后台管理系统

* 权限管理功能

1. 菜单管理：列表、添加、修改、删除
2. 角色管理：增删改查，批量删除 为角色分配菜单
3. 用户管理：增删改查，为用户分配角色
4. 权限管理表关系

权限表，角色表，用户表 权限角色关联表 用户角色关联表

* 讲师管理模块

讲师列表：增删改查 分页显示

* 课程分类模块

1. 添加课程分类

读取Excel课程分类数据，添加到数据库中 EasyExcel

1. 课程分类列表

使用树形结构显示课程分类列表

* 课程管理模块

1. 课程列表
2. 添加课程

课程发布流程：填写基本信息，添加章节和小节 课程分布信息的确认 最终课程发布

课程如何判断是否已经发布？添加字段status（draft,norm）。

课程添加过程中，中途停止添加，重新添加课程，如何找到之前没有发布完成的课程，继续进行发布？

课程列表设计了下拉列表选择已发布/未发布，对未发布课程可以进行编辑课程信息，编辑章节小节，删除等功能

1. 添加小节视频（阿里云视频点播）

* 统计分析模块

1. 生成统计数据
2. 统计数据图表显示（ECharts）

前台用户系统

1. 首页数据的显示
2. 显示幻灯片 轮播图

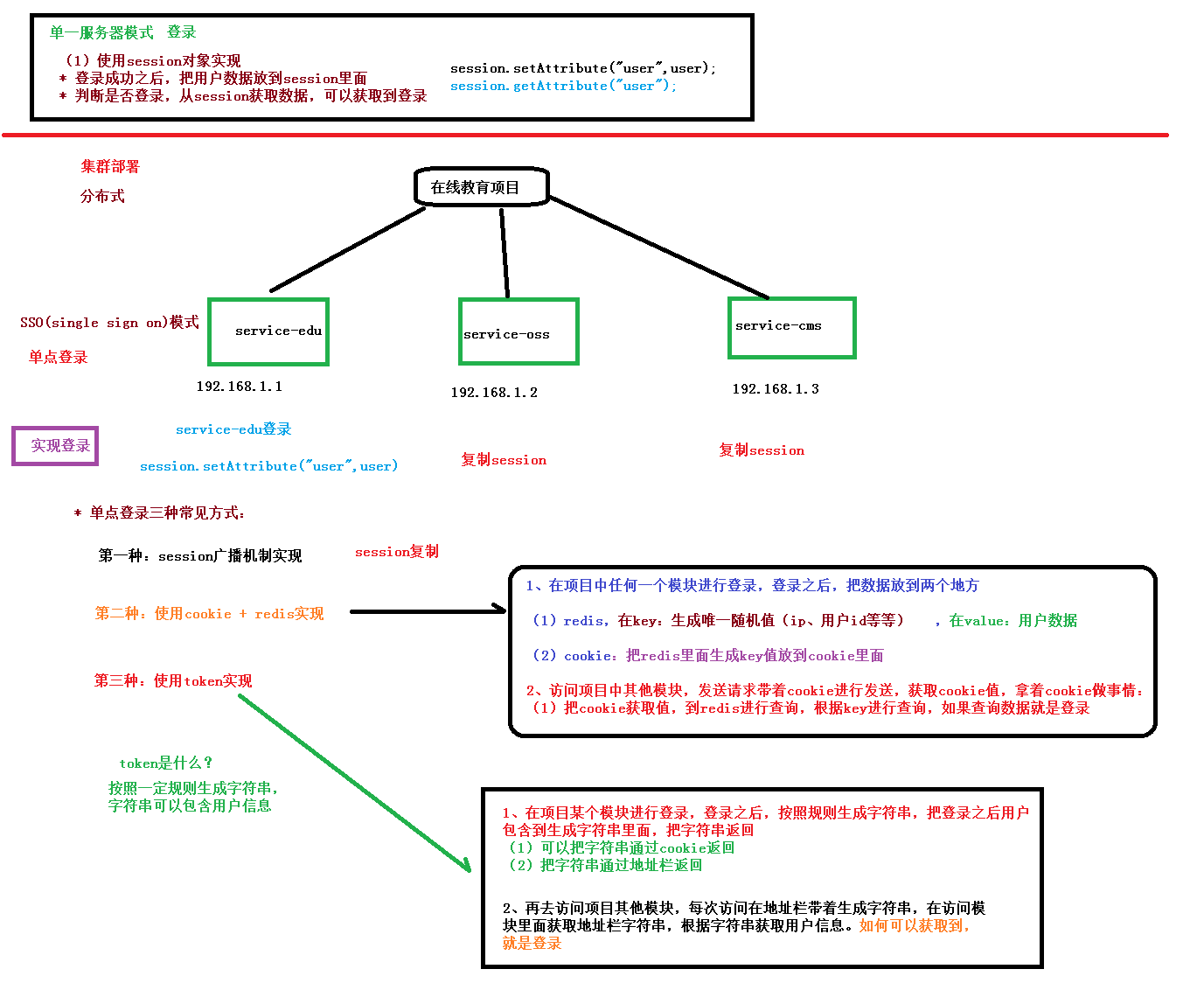
在banner表中根据id查询前两条，进行图片的播放

1. 显示热门课程
2. 显示名师
3. 注册功能

（1）获取手机验证码

1. 登录功能
2. 普通登录（手机号+密码）

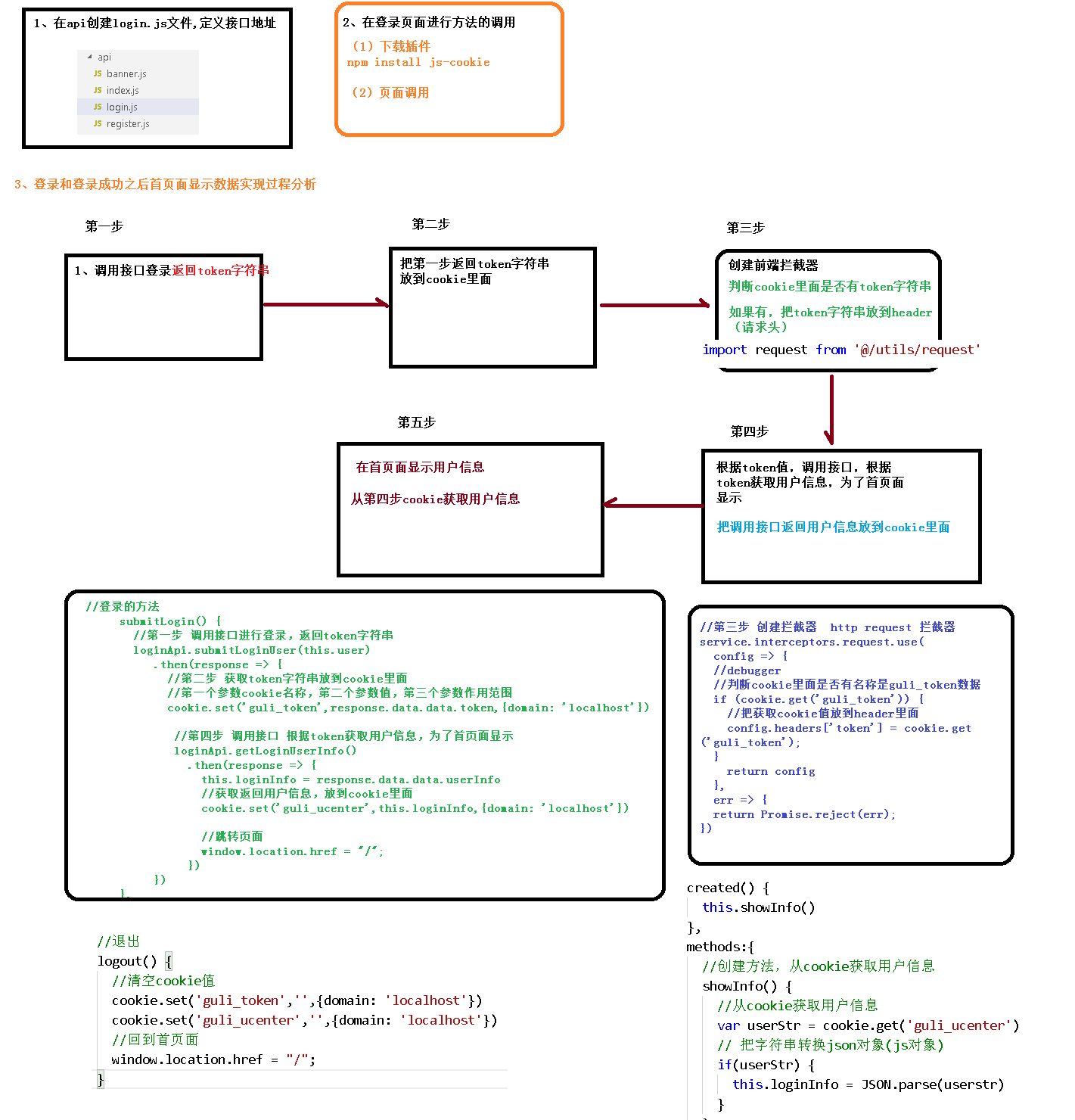
SSO 单点登录



Token生成规则，JWT生成

JWT三个组成部分（a:头信息 b:主体信息 c:签名哈希）

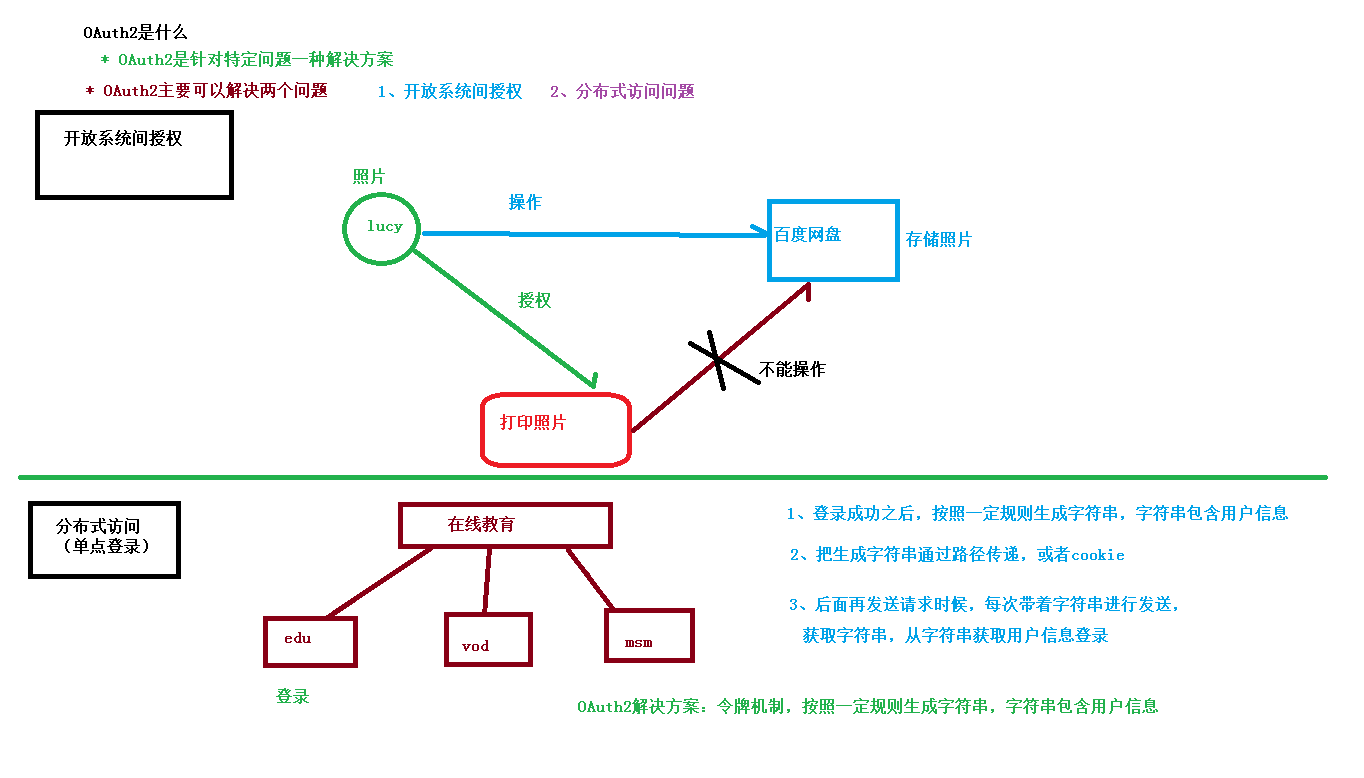
登录流程



1. 微信扫码登录

OAuth2（特定问题的解决方案不是协议）jwt是一个实现实例

解决：1：开放系统授权 2：分布式访问（单点登录）

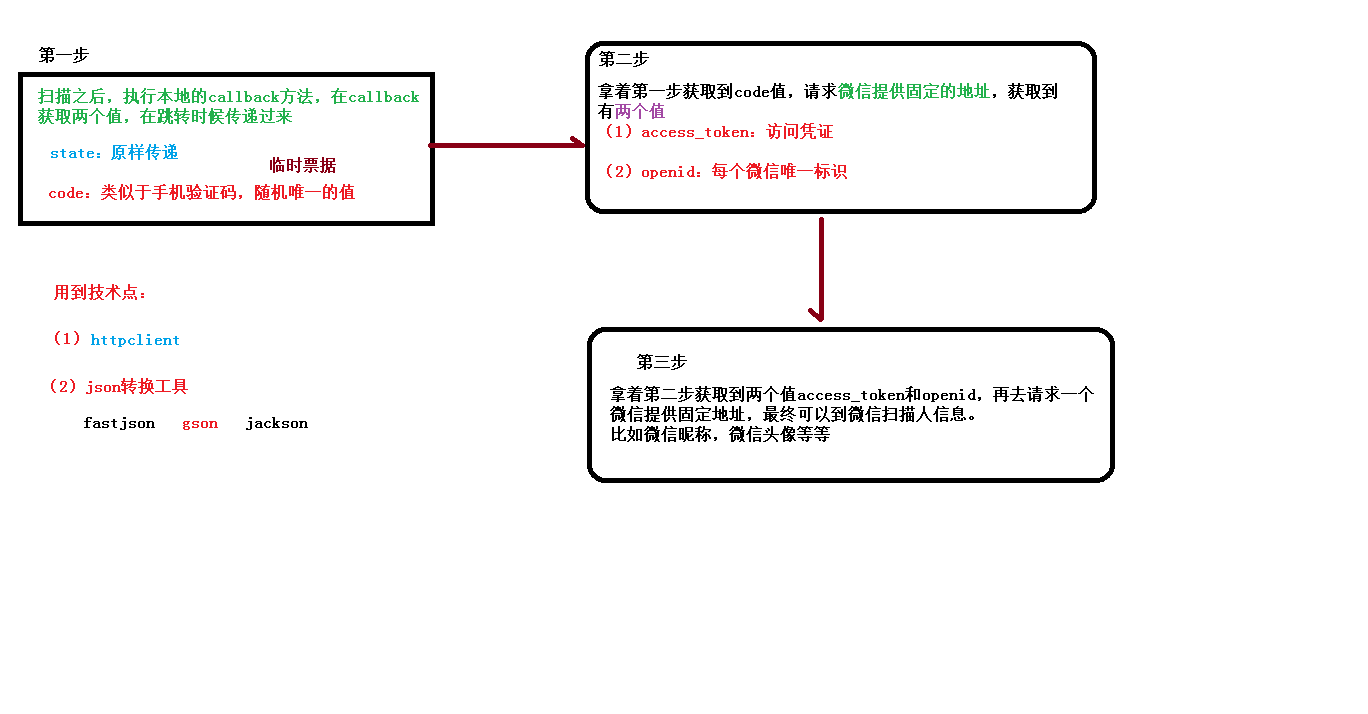


如何获取扫码人微信信息

a: 扫描后接口返回code（临时票据）

b: 拿着code请求微信的固定地址，得到Access\_token(访问凭证)+openid(微信唯一标识)

c: 根据上述返回值再去请求微信固定地址（会将上述值嵌入到请求地址中？xx&xx形式），得到扫码人微信信息



1. 课程列表

条件查询分页+排序（价格，销量，人气）

1. 课程详情页
2. 课程信息显示（课程基本信息，分类，讲师信息，课程的章节小节）
3. 判断课程是否免费
4. 课程小节视频的在线播放
5. 微信支付功能
6. 生成课程订单（跳转到新的页面）
7. 生成微信支付二维码
8. 微信最终支付

微信支付流程：

* 如果收费，点击立即购买生成课程订单
* 点击订单页面去支付，生成微信支付二维码
* 使用微信扫描支付二维码实现支付
* 支付之后，每个3秒查询订单状态（是否支付成功setinterval（“函数（）”,3000））,如果未支付成功，则等待，如果支付成功，更新订单状态，向支付记录表中添加支付成功记录

生成支付二维码



前后端分离开发

1：项目使用前端技术

1. vue框架

常见指令：v-bind v-model v-if v-for v-html

生命周期：created()页面渲染之前 mounted()页面渲染之后 data(){return{}} methods:{}

（2） Element-ui进行页面组件设计

（3） Node.js javaScript运行环境，不需要浏览器直接运行js代码，模拟服务器效果

（4） NPM 包管理工具=>Maven npm install npm init npm run dev

（5） 前端模块化 import teacherApi from ‘@/api/edu/teacher’即前端模块化编程

后端技术一：

1：采用微服务架构

2：SpringBoot

SpringBoot本质是Spring，是一个快速构建Spring工程的脚手架。

启动类包扫描机制 @ComponentScan(“包路径”) @MapperScan(“包路径”)

SpringBoot配置文件：.properties .yml

加载顺序: **bootstrap**.properties **application**.properties

3：SpringClound

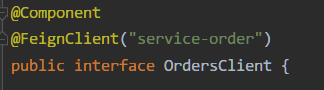
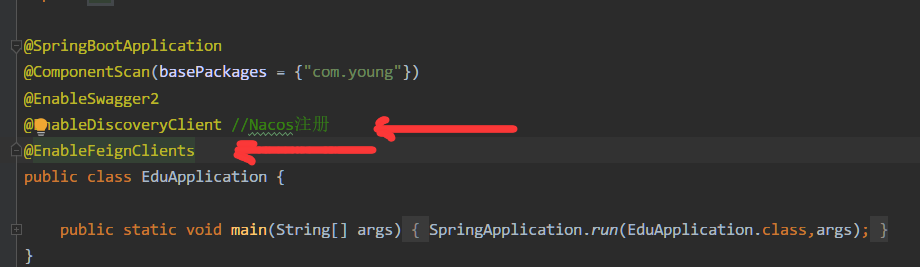
是很多框架总称，使用这些框架实现微服务架构，基于SpringBoot实现



Nacos作为注册中心，可以让一个微服务模块调用另一个微服务模块

4：Feign

服务调用，一个微服务调用另一个微服务，实现远程调用

5：MyBatisPlus

1. 对MyBatis做增强
2. 自动填充
3. 乐观锁
4. 逻辑删除
5. 代码生成器

6: EasyExcel

1. 阿里巴巴开源的Excel操作工具，效率高
2. 传统Dom加载模式是将数据全部加入内存进行解析，SAX模式采用分行读取，解析对内存消耗较少
3. 项目在添加课程分类使用，然后添加到数据库

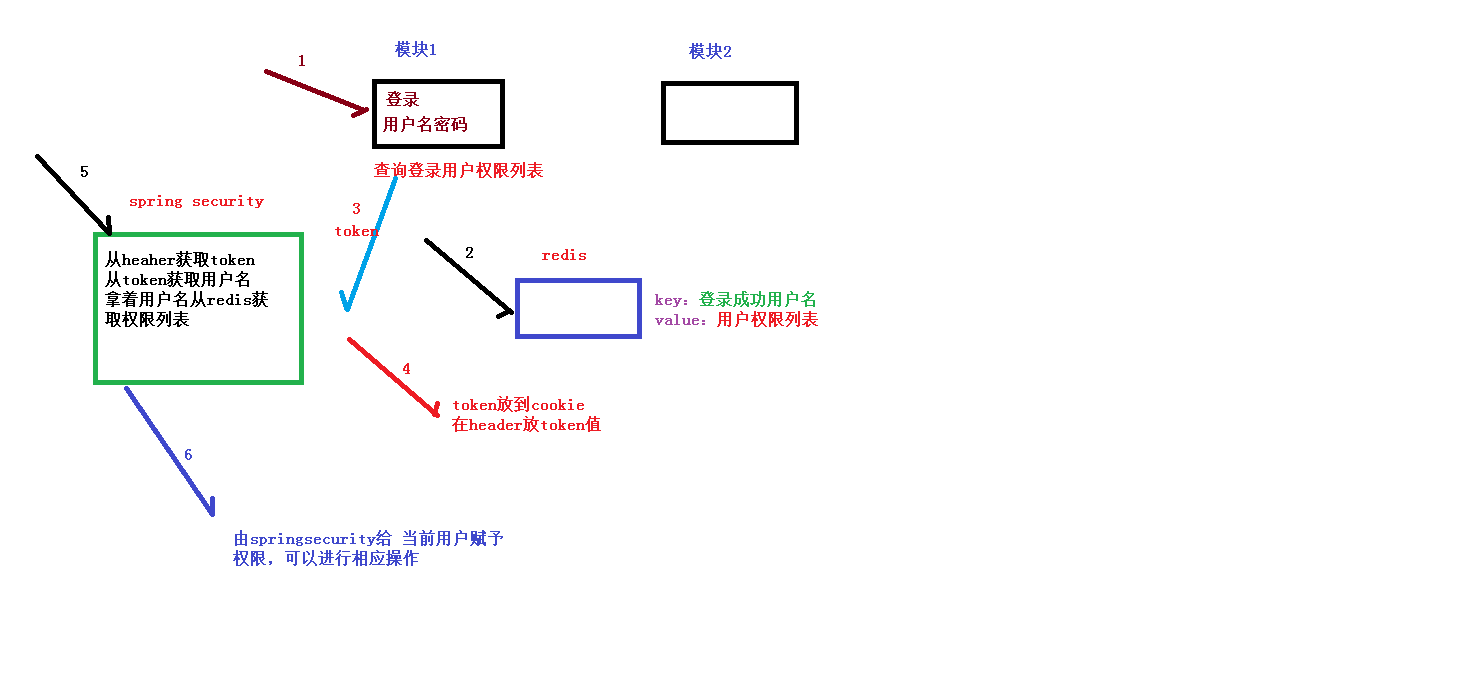
后端技术二：

1：SpringSecurity

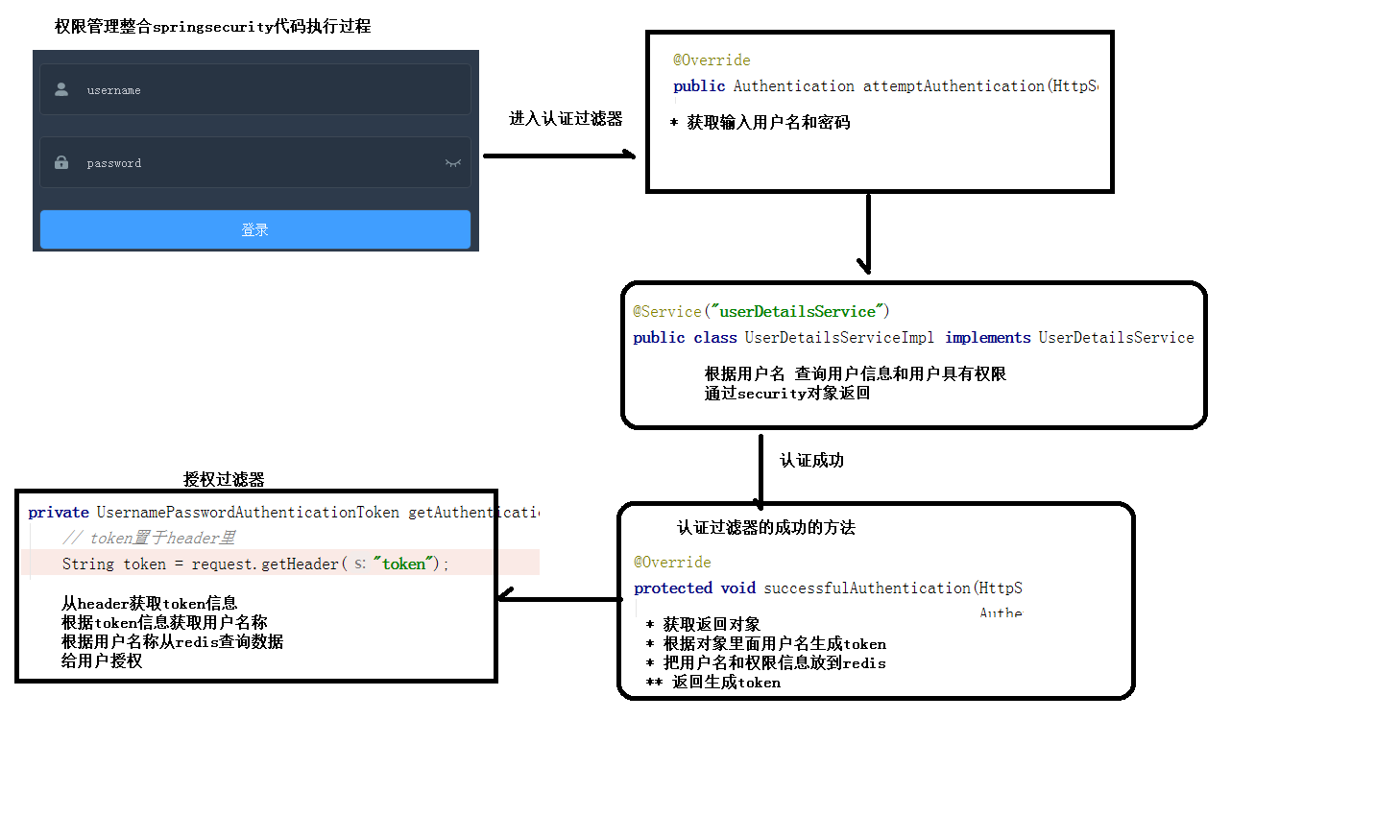
a: 在项目整合框架实现权限管理功能

b: SpringSecurity框架组成：认证和授权

c: SpringSecurity登录认证过程



d:SpringSecurity代码执行过程



2：Redis

1. 首页数据通过redis进行缓存
2. Redis数据类型
3. 使用Redis作为缓存，存放不太重要或者不经常改变的数据

3：Nginx

1. 反向代理服务器
2. 请求转发，负载均衡 动静分离—>（为了提高网站的响应速度，减轻程序服务器（Tomcat，Jboss等）的负载，对于静态资源比如图片，js，css等文件，我们可以在反向代理服务器中进行缓存，这样浏览器在请求一个静态资源时，代理服务器就可以直接处理，而不用将请求转发给后端服务器。用户请求的动态文件比如servlet,jsp则转发给Tomcat，Jboss服务器处理，这就是动静分离。这也是反向代理服务器的一个重要的作用）

4：OAuth2+Jwt

OAuth2针对特定问题的解决方案

JWT三个组成部分（a:头信息 b:主体信息 c:签名哈希）

5：HttpClient

1. 发送请求返回响应的工具，不需要浏览器完成请求和响应的过程
2. 应用场景：微信登录获取扫描人信息，微信支付查询支付状态

6：Cookie

1. Cookie特点

客户端技术，每次发送请求带着cookie值进行发送，cookie有默认时长：关闭浏览器cookie默认不存在。

7：微信登录

8：微信支付

9：阿里云oss

1. 文件存储服务器
2. 上传讲师头像进行存储

10：阿里云视频点播

（1）：视频上传，删除，播放

（2）：整合阿里云视频播放器进行播放

（3）：使用播放凭证进行播放🡪可以播放加密或者不加密视频

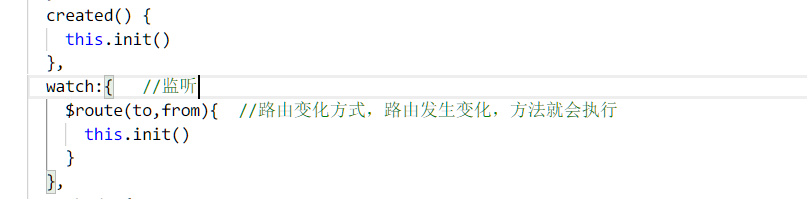
11：阿里云短信服务

注册进行手机验证码的发送

**项目开发问题**：

1：路由切换问题

问题：多次路由跳转到同一个vue页面时，压面的created（）方法只会执行一次（点一次修改，再点击添加，表单不清空的问题）



2：跨域问题

跨域：访问协议、IP地址或者端口号任一个不同都会产生跨域

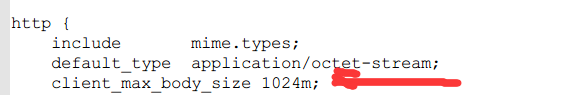
在Controller添加注解 @CrossOrigin

3：413问题

a:上传视频时。出现视频文件过大的限制

b:413:请求体过大

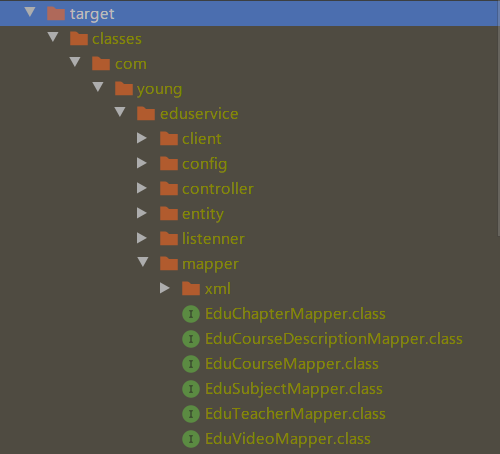
c:Nginx配置



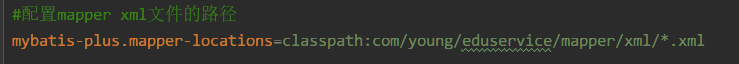
4：Maven mapper的Xml文件不会自动加载的问题

1. Maven在加载项目时，默认不会加载src-java文件夹下的xml文件

a: 直接将xml文件复制到target目录下的对应子目录



b: 通过配置文件实现





项目描述：

在线教育网站开发项目主要包络后台管理系统和前端客户系统，使用微服务架构，采用前后端分离技术进行开发。后台管理模块主要包括：课程管理，讲师管理，课程发布，统计分析等。前端主要模块包括:登录（单点登录），注册，首页、课程、名师，课程支付等。前端技术主要涉及vue框架+element-ui组件，后端技术主要涉及SpringBoot+SpringCloud(Feign,Nacos)+MyBatis-Plus+redis+HttpClient+EasyExcel+ECharts等。第三方技术包括阿里云存储服务（讲师头像存储）以及阿里云视频点播服务（试听课程播放）。