A Simple API Gateway

Using the Springframework

Configuration Server

The Server Side - @EnableConfigServer

Configure application with **location** of the configuration data:

- `spring.cloud.config.server.git.uri:` [`https://github.com/dqromney/demo-gateway`](https://github.com/dqromney/demo-gateway)
- `spring.cloud.config.server.git.searchPaths: ConfigData`

The Client Side - word-server(s) and gateway

Configure application **name** and **server** location in **bootstrap.properties/yml:**

- `spring.application.name: subject`
- `Spring.cloud.config.uri: http://localhost:8001`

Spring Cloud Eureka - Service Discovery

Passive Service Discovery

- Having services register themselves and find others automatically

Spring Cloud Eureka Server - @EnableEurekaServer

- Holds registrations, shares information on other registrants
- Synchronizes itself with other servers

Spring Cloud Eureka Client - @EnableDiscoveryClient

- Connects to server to register, and obtain information on other clients.

API Gateway

What you get:

- Built for specific client needs (“facade”): Mobile, Web, Media, etc...
- Reduces # remote calls - gateway handles calls to services
- Routes calls to specific servers - Zuul
- Handles Security / SSO
- Handles caching
- Protocol Translation (TCP/Message protocols)
- Optimizes Calls / Link Expansion (HATEOAS)

Zuul - Routing and Filtering

- Zuul - JVM-based router and Load Balancer
  - Can be use for many API Gateway needs
  - Routing - Send request to real server
    - Reverse Proxy

Zuul Basic Usage

- Dependencies: (spring-cloud-starter-zuul)
  - Includes Ribbon and Hystrix (client load-balancing, automatic Circuit Breakers)
- Annotation: @EnableZuulProxy
- Default Behavior:
  - Eureka client ids become URIs
    - /subject routes to the “subject” service
    - /verb routes to the “verb” service
    - Etc.

Client can call: ...And Zuul will call:

$$\begin{tikzcd}
\text{localhost:8080/api/subject} \arrow[r] & \text{localhost:59334/} \\
\text{/api/verb} \arrow[r] & \text{localhost:54232/}
\end{tikzcd}$$

Is Zuul an API Gateway?

What is missing?

- Zuul is a tool for creating an API Gateway
  - Specifically routing

- What parts are missing?
  - Caching (Client cache, Spring’s Method Cache (@Cachable), Server cache)
  - Protocol translation (Spring Integration)
  - Resource Expansion / Link Resolution (HATEOAS)

A Simple API Gateway Demo

Using the Spring Framework

Links

- ICS Bitbucket (EWS) demo-gateway project: https://code.ldschurch.org/git/projects/EWS/repos/demo-gateway
- Netflix Open Source Software: http://netflix.github.io/#repo
- Spring Boot Reference: https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/
- Spring Cloud Reference: http://projects.spring.io/spring-cloud/
- Spring Framework Reference: http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/