服务的注册与发现（Eureka）

Spring Cloud简介

SpringCloud为开发人员提供了快速构建分布式系统的一些工具，包括配置管理、服务发现、断路器、路由、微代理、事件总线、全局锁、决策竞选、分布式会话等等。它运行环境简单，可以在开发人员的电脑上跑。另外说明SpringCloud是基于SpringBoot的，所以需要开发中对SpringBoot有一定的了解。另外还需要了解“微服务架构”的相关理论。

创建服务注册中心

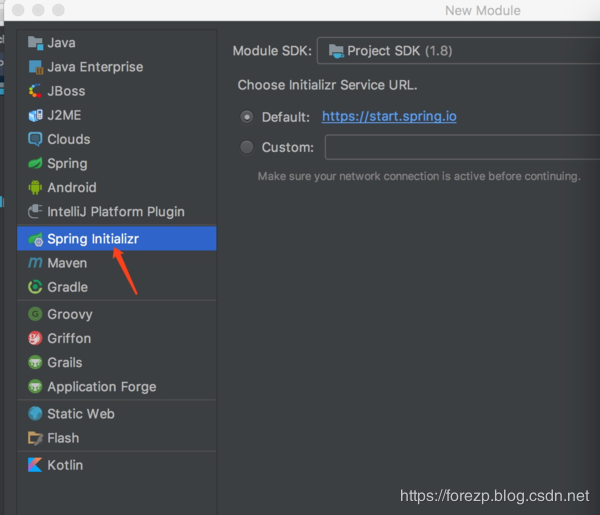
在这里，我们需要用的组件是SpringCloud Netfix的Eureka，Eureka是一个服务注册和发现模块。

1. 首先创建一个maven主工程
2. 然后创建2个model工程

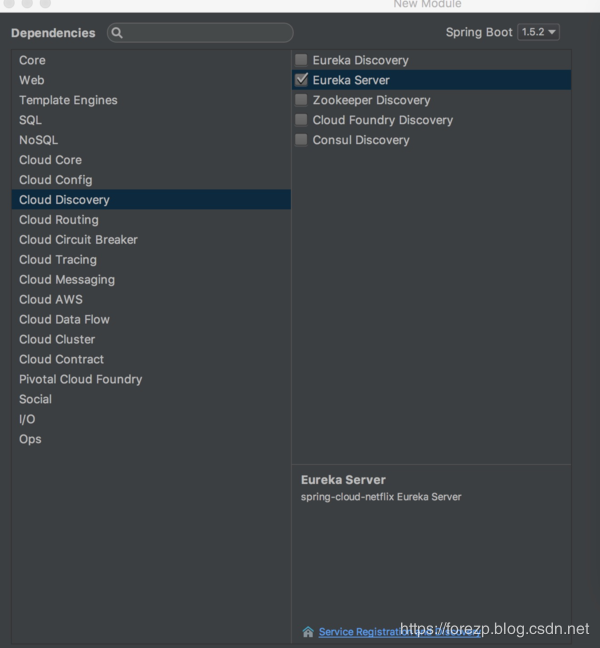
一个model工程作为服务注册中心，即Eureka Server；另一个作为Eureka Client。

下面详细说明创建过程：

右键工程->创建model->选择spring initializr



下一步 -> 选择cloud discovery -> eureka server，然后一直下一步就行了。



创建完成后工程的pom.xml文件如下：

【参考：服务注册中心/创建工程pom.txt】

启动一个服务注册中心，只需要一个注解@EnableEurekaServer，这个注解需要在springboot工程的启动application类上加上如下内容：

@EnableEurekaServer

@SpringBootApplication

public class EurekaserverApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaserverApplication.class, args);

}

}

eureka是一个高可用的组件，它没有后端缓存，每一个实例注册之后需要向注册中心发送心跳（因此可以在内存中完成），在默认情况下eureka server也是一个eureka client，必须要指定一个server。eureka server的配置文件application.yml:

server:

port: 8761

eureka:

instance:

hostname: localhost

client:

registerWithEureka: false

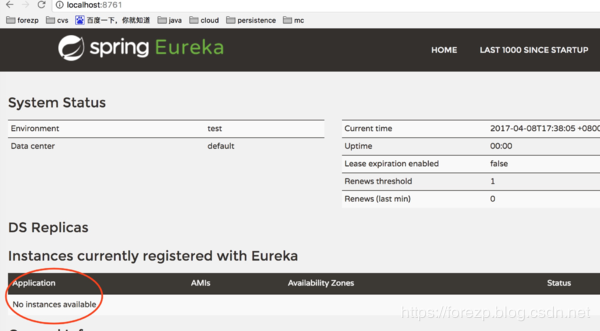
fetchRegistry: false

serviceUrl:

defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka/

通过eureka.client.registerWithEureka：false和fetchRegistry:false来表明自己是一个eureka server。

eureka server是有界面的，启动工程，打开浏览器访问：<http://localhost:8761>，界面如下:



因为没有注册服务，当然不可能有服务被发现了。

创建一个服务提供者（eureka client）

当client向server注册时，它会提供一些元数据，例如主机和端口，URL，主页等。Eureka server从每个client实例接收心跳消息。如果心跳超时，则通常将该实例从注册server中删除。

【参考：服务注册中心/创建服务提供者pom.txt】

通过注解@EnableEurekaClient表明自己是一个eureka client。

@SpringBootApplication

@EnableEurekaClient

@RestController

public class ServiceHiApplication {

public static void main(String[] args) {

SpringApplication.run(ServiceHiApplication.class, args);

}

@Value("${server.port}")

String port;

@RequestMapping("/hi")

public String home(@RequestParam String name) {

return "hi "+name+",i am from port:" +port;

}

}

仅仅@EnableEurekaClient是不够的，还需要在配置文件中注明自己的服务注册中心的地址，application.yml配置文件如下：

eureka:

client:

serviceUrl:

defaultZone: http://localhost:8761/eureka/

server:

port: 8762

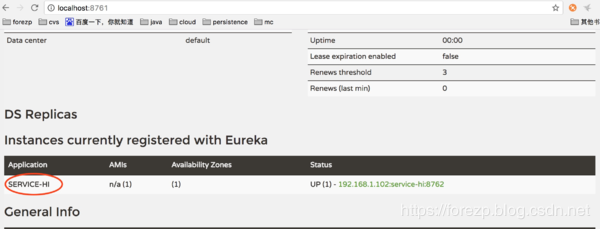
spring:

application:

name: service-hi

需要指明spring.application.name，这个很重要，这在以后的服务与服务之间相互调用一般都是根据这个name。

启动工程，打开<http://localhost:8761>，即eureka server的网址：



你会发现一个服务已经注册在服务中了，服务名为SERVICE-HI，端口为7862。这时打开<http://localhost:8762/hi?name=forezp>，你会在浏览器上看到：

hi forezp,i am from port:8762