

TP Web Serveur

Important : Une version en ligne de ce TP (potentiellement + à jour) est disponible sur <https://unicorn.artheriom.fr/>, avec également des fichiers d'aide à disposition. N'hésitez pas à consulter cette page.

Partie 1

Cette partie est à effectuer en classe, durant la première session de TP. Cela permet de cadrer le sujet et de mettre tout le monde dans le bain.

Énoncé :

Mme. Myrtille et M. Framboisier sont professeurs à l'école maternelle François Mitterrand. Dans leur classe, ils ont mis en place un système de bons/mauvais points, à base de gommette. Par exemple, une gommette verte équivaut à une bonne action effectuée par l'élève à l'attention d'un de ses collègues de classe. Une gommette blanche récompense un très fort investissement au service de la communauté, une gommette rouge quand elle sanctionnera un élève pour sa mauvaise conduite.

Actuellement, ils utilisent des gommettes papier, qu'ils collent dans le carnet de liaison des élèves. Cependant, cette approche présente plusieurs problèmes :

- Les élèves peuvent perdre leur carnet ;
- Les gommettes peuvent s'abîmer, se décoller, se déchirer etc ;
- Il est impossible pour M. Framboisier de savoir pourquoi Mme. Myrtille a mis une gommette à un élève, et vice-versa ;
- Et, par mesure de restrictions budgétaires, il a été annoncé à Mme. Myrtille que ses gommettes ne pouvaient plus être prises en charge par l'école.

Vous voici donc confié la lourde mission de créer un dispositif de gommette pour la classe de Mme. Myrtille & M. Framboisier ! Une équipe est déjà en charge de faire un frontal, vous vous occuperez donc de la partie serveur. Le reste de votre entreprise fonctionnant en Java, vous prenez la décision de faire votre projet en suivant le même socle technique que vos collègues.

Travail en groupe :

Attention, même si le travail est commun, prenez des notes !

- Quelles sont les fonctionnalités qui vous semblent importantes ?
- Quelles vont être les principales classes de votre application ?
- Quelles vont être les tables de votre base de données ? Vous pouvez produire un diagramme UML pour mettre au clair les jointures entre les tables, mais c'est facultatif.
- Quelles solutions technologiques envisagez vous ?

Partie 2

Prérequis

- Java JDK 8 ou + installé

Énoncé :

Nous allons concevoir notre application en respectant les principes vus en cours :

- Serveur sans-état
- Les URIs ne changent jamais et sont prédictibles
- Respect des verbes (dans la mesure du possible) et codes HTTP
- etc

Dans ce TP, nous utiliserons Java Spark pour gérer notre serveur Web et Thymeleaf pour templatiser nos pages HTML. Pour stocker les données, nous utiliserons une base de données H2, et nous accèderons à la base avec le driver JDBC natif de Java.

Première partie : Définir vos URIs

Dans le TP précédent, nous avons défini nos entités, nos tables et les relations entre ces dernières. La dernière étape avant de commencer à coder est de définir la structure de notre site : comment les pages vont-elles être agencées, et quels seront leurs liens d'accès ?

Par exemple, pour un outil de gestion de liste (= ce n'est pas votre TP !) :

- `GET /login` va afficher la mire de connexion et `POST /login` permet de se connecter
- `GET /lists` va afficher mes listes, `POST /lists` va me permettre de créer une liste
- `GET /lists/mon-id-de-liste` va afficher le contenu de ma liste (= mes items), `POST /lists/mon-id-de-liste` va me permettre d'ajouter un élément, `GET /lists/mon-id-de-liste/mon-id-d-item/delete` va me permettre de le supprimer
- etc

Pour vous aider à établir votre liste d'URIs, voici la liste des fonctionnalités attendues :

- M. Framboisier et Mme. Myrtille doivent pouvoir se connecter (on ne peut ajouter, modifier ou supprimer des gommettes que lorsqu'on est connecté) ;
- Les professeurs peuvent lister, ajouter, supprimer ou modifier un élève ;
- Les professeurs peuvent lister, ajouter, supprimer ou modifier une gomme, définie par une couleur et une description ;
- Les professeurs peuvent ajouter ou supprimer une gomme à un élève, avec une date et un motif ;
- Les professeurs peuvent lister les gommes attribuées à un élève, et voir qui a donné chaque gomme ;
- Tout le monde peut voir la liste des gommes configurées, et la liste des gommes de chaque élève.
- Pensez à intégrer votre liste d'URIs à côté de votre code source.

Deuxième partie : Coder !

Maintenant que vous avez défini toutes vos URIs, vos tables de données, vos classes et les liaisons entre elles, vous pouvez commencer à coder ! Comme précédemment dit, le TP est conçu pour être effectué en Java.

Attention ! Dans l'écosystème Java, il existe de nombreux frameworks capables d'abstraire la majorité du travail que vous allez effectuer (le plus connu étant Springboot). L'idée de ce TP est de vous faire comprendre le fonctionnement sous-jacent de ce genre de Framework (car la magie n'existe pas, même en informatique). Pour celles et ceux qui poursuivront en M1, vous aurez une série de TP consacré à ce genre de technologies.

Si vous êtes déjà à l'aise en Java et que vous souhaitez faire le TP dans un autre langage (mais en respectant l'interdiction de Framework), demandez à votre chargé de TP.

Bon courage !

Disclaimer : Les noms présentés dans ce TP sont fictif. Toute ressemblance avec la vraie vie serait totalement fortuite. Merci de ne pas contacter des M. Framboisier ou des Mme. Myrtille qui travailleraient réellement dans une école François Mitterand.