

---

## DEVOIR N°1 Module JEE : Développement Microservices avec Spring Cloud

- Team de développement : par binôme maximum

### Énoncé :

#### Etude de cas (1) :

Ajouter un « microservice-commandes » qui permet de réaliser les opérations CRUD sur une « COMMANDE » avec 0 ligne SQL :

- a. La version (1) de la table « COMMANDE » est composée » des colonnes suivantes [id, description, quantité, date, montant]
- b. La configuration du « microservice-commandes » doit être gérée au niveau Spring Cloud et github
- c. La configuration du « microservice-commandes » contient une propriété personnalisée « mes-config-ms.commandes-last » qui permet d’afficher les dernières commandes reçues. Dans notre cas : « mes-config-ms.commandes-last = 10 » permet d’afficher les commandes reçues les 10 derniers jours.  
  
En se basant sur le service Actuator de spring, modifier cette propriété à 20 et réaliser un chargement à chaud pour que le « microservice-commandes » affiche les commandes reçues les 20 derniers jours
- d. En se basant sur le service Actuator de spring, Implémenter la supervision la bonne santé du « microservice-commandes » : le statut à afficher « UP »

- e. Personnaliser la supervision de la bonne santé du « microservice-commandes » :  
dans notre cas, un « microservice-commandes » est en bonne santé lorsqu'il y'a des commandes dans la table « COMMANDE », dans ce cas, le statut est « UP » sinon le statut à afficher est « DOWN »

### Etude de cas (2):

La version (2) de la table « COMMANDE » est composée » des colonnes suivantes [id, description, quantité, date, montant, id\_produit]

- a. Présenter sur un schéma l'architecture de mise en place de cette application.
- b. Les microservice-commandes et microservice-produit doivent être enregistrés auprès d'Eureka
- c. Implémenter une Gateway (Zuul ou API Gateway) comme point d'accès unique à l'application.
- d. Implémenter les fonctionnalités CRUD du « microservice-commandes »
- e. Mettre en place le mécanisme de load balancing pour cette application.
- f. Simuler un Timeout d'un des deux microservices, et implémenter un mécanisme de contournement pour protéger le microservice appelant avec Hystrix.
- g. Appliquer OpenAPI et Swagger.

### Livrables:

- a. Démonstration des travaux réalisés.
- b. Code sources des projets Maven (Etude de cas 1 et 2) à déposer sur github avec un fichier Readme dont lequel vous présentez la Team de développement et toutes autres informations qui vous semblent pertinentes pour la bonne exécution des projets.
- c. Screenshots de l'application (cas 1 et 2)