

乐字节教育高级架构课程

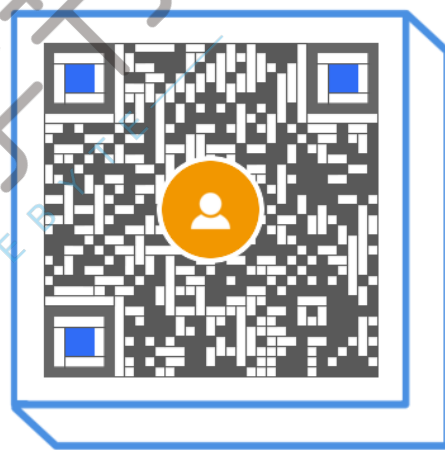
正所谓“授人以鱼不如授人以渔”，你们想要的 **Java 学习资料** 来啦！

不管你是学生，还是已经步入职场的同行，希望你们都要珍惜眼前的学习机会，奋斗没有终点，知识永不过时。

扫描下方二维码即可领取



乐字节晓啡

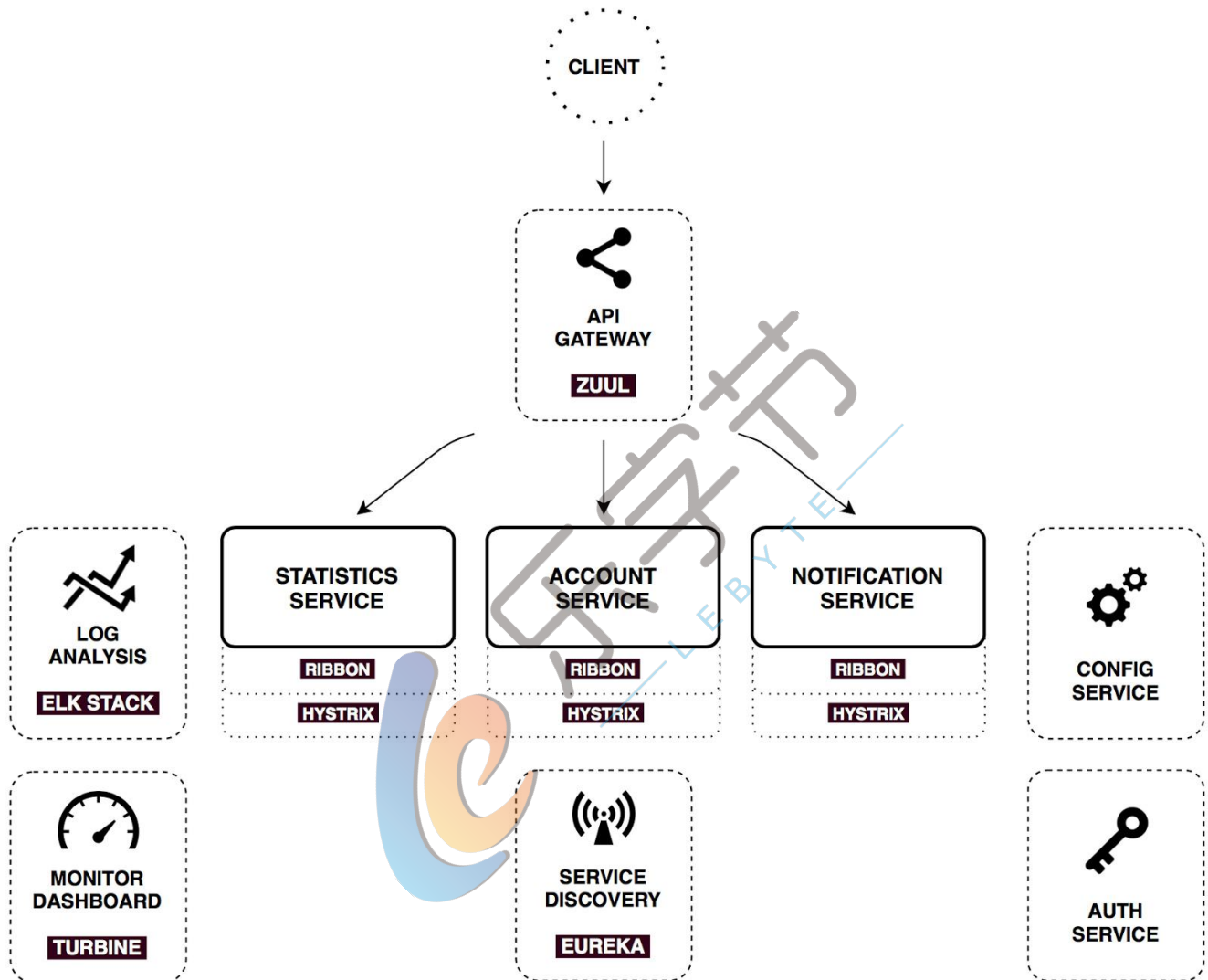


乐字节官方交流群



网关调优

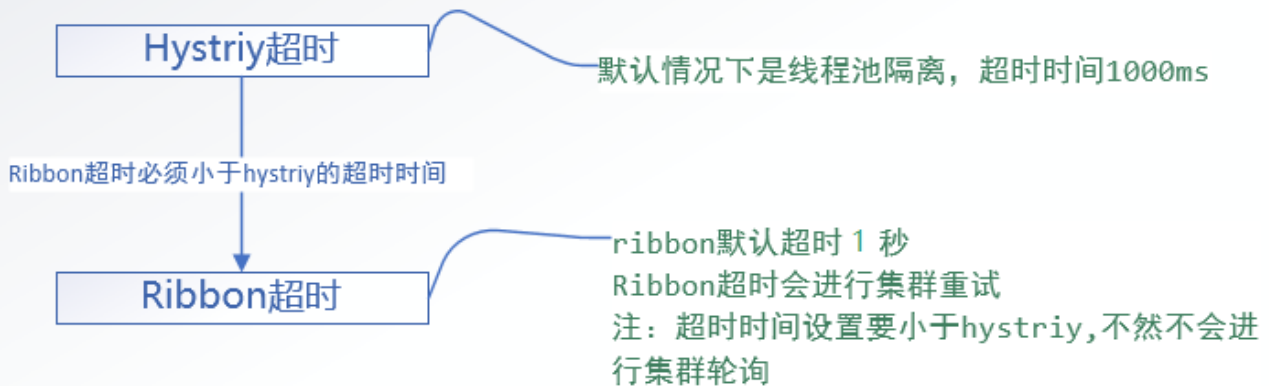
使用 Zuul 的 Spring Cloud 微服务架构图：



从上图中可以看出。整体请求逻辑还是比较复杂的，在没有 Zuul 网关的情况下，client 请求 service 的时候，也有请求超时的可能。那么当增加了 Zuul 网关的时候，请求超时的可能就更明显了。

当请求通过 Zuul 网关路由到服务，并等待服务返回响应，这个过程中 Zuul 也有超时控制。Zuul 的底层使用的是 Hystrix + Ribbon 来实现请求路由。

Zuul底层采用了ribbon和hystix来通信



Zuul 中的 Hystrix 内部使用线程池隔离机制提供请求路由实现，其默认的超时时长为 1000 毫秒。Ribbon 底层默认超时时长为 5000 毫秒。**如果 Hystrix 超时，直接返回超时异常。如果 Ribbon 超时，同时 Hystrix 未超时，Ribbon 会自动进行服务集群轮询重试，直到 Hystrix 超时为止。如果 Hystrix 超时时长小于 Ribbon 超时时长，Ribbon 不会进行服务集群轮询重试。**

配置文件

Zuul 中可配置的超时时长有两个位置：Hystrix 和 Ribbon。具体配置如下：

```

zuul:
  # 开启 zuul 网关重试
  retryable: true

  # Hystrix 超时时间设置
  hystrix:
    command:
      default:
        execution:
          isolation:
            thread:
              timeoutInMilliseconds: 10000 # 线程池隔离，默认超时时间 1000ms

  # Ribbon 超时时间设置：建议设置小于 Hystrix
  ribbon:
    ConnectTimeout: 5000 # 请求连接的超时时间：默认超时时间 1000ms
    ReadTimeout: 5000 # 请求处理的超时时间：默认超时时间 1000ms
    # 重试次数
    MaxAutoRetries: 1 # MaxAutoRetries 表示访问服务集群下原节点（同路径访问）
    MaxAutoRetriesNextServer: 1 # MaxAutoRetriesNextServer表示访问服务集群下其余节点
    (换台服务器)
    # Ribbon 开启重试
  
```

```
okToRetryOnAllOperations: true
```

添加依赖

Spring Cloud Netflix Zuul 网关重试机制需要使用 spring-retry 组件。

```
<!-- spring retry 依赖 -->
<dependency>
    <groupId>org.springframework.retry</groupId>
    <artifactId>spring-retry</artifactId>
</dependency>
```

启动类

启动类需要开启 `@EnableRetry` 重试注解。

```
package com.example;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.hystrix.dashboard.EnableHystrixDashboard;
import org.springframework.cloud.netflix.zuul.EnableZuulProxy;
import org.springframework.retry.annotation.EnableRetry;

@SpringBootApplication
// 开启 zuul 注解
@EnableZuulProxy
// 开启 EurekaClient 注解，目前版本如果配置了 Eureka 注册中心，默认会开启该注解
//@EnableEurekaClient
// 开启数据监控注解
@EnableHystrixDashboard
// 开启重试注解
@EnableRetry
public class ZuulServerApplication {

    public static void main(String[] args) {
        SpringApplication.run(ZuulServerApplication.class, args);
    }

}
```

模拟超时

商品服务模拟超时。

```
package com.example.controller;

import com.example.pojo.Product;
import com.example.service.ProductService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/product")
public class ProductController {

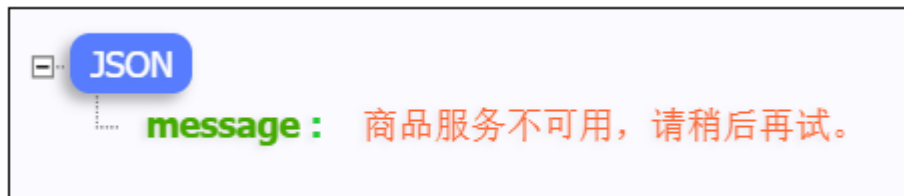
    @Autowired
    private ProductService productService;

    /**
     * 根据主键查询商品
     *
     * @param id
     * @return
     */
    @GetMapping("/{id}")
    public Product selectProductById(@PathVariable("id") Integer id) {
        // 模拟超时
        try {
            Thread.sleep(2000L);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        return productService.selectProductById(id);
    }
}
```

访问

配置前访问: <http://localhost:9000/product-service/product/1?token=abc123> 结果如下 (触发了网关服务降级) :





配置后访问：<http://localhost:9000/product-service/product/1?token=abc123> 结果如下：



思考：根据目前的配置如果访问订单服务会怎么样？

