

Steeve Sivanantham
steeve.sivanantham@gmail.com
Étudiant en M1 Informatique
Groupe 2

Damien Chesneau
ches.damien@gmail.com
Étudiant en M1 Informatique
Groupe 2 - Apprenti

Rapport de design du projet de java avancée.

Le programme contient 6 paquetages :

- exmanager : Est en charge de la récupération des exercices et de leur mise en forme.
- jinterpret : Est en charge de l'interprétation et le test de code java.
- server : En charge du traitement des requêtes et des clients celui ci contient deux autre paquetages :
 - manageclients : Celui ci est en charge de la gestion des instances des clients pour leur interpréteur respectif.
 - transacparser : permet de formater les requêtes réponses entre les clients et le serveur.

Les plus 3 premiers ont été architecturé afin de caché l'implémentation à l'extérieur. Cette manière de faire permet de pourvoir changé l'implémentation comme on le veut. Ainsi cela nous a permis de pouvoir travailler avec des implémentation factice pendant que le travail de notre collègue n'était pas encore terminé. Tout en gardant intact le code appelant vu que celui ci ne voit que l'interface. Le code de l'implémentation à doc une visibilité paquetage.

Détail de chaque classes :

exmanager :

- ExerciceImpl : Implementation du Service de récupération des exercices et le gère le fait qu'il soit en mémoire ou non.
- Exercices : Classe permettant de récupérer l'implémentation de ExerciceService et de lancer le watcher sur le dossier.
- ExerciceService : Interface contenant tout les méthodes nécessaires au client.
- Parser : Cette classe à en charge le passage des fichiers markdown avec les balises junit.
- Watcher cette classe est en charge de la détection des fichiers qui change.

jinterpret :

- InterpretedLine : représentation d'une ligne de code interprété exécuté.
- Interpreter : Interface contenant toutes les méthodes pour interpréter un code java et avoir accès à leur résultats.
- Interpreters : classe permettant de récupérer le jshell interpréter.
- JShellInterpreter : implémentation de l'interface Interpreter elle permet de faire les accès au jshell.
- JunitTestResult : Enum représentant l'état d'un test.

server.manageclients :

- Client : Class Proxy sur Interpreter permettant d'accéder a toutes les méthodes de l'interpréteur avec l'instance unique d'un client.
- ClientsManager : Classe gérant les instances de clients.

server.transacparser :

- TransactionParser : Classe qui encode des messages en json et qui les décode. elle possède deux classes interne qui sont mutable afin de pouvoir construire cette classe.
- TransactionPattern : Enum definisant les types de messages.

server :

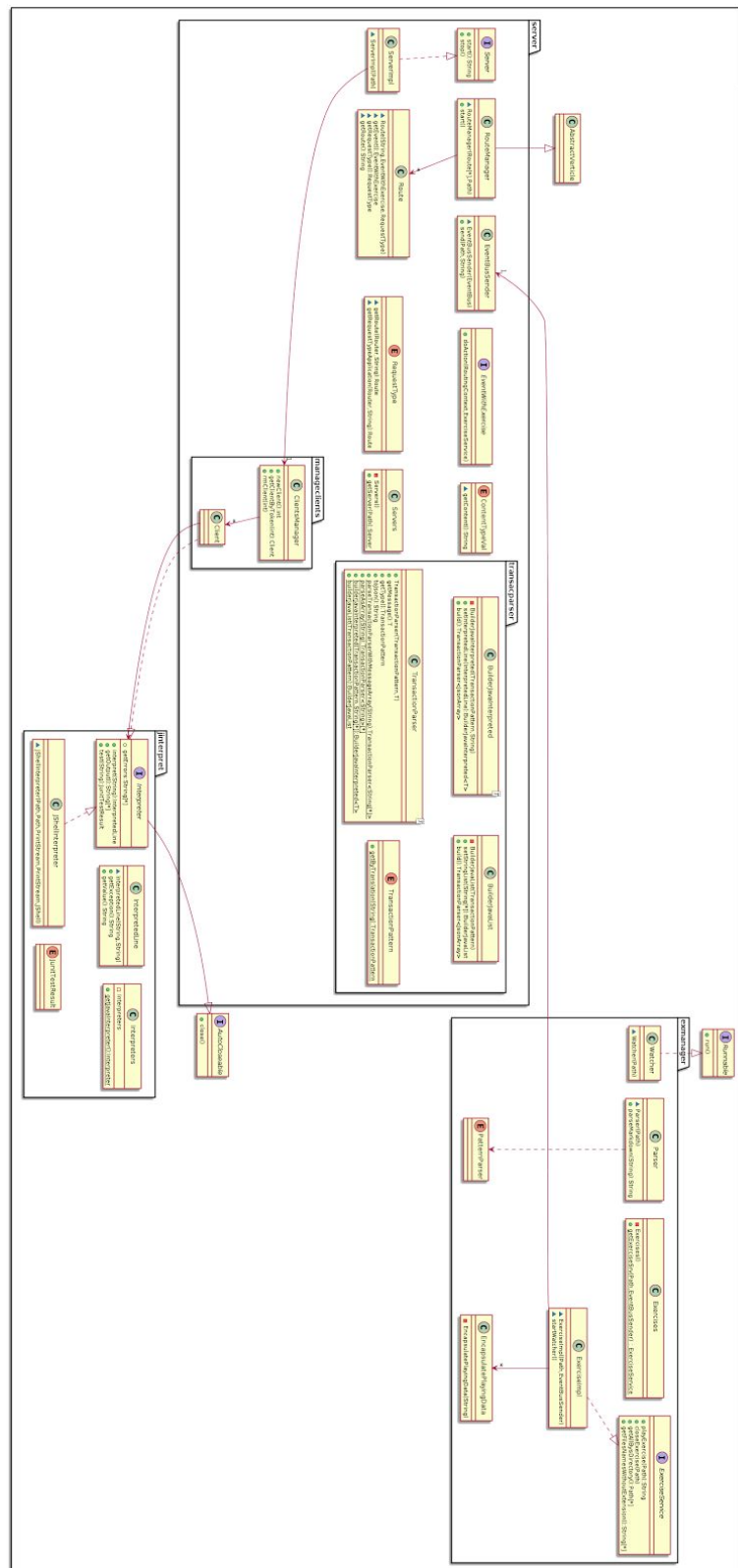
- ContentTypeVal : Enum contenant le content type des echanges entre le client et le serveur.
- EventBusSender : Cette classe permet d'envoyer un message aux clients. Elle ne peut être crée que dans le serveur et c'est le watcher qui l'utilise.
- RequestType : Enum represantant soit une méthode post ou get. Cette enum permet de voir de la même facon les deux requêtes.
- Route : Permet d'enregistrer une route spécifier par un utilisateur. Elle contient l'action a effectuer pour chaque route et son type.
- RouteManager : Cette classe permet de configurer le serveur avec les routes qui sont envoyé. Avant d'invoquer l'évènement de Route, elle sécurise l'accès.
- Server : Interface avec les méthodes permettant de contrôler le serveur.
- ServerImpl : Classe dans laquelle nous définissons les différentes routes.
- Servers : Classe permettant de récupérer l'instance de ServerImpl.

fr.upem.ijavabook :

- Main : Lancement du programme avec le répertoire courant pour les exercices au format markdown.

Le fait d'avoir crée une classe route et ensuite d'ajouter une liste de route au serveur nous a permis d'avoir une plus grande flexibilité dans notre développement. Ainsi au besoin on n'avais plus besoin de toucher au code de RouteManager.

Diagramme de classe du projet :



Disponible en grand format à cette adresse :

https://drive.google.com/file/d/0B_mzeceFSmfdVhYemhNa2Rja1U/view?usp=sharing