

Course 3 – Sprint 6: Create Single Entry point to Route the Request coming for different Microservices using Spring Cloud

Spring Cloud provides tools for developers to quickly build some of the common patterns in distributed systems (example: service discovery, circuit breakers etc.,).

Coordination of distributed systems leads to boiler plate patterns and using Spring Cloud developers can quickly stand-up services and applications that implement those patterns.

Spring Cloud Gateway aims to provide a simple, yet effective way to route to APIs and provide cross cutting concerns to them such as: security, monitoring/metrics, and resiliency.

Features

Spring Cloud focuses on providing good out of the box experience for typical use cases and extensibility mechanism to cover others.

- Built on Spring Framework 5, Project Reactor and Spring Boot 2.0
- Able to match routes on any request attribute
- Predicates and filters are specific to routes
- Circuit Breaker integration.
- Spring Cloud Discovery Client integration
- Easy to write Predicates and Filters
- Request Rate Limiting
- Path Rewriting

Why Spring Cloud?

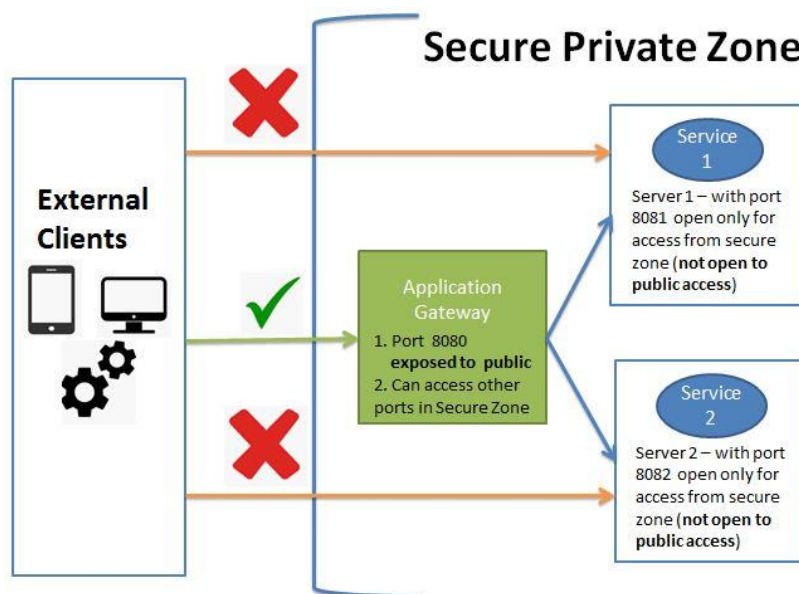
We can implement API Gateway using **Netflix ZUUL Component**, But the problem is it's a **Blocking API**.

A blocking gateway API makes use of as many threads as the number of incoming requests. So, this approach is more resource intensive. If no threads are available to process incoming request, then the request must wait in queue.

Spring Cloud Gateway is a non-blocking API. When using non-blocking API, a thread is always available to process the incoming request. These requests are then processed asynchronously in the background and once completed the response is returned. So, no incoming request never gets blocked when using Spring Cloud Gateway.

What is API Gateway? why is it needed?

API Gateway is single entry point for all microservices. No Microservices can directly connected by external clients. They connect through API Gateway only.



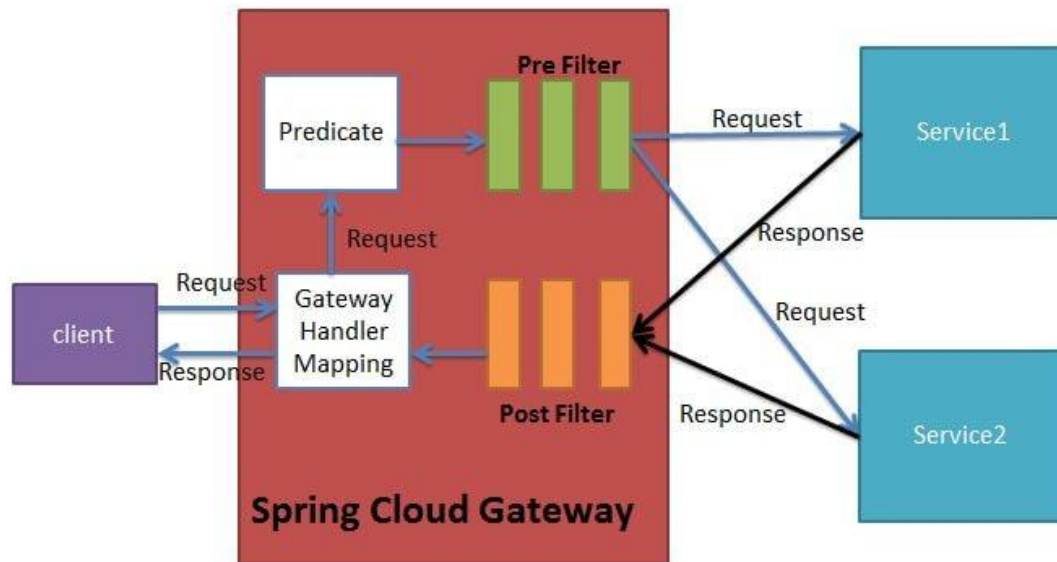
Courtesy: <https://www.javainuse.com/cloud17-1-min.JPG>

Spring Cloud Gateway Architecture

Spring Cloud Gateway is API Gateway implementation by Spring Cloud team on top of Spring reactive ecosystem. It consists of the following components.

1. Route – Basic component and has Id, destination URI, group of predicates and filters.
2. Predicate – They are used for writing some conditions to match the route (based on http request, cookies, parameters, and headers etc.,).

3. Filter – They are used for updating or modifying the request and response (like servlet filters).



Courtesy: <https://www.javainuse.com/cloud20-5-min.JPG>

When the client makes a request to the Spring Cloud Gateway, the Gateway Handler Mapping first checks if the request matches a route. This matching is done using the predicates. If it matches the predicate, then the request is sent to the filters.

For Demo:

<https://www.javainuse.com/spring/cloud-gateway>

For More in-depth understanding refer:

<https://lalverma.medium.com/spring-boot-microservices-api-gateway-e9dbcd4bb754>

<https://www.devglan.com/spring-cloud/spring-cloud-gateway>