

Compte rendue : Outils pour cycle de vie Logiciel

TP01

Selon le guide et grace à eclipse on peut créer un projet maven rapidement.

Normalement Eclipse EE est déjà intégré Maven outil. Dans le fichier pom.xml on va définir les paramètres nécessaires pour le projet, par exemple **modelVersion**, **groupId**, **artifactId** et les **dependencies**.

```
<modelVersion>4.0.0</modelVersion>
<groupId>com.springboot.secondtry</groupId>
<artifactId>spring-boot-try</artifactId>
<version>0.0.1-SNAPSHOT</version>
```

Quand on savegarde les modifications de pom.xml, eclipse detecte automatiques modifications et le recompile.

TP2 Application Spring boot : WS REST

1. Crée un projet maven pour implémenter les fonctionnalités du cas 'gestion de la réservation' basé sur le principe de l'héritage vu dans le cours.

Ceci les configurations de maven:

- Lib de springboot framew ork
- JPA pour la couche persistance
- Driver jdbc pour mysql
- Plugin spring boot

```

<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>1.5.2.RELEASE</version>
</parent>
<properties>
  <java.version>1.8</java.version>
</properties>
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <optional>true</optional>
  </dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>

```

Ceci la structure du projet:

```

▼ [icon] > src/main/java
  ▼ [icon] > com.springboot
    ▼ [icon] > client
      > [icon] RestClient.java
    ▼ [icon] > controller
      > [icon] ReservController.java
    ▼ [icon] > dao
      > [icon] IReservationDAO.java
      > [icon] ReservationDAO.java
    ▼ [icon] > entity
      > [icon] Reservation.java
    ▼ [icon] > service
      > [icon] IReservService.java
      > [icon] ReservService.java
      > [icon] Application.java
  ▼ [icon] > src/main/resources
    [icon] application.properties

```

2. Implémenter une fonctionnalité de votre choix pour l'utilisateur.

La fonctionnalité que je vais implémenter est "consulter les réservations."

Mais dans cette étape là, je fais un test du serveur REST il y a un erreur:

```
# curl -i http://localhost:8080/reserv/src/1
```

```
HTTP/1.1 500
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Tue, 24 Oct 2017 08:12 GMT
Connection: close

{
  "timestamp": 1508881513632,
  "status": 500,
  "error": "Internal Server Error",
  "exception": "org.springframework.dao.InvalidDataAccessApiUsageException",
  "message": "org.hibernate.hql.internal.ast.QuerySyntaxException: reservation is not mapped [FROM reservation as rsv OR
  "path": "/reserv/src"
}

ou {}

rien du tout avec le chemin "/reserv/1" == /reserv/{id}
```

Je ne sais pas pourquoi il n'arrive pas a mapping..j'ai pas resolu ce bug.

a. Ajouter les dépendances nécessaires pour votre client-serveur rest

.

b. Afficher les dépendances transitives.

3. Générer les packages de livraison.

4. Démarrer localement les deux jars (client – serveur Rest)