BACKEND PROGRAM

3- Spring ecosystem

ISMAIL BOUROUBA

Spring and its key components

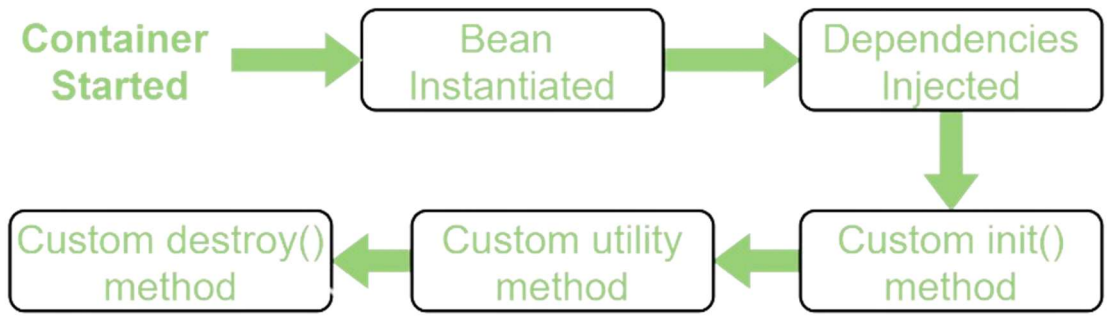
What is Spring?

Spring est un framework populaire pour le développement d'applications web en Java. Il offre un ensemble complet de fonctionnalités et d'outils qui simplifient le processus de développement et favorisent les bonnes pratiques.

What is SpringBoot?

Spring Boot fait partie de l'écosystème Spring et simplifie la configuration et le déploiement des applications Spring. Son objectif est de minimiser le code redondant et de fournir des solutions prêtes à l'emploi pour les besoins courants des applications. Spring Boot intègre un serveur embarqué et une auto-configuration, permettant aux développeurs de créer facilement des applications autonomes prêtes pour la production.

- **Beans lifecycle**



Le cycle de vie de n'importe quel objet signifie quand et comment il naît, comment il se comporte tout au long de sa vie, et quand et comment il meurt. De même, le cycle de vie du bean fait référence à quand et comment le bean est instancié, aux actions qu'il effectue tout au long de sa vie, et quand et comment il est détruit. Dans cet article, nous discuterons du cycle de vie du bean. Le cycle de vie du bean est géré par le conteneur Spring. Lorsque nous exécutons le programme, tout d'abord, le conteneur Spring se lance. Ensuite, le conteneur crée l'instance d'un bean selon la demande, puis les dépendances sont injectées. Enfin, le bean est détruit lorsque le conteneur Spring est fermé. Par conséquent, si nous voulons exécuter un certain code lors de l'instanciation du bean et juste après la fermeture du conteneur Spring, nous pouvons écrire ce code à l'intérieur de la méthode d'initialisation personnalisée (init()) et de la méthode de destruction (destroy()).

1. **Instantiation :**
    - Lorsque le conteneur Spring démarre, il crée une instance du bean selon la configuration définie dans le fichier de configuration (généralement XML ou annotations).
2. **Injection de dépendances :**
    - Une fois l'instance du bean créée, le conteneur Spring injecte les dépendances du bean. Cela se fait généralement par des annotations telles que **@Autowired** ou par des configurations XML.
3. **Initialisation :**
    - Après l'injection des dépendances, si le bean implémente l'interface **InitializingBean** ou s'il est configuré avec une méthode d'initialisation personnalisée à l'aide de l'annotation **@PostConstruct** ou de la méthode init() (selon la configuration), cette méthode est appelée pour effectuer des opérations d'initialisation personnalisées.
4. **Utilisation :**
    - Le bean est maintenant prêt à être utilisé dans l'application. Les méthodes métier peuvent être appelées, et le bean peut interagir avec d'autres composants de l'application.
5. **Destruction :**
    - Lorsque le conteneur Spring est fermé, il détruit les beans. Si le bean implémente l'interface **DisposableBean** ou s'il est configuré avec une méthode de destruction personnalisée à l'aide de l'annotation **@PreDestroy**

ou de la méthode destroy() (selon la configuration), cette méthode est appelée pour effectuer des opérations de nettoyage.

What is Microservice?

Les microservices, ou architecture de microservices, représentent une technique de développement logiciel où une grande application est divisée en petits services modulaires. Chacun de ces services fonctionne de manière indépendante et exécute un processus métier spécifique. Cette décentralisation permet aux équipes de développer, déployer et mettre à l'échelle les services de manière indépendante, améliorant l'isolation des erreurs et offrant divers autres avantages. Cette architecture est particulièrement adaptée aux entreprises cherchant à faire évoluer leur architecture système en parallèle de leur croissance, ce qui leur permet de s'adapter aux évolutions des technologies et des exigences commerciales.

NB : i will drop a project in my gitlab account implement the microservice archi