



UNIVERSITÉ SAINT JEAN  
SAINT JEAN INGÉNIEUR

GROUPE 2, LICENCE 3  
2023/2024

# APPRENTISSAGE DU FRAMEWORK SPRING BOOT

## MEMBRES DU GROUPE 3

- TSIDIE PIERRE BERKAM 20%
- KOM MONJOH IVAN MARTIAL 20%
- NGUEWO CHRISTIAN 20%
- TAKOUTCHE ARMEL 20%
- DJOUMESSI FRANKLIN 20%

Spring boot groupe 2

## Table des matières

Introduction .....	2
I. Définition et historiques de l'api .....	2
1. Définition.....	2
2. Historiques de l'API.....	3
II. Rôle et mécanismes d'utilisation.....	4
1. Rôle .....	4
2. Mécanismes d'utilisation.....	5
III. Implémentation.....	6
IV. Recommandation et mode d'emploi.....	7
1. Recommandation.....	7
2. Mode d'emploi .....	8
Conclusion.....	10

# Introduction

Spring est un **framework**, c'est-à-dire un **cadre de travail existant** que les développeurs peuvent utiliser. Spring Framework : Un socle solide pour le développement d'applications. C'est un framework open-source largement utilisé pour simplifier et accélérer le développement d'applications Java, en particulier les applications d'entreprise. Il s'agit d'un ensemble de modules et de bibliothèques qui offrent des fonctionnalités prêtes à l'emploi pour de nombreux aspects du développement, tels que la gestion des dépendances, l'injection de dépendances, la configuration, la gestion des transactions, la sécurité, l'accès aux données et le développement web. Grâce à son mécanisme de configuration automatique basé sur des conventions intelligentes, Spring Boot prend en charge de nombreux aspects du développement, ce qui permet aux développeurs de se concentrer sur la logique métier de leurs applications plutôt que sur les détails techniques. En combinant la puissance du framework Spring avec des fonctionnalités telles que les starters pré-configurés, l'embarquabilité, la gestion avancée des propriétés et une intégration transparente avec l'écosystème Spring, Spring Boot offre une expérience de développement fluide et efficace pour créer une large gamme d'applications, des microservices aux applications web en passant par les services cloud.

## I. Définition et historiques de l'api

### 1. Définition

Spring Boot est un framework open-source pour le développement d'applications Java. Il vise à simplifier le processus de création, de configuration et de déploiement d'applications Java en fournissant des outils et des conventions intelligentes par défaut. Conçu pour être facile à utiliser, Spring Boot élimine la nécessité de configurer manuellement de nombreux aspects d'une application Java, en offrant une configuration automatique basée sur des conventions, tout en restant extensible pour répondre aux besoins spécifiques des développeurs. Quelques points clés qui définissent Spring Boot :

- ✚ **Productivité accrue** : Spring Boot vise à accroître la productivité des développeurs en réduisant la quantité de code boilerplate nécessaire pour démarrer un projet. Avec Spring Boot, les développeurs peuvent créer rapidement des applications Java sans avoir à configurer manuellement de nombreuses fonctionnalités.

- ✚ **Configuration automatique :** Spring Boot offre une configuration automatique intelligente basée sur des conventions. Il détecte automatiquement les bibliothèques et les frameworks utilisés dans le projet et configure les beans Spring correspondants en conséquence, ce qui réduit la charge de configuration manuelle.
- ✚ **Intégration simplifiée :** Spring Boot facilite l'intégration avec d'autres frameworks et bibliothèques populaires. Il fournit des starters préconfigurés pour une variété de technologies, ce qui simplifie l'ajout de fonctionnalités telles que la connectivité avec les bases de données, la sécurité, les services web, etc.
- ✚ **Microservices :** Spring Boot est souvent utilisé pour le développement de microservices en raison de sa facilité de configuration et de déploiement. Il offre des fonctionnalités telles que la gestion des dépendances, l'intégration avec Spring Cloud pour la mise en œuvre de modèles de microservices, et la prise en charge de conteneurs légers comme Docker.
- ✚ **Écosystème Spring:** Spring Boot s'intègre parfaitement à l'écosystème Spring, notamment Spring Framework, Spring Data, Spring Security, etc. Cela permet aux développeurs de bénéficier des fonctionnalités avancées de Spring tout en utilisant les fonctionnalités de simplification fournies par Spring Boot.

## 2. Historiques de l'API

Spring Boot est une partie intégrante de l'écosystème Spring développé par Pivotal Software (maintenant VMware) pour faciliter et accélérer le développement d'applications Java. Voici un bref historique de l'API Spring Boot :

- ✚ **2013 : Lancement initial** - Spring Boot a été annoncé pour la première fois par l'équipe de Spring lors de la conférence SpringOne 2GX en 2013. L'objectif principal était de simplifier le processus de développement d'applications basées sur Spring en fournissant des conventions intelligentes, une configuration automatique et des fonctionnalités de démarrage rapide.
- ✚ **2014 : Première version publique** - La première version publique de Spring Boot, la version 1.0, a été publiée en avril 2014. Cette version initiale a posé les bases du framework et a introduit des fonctionnalités telles que la configuration automatique, les starters, et la prise en charge des environnements de développement.
- ✚ **Versions successives** : Depuis sa première version, Spring Boot a connu un développement rapide avec des mises à jour régulières et des améliorations continues. Les versions successives ont apporté de nouvelles fonctionnalités, des améliorations de performances, des correctifs de bugs et une compatibilité avec les dernières versions de Spring Framework.

- ✚ **2015 : Lancement de Spring Boot 1.3** - Cette version a introduit des fonctionnalités telles que la prise en charge de Groovy dans les scripts de configuration, la gestion améliorée des propriétés de configuration, et des améliorations de performances significatives.
- ✚ **2016 : Spring Boot 1.4 et au-delà** - Les versions ultérieures ont continué à étendre les fonctionnalités de Spring Boot avec des améliorations de la gestion des dépendances, une meilleure intégration avec Spring Cloud pour le développement de microservices, des outils de surveillance et de gestion, et une compatibilité accrue avec les dernières technologies et normes de l'industrie.
- ✚ **2018 : Annonce de Spring Boot 2.0** - Spring Boot 2.0 a été annoncé, apportant des fonctionnalités importantes telles que la compatibilité avec Java 9, la mise à jour vers Spring Framework 5, la prise en charge de la programmation réactive, et des améliorations de performances.
- ✚ **Versions récentes** : Les versions les plus récentes de Spring Boot continuent à évoluer avec des mises à jour régulières, des corrections de bugs, des améliorations de performances et le support des dernières versions de Java et des autres technologies. la version la plus récente est : **Spring Boot 3.2.3**

## II. Rôle et mécanismes d'utilisation

### 1. Rôle

Il s'appuie sur le Framework Spring Framework en offrant des configurations par défaut et en automatisant de nombreuses tâches répétitives. Le rôle de Spring Boot dans le développement d'applications Java est crucial et multifacette. Voici les principaux aspects de son rôle :

- **Simplification du développement** : Spring Boot vise à simplifier le processus de développement d'applications Java en réduisant la quantité de configuration nécessaire. Il fournit des conventions intelligentes et une configuration automatique, ce qui permet aux développeurs de démarrer rapidement un nouveau projet sans avoir à passer du temps sur des tâches de configuration fastidieuses.
- **Accélération du développement** : En offrant des starters préconfigurés pour diverses tâches et technologies, Spring Boot accélère le processus de développement. Les starters incluent toutes les dépendances nécessaires pour un scénario donné, ce qui permet aux développeurs de se concentrer sur la logique métier de leur application plutôt que sur la résolution des dépendances et la configuration.




- **Facilitation du déploiement** : Spring Boot permet d'embarquer un serveur d'application dans l'application elle-même, ce qui facilite le déploiement de l'application. Les applications Spring Boot peuvent être déployées sous forme de fichiers JAR exécutables, ce qui simplifie la mise en œuvre et réduit la dépendance à des serveurs d'application externes.
- **Intégration transparente** : Spring Boot s'intègre facilement avec l'écosystème Spring ainsi qu'avec d'autres Frameworks et bibliothèques Java populaires. Cela permet aux développeurs de bénéficier des fonctionnalités avancées de Spring et d'intégrer facilement d'autres technologies dans leurs applications.
- **Gestion de la configuration** : Spring Boot offre des mécanismes de gestion de la configuration flexibles et puissants. Les propriétés de configuration peuvent être définies à l'aide de fichiers de propriétés, de variables d'environnement, de profils, etc., ce qui permet de personnaliser le comportement de l'application en fonction de différents environnements.
- **Surveillance et gestion des applications** : Spring Boot Actuator fournit des fonctionnalités pour surveiller et gérer les applications en cours d'exécution. Il expose des points de terminaison HTTP qui permettent de vérifier l'état de l'application, de surveiller les métriques, de gérer les journaux, etc., ce qui facilite la surveillance et la gestion des applications en production.

## 2. Mécanismes d'utilisation

Spring Boot propose plusieurs mécanismes pour faciliter l'utilisation du Framework et accélérer le développement des applications Java. Voici quelques-uns des principaux mécanismes d'utilisation de Spring Boot :

- ✚ **Annotations et conventions de configuration** : Spring Boot utilise des annotations et des conventions de configuration pour simplifier le développement. Par exemple, l'annotation `@SpringBootApplication` sur la classe principale de l'application configure automatiquement divers aspects de l'application, tels que la recherche de composants Spring, l'activation de la configuration automatique, etc.
- ✚ **Starters** : Les starters sont des dépendances prédéfinies qui regroupent les bibliothèques et les configurations couramment utilisées pour des tâches spécifiques. Par exemple, le starter `spring-boot-starter-web` inclut toutes les dépendances nécessaires pour créer une application web, tandis que le starter `spring-boot-starter-data-jpa` inclut les dépendances nécessaires pour accéder aux bases de données avec JPA.
- ✚ **Embarquabilité (Embeddability)** : Spring Boot permet d'embarquer un serveur d'application dans l'application elle-même. Cela signifie que vous pouvez exécuter votre application Java en

tant qu'application autonome, sans avoir besoin de déployer un serveur d'application séparé tel que Tomcat ou Jetty. Vous pouvez créer un fichier JAR exécutable qui contient tout ce dont votre application a besoin pour s'exécuter.

-  **Configuration externe** : Spring Boot facilite la gestion de la configuration en permettant de définir des propriétés dans des fichiers de propriétés externes, des variables d'environnement, des arguments de ligne de commande, etc. Les propriétés de configuration peuvent être définies dans différents profils pour personnaliser le comportement de l'application dans différents environnements.
-  **Actuator** : Spring Boot Actuator fournit des fonctionnalités pour surveiller et gérer l'application en cours d'exécution. Il expose des points de terminaison HTTP qui permettent de vérifier l'état de l'application, de surveiller les métriques, de gérer les journaux, etc. Actuator facilite la surveillance et la gestion des applications Spring Boot en production.
-  **Spring Initializr** : Spring Initializr est un outil en ligne qui permet de générer rapidement un projet Spring Boot personnalisé. Vous pouvez sélectionner les dépendances et les technologies que vous souhaitez utiliser dans votre application, puis télécharger un projet prêt à l'emploi.

### III. Implémentation

Pour implémenter un projet utilisant Spring Boot, voici les étapes générales à suivre :

#### I. Configurer votre environnement de développement :

- Assurez-vous d'avoir installé Java JDK sur votre machine.
- Utilisez un IDE tel qu'Eclipse, IntelliJ IDEA ou Spring Tool Suite pour faciliter le développement.

#### II. Créer un nouveau projet Spring Boot :

- Vous pouvez utiliser Spring Initializr pour générer un projet Spring Boot personnalisé. Visitez <https://start.spring.io/>.
- Sélectionnez les dépendances nécessaires pour votre projet, telles que Web, JPA, Security, etc.
- Téléchargez le projet ZIP généré et extrayez-le dans votre espace de travail.

#### III. Configurer votre application :

- Ouvrez le projet dans votre IDE et explorez la structure du projet.
- Modifiez le fichier `application.properties` ou `application.yml` pour configurer votre application en fonction de vos besoins (base de données, port d'écoute, etc.).

#### **IV. Écrire votre code :**

- Commencez par écrire votre logique métier dans les classes de service, les contrôleurs, etc.
- Utilisez les annotations Spring appropriées pour définir les composants de votre application, tels que `@RestController`, `@Service`, `@Repository`, etc.
- Implémentez vos endpoints REST si vous construisez une API web.

#### **V. Exécuter votre application :**

- Vous pouvez exécuter votre application directement depuis votre IDE ou utiliser Maven ou Gradle en ligne de commande.
- Si vous utilisez Maven, exécutez `mvn spring-boot:run`.
- Si vous utilisez Gradle, exécutez `./gradlew bootRun`.

#### **VI. Tester votre application:**

- Utilisez un outil de test comme Postman pour tester les endpoints de votre application.
- Assurez-vous que toutes les fonctionnalités de votre application fonctionnent comme prévu.

#### **VII. Déployer votre application :**

- Vous pouvez déployer votre application Spring Boot sur divers environnements, tels que des serveurs d'application traditionnels, des conteneurs Docker, des services cloud, etc.
- Pour un déploiement local, vous pouvez simplement exécuter le fichier JAR généré par Spring Boot en utilisant la commande `java -jar mon_application.jar`.
- Pour un déploiement sur des services cloud, suivez les instructions spécifiques au fournisseur de cloud que vous utilisez (AWS, Azure, Google Cloud, etc.).

## **IV. Recommandation et mode d'emploi**

### **1. Recommandation**

Spring Boot est un framework très populaire dans l'écosystème Java, et il est largement recommandé pour le développement d'applications Java, en particulier pour les microservices et les applications web. Voici quelques-unes des principales raisons pour lesquelles Spring Boot est recommandé :



**Productivité élevée :** Spring Boot simplifie considérablement le processus de développement en offrant une configuration automatique basée sur des conventions. Cela permet aux développeurs de se concentrer sur la logique métier de leur application plutôt que sur la configuration de l'infrastructure.

**Facilité de démarrage :** Avec Spring Boot, il est facile de démarrer un nouveau projet grâce à l'outil Spring Initializr qui génère un squelette de projet prêt à l'emploi avec les dépendances nécessaires.

**Intégration transparente :** Spring Boot s'intègre facilement avec d'autres frameworks et technologies Java, ainsi qu'avec l'écosystème Spring. Cela permet aux développeurs de bénéficier des fonctionnalités avancées de Spring tout en utilisant les fonctionnalités de simplification fournies par Spring Boot.

**Embarquabilité (Embeddability) :** Spring Boot permet d'embarquer un serveur d'application dans l'application elle-même, ce qui facilite le déploiement de l'application sous forme de fichier JAR exécutable.

**Écosystème robuste :** Spring Boot fait partie de l'écosystème Spring, qui offre une gamme étendue de projets et de bibliothèques pour répondre à divers besoins de développement. Cela inclut Spring Framework, Spring Data, Spring Security, Spring Cloud, et bien d'autres.

**Documentation complète et communauté active :** Spring Boot bénéficie d'une documentation complète et bien entretenue, ainsi que d'une communauté active de développeurs prêts à aider et à partager leurs connaissances.

**Sécurité :** Spring Boot propose des fonctionnalités de sécurité robustes, notamment l'intégration avec Spring Security, qui facilite la mise en œuvre de l'authentification, de l'autorisation et d'autres aspects de la sécurité des applications.

## 2. Mode d'emploi

Le mode d'emploi de Spring Boot peut être divisé en plusieurs étapes pour vous aider à démarrer rapidement avec le framework. Voici un guide pas à pas pour utiliser Spring Boot efficacement :

### **Étape 1 :** Configuration de l'environnement de développement

1. Assurez-vous d'avoir Java JDK installé sur votre système.
2. Utilisez un IDE tel qu'Eclipse, IntelliJ IDEA ou Spring Tool Suite pour faciliter le développement.

### **Étape 2 :** Création d'un nouveau projet Spring Boot

1. Utilisez Spring Initializr pour générer un projet Spring Boot.
2. Sélectionnez les dépendances nécessaires pour votre projet (par exemple, Web, JPA, Security) et téléchargez le projet ZIP généré.

### **Étape 3 : Configuration de votre application**

1. Explorez la structure du projet et familiarisez-vous avec les fichiers générés.
2. Modifiez le fichier `application.properties` ou `application.yml` pour configurer votre application (par exemple, définissez les paramètres de base de données, les propriétés du serveur, etc.).

### **Étape 4 : Écriture de votre code**

1. Implémentez la logique métier de votre application en créant des classes de service, des contrôleurs, des entités, etc.
2. Utilisez les annotations Spring pour définir les composants de votre application (par exemple, `@RestController`, `@Service`, `@Repository`).
3. Développez vos endpoints REST si vous construisez une API web.

### **Étape 5 : Exécution de votre application**

1. Exécutez votre application à partir de votre IDE ou en utilisant Maven ou Gradle en ligne de commande.
2. Utilisez Maven avec la commande `mvn spring-boot:run` ou Gradle avec `./gradlew bootRun`.

### **Étape 6 : Test de votre application**

1. Utilisez des outils de test comme Postman pour tester les endpoints de votre application.
2. Assurez-vous que toutes les fonctionnalités de votre application fonctionnent comme prévu.

### **Étape 7 : Déploiement de votre application**

1. Pour un déploiement local, exécutez simplement le fichier JAR généré par Spring Boot en utilisant la commande `java -jar mon_application.jar`.
2. Pour un déploiement sur un service cloud ou un serveur, suivez les instructions spécifiques au fournisseur de services que vous utilisez.

## Conclusion

En conclusion, Spring Boot est un framework Java puissant et largement utilisé qui simplifie considérablement le processus de développement d'applications. Grâce à sa configuration automatique, ses starters pré-configurés, son embarquabilité, son intégration transparente et son écosystème robuste, Spring Boot permet aux développeurs de créer rapidement des applications Java de haute qualité avec moins d'efforts de configuration et de déploiement. Avec sa documentation complète, sa communauté active et ses fonctionnalités avancées, Spring Boot reste un choix incontournable pour le développement d'une large gamme d'applications, des microservices aux applications web en passant par les services cloud.