**Spring Cloud API Gateway – 2024**

Spring Cloud API Gateway is a crucial component in microservices architecture, providing a single entry point for requests. Here’s a concise overview:

1. **Routing**: Directs incoming requests to appropriate microservices based on rules.
2. **Security**: Implements authentication, authorization, and other security measures.
3. **Rate Limiting**: Controls the number of requests to prevent overloading services.
4. **Load Balancing**: Distributes requests evenly across multiple service instances.
5. **API Monitoring**: Tracks and logs request metrics for analysis.

**How to create API Gateway**

**Pom.xml Relevant Portion**

<parent>

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

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

<version>3.2.3</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<properties>

<java.version>17</java.version>

<spring-cloud.version>2023.0.0</spring-cloud.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</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>

**application.properties**

**spring.cloud.gateway.routes[0].id**=sample-app1-service

**spring.cloud.gateway.routes[0].uri**=http://localhost:8081/

**spring.cloud.gateway.routes[0].predicates[0]=**Path=/api/sample-app1-service/\*\*

**spring.cloud.gateway.routes[0].filters[0]=StripPrefix=1**

spring.cloud.gateway.routes[1].id=sample-app2-service

spring.cloud.gateway.routes[1].uri=http://localhost:8082/

spring.cloud.gateway.routes[1].predicates[0]=Path=/api/sample-app2-service/\*\*

**spring.cloud.gateway.routes[1].filters[0]=StripPrefix=1**

**SpringBoot Main Class**

@SpringBootApplication

**public** **class** SpringAPIGatewayApp {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(SpringAPIGatewayApp.**class**, args);

}

}

A Filter Class to log the incoming request

@Slf4j

@Component

**public** **class** APIRequestFilter **implements** GlobalFilter {

@Override

**public** Mono<Void> filter(ServerWebExchange exchange, GatewayFilterChain chain) {

Set<URI> uris = exchange.getAttributeOrDefault(***GATEWAY\_ORIGINAL\_REQUEST\_URL\_ATTR***, Collections.*emptySet*());

String originalUri = (uris.isEmpty()) ? "Unknown" : uris.iterator().next().toString();

**Route route = exchange.getAttribute(*GATEWAY\_ROUTE\_ATTR*);**

**URI routeUri = exchange.getAttribute(*GATEWAY\_REQUEST\_URL\_ATTR*);**

***log***.debug("Incoming Request URI: {}", originalUri);

***log***.debug("Application Id: {}", route.getId());

***log***.debug("Routed URI: {}", routeUri);

***log***.debug("Incoming Request Http Method: {}", exchange.getRequest().getMethod());

**return** chain.filter(exchange);

}

}

We will create the below ordinary microcservices with different port.

|  |  |  |
| --- | --- | --- |
| Name | Port | Description |
| sample-app1-service | 8081 | Ordinary Microservice |
| sample-app2-service | 8082 | Ordinary Microservice |

**There will be a simple Main class annotated with @SpringBootApplication**, no other annotations required.

**For sample-app1-service and sample-app2-service microservices, code given below.**

**Pom.xml**

<parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.2.4</version>  
 <relativePath />  
</parent>  
<properties>  
 <java.version>17</java.version>  
 <spring-cloud.version>2023.0.0</spring-cloud.version>  
</properties>  
  
<dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
<dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 <optional>true</optional>  
 </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>

**application-dev.properties for sample-app1-service**

spring.application.name=sample-app1-service  
server.port=8081  
server.servlet.contextPath=/sample-app1-service

**application-dev.properties for sample-app2-service**

spring.application.name=sample-app2-service  
server.port=8082  
server.servlet.contextPath=/sample-app2-service

To access sample-app1-service: <http://localhost:8081/sample-app1-service/v1/info>

To access sample-app2-service: <http://localhost:8082/sample-app2-service/v1/info>

**Using API Gateway, you can access the application as** [**http://localhost:8080/api/sample-app2-service/v1/info**](http://localhost:8080/api/sample-app2-service/v1/info)