我们这个课程，基于hystrix，如何来构建高可用的分布式系统的架构，项目实战

模拟真实业务的这么一个小型的项目，来全程贯穿，用这个项目中的业务场景去一个一个的讲解hystrix高可用的每个技术

纯讲hystrix，脱离实际的业务背景，听起来有点枯燥，大家学完了hystrix以后，可能没法完全感受到技术是如何融入我们的项目中的

大背景：电商网站，首页，商品详情页，搜索结果页，广告页，促销活动，购物车，订单系统，库存系统，物流系统

小背景：商品详情页，如何用最快的结果将商品数据填充到一个页面中，然后将页面显示出来

分布式系统：商品详情页，缓存服务，+底层源数据服务，商品信息服务，店铺信息服务，广告信息服务，推荐信息服务，综合起来组成一个分布式的系统

1、电商网站的商品详情页系统架构

（1）小型电商网站的商品详情页系统架构（不是我们要讲解的）

（2）大型电商网站的商品详情页系统架构

（3）页面模板

举个例子

将数据动态填充/渲染到一个html模板中，是什么意思呢？

<html>

<title>#{name}的页面</title>

<body>

商品的价格是：#{price}

商品的介绍：#{description}

</body>

</html>

上面这个就可以认为是一个页面模板，里面的很多内容是不确定的，#{name}，#{price}，#{description}，这都是一些模板脚本，不确定里面的值是什么？

将数据填充/渲染到html模板中，是什么意思呢？

{

"name": "iphone7 plus（玫瑰金+32G）",

"price": 5599.50

"description": "这个手机特别好用。。。。。。"

}

<html>

<title>iphone7 plus（玫瑰金+32G）的页面</title>

<body>

商品的价格是：5599.50

商品的介绍：这个手机特别好用。。。。。。

</body>

</html>

上面这个就是一份填充好数据的一个html页面

2、缓存服务

缓存服务，订阅一个MQ的消息变更，如果有消息变更的话，那么就会发送一个网络请求，调用一个底层的对应的源数据服务的接口，去获取变更后的数据

将获取到的变更后的数据填充到分布式的redis缓存中去

高可用这一块儿，最可能出现说可用性不高的情况，是什么呢？就是说，在接收到消息之后，可能在调用各种底层依赖服务的接口时，会遇到各种不稳定的情况

比如底层服务的接口调用超时，200ms，2s都没有返回; 底层服务的接口调用失败，比如说卡了500ms之后，返回一个报错

在分布式系统中，对于这种大量的底层依赖服务的调用，就可能会出现各种可用性的问题，一旦没有处理好的话

可能就会导致缓存服务自己本身会挂掉，或者故障掉，就会导致什么呢？不可以对外提供服务，严重情况下，甚至会导致说整个商品详情页显示不出来

缓存服务接收到变更消息后，去调用各个底层依赖服务时的高可用架构的实现

我们刚才讲解的整套大型电商网站的商品详情页的缓存架构，完整的那个流程，《亿级流量电商详情页系统的大型高并发与高可用缓存架构实战》

3、框架结构

围绕着缓存服务去拉取各种底层的源数据服务的数据，调用其接口时，可能出现的系统不可用的问题

从简

spring boot，微服务的非常快速，非常好用的技术框架，脱胎于spring，具体的东西就不讲解，直接带着大家上手搭建一个spring boot的框架

2个服务，缓存服务，商品服务，缓存服务依赖于商品服务

模拟各种商品服务可能接口调用时出现的各种问题，导致系统不可用的场景，然后用hystrix完整的各种技术点全部贯穿在里面

解决了一大堆设计业务背景下的系统不可用问题，hystrix整个技术体系，知识体系，也就讲解完了

消息队列，redis，咱们都不搞了

分布式系统，微服务，dubbo，不用dubbo，目前比较明显的一个趋势是，行业里，未来主要还是spring boot，spring cloud，主流的开源技术，去构建微服务的分布式系统

基于dubbo，官方很久之前就停止更新了，支持也不是太好

spring boot + http client + hystrix

《亿级流量电商详情页系统的大型高并发与高可用缓存架构实战》

龙果学院全网独家发布：<http://www.roncoo.com/>

观看地址：<http://www.roncoo.com/course/view/af7d501667fe4a19a60e9467e6d2b3d9>

1、亿级流量的电商网站的商品详情页系统架构

2、大型的企业级缓存架构，支撑高并发与高可用

3、几十万QPS的高并发+99.99%高可用+1T以上的海量数据+绝对数据安全的redis集群架构

4、高并发场景下的数据库+缓存双写一致性保障方案

5、大缓存的维度化拆分方案

6、基于双层nginx部署架构的缓存命中率提升方案

7、基于kafka+spring boot+ehcache+redis+nginx+lua的多级缓存架构

8、基于zookeeper的缓存并发更新安全保障方案

9、基于storm+zookeeper的大规模缓存预热解决方案

10、基于storm+zookeeper+nginx+lua的热点缓存自动降级与恢复解决方案

11、基于hystrix的高可用缓存服务架构

12、hystrix的进阶高可用架构方案、架构性能优化以及监控运维

13、基于hystrix的大规模缓存雪崩解决方案

14、高并发场景下的缓存穿透解决方案

15、高并发场景下的缓存失效解决方案