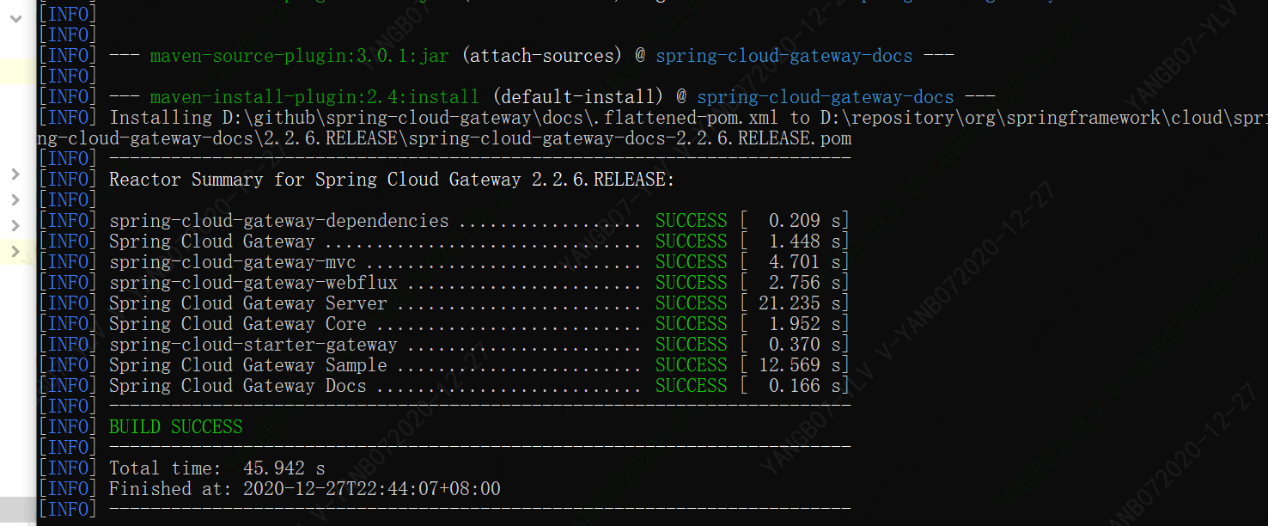
# springcloud-gateway-入门使用

## 1 fork spring cloud gateway到自己分支

链接：<https://github.com/50133142/spring-cloud-gateway.git>

建立自己的学习分支：

## 2：mvn install 成功



坑：需要注释掉 maven-checkstyle-plugin 才能install成功

## 3：运行gateway的demo一

在himly-demo工程下，新建gateway工程，服务注册到eureka上，

github链接：[https://github.com/50133142/hmily/tree/master/hmily-demo/hmily-demo-springcloud](https://github.com/50133142/hmily/tree/master/hmily-demo/hmily-demo-springcloud" \t "_blank)

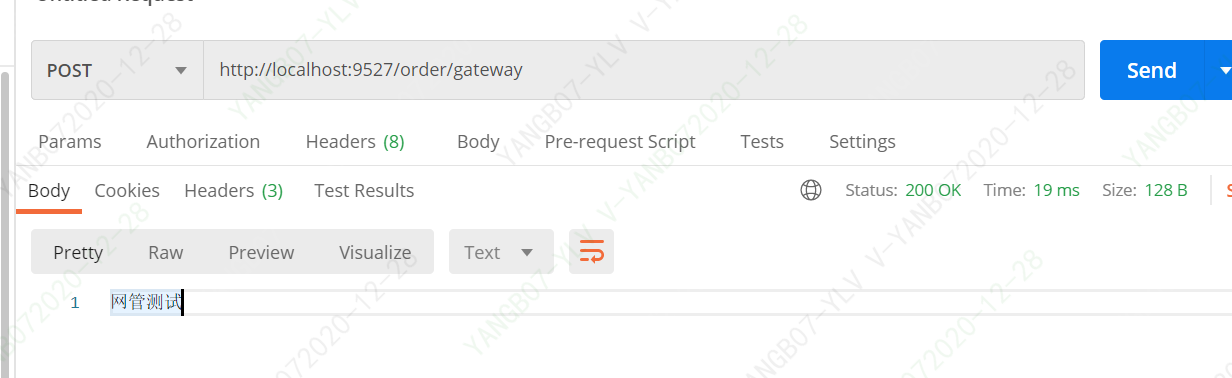
yml配置如下:

核心配置

* ID：编号，路由的唯一标识。
* URI：路由指向的目标 URI，即请求最终被转发的目的地。
* [Predicate](https://github.com/spring-cloud/spring-cloud-gateway/blob/master/spring-cloud-gateway-core/src/main/java/org/springframework/cloud/gateway/handler/AsyncPredicate.java)：谓语，作为路由的匹配条件。Gateway 内置了多种 Predicate 的[实现](https://github.com/spring-cloud/spring-cloud-gateway/blob/master/spring-cloud-gateway-core/src/main/java/org/springframework/cloud/gateway/handler/predicate/" \t "_blank)，提供了多种请求的匹配条件，比如说基于请求的 Path、Method 等等。
* [Filter](https://github.com/spring-cloud/spring-cloud-gateway/blob/master/spring-cloud-gateway-core/src/main/java/org/springframework/cloud/gateway/filter/GatewayFilter.java)：过滤器，对请求进行拦截，实现自定义的功能。Gateway 内置了多种 Filter 的[实现](https://github.com/spring-cloud/spring-cloud-gateway/blob/master/spring-cloud-gateway-core/src/main/java/org/springframework/cloud/gateway/filter/factory/" \t "_blank)，提供了多种请求的处理逻辑，比如说限流、熔断等等

post测试：

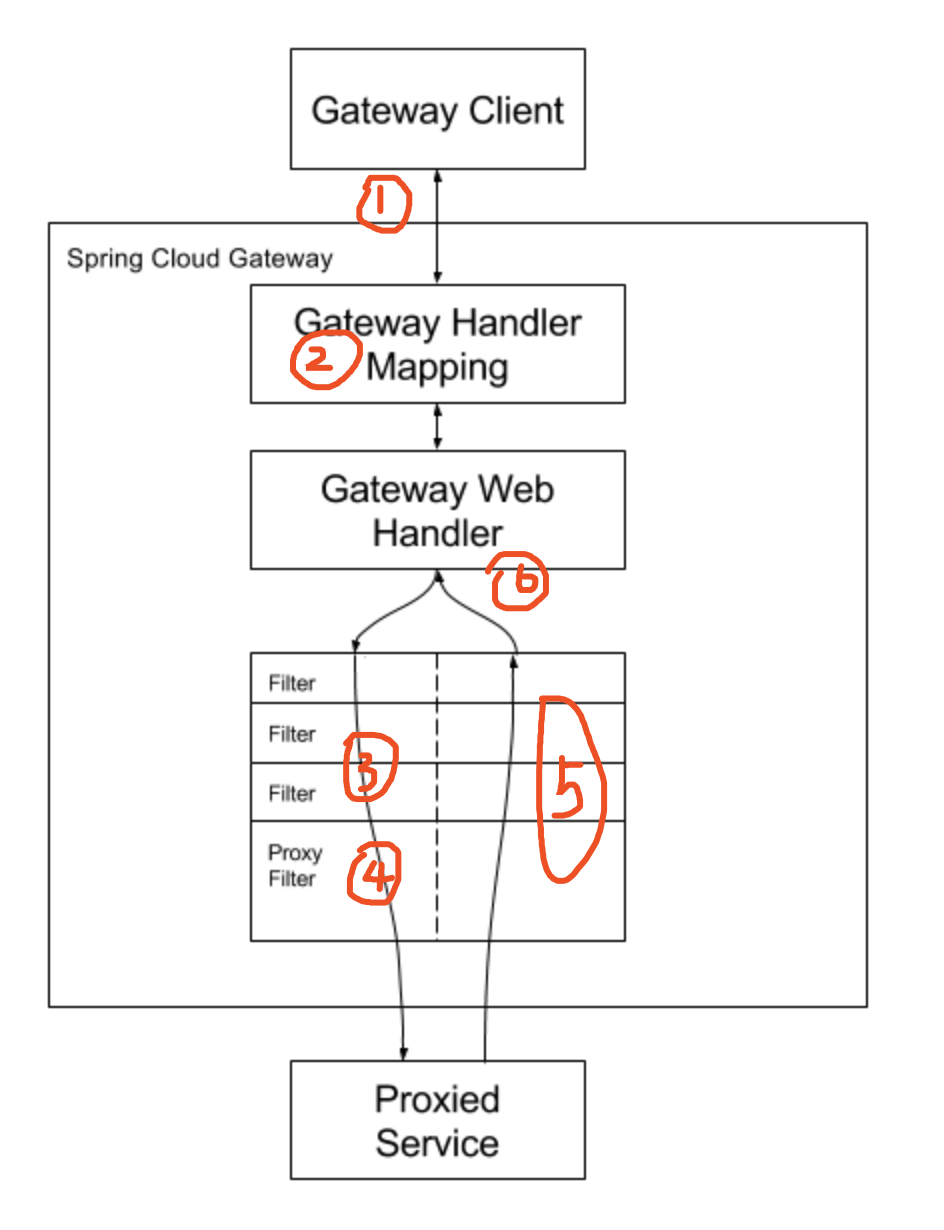
本地起eureka，order ，gateway三个服务



## 4：Gateway 的整体工作流程中的作用

（摘抄：<http://www.iocoder.cn/Spring-Cloud/Spring-Cloud-Gateway/?self>）

，如下图所示：



① Gateway 接收客户端请求。

② 请求与 Predicate 进行匹配，获得到对应的 Route。匹配成功后，才能继续往下执行。

③ 请求经过 Filter 过滤器链，执行前置（prev）处理逻辑。

例如说，修改请求头信息等。

④ 请求被 Proxy Filter 转发至目标 URI，并最终获得响应。

一般来说，目标 URI 是被代理的微服务，如果是在 Spring Cloud 架构中。

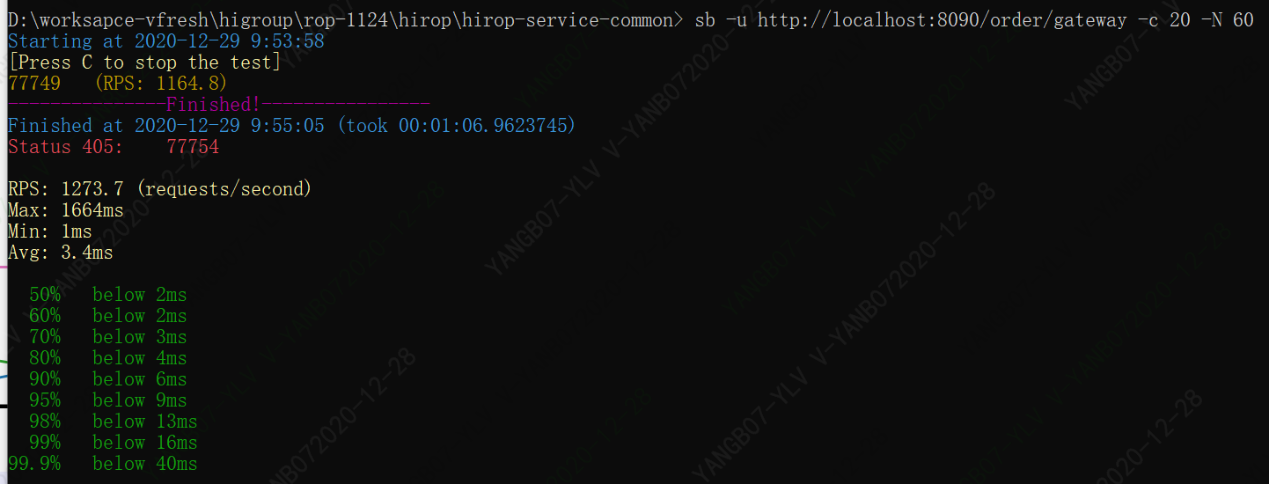
⑤ 响应经过 Filter 过滤器链，执行后置（post）处理逻辑。

⑥ Gateway 返回响应给客户端。

## 5 压测

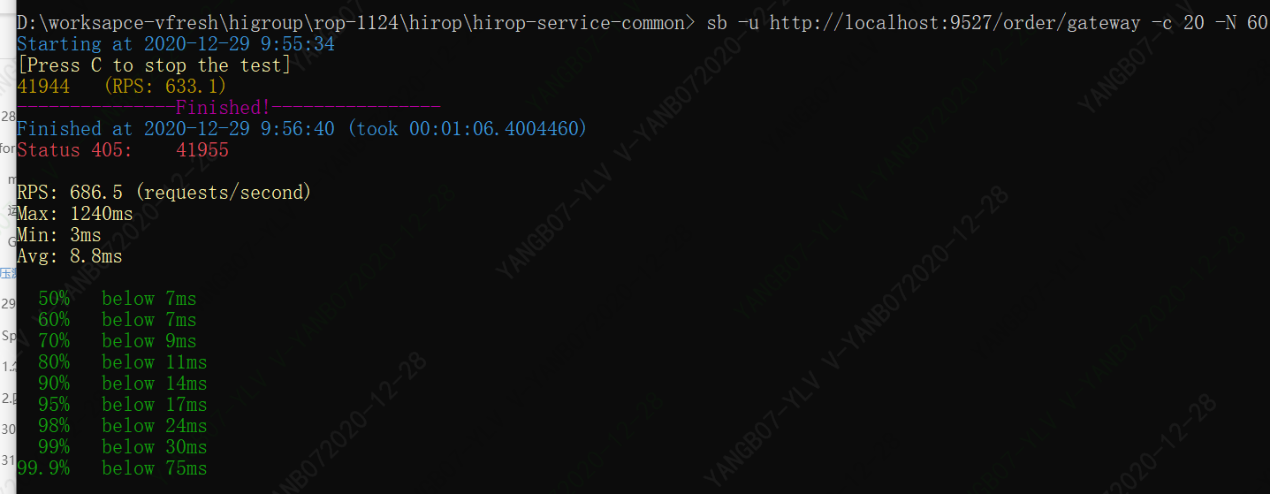
直连order服务压测 （在本地window环境）

命令：sb -u http://localhost:8090/order/gateway -c 20 -N 60



请求网关压测

命令：sb -u http://localhost:9527/order/gateway -c 20 -N 60



压测对比：请求网关是相对直连服务QPS性能的50%

总结：请求网关多出IO消耗，代理调用处理和从eureka获取访问ip信息