示例代码说明文档

功能实现	包含工程	工程启动说明	工程操作方式
eureka 分布式服务注册中心	eureka-server-center eureka-server-center2 eureka-client01	首先启动 eureka-server-center 和 eureka-server-center2，然后启动 eureka-client01（本示例需要配置 host 文件，用来模拟不同的服务器,具体说明见代码注释）	浏览器分别输入 locakhost:8080 和 localhost：8081 将会显示服务注册中心，服务注册中心的数据共同拷贝对方，且都有注册的服务客户端信息
Ribbon 实现负载均衡	eureka-server-simple-center simple-service-provide simple-service-provide2 ribbon-simple-loadbalanced	首先启动 eureka-server-simple-center，然后启动 simple-service-provide，simple-service-provide2，最后启动 ribbon-simple-loadbalanced	浏览器输入 localhost:8761,然后重新打开一个界面输入 localhost:8766/show,多次刷新，将会看到不同的结果,原因在于采用负载均衡方式获取两个服务的不同数据。
Feign 实现负载均衡	eureka-server-simple-center simple-service-provide simple-service-provide2 feign-simple-loadbalanced	首先启动 eureka-server-simple-center，然后启动 simple-service-provide，simple-service-provide2，最后启动 feign-simple-loadbalanced	浏览器输入 localhost:8761,然后重新打开一个界面输入 localhost:8769/feignshow,多次刷新，将会看到不同的结果原因在于采用负载均衡方式获取两个服务的不同数据。
Hystrix 断路器在 Ribbon 负载均衡下实现	eureka-server-simple-center simple-service-provide simple-service-provide2 hystrix-ribbon-simple-circuit-breaker	首先启动 eureka-server-simple-center，然后启动 simple-service-provide，simple-service-provide2，最后启动 hystrix-ribbon-simple-circuit-breaker	localhost:8761,然后重新打开一个界面输入 localhost:8766/show,多次刷新，将会看到不同的结果，关闭 simple-service-provide simple-service-provide2 将会显示断路处理结果，断路器能够有效的防止雪崩效应的发生
Hystrix 断路器在 Feign 负载均衡下实现	eureka-server-simple-center simple-service-provide simple-service-provide2 hystrix-feign-simple-circuit-breaker	首先启动 eureka-server-simple-center，然后启动 simple-service-provide，simple-service-provide，最后启动 hystrix-feign-simple-circuit-breaker	浏览器输 localhost:8761,然后重新打开一个界面输入 localhost:8769/feignshow, 多次刷新，将会看到不同的结果，关闭 simple-service-provide 和 simple-service-provide2 将会显示断路处理结果，断路器能够有效的防止雪崩效应的发生
断路器监控在 Ribbon 负载均衡下实现	eureka-server-simple-center simple-service-provide simple-service-provide2 hystrix-dashboard-ribbon	首先启动 eureka-server-simple-center，然后启动 simple-service-provide，simple-service-provide2，最后启动 hystrix-dashboard-ribbon	浏览器输 localhost:8761,然后重新打开一个界面输入 http://localhost:8001/hystrix，然后在打开的界面里面设置地址为 http://localhost:8001/hystrix.stream，delay 值设为 2000，设置标题，后点击 monitor Stream，进入监控面板界面 ，在上述完成的基础上，打开新的界面请求 http://localhost:8001/hystrixshow，此时监测面板里面就会有数据变化

断路器监控在在 Feign 负载均衡下实现	eureka-server-simple-center simple-service-provide simple-service-provide2 hystrix-dashborad-feign	首先启动 eureka-server-simple-center，然后启动 simple-service-provide，simple-service-provide，最后启动 hystrix-dashborad-feign	浏览器输 localhost:8761,然后重新打开一个界面输入 http://localhost:8002/hystrix，然后在打开的界面里面设置地址为 http://localhost:8002/hystrix.stream，delay 值设为 2000，设置标题，后点击 monitor Stream，进入监控面板界面 ，在上述完成的基础上，打开新的界面输入 http://localhost:8002/hystrixshow，此时监测面板里面就会有数据变化
Turbine 断路器聚合监控 ribbon 负载均衡下实现	eureka-server-simple-center simple-service-provide simple-service-provide2 hystrix-turbine-simple-service hystrix-turbine-simple-service2 hystrix-turbine-actuator	首先启动 eureka-server-simple-center 然后是 simple-service-provide simple-service-provide2，之后 hystrix-turbine-simple-service、hystrix-turbine-simple-service2，然后启动 hystrix-turbine-actuator	浏览器输 localhost:8761,然后重新打开一个界面输入 http://localhost:8099/hystrix，然后在打开的界面里面设置地址为 http://localhost:8099/turbine.stream，delay 值设为 2000，设置标题，后点击 monitor Stream，进入监控面板界面 ，在上述完成的基础上，打开新的界面输入 http://localhost:8005/simple 和 http://localhost:8004/simple 此时监测面板里面就会有数据变化
Zuul 路由网关转发实现	eureka-server-simple-center simple-service-provide simple-service-provide2 feign-simple-loadbalanced ribbon-simple-loadbalanced zuul-dispatch-center-server	首先启动 eureka-server-simple-center，然后启动 simple-service-provide，simple-service-provide2，之后启动 feign-simple-loadbalanced、ribbon-simple-loadbalanced，最后启动 zuul-dispatch-center-server	浏览器输 localhost:8761,然后分别打开一个界面输入 localhost:8011/api-a/show, localhost:8011/api-b/feignshow,将会分别转发到 service-ribbon（ribbon-simple-loadbalanced）和 service-feign（feign-simple-loadbalanced1)执行的功能。
Zuul 路由网关过滤实现	eureka-server-simple-center simple-service-provide simple-service-provide2 feign-simple-loadbalanced ribbon-simple-loadbalanced zuul-filter-center-server	首先启动 eureka-server-simple-center，然后启动 simple-service-provide，simple-service-provide2，之后启动 feign-simple-loadbalanced、ribbon-simple-loadbalanced，最后启动 zuul-filter-center-server	浏览器输 localhost:8761,然后分别打开一个界面输入 localhost:8012/api-a/show?token=123, localhost:8012/api-b/feignshow?token=123,将会分别转发到 service-ribbon（ribbon-simple-loadbalanced）和 service-feign（feign-simple-loadbalanced2)执行的功能.本示例做了一个过滤，即将不带 token 的请求都过滤掉。如果请求中不带 token 将不会被转发。
ServerConfig 分布式配置中心实现	eureka-server-simple-center server-config-center server-config-center2 server-config-client	首先启动 eureka-server-simple-center 然后 server-config-center server-config-center2，然后启动 server-config-client（本示例中需要使用 git 仓库）	浏览器输 localhost:8761,然后打开新的界面输入 localhost:8019/userinfo，就可以获得配置文件的内容

BUS 消息总线下的配置更新实现	eureka-server-simple-center server-config-center server-config-center2 bus-server-config-client bus-server-config-client2	首先启动 eureka-server-simple-center 然后 server-config-center、server-config-center2，之后启动 bus-server-config-client、bus-server-config-client2（本示例需要使用 RabbitMQ 和 git）	浏览器输 localhost:8761,然后打开新的界面输入 localhost:8023/userinfo 和 localhost：8066/userinfo 就可以看到配置文件的内容，修改配置文件，然后在发送一个 post 请求 http://localhost:8023/actuator/bus-refresh，重新请求之后的 userinfo 就会发生变化，而服务没有重新启动
zipkin 服务链路追踪实现	eureka-server-simple-center sleuth-zipkin-server sleuth-zipkin-client sleuth-zipkin-client2	首先启动 eureka-server-simple-center ，然后 sleuth-zipkin-server，之后启动 sleuth-zipkin-client、sleuth-zipkin-client2	首先打开浏览器输 localhost:8761，然后打开 http://localhost:9411/点击切换到依赖分析，然后分别请求 http://localhost:8055/show2 和 http://localhost:8057/show1，回到依赖分析页面刷新就可以看到服务之间的依赖关系
stream 消息驱动下的服务消费者与服务提供者通信	eureka-server-simple-center stream-provider-server stream-consumer-server	首先启动 eureka-server-simple-center ，然后启动 stream-provider-server ，最后启动 stream-consumer-server(本示例需要使用 RabbitMQ）	首先打开浏览器输 localhost:8761，然后请求 http://localhost:8401/produceMessage/hello1 ，然后就会在 stream-consumer-server 中打出日志。