> EPSI Lille - B3 - 2016-2017

Étude de cas JEE

Date de rendu : **28**/**05**/2017 23h55. Consignes de rendu détaillées plus bas.

Sujet : développement d'une API Rest avec les technologies Jee suivant une spécification Swagger.

Il s'agit de développer une application où les utilisateurs pourront créer et voter pour des podiums. Par exemple, un utilisateur peut créer un podium avec les 3 éléments suivants : "fraise", "chocolat" et "vanille"; les autres utilisateurs pourront alors voter pour donner leur ordre de préférence.

Il s'agit pour votre **équipe de 3 personnes** de développer le back-end de cette application, et de fournir une api REST qui sera consommée par le front-end. Cette api devra respecter strictement le contrat de service Swagger qui vous est fourni. Cette spécification décrit les différentes routes que vous devrez mettre en place, ainsi que les entrées et sorties attendues : il est important de la respecter strictement. L'application devra écouter les connexions `HTTP` sur le port `8080`.

La stack technique suivante est conseillée, mais non obligatoire (vous pouvez notamment choisir un autre serveur d'application ou d'utiliser Spring) :

- Serveur : Jetty
- ORM : Hibernate
- API Rest : Jersey
- BDD : MySQL

En cas de doute sur la possibilité d'utiliser une technologie dans le projet, demandez confirmation aux intervenants.

Pour faire ce projet, vous seront fournis :

- La spécification Swagger au format yaml; vous pouvez utiliser swagger editor pour en avoir une vue plus human-friendly.
- Des données de tests au format Json afin que vous puissiez tester votre API. L'utilisation de Postman est conseillée.
- Une version modifiée du programme de test qui sera utilisé pour la notation automatique vous sera donnée au cours du projet.

Les livrables attendus sont :

- Le code source de l'application, y compris les scripts SQL
- Votre application compilée sous forme de jar/war;
- Une documentation technique expliquant notamment comment installer et lancer votre projet
- un bref rapport (1 ou 2 pages), explicitant les problèmes que vous avez rencontrés, les choix que vous avez dû effectuer et la répartition des tâches au sein de votre groupe.

Ces livrables seront rendus dans une archive zip nommée `EDC-[NOM1]-[NOM2]-[NOM3].zip`, et envoyée à l'addresse suivante : **remi.even@zenika.com**.

Pour ce projet, **les groupes seront les mêmes** que pour le projet de développement d'une plateforme de services avec QRCode.

Une partie de la notation se fera suivant des tests automatiques de votre API, qui permettront de déterminer si votre application correspond bien au swagger. Le reste de la notation se fera à la main et portera notamment sur la qualité du code et le respect des conventions.

Vous pourrez avoir des points bonus si vous réalisez des tâches supplémentaires, comme par exemple :

- Valider les objets envoyées par le client
- Utiliser docker pour déployer l'application
- Tests unitaires et d'intégration
- Mise en cache de certaines données

Il est suggéré de répartir le travail de la façon suivante :

- une personne pour la couche présentation
- une personne pour la couche business et le déploiement de l'application
- une personne pour la couche données

Il est conseillé de bien réfléchir à la modélisation des données dans la base: servez-vous du fait que les spécifications du sujet soient fixées, et ne cherchez pas la généricité/modularité à tout prix.

Comme pour le reste du module, vous pouvez transmettre vos questions à l'adresse mail suivante : `remi.even@zenika.com`