监控仪表盘功能页面

<dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-netflix-hystrix</artifactId>  
</dependency>  
<dependency>  
 <groupId>org.springframework.cloud</groupId>  
 <artifactId>spring-cloud-starter-netflix-hystrix-dashboard</artifactId>  
</dependency>

@EnableHystrixDashboard

<http://localhost:9000/hystrix>

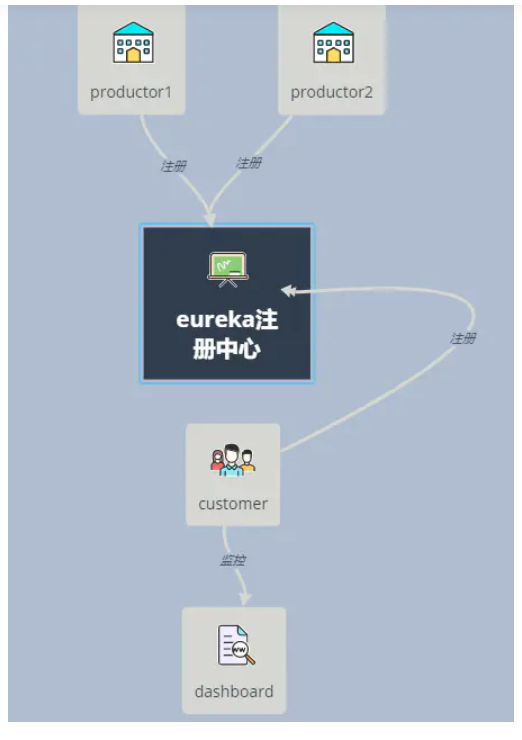
要监控的对象

@Bean  
public ServletRegistrationBean getServlet(){  
 HystrixMetricsStreamServlet streamServlet = new HystrixMetricsStreamServlet();  
 ServletRegistrationBean registrationBean = new ServletRegistrationBean(streamServlet);  
 registrationBean.setLoadOnStartup(1);  
 registrationBean.addUrlMappings("/actuator/hystrix.stream");  
 registrationBean.setName("HystrixMetricsStreamServlet");  
 return registrationBean;  
}

<http://localhost:9000/actuator/hystrix.stream>

把<http://localhost:9000/actuator/hystrix.stream>填到<http://localhost:9000/hystrix>

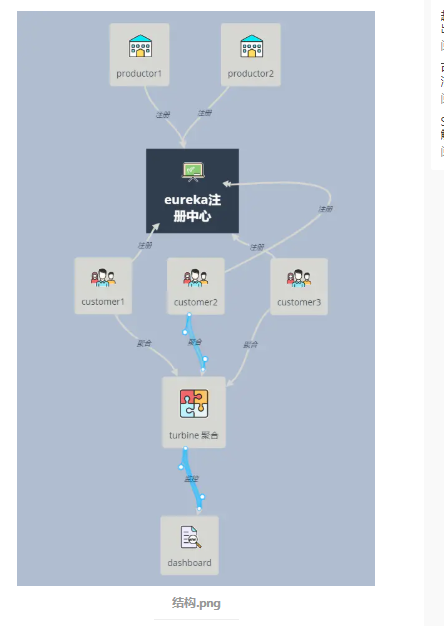
并调用controller就会有数值



一般单独建一个项目进行dashboard监控

<https://www.jianshu.com/p/59679a6838c8>

# **聚合监控Turbine**



就是将多个聚合起来监控

<https://www.jianshu.com/p/03863d219fbb>

开启熔断

方式1 独立hystrix 单个方法级别

@EnableCircuitBreaker

或者@EnableHystrix

在controller的接口方法上使用

@HystrixCommand(fallbackMethod ="loginByFeignFallback")

指定方法

public String loginByFeignFallback() {  
 return "loginByFeignFallback";  
}

方式2 与feign结合

开启feign.hystrix.enabled=**true**

@GetMapping("/loginByFeign")  
public String loginByFeign() {  
 return loginService.login();  
}

**LoginService接口代码：**

**@FeignClient(value = "eureka-client-provider", configuration = FeignConfiguration.class, fallback = LoginServiceFallback.class)//指定要调用的提供者应用名称，也就是服务名称**

**public interface LoginService {**

**@GetMapping("/service/login")**

**public String login();**

**}**

**其中**fallback指定LoginServiceFallback

**该类直接重写LoginService 出问题返回的结果**

Turbine集群监控

再建一个turbine监控工程 通过配置application.properties

turbine.app-config=eureka-client-consumer

turbine.combine-host-port=true

turbine.cluster-name-expression=new String("default")

说明要监控哪些服务，监控集群的名称default

这样就相当于把多个单个服务汇集在一块，启动turbine项目，浏览器输入

<http://localhost:20000/turbine.stream>可以看到请求情况的数据；

打开dashboard工程启动，在<http://localhost:10000/hystrix>中输入

<http://localhost:20000/turbine.stream>可以可视化看到情况；

<http://localhost:20000/turbine.stream>类比于 <http://localhost:9000/actuator/hystrix.stream>

注意点

1.被监控消费者需要向注册中心注册自己eureka.client.register-with-eureka=true

2、turbine.cluster-name-expression=new String("default") 这个default不能改（这是自己测试发现的）

总结

Dashboard监控工程提供了一个可视化页面 <http://localhost:10000/hystrix>

监控单个项目只需要消费者使用了hystrix, 并且需要管理ServletRegistrationBean，对应展示json格式的监控地址就是 <http://localhost:9000/actuator/hystrix.stream>，填入可视化页面即可可视化；

监控集群需要另开turbine监控工程，配置一下application.properties，并且有2个注意点，

然后启动项目，对应展示json格式的监控地址就是<http://localhost:20000/turbine.stream>，填入可视化页面即可可视化；