Applications réflexives

TD 4 La plateforme BretteRossit initiative (BRi)

Introduction

Le but du programme que vous allez réaliser est d'offrir une plateforme dynamique de services qui peuvent être invoqués de manière distante par un client. La mise en œuvre du système se fait en deux temps pour un service donné : chargement de la classe de service via un serveur ftp, lancement d'un service à la demande d'un client distant.

L'installation et le test simple du serveur ftp est décrit dans le document ***ftp apache***. Cette prise en main préalable ne devrait pas vous prendre plus de 10 mns. Pour ce tp, vous lancerez le serveur ftp sur un poste, le serveur BRi sur un second poste et un client sur un 3ème. A défaut, tout sur le même poste...

Chargement dynamique d'un service

Une classe de service (implémentant bri.Service) développée par un développeur indépendant est intégrée à votre logiciel, pour ce tp, « manuellement » : le fichier compilé doit être placé sur votre serveur ftp et le nom de la classe saisie au clavier dans la console du serveur BRi. A partir de là, un URLClassLoader charge cette classe et elle est déclarée à `ServiceRegistry`.

Lancement d'un service

Le lancement du service est réalisé à la demande d'un client qui se connecte au port PORT_SERVICE au ServeurBRi du type de ceux vus en S3-AppServ, qui instancie et lance un ServiceBRi. Celui-ci va dialoguer via socket avec le client pour lui demander quel service il souhaite avoir parmi ceux disponibles. Le service choisi est instancié (avec la socket) et le ServiceBRi appelle le run() de ce service choisi. Instanciation et appel du run se font par introspection de la classe de service.

La norme BRi

Le développeur de services BRi doit respecter la norme qui lui permettra de soumettre sa classe Java à notre plateforme. Une classe de service BRi doit :

implémenter l'interface BRi.Service
ne pas être abstract
être publique
avoir un constructeur public (Socket) sans exception
avoir un attribut Socket private final
avoir une méthode public static String toStringue() sans exception

L'introspection de java.lang.reflect sera utilisée pour contrôler ces éléments. Bien entendu, si un élément de cette norme n'est pas respecté, le message de refus devra dire clairement pourquoi.

Un exemple : le service… d'inversion d'un texte !

Dans le package ***examples*** se trouve la classe ServiceInversion développée suivant la norme de cette plateforme. Cette classe est totalement opérationnelle et ne nécessite aucune modification, sinon peut-être pour mieux gérer la fin de connexion. Vous pouvez la placer (après compilation) sur le serveur ftp dans le répertoire home pour les tests.

Le code du client

Client.jar contient l'application cliente (qui se connecte au port PORT_SERVICE au ServeurBRi. Cette application est opérationnelle modulo la même remarque que ServiceInversion pour la fin de connexion.