Introduction à SpringBoot  
première webapp

### Généralités J2EE, JEE, Spring, Spring Boot

Java est à la base un langage et un ensemble de classe prévues pour la programmation générale. Très rapidement, il a fallu mettre en place des extensions permettant de programmer plus rapidement dans certains domaines, en particulier ce qui était vu à l’époque comme les domaines professionnels (en simplifiant) :

* connexions aisées avec des bases/sources de donnée ;
* facilitation d’architecture MVC pour le Web avec les Servlets (Contrôleurs) et JSP (templates pour les vues) ;
* divers aspects d’architecture et outils (inversion de dépendance [voir plus loin], XML, convention “Beans”, connecteur mail, etc).

Cela a donné lieu à Java 2 Entreprise Edition (J2EE), qui sera renommé par la suite en Java Entreprise Edition (cf <https://en.wikipedia.org/wiki/Java_EE_version_history>).

Les premières versions de J2EE/JEE étaient considérées à la fois très intéressantes et très difficiles à appréhender et à mettre en œuvre. Des efforts concurrents multiples ont été menés et un en particulier s’est fortement inspiré de J2EE pour chercher à le simplifier tout en en gardant l’esprit : Spring ( cf <https://fr.wikipedia.org/wiki/Spring_(framework)>).

L’idée principale est d’avoir des composants  - Beans - d’une application (en général Web) qui peuvent être assemblés de façon dynamique pour réaliser cette application. Par exemple, tel composant de sources de données avec tel composant d’accès à un service externe, etc. **Ces composant sont définis par l’interface qu’ils implémentent (contrat)** et l’instance effective des composants découlent de la configuration de l’application. Cela permet de facilement changer les composants d’une application (découplage) pour, par exemple, avoir une version avec stockage local, une version avec un stockage dans le Web dans une base MongoDB… ou replacer un composant qui sera disponible en production par un composant “mimant” son fonctionnement, permettant de tester le reste de l’application (principe des “mock/stub” ou  “simulacre/bouchon”, cf http://www.test-recette.fr/tests-techniques/deployer-tests-unitaires/simulacres-bouchons.html)

Nous n’allons pas couvrir les différentes versions ou étapes de ces frameworks qui sont parmi les plus utilisés dans l’environnement Java. Nous allons au contraire nous concentrer sur le framework Spring Boot qui simplifie (certes en limitant) l’utilisation de Spring. En quelque sorte, Spring Boot pour Java est ce qu’est Laravel pour PHP.

# Introduction :

Nous allons, au cours de ce module d’alternance, utiliser SpringBoot afin de développer des microservices. Le travail va s’effectuer par étapes :

* Nous allons tout d’abord prendre en main l’architecture générale de SpringBoot.
* Nous élaborerons ensuite des API REST simples
* Nous consommerons ces API via SpringBoot
* Puis, ces API accéderont à une base de données via JPA
* Nous aborderons enfin les microservices
* Et nous mettrons tout cela en œuvre via une application de location de véhicules.

Dans ce qui va suivre, nous allons bien entendu utiliser un IDE. Plusieurs IDE peuvent être utilisés lors de développement avec SpringBoot : Eclipse, NetBeans, IntelliJ ou STS (Spring Tool Suite)… Nous allons choisir (arbitrairement !) d’utiliser IntelliJ. Il vous faudra donc adapter les tutos que vous allez suivre à l’IDE IntelliJ.

### Objectifs de l’activité :

* Découvrir SpringBoot
* Configurer son IDE pour la suite du module
* Créer une première application WEB avec SpringBoot
* Aborder l’utilisation du moteur de template Thymeleaf pour l’élaboration des vues

### Modalités :

* Travail en autonomie
* Production individuelle

### Consignes:

1. **Etude de tutoriels :**

Openclassroom : construisez des micro services (chapitre 2 + début du chapitre 3) :

* [*https://openclassrooms.com/fr/courses/4668056-construisez-des-microservices/5122425-decouvrez-le-framework-spring-boot*](https://openclassrooms.com/fr/courses/4668056-construisez-des-microservices/5122425-decouvrez-le-framework-spring-boot)
* [*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-microservices/5122884-creez-un-microservice-grace-a-spring-boot)

# Video : Learn Spring Boot - Hello World in 2018 With IntelliJ :

* [*https://www.youtube.com/watch?v=27I1M5RLplE*](https://www.youtube.com/watch?v=27I1M5RLplE)

# Tuto démarrage Spring Boot :

# [*https://o7planning.org/fr/11267/le-tutoriel-de-spring-boot-pour-les-debutants*](https://o7planning.org/fr/11267/le-tutoriel-de-spring-boot-pour-les-debutants)

1. **Création d’une première webapp**

Nous allons créer une première application web afin de prendre en main l’architecture de SpringBoot. Les différentes vues de cette application seront générées par le moteur de template ***thymeleaf.*** (on aurait pu aussi utiliser directement du JSP, ou JSF, etc…)

Développez une application qui permet l’affichage d’une liste de modèles de voitures, et l’ajout d’un nouveau modèle dans cette liste (avec affichage d’un message d’erreur si les données ne sont pas saisies). Les modèles ont un id, et sont référencés par une marque et un modèle (ex : Audi A7, Audi A8, Renault Clio, Renault Espace, Peugeot 208….). En vous inspirant du tuto suivant :

* [*https://o7planning.org/fr/11545/tutoriel-spring-boot-et-thymeleaf*](https://o7planning.org/fr/11545/tutoriel-spring-boot-et-thymeleaf)

### Livrables :

* Codes commentés des différentes classes de votre application, à déposer sous git.
* Glossaire des différentes syntaxes utilisées. Commencez à rédiger un glossaire des différentes annotations Spring au cours de cette activité. Vous le ferez évoluer au fil du module.

### Compétences associées :

Les compétences validées par cette activité sont les compétences 1 et 2.

### Durée :

1,5 jour

### Ressources :

Openclassroom : construisez des micro services (chapitres 2, et début du chapitre 3) :

* [*https://openclassrooms.com/fr/courses/4668056-construisez-des-microservices/5122425-decouvrez-le-framework-spring-boot*](https://openclassrooms.com/fr/courses/4668056-construisez-des-microservices/5122425-decouvrez-le-framework-spring-boot)
* [*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-microservices/5122884-creez-un-microservice-grace-a-spring-boot)

# Video : Learn Spring Boot - Hello World in 2018 With IntelliJ :

* [*https://www.youtube.com/watch?v=27I1M5RLplE*](https://www.youtube.com/watch?v=27I1M5RLplE)

Le Tutoriel de Spring Boot et Thymeleaf :

* [*https://o7planning.org/fr/11545/tutoriel-spring-boot-et-thymeleaf*](https://o7planning.org/fr/11545/tutoriel-spring-boot-et-thymeleaf)

Les annotations Spring :

* [*https://jrebel.com/rebellabs/spring-framework-annotations-cheat-sheet/*](https://jrebel.com/rebellabs/spring-framework-annotations-cheat-sheet/)

## Ressources complémentaires

Voici 2 tutoriels assez complets pour Spring Boot :

* <https://projects.spring.io/spring-boot/#quick-start>
* <https://spring.io/guides/gs/serving-web-content/>

D’autres tutoriels peuvent être intéressant en complément ou avec une présentation légèrement différente :

* <https://www.ibm.com/developerworks/library/j-spring-boot-basics-perry/>
* <https://www.callicoder.com/spring-boot-rest-api-tutorial-with-mysql-jpa-hibernate/>
* ht[tps://www.youtube.com/watch?v=msXL2oDexqw&list=PLqq-6Pq4lTTbx8p2oCgcAQGQyqN8XeA1x&index=1](https://www.youtube.com/watch?v=msXL2oDexqw&list=PLqq-6Pq4lTTbx8p2oCgcAQGQyqN8XeA1x&index=1)