

云原生软件技术 Lab3 微服务拆分实践

张皓捷 胡明明 杨特 王浩睿

本次lab中，我们将对微服务拆分进行实践。考虑到同学们的时间和精力有限，本次lab不要求同学实际拆分微服务，只需要提交拆分的思路和设计即可。

本lab是小组作业。

0. 要拆分的单体项目

实验中要拆分的项目是一个在线考试系统。其主要功能有：

- 支持单选题、多选题、判断题
- 支持学生(student)、教师(teacher)、管理员(admin)三种角色
 - 学生：参加考试和查看我的考试
 - 教师：学生的所有权限+创建/编辑题目+创建/编辑考试
 - 管理员：教师的所有权限+管理用户

项目源代码地址：<https://github.com/lsgwr/spring-boot-online-exam/tree/master>

任务1 理解现有代码

在开始拆分之前，你需要先理解现有的单体应用。你需要通过阅读代码和运行应用来了解应用的功能和架构。

两个项目的README中都提供了如何运行应用的指导。这里就不再赘述。

你需要写一个报告，描述你对这个单体应用运行机制的理解。你可以描述应用的功能、架构、技术栈等等。此外，你还需要思考现有的架构在拆分成微服务应用时是否会遇到什么问题，以及如何解决这些问题。将这些内容写在一个文档中。

任务2 拆分微服务

在理解了现有代码之后，你需要开始拆分微服务。考虑到同学们的时间和精力有限，本次lab不要求同学实际拆分微服务，只需要提交拆分的思路和设计即可。将你的拆分思路和设计写在一个文档中。

你的拆分思路和拆分设计可以包括以下内容：

- 你计划将单体应用拆分为多少个微服务？每个服务的职责是什么？你是如何做出这个决定的？
- 你计划如何实现服务间的通讯？你会使用哪些技术栈？
- 你计划如何实现服务注册与发现？你会使用哪些技术栈？
- 每个服务的数据如何存储？每个服务的数据模型（可以理解为表结构）是什么样的？拆分后的分布式系统是否有数据一致性的问题？如何解决这些问题？
- 每个微服务中主要类的设计和UML类图
- 每个微服务的内部和外部接口设计，说明每个接口的功能和参数。对于关键的接口，可以给出伪代码实现
- 微服务间的调用关系图
- 某几个典型的用例(Use Case)的UML时序图，描述微服务间的调用关系，调用了哪些内部和外部接口

评分标准

名称	占比	备注
对现有代码的理解	40%	pdf文档
拆分思路和设计	60%	pdf文档

提交内容与方式

将任务1和任务2的文档导出为两个pdf文件，分别命名为 lab3-task1-group-X.pdf 和 lab3-task2-group-X.pdf。将这两个文件打包成一个zip文件，命名为 lab3-group-X.zip，其中X为你的小组编号。将这个zip文件提交到elearning上。

截止日期

2024年5月26日 23:59

附录

附录1：技术指南

- 服务注册与发现：**一些中间件如Consul、Zookeeper、Eureka和Nacos等，能够实现服务的注册与发现。此外，Kubernetes和Istio服务网格也具备这样的功能，它们将服务注册与发现的职责从应用层转移到了基础设施层。
- 服务间通讯：**微服务间的通讯主要有同步和异步两种方式。同步通讯主要包含两种流派，分别是基于HTTP的通讯和基于RPC的通讯。异步通讯可以使用消息队列来实现，例如RabbitMQ、Kafka等。由于本次实践的项目业务比较简单，同步通讯即可满足需求。基于HTTP的通讯可以使用Spring Cloud OpenFeign、RestTemplate等。基于RPC的通讯可以使用Dubbo、gRPC等。
- 分布式事务：**在微服务系统中，由于服务是分布式的，每个服务可能有自己的数据库，因此在处理跨服务的业务操作时，可能需要在多个数据库中进行操作。这就引入了分布式事务的需求。目前有一些解决方案可以实现分布式事务，例如Seata、TCC等。在本次实践中，我们可以使用Seata来实现分布式事务。
- 服务治理：**服务治理是微服务架构中熔断、降级、限流等。服务治理可以在应用层实现，典型的有Hystrix、Sentinel等。也可以在基础设施层实现，例如Istio等。
- 网关：**在本次lab中，网关主要的功能是将请求转发到对应的微服务。Spring Cloud Gateway、Zuul等都可以的一个重要组成部分，它包括服务注册与发现、服务间通讯、负载均衡、实现网关的功能。此外，Kubernetes中的Ingress和Istio中的Ingress Gateway也可以在基础设施层实现网关的功能。

附录2：Running Examples

有一些可运行的代码示例可以帮助你快速上手微服务的开发技术栈和框架。

- 使用Spring Cloud OpenFeign实现微服务间的通讯：[feign-demo](#)
- Spring Cloud OpenFeign与Seata集成：[seata-demo](#)的seata-integration分支、[seata官方文档](#)
- Dubbo与Seata集成：[seata-samples/at-samples](#)、[seata官方文档](#)
- 使用Spring Cloud Gateway实现网关：lab2的ase-gateway代码