



# **RAPPORT DE LA PREMIERE VERSION**

« Conception agile d'applications web en Java : biblio »

Etudiant : HA Viet Long

Professeur : OLIVIER Cailloux

**Juliet – 2017**

## 1. Création d'un dépôt sur Git

Je crée un nouveau dépôt sur Git dont le nom est Rattrapage\_JEE afin de commencer mon chemin de continuer mon projet BIBLIO effectué durant l'année.

Voici le lien de mon travail sur Git :

[https://github.com/VietLongHA/Rattrapage\\_JEE](https://github.com/VietLongHA/Rattrapage_JEE)

## 2. L'importation le projet existant « Biblio »

Comme vous m'avez demandé, j'ai importé le projet de mon équipe a mon nouveau dépôt. J'ai occupé ce projet par le lien :

<https://github.com/iliasskahli/biblio>

## 3. Configuration d'Eclipse

Je continue travailler sur Eclipse. En effet du Maven, le projet biblio est automatiquement télécharger et mise à jour tous les package marques dans le fichier pom.xml.

Afin de préparer utiliser CDI, je rajoute la partie du CDI dans le fichier pom.xml.

```
<!-- https://mvnrepository.com/artifact/javax.enterprise/cdi-api -->
<dependency>
<groupId>javax.enterprise</groupId>
<artifactId>cdi-api</artifactId>
<version>2.0</version>
<scope>provided</scope>
</dependency>
```

## 4. Servlets : choix doGet () ou doPost () en fonction de sémantique http

### Various (S8) #11



oliviercailloux opened this issue on Apr 6 · 0 comments



oliviercailloux commented on Apr 6



- pom.xml incorrect. Voir mes échantillons et best practices
- persistence.xml : utiliser propriété standard JPA pour création des tables (cf. diapos cours) ; ne pas lister les classes (auto-detect) ; utiliser connexion à BD par défaut du serveur d'application
- utiliser transactions JTA plutôt que resource-local, cf cours JPA
- coder en anglais
- servlets : get ou post en fonction de sémantique HTTP, cf doc cours Servlets.
- utiliser CDI, etc.
- activer warnings stricts eclipse, cf Best practices fichier epf

Compter sur vos derniers commentaires sur notre projet BIBLIO effectué durant l'année. Je vérifie tous les fichier servlet.

On utilise que doGet() ou que doPost() pour faire une requête http.

Donc je enlever quelque doGet() et doPost() qu'on n'a pas besoin.

## **5. Utiliser CDI (Context Dependency Injection)**

Après installer bien CDI par Maven, je mise en place CDI par préciser sa portée contextuelle (contextualscope) : @RequestScoped. Dans la package « modele » et au début des classes.

Grace à CDI, on peut éviter de la coder en dur chaque fois on crée un objet.

@Inject sur champ ou méthode ou constructeur.