

Projet RMI – Blaguemule – Part 1

L'objectif du projet est de proposer un blaguemule™, à savoir un système de partage de blagues permettant à chacun de se réjouir des blagues cocasses partagées par les autres utilisateurs et de partager à son tour ces grands instants de bonheur.

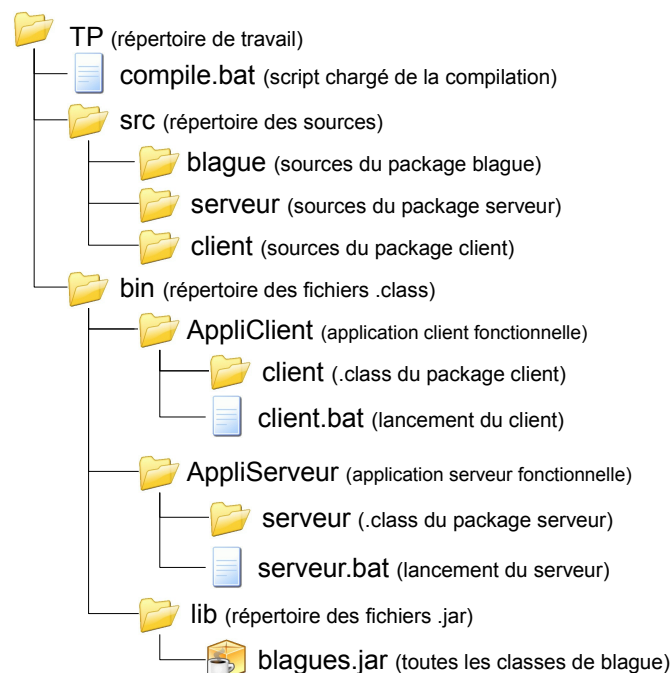
Le projet utilise RMI et fonctionnera dans cette partie de la manière suivante : on vous demande de créer un **BlagueProviderServeur** permettant de partager des blagues et d'écrire un **BlagueProviderClient** capable de récupérer les blagues du serveur.

1 Structuration en package et makefile

Un des problèmes rencontrés lors de l'utilisation de RMI réside dans le fait qu'il y a plusieurs classes à compiler pour générer le client et le serveur. Ainsi, il arrive de temps en temps que les classes communes entre le client et le serveur (par exemple les blagues à échanger) ne correspondent pas à la même version compilée du code (ex seul le serveur a été recompilé).

Pour éviter les désagréments occasionnés – problème de sérialisation – nous vous proposons

1. de séparer les sources des fichiers binaires résultants
2. de structurer vos sources en plusieurs package (serveur, client et blague)
3. de générer les deux applications distinctes dans deux répertoires différents: le serveur dans un répertoire bin/AppliServeur et le client dans un répertoire bin/AppliClient)
4. d'écrire un makefile (ou un script .bat sous windows) capable de compiler et d'organiser tous les fichiers pour être sur que les applications possèdent après chaque compilation les bonnes versions des fichiers (complément java – TD3).



2 Classe Blague

La classe **Blague** permet de représenter une blague (les connaisseurs remarquerons qu'il s'agit techniquement d'une devinette plus que d'une blague). La classe **Blague** contient trois attributs privés

- **String nom** : le nom de la devinette. Par exemple « Série télé »
- **String question** : la question définissant la devinette. Par exemple : « Quelle est la série télé préférée des informaticiens ? »
- **String reponse** : la réponse cocasse à la question. Par exemple « RMI sans famille »

Question a :

Ecrire la classe **Blague**

3 Fonctionnement du BlagueProviderServeur

La classe **BlagueProviderServeur** a pour objectif de fournir des blagues. Les blagues sont stockées sous la forme d'une **HashMap** qui à chaque nom de blague (type **String**) associe la blague correspondante (type **Blague**)

Voici les méthodes distantes à implémenter

- **String[] getAllNames()** qui retourne un tableau contenant tous les noms des blagues disponibles sur le serveur
- **Blague getBlague(String nom)** qui retourne une blague à partir du nom donné en paramètre. Si la blague n'existe pas, la méthode génère une **BlagueAbsenteException**

Question b :

En déduire l'interface **BlagueProviderInterface** à implémenter.

Question c :

Ecrire la classe **BlagueProviderServeur** qui implémente cette interface. Cette classe doit être déclarée dans un package **Serveur**.

Question d :

Quel est le codebase à passer pour enregistrer correctement la référence distante du **BlagueProviderServeur** et quelles sont les contraintes à avoir sur les classes ? Mettez l'ensemble du codebase dans un package spécifique **blague** et créez un fichier jar.

Question e :

Ecrire une méthode main qui permet au **BlagueProviderServeur** d'enregistrer sa référence distante sous le nom « blagueProvider »

4 Ecriture de la classe BlagueProviderClient

La classe **BlagueProviderClient** consiste simplement à récupérer la référence distante du **BlagueProviderServeur** et à récupérer et afficher une blague enregistrée dans le **BlagueProviderServeur**.

Question f :

Écrire la classe **BlagueProviderClient**. Vous penserez à gérer correctement les exceptions (ex : **BlagueAbsenteException**). Cette classe doit être écrite dans un package **Client**