

# Rapport Projet Services Web Master 2 Informatique

---

Mateusz SZCZECH, Mael HANQUEZ, Lilian BERRUET

---

## 1. Les choix de conception

Le projet est divisé en deux parties:

- RMI avec le service EiffelBikeCorpService et le client EiffelBikeCorpClient se trouvant dans le répertoire EiffelBikeCorpProject
  - EiffelBikeCorpService qui gere la location des vélos
  - EiffelBikeCorpClient qui communique avec EiffelBikeCorpService pour pouvoir louer des vélos aux utilisateurs
- Web service qui regroupe deux services et un client:
  - GustaveBikeService qui gere l'achat des vélos
  - BankService qui gere les fonds des utilisateurs
  - GustaveBikeClient qui permet de communiquer avec GustaveBikeService pour acheter des vélos

### a) La partie RMI

La communication entre EiffelBikeCorpClient et EiffelBikeCorpService se fait grâce aux interfaces dans le package "common".

EiffelBikeCorpService implémente deux interfaces:

- EiffelBikeCorpInterface qui permet de louer des vélos, de les rendre et de lister les vélos à louer, c'est cette interface qu'utilise EiffelBikeCorpClient
- EiffelBikeCorpAccessInterface qui permet d'avoir accès à un vélo spécifique et de retirer des vélos. En effet on a besoin de ces fonctionnalités pour que GustaveBikeService puisse retirer les vélos que l'utilisateur veut acheter sans pour autant de donner accès à ces méthodes à EiffelBikeCorpClient

### b) La partie Web Service

GustaveBikeService communique directement avec EiffelBikeCorpService grâce aux interfaces du package "common" (qui a été copié de la partie RMI) mais aussi avec BankService pour savoir si le client peut ou non acheter un vélo et avec FxtopServices qui est un service en ligne permettant d'échanger la monnaie en temps réel.

GustaveBikeClient communique avec GustaveBikeService pour pouvoir acheter des vélos dans la monnaie souhaitée, rajouter des fonds, constituer son panier, afficher les vélos disponibles pour l'achat

## 2. les difficultés rencontrées

### a) les difficultés de la partie RMI

On n'a pas rencontré de difficultés particulières pour la partie RMI.

## a) les difficultés de la partie Web Service

On a rencontrées plusieurs difficultés pour la partie Web Service:

- La compréhension de comment fonctionne eclipse pour crée un web service et un web service client
- Quand on essayais de crée un web service client, eclipse generais des classes dans un package. Le code generé possedais des conflits d'import. Le module se trouvais a la fois dans les jars et dans java.util.xml. On a du changer la version de java en JEE 1.5 ce qui a réglé le souci
- Pour la creation des methodes dans GustaveBikeService, eclipse ne nous laissais pas de generer le fichier .wsdl sans donner l'erreur precise. Apres 4h d'essais on a compris qu'on a pas le droit d'utilise de lambdas, de return seulement des tableaux et pas de list et qu'il fallait remplir le tableau a la main et pas avec la methode list.toArray()
- La serialisation de la classe GustaveBike ne fonctionnais pas, grace a votre aide on a compris que les noms de nos packages commencaient par une majuscule ce qui causais le probleme
- Le maintien de la session avec GustaveBikeService et BankService ne fonctionnais pas, il fallait rajouter a la main le "ns1:parameter" avec le "name: scope" et la "value: session" dans le server-config.wsdd
- Il fallait faire attention de ne pas avoir deja un serveur lancé sur le port 8080, une personne d'entre nous avec spring lancé sur ce port qui empechais le lancement du serveur avec les services