Consommer une API REST en utilisant

Springboot

Au cours de l'activité précédente vous avez créé une web-app exposant sa plus belle API REST aux yeux du monde (local). Pour la tester vous avez utilisé Postman et vérifié que tout fonctionnait correctement. Nous allons maintenant faire un pas de plus dans le monde des microservices.

### 

### **Objectifs de l’activité :**

* Créer une nouvelle web-app qui tournera sur un port différent de la première web-app.
* Cette web-app sera capable de “consommer” l’API REST exposée par un autre service en lui envoyant des requêtes CRUD.
* Utilisation de la classe RestTemplate

### **Modalités :**

* Travail en autonomie
* Production individuelle

### **Consignes:**

Nous allons plus ou moins découper la web app que vous avez développé en deux services (microservices!) distincts :

* Un premier service tournant sur le port 8081 de votre machine. Ce service expose une API REST telle que développée dans l’activité 2.
  + un GET renvoie la liste des modèles de voiture
  + un GET avec {id} renverra les infos du modèle id,
  + un POST sert à ajouter un modèle,
  + un PUT sert à modifier un modèle,
  + un DELETE avec {id} sert à supprimer un modèle
* Le second tourne sur le port 8082 de votre machine et “consomme” l’API de localhost:8081 puis sert une vue à l’utilisateur.

Pour faire simple vous pouvez commencer par n’effectuer que la requête GET pour récupérer la liste des modèles de voiture [ utiliser la classe RestTemplate et sa méthode getForObject ]

### **Livrables :**

* Codes commentés des différentes classes de votre application, à déposer sous git.
* Glossaire des différentes syntaxes utilisées. Continuez à rédiger votre glossaire des différentes annotations Spring.

### **Compétences associées :**

La compétence validée par cette activité est la compétence 6

### **Durée :**

1 jour

### **Ressources :**

La doc officielle :

* <https://spring.io/guides/gs/consuming-rest/>

Les méthodes de la classe RestTemplate :

* <https://docs.spring.io/autorepo/docs/spring/3.2.3.RELEASE/javadoc-api/org/springframework/web/client/RestTemplate.html>

Un cours bien fait sur youtube : Java Brains **(in english)** :

* Cours sur les microservices en Spring Boot (playlist youtube). Pour cette partie ce qui nous intéresse correspond aux vidéos 09 - 10 et 11 de la playlist. Si vous vous sentez perdus, vous pouvez regarder la playlist en entier (ce n’est pas très long).

# 09 Configuring server port for microservices - Spring Boot Microservices Level 1

<https://www.youtube.com/watch?v=i6ipBFb-5YM&list=PLqq-6Pq4lTTZSKAFG6aCDVDP86Qx4lNas&index=9>

# 10 Coding the third service and discussing communication - Spring Boot Microservices Level 1

<https://www.youtube.com/watch?v=UBnSkjsJ-ow&list=PLqq-6Pq4lTTZSKAFG6aCDVDP86Qx4lNas&index=10>

# 11 Using RestTemplate to call an external microservice API - Spring Boot Microservices Level 1

<https://www.youtube.com/watch?v=WPKv8NA-ZhE&list=PLqq-6Pq4lTTZSKAFG6aCDVDP86Qx4lNas&index=11>

* Si vous êtes perdus sur la vidéo 11 : une explication de stream(), map() et collect()

<https://dzone.com/articles/how-to-use-map-filter-collect-of-stream-in-java-8>

Une version très simple et synthétique de consommer un service RESTful (mais sans explication orale) :

* <https://www.youtube.com/watch?v=W_PJJVt23I8>
* Appel Rest, mais est-ce un microservice ? Pourquoi ?