Compte rendu IOS.TP3 :

Thibault Coilliaux

Première séance TP :
L'idée de créer de un bundle pour chaque mode souhaité (ici un pour ServletMode et un pour FileMode) m'est venue assez rapidement. Il n'y a pas mille façons d'implémenter la solution avec des bundles.
En revanche je n'ai pas avancé plus loin dans le développement lors de cette séance. Je ne voyais pas comment j'allais pouvoir bundle-iser le serveur Comanche de base, malgré de nombreuses réflexions.

Travail à domicile :
J'ai exploré des pistes, et notamment la bonne qui est d'abord de créer un bundle interface (request_service). Cependant je n'arrivais toujours pas à voir la suite, comment j'allais pouvoir faire changer le mode du serveur (File -> Servlet) juste en lançant un autre bundle. Chaque fois que j'entreprends quelquechose, je me retrouve à devoir modifier le code source de base. Je ne suis donc pas allé plus loin et je n'ai pas sauvegardé mon travail. Cela m'a même fait douter de ma solution d'amélioration du serveur comanche de base.

Deuxième séance TP :
Dès le début de séance je me renseigne si on est autorisé à modifier davantage le code de base du serveur Comanche. Il s'avère que oui, et j'ai donc repris mes pistes du travail à la maison.
J'ai pu ajouter une méthode setRequestAnalyzer dans la classe RequestAnalyzer et à partir de ce moment cela a été simple. J'ai repris le principe du TP2, en remarquant que le serveur jouait le rôle de Client du TP2.
La solution fonctionne du premier coup.
Suite à vos précisions concernant les comportements du serveur en fonction du démarrage et de l'arrêt des bundles File et Servlet, j'ai fait des petits tests dans les méthodes addingService et removedService pour bien comprendre ce qui se passe derrière le démarrage et l'arrêt des ces bundles.
Et pour garder en mémoire les types de RequestHandler lancés au fil du temps, j'ai décidé de les mettre dans une liste, et d'ajouter/enlever les RequestHandler en fonction des évènements.

Travail à domicile :
Je me suis penché sur le problème du socket ("adresse déjà utilisée"). Après des petits tests pour comprendre comment s'enchaîne le code de la classe WebServer, j'ai très vite eu l'idée d'ajouter une méthode stop() qu'on pourrait utiliser dans la classe Activator pour la fermeture du Bundle, et donc de déclarer la variable 'ss' comme une variable de la classe.
La solution fonctionnait mais toujours avec une trace d'erreur dans les logs. Et pour être honnête ayant entendu parler de "SocketException" pendant la deuxième séance, ça n'a pas été trop long avant de penser à modifier les clauses catch.