



Jour 1 : Introduction à Spring Boot et Fondamentaux



Job 1

Rendez-vous sur le site de **Spring Initializr**. Sélectionnez **Maven** comme système de construction, la dernière version de Spring Boot, et ajoutez les **dépendances Web** et **Thymeleaf**. Ensuite, téléchargez et extrayez le projet.



Project

☐ Gradle - Groovy

☐ Gradle - Kotlin

☒ Maven

Language

☒ Java

☐ Kotlin

☐ Groovy

Spring Boot

☐ 3.5.0 (SNAPSHOT)

☐ 3.5.0 (M2)

☐ 3.4.4 (SNAPSHOT)

☒ 3.4.3

☐ 3.3.10 (SNAPSHOT)

☐ 3.3.9

Project Metadata

Group

com.example

Artifact

demo

Name

demo

Description

Demo project for Spring Boot

Dependencies

ADD DEPENDENCIES... CTRL + B

No dependency selected

GENERATE CTRL + G

EXPLORE CTRL + SPACE

...



Objectif : Avoir un projet Spring Boot prêt à être importé dans votre IDE.

Question : Qu'est-ce que Spring Initializr et comment peut-il faciliter la création d'un nouveau projet Spring?

Job 2

Ouvrez le projet dans votre IDE préféré. Examinez les fichiers et les dossiers générés, en particulier le dossier src et le fichier pom.xml.

Objectif : Comprendre la structure de base d'un projet Spring Boot.

Question : Pourquoi le fichier pom.xml est-il crucial dans un projet Maven ?

Job 3

Créez un nouveau package nommé controller. Dans celui-ci, créez une classe HelloController qui retourne "Hello World" à la racine (/).

Objectif : Avoir une application qui affiche "Hello World" à l'adresse localhost:8080.

Question : Qu'est-ce qu'un contrôleur dans le contexte de Spring MVC ?

Job 4



Convertissez `application.properties` en `application.yml` et définissez une propriété, par exemple `greeting.message`. Utilisez `@Value` pour injecter cette propriété dans votre contrôleur.

Objectif : Afficher un message personnalisé défini dans `application.yml` sur votre route.

Question : Comment Spring permet-il l'injection de propriétés depuis des fichiers de configuration ?

Job 5

Définissez deux profils dans `application.yml`: `dev` et `prod`. Chacun ayant un message de salutation différent. Faites en sorte que `dev` soit le profil par défaut.

Objectif : Afficher le message de salutation du profil `dev` par défaut.

Question : Pourquoi serait-il utile d'avoir différents profils dans une application Spring ?

Job 6

Ajoutez la dépendance `DevTools` à votre projet. Modifiez votre code et observez le comportement de l'application.

Objectif : Découvrir le rechargement automatique des modifications grâce à `DevTools`.



Question : En quoi la dépendance DevTools est-elle bénéfique pour le développement ?

Rendu

Le projet est à rendre sur https://github.com/prenom-nom/runtrack_java_spring/jour1

Compétences visées

- Java Spring
- Spring Boot

Base de connaissances

- [Github](#)
- [Wiki GIT](#)
- [GIT](#)
- [Spring](#)
- [Spring Initializr](#)