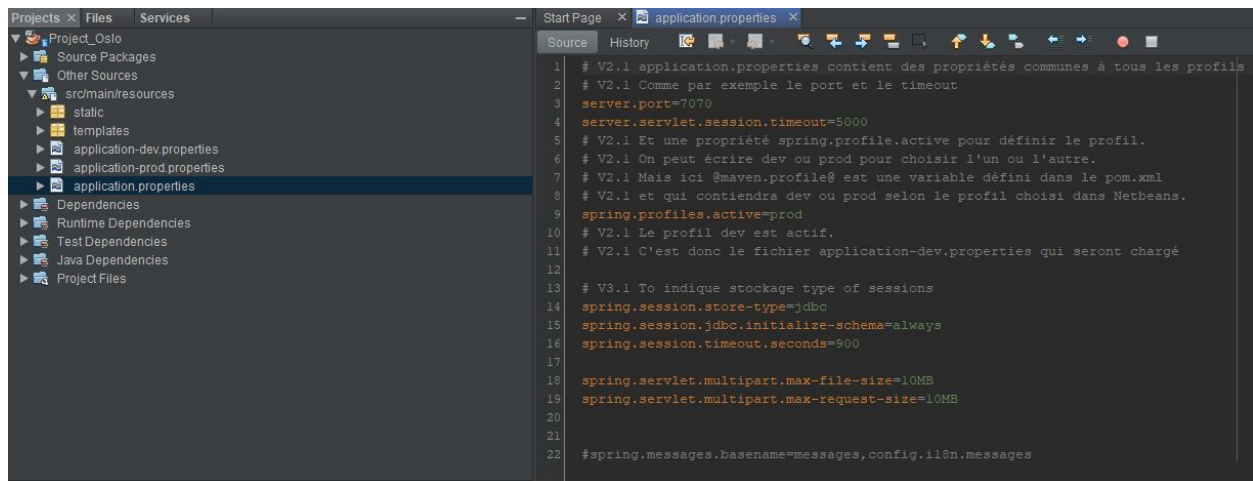
Il est bon de remarquer que le dossier est reconnu par Netbeans comme étant un projet Maven (l'icône "Ma" avant le nom du dossier).

Si vous avez choisi le bon dossier, vous devriez avoir une arborescence dans la partie "Projet" de Netbeans (à gauche).



Il va donc falloir maintenant configurer le projet et notamment l'accès à la base de données.

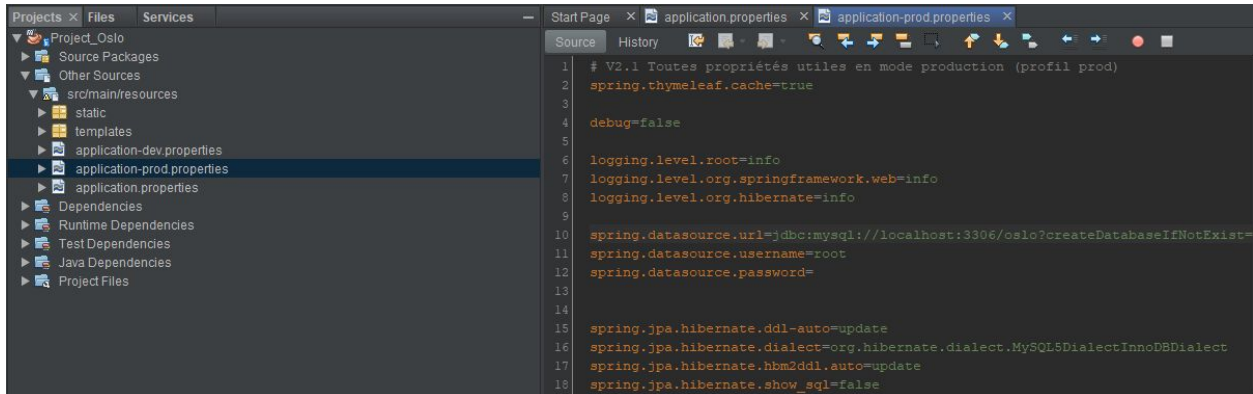
Pour faire cela, rendez vous dans "Other Sources" puis ouvrez le fichier "application.properties":



Dans ce fichier, veillez à ce que la valeur de "spring.profiles.active" soit bien mise à "prod"

Vous pouvez aussi changer le port sur lequel va être lancée l'application en changeant la ligne "server.port". Ici nous le laisserons à 7070.

Rendez vous ensuite dans le fichier “application-prod.properties”:



```
1 # V2.1 Toutes propriétés utiles en mode production (profil prod)
2 spring.thymeleaf.cache=true
3
4 debug=false
5
6 logging.level.root=info
7 logging.level.org.springframework.web=info
8 logging.level.org.hibernate=info
9
10 spring.datasource.url=jdbc:mysql://localhost:3306/oslo?createDatabaseIfNotExist=
11 spring.datasource.username=root
12 spring.datasource.password=
13
14
15 spring.jpa.hibernate.ddl-auto=update
16 spring.jpa.hibernate.dialect=org.hibernate.dialect.MySQL5DialectInnoDBDialect
17 spring.jpa.hibernate.hbm2ddl.auto=update
18 spring.jpa.hibernate.show_sql=false
```

Ici, vous pouvez indiquer la base de données à utiliser / créer dans le champ “spring.datasource.url”.

Ici nous avons choisi une base de données locale sur le port 3306 (port par défaut de MySQL. Changer le par la valeur du port noté précédemment dans la partie “Vérification de la configuration WAMP” si nécessaire) ayant pour nom “oslo”. Le reste de la ligne permet de créer la base de données si elle n’existe pas.

Vous pouvez donc modifier cette ligne pour renommer la base de données à votre souhait et aussi changer l’adresse et le port de la base de données en fonction de vos besoin.

Dans les deux lignes suivantes, vous devez spécifier le nom d’utilisateur et le mot de passe permettant d’accéder à la base de données. Dans notre cas, le nom de compte est “root” et il n’y a pas de mot de passe. A vous de le changer en fonction de vos besoins.

Maintenant que la configuration du projet a été faite, il ne vous reste plus qu'à "build" le projet en cliquant sur le marteau situé en haut dans la barre d'outil:



Si tout c'est bien passé, vous devriez avoir, dans la console en bas de l'IDE un message ressemblant à celui-ci:

```

Output - Build (Project_Oslo) x
--- maven-jar-plugin:3.2.0:jar (default-jar) @ Project_Oslo ---
Building jar: D:\Code\Example\Project_Oslo\target\Project_Oslo.jar
--- spring-boot-maven-plugin:2.3.4.RELEASE:repackage (repackage) @ Project_Oslo ---
Replacing main artifact with repackaged archive
--- maven-install-plugin:2.5.2:install (default-install) @ Project_Oslo ---
Installing D:\Code\Example\Project_Oslo\target\Project_Oslo.jar to C:\Users\Psykroz\AppData\Local\Temp\maven\20210119T1646360100\Project_Oslo.jar
Installing D:\Code\Example\Project_Oslo\pom.xml to C:\Users\Psykroz\AppData\Local\Temp\maven\20210119T1646360100\Project_Oslo.pom.xml

BUILD SUCCESS

Total time: 8.788 s
Finished at: 2021-01-19T16:46:36+01:00
  
```

Si le message "BUILD SUCCESS" est affiché, c'est que vous avez bien réussi à compiler le projet.

Vous pouvez aussi vérifier dans le dossier du projet qu'un nouveau répertoire a bien été créé et s'appelle "target".

Si le fichier "Project_Oslo.jar" est présent, c'est que vous avez réussi à compiler le projet correctement

» Ce PC » HDD (D:) » Code » Example » Project_Oslo » target

	Nom	Modifié le	Type	Taille
	classes	19/01/2021 16:37	Dossier de fichiers	
	generated-sources	19/01/2021 16:46	Dossier de fichiers	
	maven-archiver	19/01/2021 16:34	Dossier de fichiers	
	maven-status	19/01/2021 16:34	Dossier de fichiers	
	test-classes	19/01/2021 16:37	Dossier de fichiers	
	Project_Oslo.jar	19/01/2021 16:46	Executable Jar File	45 752 Ko
	Project_Oslo.jar.original	19/01/2021 16:46	Fichier ORIGINAL	218 Ko

Lancement de l'application

Pour pouvoir maintenant lancer l'application, il faut au préalable bien vérifier que WAMP soit lancé (et son icône en vert comme montré dans la partie "Vérification de la configuration de WAMP").

Une fois le serveur mysql bien lancé, il suffit de se diriger dans le dossier target (précédemment créé par Netbeans dans la partie "Compilation des sources"):

```
D:\Code\Example\Project_Oslo>cd target  
D:\Code\Example\Project_Oslo\target>_
```

Ensuite il suffit d'écrire la ligne de commande suivante:

```
D:\Code\Example\Project_Oslo\target>java -jar Project_Oslo.jar_
```

L'application peut prendre du temps pour se lancer entièrement, patientez quelque instant.

Ensuite, accéder à l'application par l'url suivante: localhost:[Votre PORT]

Dans notre cas, cela serait donc: localhost:7070

Si une page d'accueil vous demandant de vous connecter s'affiche, ce que vous avez bien installé l'application!

Pour une première utilisation, créer un compte puis connecter vous avec les identifiants de base administrateur (nom de compte = admin et mot de passe = admin123) puis mettez votre compte nouvellement créé en rôle administrateur dans le panel administration.

Déconnectez-vous du compte administrateur de base puis reconnectez-vous avec le compte que vous avez créé.

Il ne vous reste plus qu'à supprimer le compte administrateur de base pour éviter de potentiel accès non voulu fait par des personnes malveillantes! (Pour le moment, il faut supprimer directement dans la base de données la personne "Arthas Menethil" et le compte "admin")