

PR6 – Programmation réseaux

TP n° 11 : Serialization

Le but de ce TP est d'explorer plusieurs solutions (plus ou moins bonnes) pour transférer des données complexes. Pour l'ensemble de cette séance, vous disposez de deux classes

- `MyImage` qui représente une image ;
- `Filtre` qui permet d'appliquer certains filtres sur une image donnée comme une instance de la classe `MyImage`.

Vous pouvez consulter la documentation de ces classes sur la page du cours.

La classe `MyImage` n'est pas complète, ce sera à vous de la compléter au fur et à mesure des exercices.

Dans chacun des trois exercices ci-dessous, il s'agit d'écrire un serveur qui reçoit une image sous une certaine forme, la transforme en niveaux de gris et la renvoie sous une forme similaire, et le client correspondant qui envoie au serveur l'image, la récupère modifiée par le serveur et la sauvegarde dans un fichier. Le codage de ces échanges varie selon l'exercice et il faudra en particulier réfléchir au protocole d'échange.

Pour tester vos programmes, récupérez une image quelconque sur internet au format jpeg.

Exercice 1 : Premier échange : fichier régulier

La classe `MyImage` implémentant l'interface `Serializable`, on peut en serialiser ses instances. En particulier, on peut utiliser un encodeur XML. Utilisez `XMLEncoder/XMLDecoder` pour transmettre un fichier texte entre votre client et votre serveur, en complétant les méthodes `serializeXML` et `deserializeXML` de la classe `MyImage`. Attention : `XMLEncoder` est très lent (ça vous convaincra sûrement que c'est une assez mauvaise idée de l'utiliser).

Exercice 2 : Deuxième échange : fichier binaire

Dans cet exercice : le client envoie un fichier binaire au format jpeg au serveur et celui-ci lui renvoie un fichier au même format.

Exercice 3 : Troisième échange : objet

Utilisez un `ObjectInputStream` et un `ObjectOutputStream` pour compléter les méthodes `serializeOOS` et `deserializeOOS` de la classe `MyImage` et ainsi faire l'échange de données en serialisant des instances de `MyImage`.