

TD3

mardi 5 novembre 2024 13:22

Q1. Rôle des stubs et des skeletons dans le RPC :

Le stub client remplace la vraie procédure distante / objet distant
c'est un représentant de l'objet distant localement

Le skeleton gère les appels, le stockage, ... coté serveur

L'interface `ObjetDistant` représente le contrat entre le serveur et le client, il se situe sur les deux sites, c'est une connaissance qui est partagée entre tous les partenaires qui veulent utiliser ce service

Tous les éléments de l'interface sont communs aussi, ils doivent donc être connus par le client et le serveur

L'implémentation de l'interface est nommé par convention `NomInterfaceImpl`

1. Le contrat
2. Les stubs et les skeletons
3. Classe d'objet qui implémente l'interface (cette classe se situe côté serveur uniquement)

Le **servant** est l'objet qui implémente le service, ici c'est un objet distant

Q2. Différence entre passage par valeur et passage par référence :

Une valeur est un entier, un réel, une chaîne de caractère, ...

Une référence est un pointeur, une adresse, qui contient une valeur

Quand une référence est envoyé on ne peut pas travailler avec, on va donc d'abord récupérer la valeur référencée par la référence et c'est celle-ci qui va être envoyé

On clone donc l'objet, on le copie et on envoie cette copie via le réseau

Q3. Modifications à apporter aux classes `ObjetParam1` et `ObjetParam2` pour que le passage par valeur soit possible :

Les objets doivent être sérialisables car on envoie l'objet entier

```
public class ObjetParam1 implements Serializable {}  
public class ObjetParam2 implements Serializable {}
```

Une socket simple permet de récupérer les données en entrée et en sortie

D