Créer une API REST avec SpringBoot

### **Introduction:**

Nous allons repartir de la webapp précédemment développée. Nous allons développer une API REST afin d’élaborer un CRUD pour les modèles de véhicule précédemment manipulés.

### **Modalités :**

* Travail en autonomie
* Production individuelle

### **Objectifs de l’activité :**

* Créer une API REST qui va nous permettre de lister, visualiser, ajouter, modifier et supprimer les modèles de véhicules.
* Créer les contrôleurs correspondants aux différentes routes
* Implémenter une architecture DAO pour gérer les modèles de véhicules.
* Comprendre les différentes annotations utilisées par Springboot (@autowiring, @controller, @repository, @service….)
* Utiliser POSTMAN pour tester une API
* Documenter une API avec SWAGGER

### **Consignes:**

1. **Créez une API REST**

Nous allons élaborer un CRUD. Pour ce faire, nous allons créer les mappings suivants :

/modeles en GET, qui renverra la liste des modèles

/modeles/{id} en GET, qui renverra les infos du modèle id,

/modeles en POST, qui sert à ajouter un modèle,

/modeles en PUT, qui sert à modifier un modèle,

/modeles/{id} en DELETE, pour supprimer un modèle

Les différents retours s’effectueront en JSON.

On implémentera un DAO pour accéder aux modèles, avec les méthodes findAll(), findById(**int** id), save(Modele m) et delete(**int** id).

Dans un premier temps, la liste des modèles sera stockée « en dur » dans la classe Dao (ultérieurement, on ira la stocker en base de données).

Vous étudierez tout d’abord les tutos suivants :

* Développer une API REST avec SpringBoot :

<https://atomrace.com/developper-une-api-rest-avec-spring-boot>

* Openclassroom : créer un microservice grâce à SpringBoot (fin du chapitre 3) :

[https://openclassrooms.com/fr/courses/4668056-construisez-des- microservices/5122884-creez-un-microservice-grace-a-spring-boot](https://openclassrooms.com/fr/courses/4668056-construisez-des-%20%20microservices/5122884-creez-un-microservice-grace-a-spring-boot)

1. **Testez votre API avec POSTMAN**

Installez et utilisez POSTMAN pour tester votre API.

* Tuto Openclassroom : testez votre API avec Postman :

<https://openclassrooms.com/fr/courses/4668056-construisez-des-microservices/5123020-testez-votre-api-grace-a-postman>

1. **Documentez votre API avec SWAGGER**

Installez et utilisez SWAGGER pour documenter votre API.

* Tuto Openclassroom : documentez votre microservice avec Swagger2 :

<https://openclassrooms.com/fr/courses/4668056-construisez-des-microservices/5123565-documentez-votre-microservice-avec-swagger-2>

* Plus éventuellement Tuto Swagger

<http://www.baeldung.com/swagger-2-documentation-for-spring-rest-api>

### **Livrables :**

* Codes commentés des différentes classes de votre application, à déposer sous git.
* Documentation de l’API avec Swagger
* Enrichissement de votre glossaire des différentes syntaxes et annotations SpringBoot utilisées.

### **Durée :**

1 jour

### **Compétences associées :**

Les compétences validées par cette activité sont les compétences 3, 4 et 5.

### **Ressources :**

* Développer une API REST avec SpringBoot :

<https://atomrace.com/developper-une-api-rest-avec-spring-boot>

* Openclassroom : construisez des micro services (fin du chapitre 3, chapitre 4 et chapitre 9) :

[https://openclassrooms.com/fr/courses/4668056-construisez-des- microservices/5122884-creez-un-microservice-grace-a-spring-boot](https://openclassrooms.com/fr/courses/4668056-construisez-des-%20%20microservices/5122884-creez-un-microservice-grace-a-spring-boot)

<https://openclassrooms.com/fr/courses/4668056-construisez-des-microservices/5123020-testez-votre-api-grace-a-postman>

<https://openclassrooms.com/fr/courses/4668056-construisez-des-microservices/5123565-documentez-votre-microservice-avec-swagger-2>

* Tuto Swagger :

<http://www.baeldung.com/swagger-2-documentation-for-spring-rest-api>