

TP 2 - Design Patterns et Conception

Pour rendre le TP:

- Email: andre.martel@niji.fr / andre.martel09@gmail.com
- Objet: [COO-TP2] Nom Prénom
- Contenu: Une URL vers un dépôt Git ou une archive du code (**sans les node_modules par pitié !**)

Si des réponses du TP doivent être écrites ou si vous avez des notes personnelles à ajouter, alimentez le fichier *README.md*.

Le projet contient un serveur Web, écrit en NodeJS (aidé du framework ExpressJS) avec du TypeScript. Le serveur ne renvoie que de la donnée brute, sans UI, nous sommes sur une API qui gère des commandes. Le point d'entrée de l'application est le fichier à la racine *www.ts*.

#1

Le serveur a été écrit pour l'instant de manière très déclarative sans aucune approche objet. En vous basant sur le diagramme de classe fourni à la racine du projet, réécrivez le code des fichiers *www.ts* et *app.ts* en utilisant des objets.

#2

Réarranger le code du projet en utilisant un pattern d'architecture adapté à notre besoin et justifiez votre choix. (Les routes restent au même endroit dans le dossier *routes*)

#3

Profitons du TypeScript pour être plus rigoureux dans la manipulation de nos commandes. Créez un type *Order* dans un fichier *types.ts* à la racine du projet.

Ce type est similaire à une interface et peut donc respecter un des principes du SOLID, lequel ?

#4

Lorsque l'on récupère la totalité des commandes, on souhaite que chaque information présente dans la partie contact soit anonyme (remplacée par une donnée fake, ce que vous voulez). Pour obtenir ce résultat, un Design Pattern nous sera utile, lequel ?

#5

Faites en sorte que nous n'ayons accès, tout au long de l'exécution de l'application, qu'à une seule et même instance de notre stockage. Nommez le Design Pattern ainsi mis en oeuvre.

#6

Réalisez le diagramme de classe de l'architecture obtenu au fil des questions.