云南大学软件学院期末课程报告

Final Course Report

School of Software, Yunnan University

**个人成绩**

学　　期： 2023年春季学期

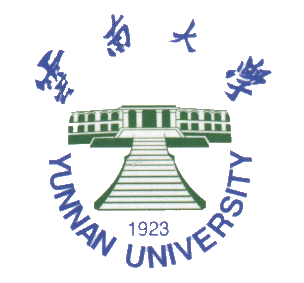
课程名称: 软件服务工程

姓 名： 陈乐

学 号： 20201120309

成 绩:

承担分工： 后端微服务环境搭建及部分接口的编写，并设计微服务间调用的限流、熔断，约占总工作量的35%

云南大学软件学院期末课程报告

Final Course Report

School of Software, Yunnan University

**个人成绩**

学　　期： 2023年春季学期

课程名称: 软件服务工程

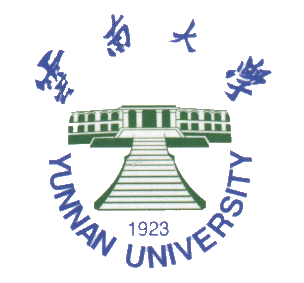
姓 名： 陈远豪

学 号： 20201120498

成 绩:

承担分工： 后端代码中大部分功能接口的编写，并设计微服务间调用的降级、灰度发布，

约占总工作量的35%

云南大学软件学院期末课程报告

Final Course Report

School of Software, Yunnan University

**个人成绩**

学　　期： 2023年春季学期

课程名称: 软件服务工程

姓 名： 马金来

学 号： 20201120480

成 绩:

承担分工： 项目主要前端代码的编写，并与后端接口进行联调，约占总工作量的30%

目录

项目地址：<https://github.com/chenle2002/soft-project-mooc>

**git成员：**

**账号：[chenle2002](https://github.com/chenle2002) 陈乐 20201120309**

**账号：Chenyh20040619 陈远豪 20201120498  
账号：Surpmen  马金来 20201120480**

[一、 背景及业务分析 5](#_Toc20126)

[1.1 项目简介 5](#_Toc26612)

[1.2 背景分析 5](#_Toc3726)

[1.3 业务分析 6](#_Toc25054)

[1.3.1 管理员功能 6](#_Toc30145)

[1.3.2 用户功能 7](#_Toc10002)

[二、 领域分析及服务设计 7](#_Toc18664)

[2.1业务流程 7](#_Toc3063)

[2.1.1 管理员新增课程 7](#_Toc11074)

[2.1.2 用户主要功能 9](#_Toc11628)

[2.2 API协作图 10](#_Toc31361)

[2.2.1 主页面信息展示 10](#_Toc32117)

[2.2.2 课程详情信息展示 11](#_Toc18226)

[三、 微服务节点划分及设计 12](#_Toc32417)

[3.1 微服务逻辑节点划分 12](#_Toc13583)

[3.2 微服务物理节点划分 12](#_Toc27524)

[3.2.1 微服务所用组件 12](#_Toc11035)

[3.2.2 网关服务 13](#_Toc13030)

[3.2.3 核心服务 13](#_Toc14723)

[3.2.4 低负载服务 13](#_Toc13986)

[3.3 节点API规划 13](#_Toc2961)

[3.3.1 service-course节点 13](#_Toc27211)

[3.3.2 service-member节点 14](#_Toc11835)

[3.3.3 service-sort节点 15](#_Toc6148)

[3.3.4 service-oss节点 16](#_Toc30731)

[四、 环境搭建及部署结构 17](#_Toc1344)

[4.1 部署环境 17](#_Toc1401)

[4.2 项目结构 18](#_Toc15926)

[4.3 代码展示 19](#_Toc5361)

[五、 观测及治理 22](#_Toc18007)

[5.1 服务限流 22](#_Toc10260)

[5.2 服务发布与负载均衡 24](#_Toc7427)

[5.3 服务熔断与降级 25](#_Toc15350)

[5.4 灰度发布 28](#_Toc16715)

[六、项目总结 30](#_Toc14445)

# 背景及业务分析

## 1.1 项目简介

项目为大创项目 法途——法学在线学习平台的代码部分。项目用于法律相关课程的学习、法学相关资料的查询以及法律相关课程的考核。以帮助法学院进行课程教学。

用户在注册账号后，可以登录用户端查看法律相关课程，下载其课程资料，并且观看其课程视频，最终通过该课程的考核从而完成该课程的学习，并且可以在学习过程中发表自己的评论。

## 1.2 背景分析

法学在当今社会扮演着越来越重要的角色，不仅具有维护社会正义和秩序的作用，而且对于维护国家的长治久安也有着至关重要的作用。但是，法学作为一门复杂而且高深的学科，其学习和教学都面临着许多困难和挑战。特别是在课程内容，教学方法和教材资源等方面，还存在许多问题。因此，为了提高法学教学的质量和效率，我们需要借助现代技术手段，建立一个高效便捷的法学在线学习平台。

我们的法学在线学习平台旨在为学生提供一个全面而深入的学习环境，使他们能够更好地理解和掌握法学知识，增强对法律的理解和应用能力。此外，该平台还将作为法学院开展教学和科研的重要平台，提供多元化的教学资源，帮助学生充分发展自己的知识和技能，成为具有良好法律素质的复合型人才。

## 1.3 业务分析

### 1.3.1 管理员功能

（1）课程管理

课程结构分为课程、目录、章节三部分，管理员可以对课程的树形结构进行创建课程、新增章节、添加课程资料、课程描述等操作

（2）章节管理

章节信息在数据库上独立于课程结构，管理员可以对课程各章节的视频信息、章节名称等数据进行增删查改等操作

1. 分类管理

为方便用户的查看，更好地展示课程信息，对各课程进行分类，对课程进行分类，依照不同分类展示具体地课程信息

1. 教师信息管理

各个课程应有其上课教师，管理员可以对教师信息进行增删查改，并为课程选择其上课教师

1. 评论信息管理

管理员可以对用户在课程下地评论进行管理，如果用户地评论不符合要求，则对其进行删除地评论管理操作

1. 考试功能

管理员可以为各个课程选择其考试内容，供用户在课程学习之后进行考试行为，并得出用户的考试成绩

### 1.3.2 用户功能

（1）课程信息查看

用户登录后可以查看用户端主页的课程分类列表，并查看该课程的详情信息、教师信息等数据，并可以下载该课程的教学资料

（2）课程视频播放

用户选择课程后，可以观看该课程章节的各个视频

1. 课程评论

用户在查看课程信息、观看该课程的视频后，可以进入该评论区发表自己的评论；也可以为别人的评论点赞

（4）课程考核

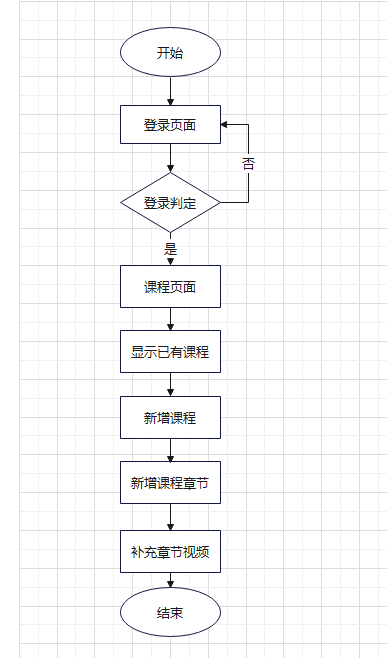
在用户观看完视频后，可以在该课程考核功能中完成该课程的试卷，完成试卷后可以得到这门课程的成绩

# 领域分析及服务设计

## 2.1业务流程

### 2.1.1 管理员新增课程

管理员端多为对数据的单一管理，跨服务的业务逻辑较少。下图管理员进行管理的业务流程图：

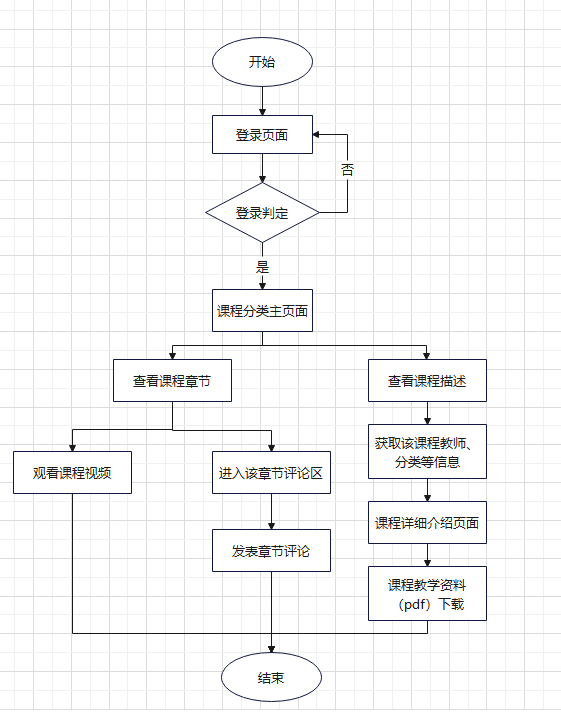


下图为管理员进行管理的课程树结构：



### 2.1.2 用户主要功能

用户可以在用户端登录后完成如课程视频查看、课程评论、课程课件下载等功能，具体业务流程图如下所示：

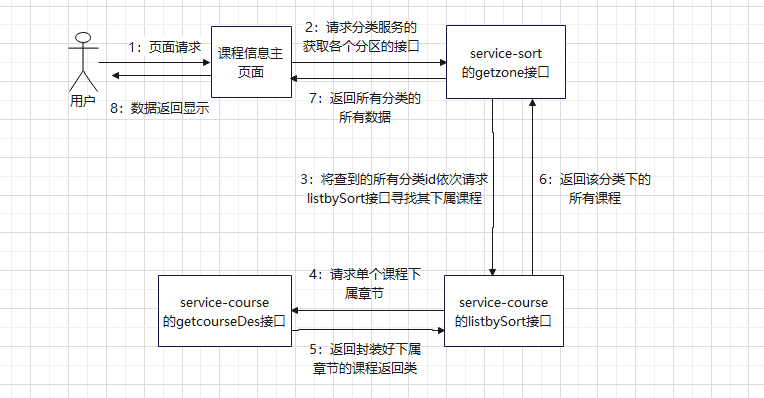


课程分类主页面（按照分类将课程进行展示）:



## 2.2 API协作图

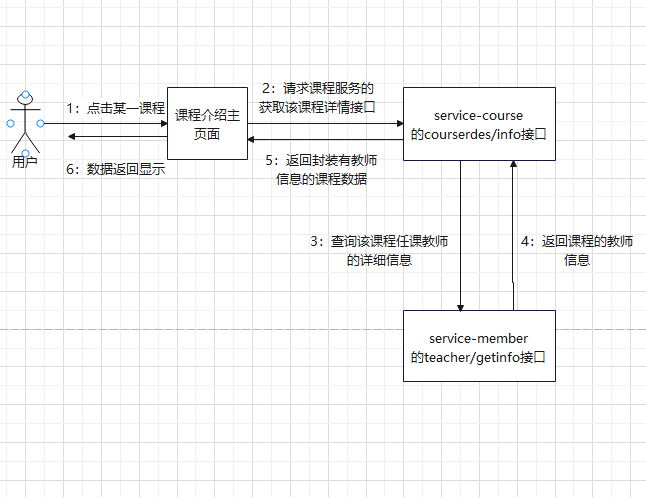
### 2.2.1 主页面信息展示



在次API调用中后端返回的数据如下所示：



### 2.2.2 课程详情信息展示



课程信息展示的页面如下所示：



# 微服务节点划分及设计

## 3.1 微服务逻辑节点划分

（1）api-gateway：根据请求前缀的不同将请求分发到不同微服务中

（2）service-course：课程信息微服务。包括课程数据、章节视频数据、课程有关信息的微服务，在获取教师信息时调用人员管理服务，上传数据调用第三方微服务

（3）service-member：人员管理服务。包括教师信息和用户信息，用于教师和用户信息的增删查改，查询老师教授的课程时调用其他微服务

（4）service-oss：第三方调用微服务。包括固定数据信息的获取以及阿里云OSS接口的的调用，该服务主要用于被调用

（5）service-sort：分类管理服务。用于管理课程分类信息，查找某一分类下的课程及其教师时调用其他微服务

（6）service-admin：基于开源模板生成的微服务。为默认微服务，用于客户端管理，除基本功能外，各种具体功能调用其他微服务

## 3.2 微服务物理节点划分

### 3.2.1 微服务所用组件

mysql、nacos、sentinel等项目所用组件使用docker直接部署到服务器上每个组件一个物理节点

### 3.2.2 网关服务

使用docker部署，网关服务用一个docker容器，设置一个物理节点

### 3.2.3 核心服务

service-course、service-member等，每个服务部署三个以上节点以防止应对请求，并用来演示项目负载均衡

### 3.2.4 低负载服务

此类服务使用频率较低，压力小一个服务两个物理节点

## 3.3 节点API规划

### 3.3.1 service-course节点

主要API:

1、课程表的增删查改

2、获取所有信息封装为树形结构

3、根据课程id寻找子节点、父节点

4、根据名称、id、节点等参数获取课程的章节信息

5、oss服务调用接口

**service-course节点的API图：**



### 3.3.2 service-member节点

主要API:

1、用户表、教师表的增删查改

2、用户的登录注册

3、根据课程id获取教授该课程的老师信息

4、根据教师id返回该教师信息

**service-member节点的API图：**



### 3.3.3 service-sort节点

主要API:

1、分类表的增删查改

2、根据分类id进行分类信息获取

3、将用户主页的所有课程信息按照不同分类进行结构封装

4、用户主页轮播图数据返回

**service-sort节点的API图：**



### 3.3.4 service-oss节点

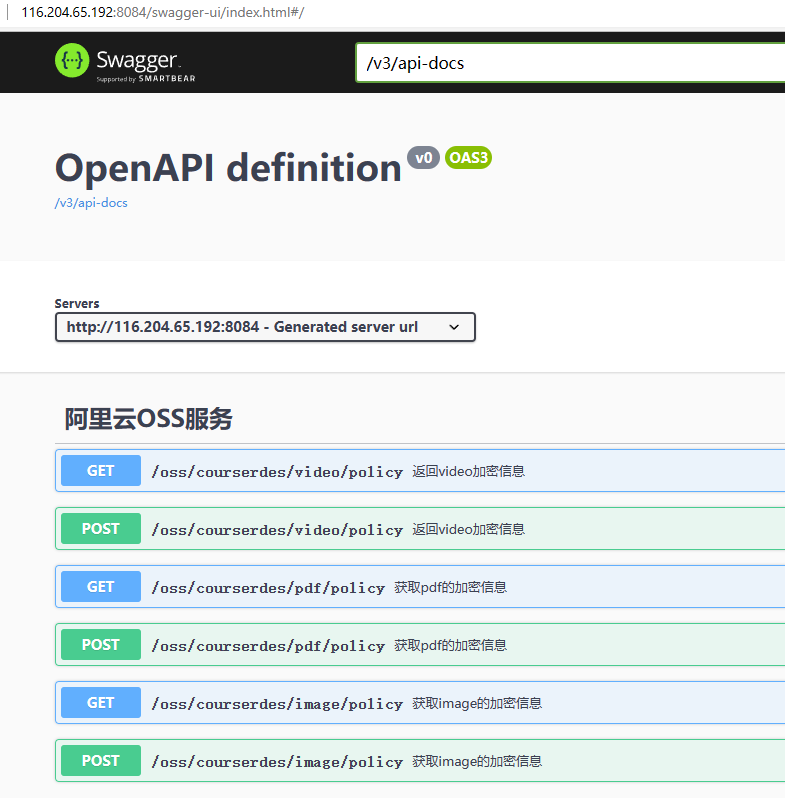
主要API:

1、video签名返回前端，前端直传

2、pdf签名返回前端，前端直传

3、image签名返回前端，前端直传

**service-oss节点的API图：**



# 环境搭建及部署结构

## 4.1 部署环境

nacos、mysql、sentinel均使用docker在服务器上部署较新版本

微服务：将各个微服务打成jar包后，进行镜像制作并上传DockerHub

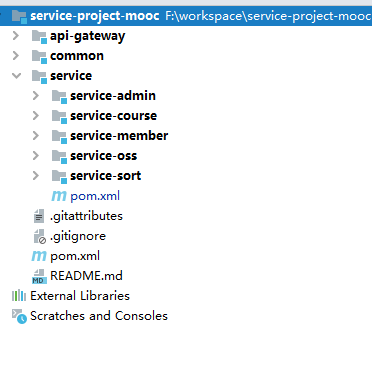
使用Dockerfile进行微服务快速部署

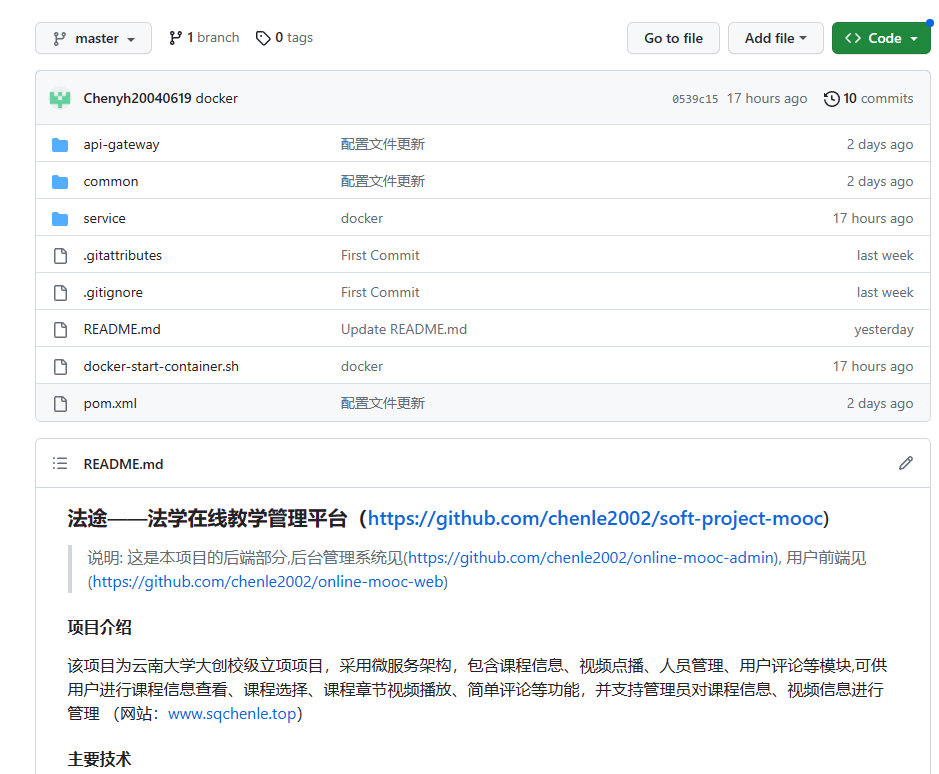
**下图为某Dockerfile：**



## 4.2 项目结构

**项目地址：<https://github.com/chenle2002/soft-project-mooc>**





## 4.3 代码展示

**树形结构封装：**

****

**网关路由规则：**

****

**返回各个分类下的课程集合：**

****

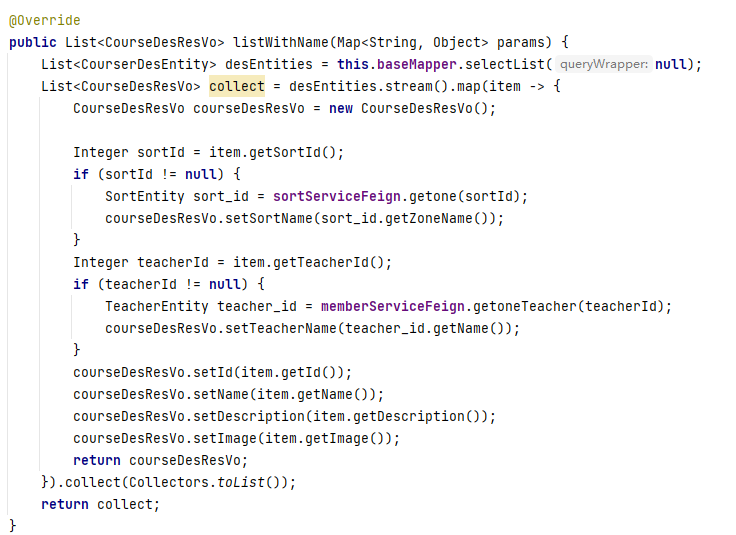
**oss上传内容封装：**

****

**带有事务的课程信息更新：**

****

**根据筛选条件返回课程章节集合：**

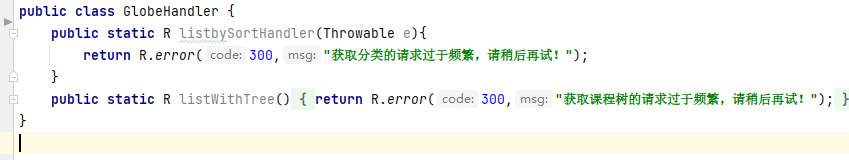
****

# 观测及治理

## 5.1 服务限流

获取课程树结构运算比较复杂，多次请求对服务器压力较大，对获取课程树结构的接口进行限流，使用@SentinelResource注解，标明fallback方法





对该接口使用sentinel设置流控规则，每秒请求超过3次则执行fallback方法



使用Postman发送30个请求，从第四个请求时出现限流拒绝策略



## 5.2 服务发布与负载均衡

测试接口：获取单个课程详细信息时会访问Member的获取教师详情的接口，在获取教师详情的接口返回的信息包括机器号



服务Member（version2.0）发布前，Nacos中只有一个member实例



此时请求课程详情信息接口时调用的教师详情接口返回的机器号machineCode均为‘111’

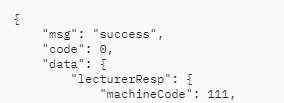


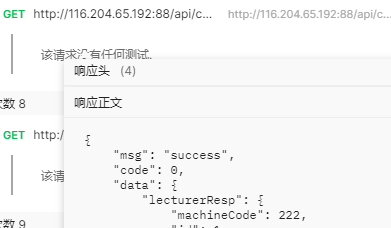


此时使用docker发布服务Member（version2.0），Nacos中有两个member服务实例



此时请求课程详情信息接口时调用的教师详情接口返回的机器号machineCode在‘111’和‘222’间进行随机访问





## 5.3 服务熔断与降级

服务熔断:当服务A调⽤的某个服务B不可⽤时，上游服务A为了保证⾃⼰不受影响，从⽽不再调⽤服务B，直接返回⼀个结果，减轻服务A和服务B的压⼒，直到服务B恢复

服务降级:当发现系统压⼒过载时，可以通过关闭某个服务，或限流某个服务来减轻系统压⼒，这就是服务降级

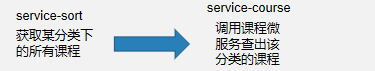
相同点：

1. 都是为了防⽌系统崩溃

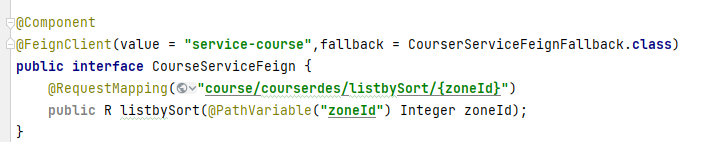
2. 都让⽤户体验到某些功能暂时不可⽤

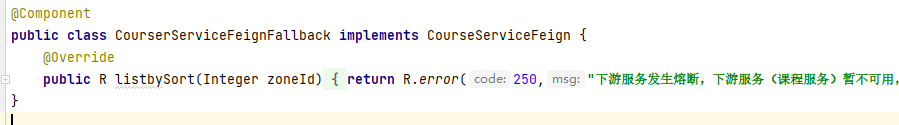
不同点：熔断是下游服务故障触发的，降级是为了降低系统负载

熔断测试接口：在输入某个分类时，分类服务调用课程服务，获得该分类下的所有课程。如：分类服务向课程服务发送一个“大学课程”分类，课程服务查出该分类下的所有课程并封装返回



在service-sort中使用openfeign进行降级



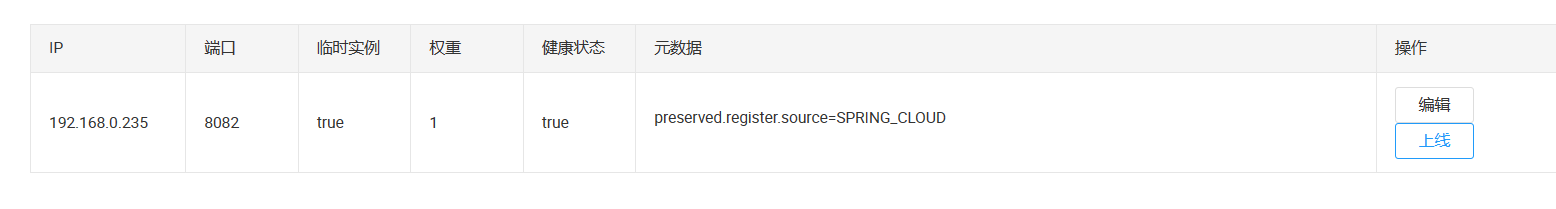


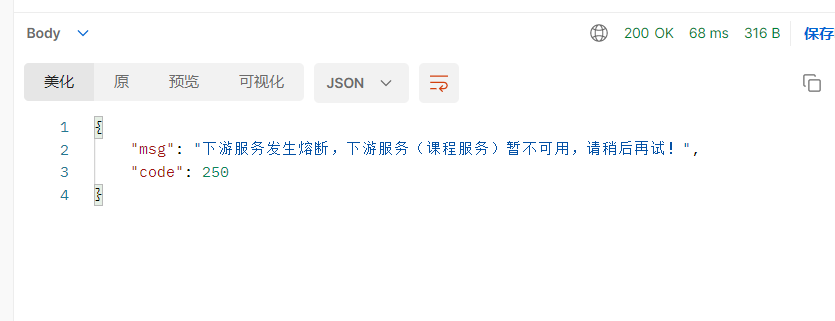
**服务降级测试一：下游服务反应速度过慢，超过在sentinel上设置的熔断阈值**





**熔断测试二：下游服务直接down，此时openfeign进行调用时也会直接进行降级**



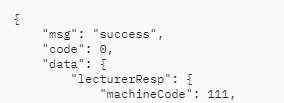


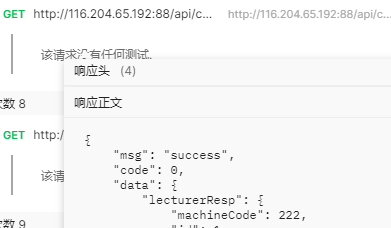
## 5.4 灰度发布

测试服务（与负载均衡接口相同）：获取单个课程详细信息时会访问Member的获取教师详情的接口，在获取教师详情的接口返回的信息包括机器号。初始状态下Nacos中两个member服务权重均为1

此时访问该接口机器码‘111’和‘222’以1：1比例返回，如下图所示





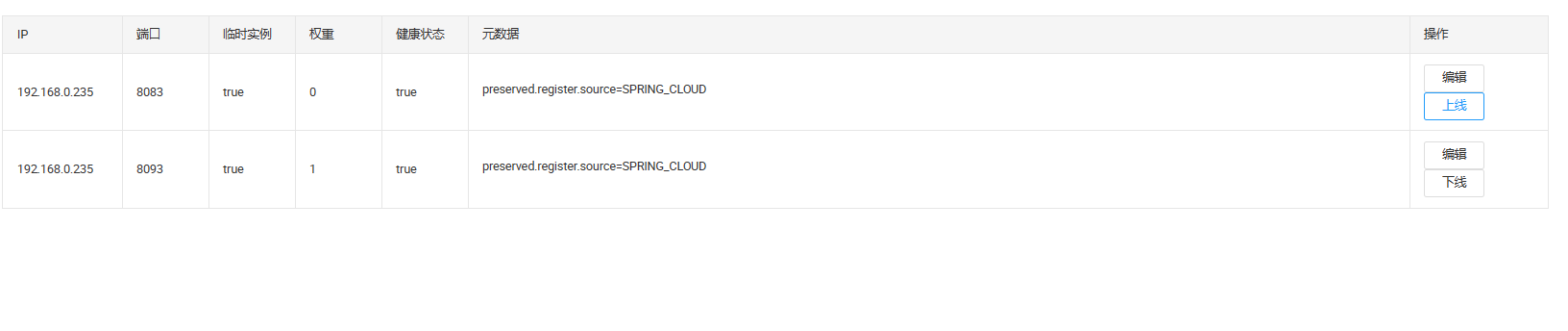


在Spring中开启Nacos默认的负载均衡策略，并将Nacos中将两个服务权重改为3：7





再次访问该接口即可发现机器码‘111’与‘222’以3：7的比例返回，最终逐渐将一个服务权重设为0，在没有人访问该服务后将该服务下线，此时再次访问该接口机器码均为‘222’



# 六、项目总结

本项目是一个基于Springcloud Alibaba的在线学习平台，实现了管理员和用户两个角色的功能。管理员功能包括课程管理、章节管理、分类管理、教师信息管理、评论信息管理和考试功能。用户功能包含课程信息查看、课程视频播放、课程评论和课程考核的功能。

我们的微服务架构是这样的：我们有api-gateway网关，它根据请求前缀的不同将请求分发到不同的微服务中。微服务包括课程信息微服务service-course、人员管理服务service-member、第三方调用微服务service-oss、分类管理服务service-sort和客户端管理服务service-admin。

我们的微服务在物理节点上的部署采用了不同的策略。如核心服务如service-course、service-member等，每个服务部署了三个以上的节点以应对请求，并演示项目负载均衡。低负载的服务则部署了两个物理节点。

环境搭建和部署方面，我们使用了docker来部署各种组件和服务，包括nacos、mysql、sentinel等。微服务则是将各个微服务打成jar包后，进行镜像制作并上传DockerHub，使用Dockerfile进行微服务快速部署。

服务治理方面，我们实现了服务限流、服务发布与负载均衡、服务熔断与降级以及灰度发布等功能。限流方面，我们对获取课程树结构的接口进行限流，使用了@SentinelResource注解，并设置了流控规则。服务发布与负载均衡则是通过Nacos和docker来实现。服务熔断与降级是在服务A调用服务B出现问题时，服务A可以不再调用服务B，直接返回结果，以此来减轻服务A和服务B的压力。而灰度发布则是通过改变Nacos中的服务权重来实现。

总体来说，法途在线学习平台实现了丰富的功能，包括管理员管理课程、用户学习课程等，微服务的设计和实现也充分考虑了高可用性和可扩展性。服务治理方面，我们实现了限流、熔断、降级和灰度发布等功能，为提供稳定的服务打下了坚实的基础。