

ALG2ir - Examen

Liste de todos

Examen d'ateliers logiciels, section industrielle et réseaux

(Mercredi 2 septembre 2009)

Remarques :

- ↪ vous pouvez utiliser **toutes les notes** que vous voulez ainsi que vos travaux,
- ↪ vous disposez de 4 heures,
- ↪ vous travaillez sur machine et remettez votre programme au terme du temps imparti,
- ↪ vous êtes responsable de la sauvegarde régulière de votre travail,
- ↪ soyez attentif au **style**, à la **clarté** et à la **robustesse** du code

20

ÉNONCÉ. Nous allons écrire une application permettant de gérer une liste de tâches (*todos list*). Un serveur possède une liste de todos par utilisateur. Un client, se connectant au serveur, peut obtenir sa liste de todo, en ajouter et en supprimer. Le serveur ne sauvegarde pas cette liste lorsqu'il s'arrête, il la conserve simplement en mémoire. Les todo seront représentés de manière très simple ; ils se composent d'un identifiant et d'une description ¹.

Le client graphique a l'allure présentée sur la figure 1. Le serveur est simplement textuel.

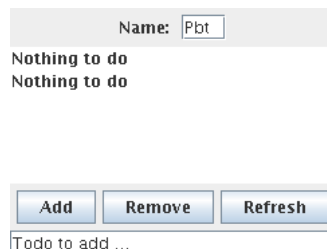


FIG. 1 – Capture d'écran de l'application, partie cliente

Nous vous fournissons des parties de code que nous vous demandons de compléter. Vous trouverez dans l'archive jointe, un projet « client » et un projet « serveur » que vous complèterez. Vous trouverez également une bibliothèque (todoLibrary.jar) – ainsi que sa javadoc – que vous incluez à votre projet. Le tout devrait être *selfcompréhensible*.

¹Pas d'échéance, de priorité, de catégorie, ...

La bibliothèque

Comme vous le voyez dans la javadoc, elle se compose d'une classe `pbt.todo.dto.TODO` et d'une classe `pbt.todo.protocol.Protocol`. La classe `TODO` n'appelle pas de commentaires et sera utilisée dans le client et le serveur.

Protocole de communication

Le protocole de communication entre le client et le serveur se compose

- ↪ d'une requête de demande de liste de todos pour un utilisateur,
- ↪ d'une requête d'ajout d'un todo,
- ↪ d'une requête de suppression d'un todo et
- ↪ d'une demande de déconnexion.

Les requêtes sont de la forme

```
Protocol.CMD_<name> [<arg>Protocol.FS<arg>...]Protocol.EOL
```

En réponse à ces requêtes le serveur acquittera (`Protocol.ACK`) ou pas (`Protocol.NACK`) la demande.

Vous trouverez également dans le protocole le **port** utilisé par défaut.

Requête de demande de liste

Le client demande la liste des todos pour un utilisateur donné, il reçoit cette liste. En cas d'erreur le serveur précise le type d'erreur rencontré.

```
Requête: CMD_GET<user>EOL
Réponse: ACK<todo1-id>FS<todo1-description>[FS<todo2-id>FS ...]FSEOL ou
         NACK<message d'erreur>EOL
```

Requête d'ajout

Permet l'ajout d'un todo.

```
Requête: CMD_ADD<user>FS<todo-description>EOL
Réponse: ACK ou
         NACK<message d'erreur>EOL
```

Requête de suppression

Permet la suppression d'un todo sur base de son identifiant.

```
Requête: CMD_DEL<user>FS<todo-id>EOL
Réponse: ACK ou
         NACK<message d'erreur>EOL
```