**Dependency Need to Know**

<dependency> <groupId>org.springframework.boot</groupId><artifactId>spring-boot-configuration-processor</artifactId><optional>true</optional></dependency>

**Spring CLoud**

Spring Cloud provides tools for developers to build some of the **common patterns** in distributed systems quickly. For example, configuration management, service discovery, circuit breakers, intelligent routing, micro-proxy, a control bus, one-time tokens, global locks, leadership election, distributed sessions, cluster state.

**Spring Boot - Eureka Server**

<https://www.tutorialspoint.com/spring_boot/spring_boot_eureka_server.htm>

Eureka Server is an application that holds the information about all client-service applications. Every Micro service will register into the Eureka server and Eureka server knows all the client applications running on each port and IP address. Eureka Server is also known as Discovery Server.

Eureka Server comes with the bundle of Spring Cloud. For this, we need to develop the Eureka server and run it on the default port 8761.

Important Vedio

Eureka server Load Balanced <https://www.youtube.com/watch?v=ueyVjOnDHYQ>

**Using Feign REST Client for Service Invocation**

<https://www.javatpoint.com/using-feign-rest-client-for-service-invocation>

Feign

The Feign is a declarative web service (HTTP client) developed by **Netflix**. Its aim is to simplify the HTTP API clients. It is a Java to HTTP client binder. If you want to use Feign, create an interface, and annotate it. It provides pluggable annotation support, including Feign annotations and JAX-RS annotations.

It is a library for creating REST API clients. It makes web service clients easier. The developers can use declarative annotations to call the REST services instead of writing representative boilerplate code.

Spring Cloud OpenFeign

**Spring Cloud OpenFeign** provides OpenFeign integrations for Spring Boot apps through auto-configuration and binding to the Spring Environment. Without Feign, in Spring Boot application, we use **RestTemplate** to call the User service. To use the Feign, we need to add **spring-cloud-starter-openfeign** dependency in the pom.xml file.

When we work with microservices, there will be a lot of calls to other services. We need not to code like the previous one. Feign is a component that solves this problem. Feign makes it easy to invoke other microservices.

The other additional thing that Feign provides is:  it integrates with the **Ribbon**(client-side load balancing framework).

**Spring Cloud Sleuth**

<dependency>

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

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

</dependency>

<dependency>

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

<artifactId>spring-cloud-sleuth-zipkin</artifactId>

</dependency>

**Zuul Server**

<https://www.youtube.com/watch?v=cRhoODRCZAo>