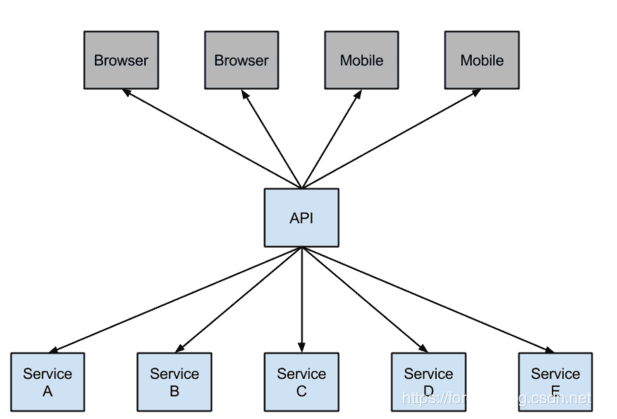
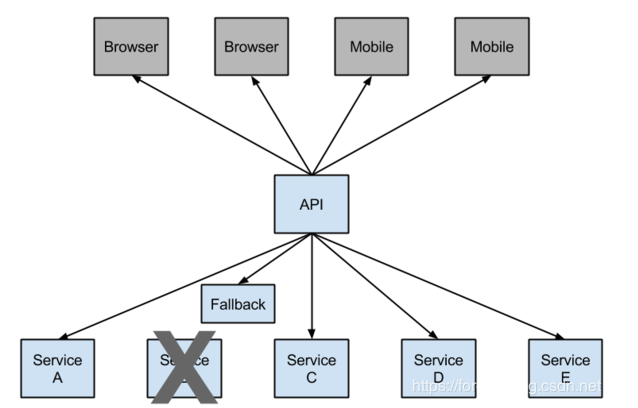
Hystrix：hes str

微服务中，服务间相互调用，如果单个服务出现问题，会造成该服务线程阻塞，若大量请求进入，最终导致的结果就是线程耗尽，当前服务瘫痪，由于服务间存在相互调用的依赖关系，最终造成整个微服务系统造成灾难性的严重后果，这就是服务故障的“雪崩”效应。

那么为了解决这个问题，业界提出了断路器模型。Netflix开源了Hystrix组件，实现了断路器模式，SpringCloud对这一组件进行了整合。



较底层的服务如果出现故障，会导致连锁故障。当对特定的服务的调用的不可用达到一个阀值（**Hystric 是5秒20次**） 断路器将会被打开。断路打开后，可用避免连锁故障，**fallback方法可以直接返回一个固定值，控制容器的线程阻塞。**



1、基于Ribbon进行

关键依赖

spring-boot-starter-parent

spring-cloud-starter-netflix-eureka-server

spring-boot-starter-web

**spring-cloud-starter-netflix-hystrix**

2、配置文件

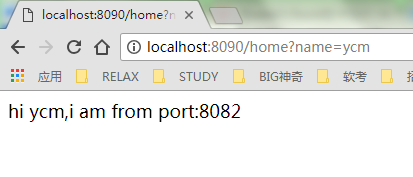
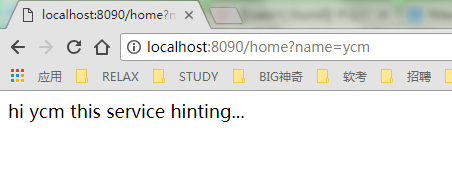


3、启动类



4、service，注意fallbackMethod值对应的方法，参数要跟接口的一致，否则找不到fallbackMethod指定的方法





2、基于Feign进行，Feign是自带断路器的，默认不开启

关键依赖

spring-boot-starter-parent

spring-cloud-starter-netflix-eureka-server

spring-boot-starter-web

**spring-cloud-starter-openfeign**

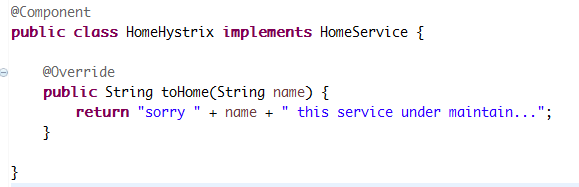
3、配置文件

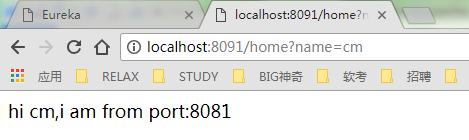
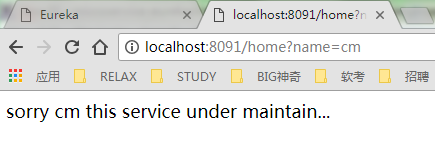


4、service



5、serviceimpl，要向IoC容器中注入





断路器仪表盘 – Hystrix Dashboard

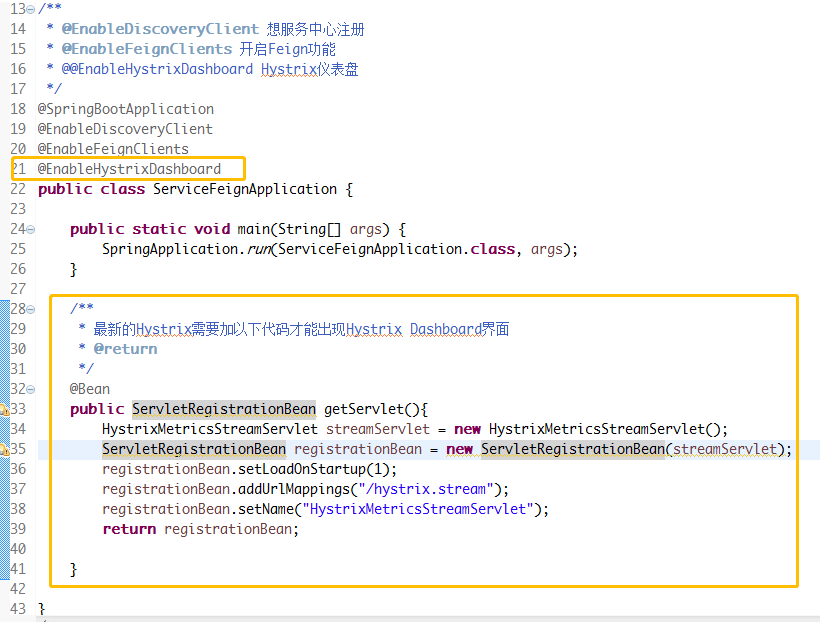
Feign与Ribbon搭建一样，这里以Feign为例

1、关键依赖

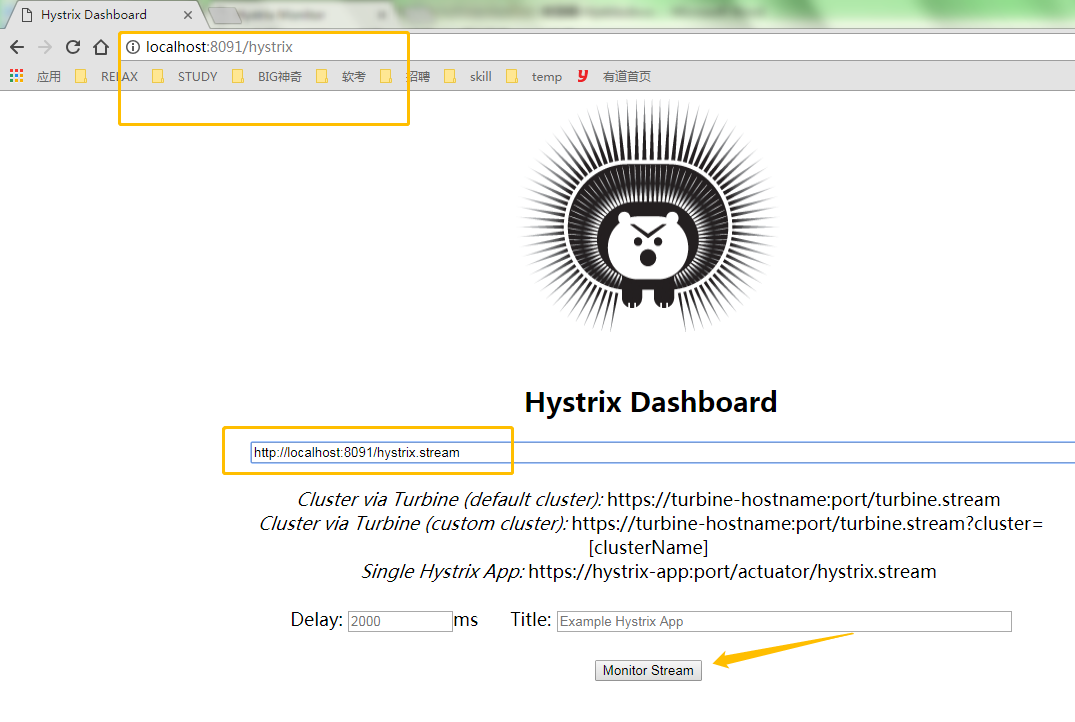
**spring-boot-starter-actuator**

**spring-cloud-starter-netflix-hystrix-dashboard**

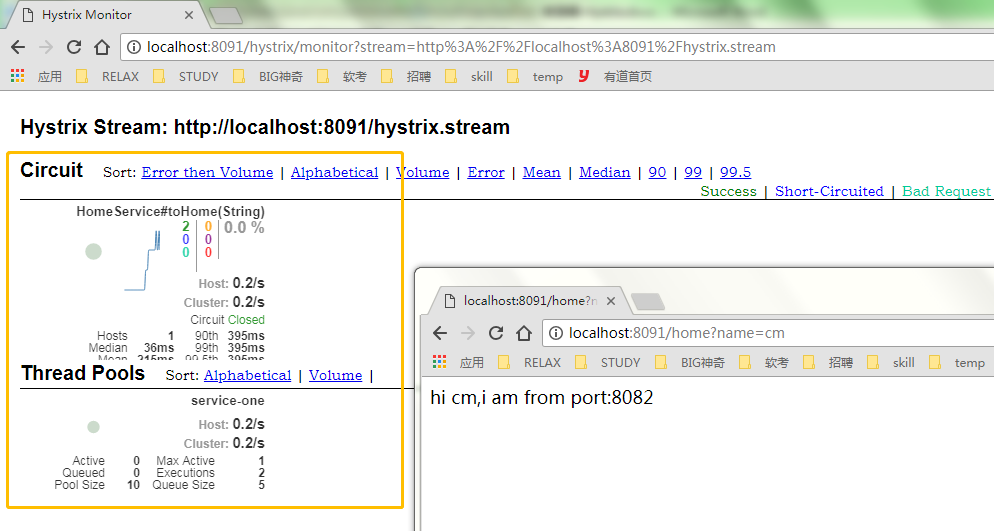
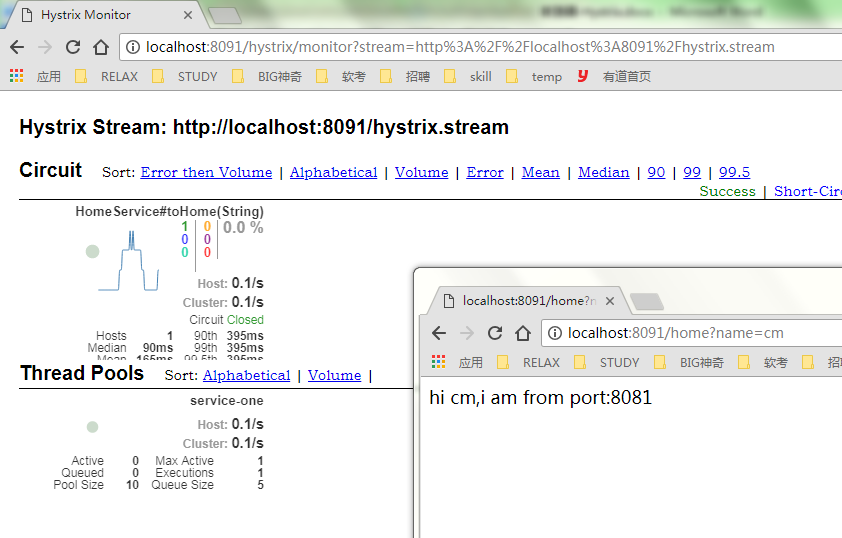
2、启动类



输入http://localhost:8091/hystrix



可以查看具体的请求信息



[**原文：史上最简单的SpringCloud教程 | 第四篇:断路器（Hystrix）**](https://blog.csdn.net/forezp/article/details/69934399)