Spring Cloud est un framework open source pour le développement de systèmes distribués basés sur Spring Boot. Il fournit des outils pour la création de microservices et de solutions cloud évolutives.

Voici une documentation de base pour Spring Cloud :

Architecture : Spring Cloud utilise une architecture de microservices pour le développement d'applications distribuées. Il fournit des outils pour la configuration, le routage, la découverte de services et la gestion de la charge pour les microservices.

Modules : Spring Cloud se compose de plusieurs modules, chacun fournissant une fonctionnalité spécifique. Certains des modules les plus couramment utilisés sont :

Spring Cloud Config : Il permet de stocker les configurations des applications dans un référentiel centralisé, ce qui facilite la gestion et la mise à jour des configurations.

Spring Cloud Netflix : Il fournit des intégrations avec les outils Netflix OSS tels que Eureka pour la découverte de services, Ribbon pour la gestion de la charge, et Hystrix pour la résilience des applications.

Spring Cloud Bus : Il fournit un canal de communication entre les microservices pour la propagation des événements de configuration.

Spring Cloud Stream : Il fournit un modèle de programmation pour la production et la consommation de flux de données en utilisant des brokers de messages tels que Apache Kafka ou RabbitMQ.

Spring Cloud Sleuth :Il fournit des fonctionnalités de traçage distribué pour les requêtes entre les microservices.

Fonctionnalités : Spring Cloud offre de nombreuses fonctionnalités pour les microservices, notamment :

La découverte de services : permet aux microservices de découvrir les autres microservices disponibles dans le système.

La gestion de la charge : permet de distribuer la charge entre les différents microservices pour une utilisation efficace des ressources.

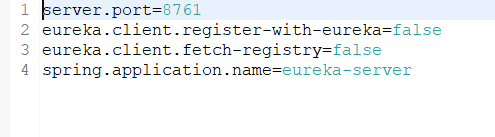
La résilience des applications : permet de gérer les erreurs et les pannes dans les microservices pour garantir la continuité du service.

La sécurité : permet de sécuriser les communications entre les microservices et de gérer l'accès aux ressources.

Utilisation : Pour utiliser Spring Cloud, il faut ajouter les dépendances appropriées dans votre projet Spring Boot. Ensuite, vous pouvez configurer les différents modules selon les besoins de votre application. Vous pouvez également utiliser des annotations et des classes utilitaires fournies par Spring Cloud pour simplifier le développement de microservices.

1-Discovery service

Configuration(Application.properties)



Dependencies(pom.xml)

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>

</dependency>

2-api-gatway

Configuration des routes ( application.properties)

server.port=8888

spring.application.name= api-gateway

#Gateway configuration

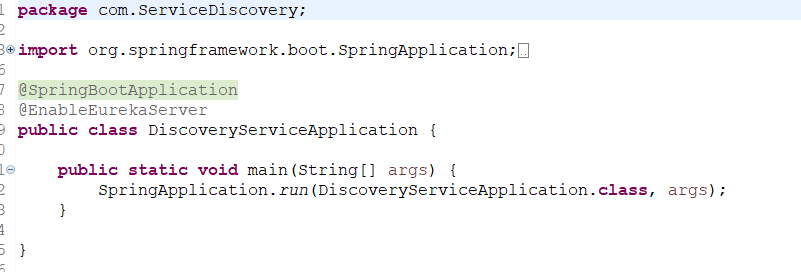
spring.cloud.gateway.routes[0].id=authentication-service

spring.cloud.gateway.routes[0].uri=http://localhost:1515/

spring.cloud.gateway.routes[0].predicates[0]=Path=/\*\*

#spring.cloud.gateway.routes[1].id=another\_route

Configuration classe main



#spring.cloud.gateway.routes[1].uri=http://another.com

#spring.cloud.gateway.routes[1].predicates[0]=Path=/another/\*\*

#spring.cloud.gateway.routes[0].id=account-service

#spring.cloud.gateway.routes[0].uri=http://localhost:1444/

#spring.cloud.gateway.routes[0].predicates[0]=Path=/\*\*

# Eureka configuration (Optional)

spring.cloud.gateway.discovery.locator.enabled=true

Dependencies(pom.xml)

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-gateway</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

</dependencies>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>${spring-cloud.version}</version>

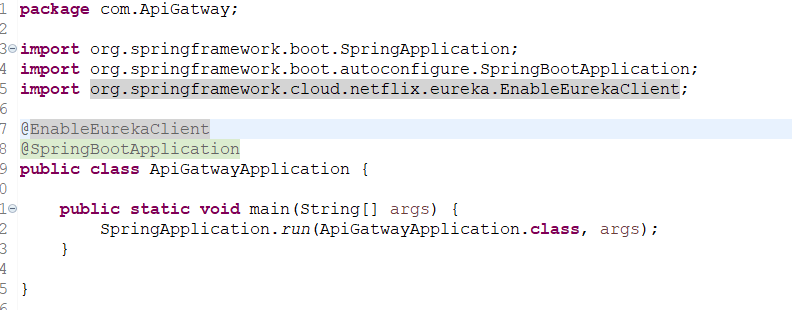
<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>



Exemple de microservice:

Configuration (application.properties)

server.port = 1515

spring.application.name=authentication-service

eureka.client.service-url.default-zone=http://localhost:8761/eureka

Dependencies(pom.xml)

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

En résumé, Spring Cloud est un framework essentiel pour le développement de microservices et de solutions cloud évolutives. Il fournit des outils pour la configuration, la découverte de services, la gestion de la charge, la résilience des applications et la sécurité des microservices. Il est facile à utiliser avec Spring Boot et fournit une architecture modulaire qui permet aux développeurs de choisir les fonctionnalités dont ils ont besoin pour leur application. La documentation officielle de Spring Cloud est disponible sur le site web de Spring à l'adresse suivante : https://spring.io/projects/spring-cloud. Vous y trouverez des guides de démarrage rapide, des exemples de code, des tutoriels et une documentation complète pour chaque module de Spring Cloud.