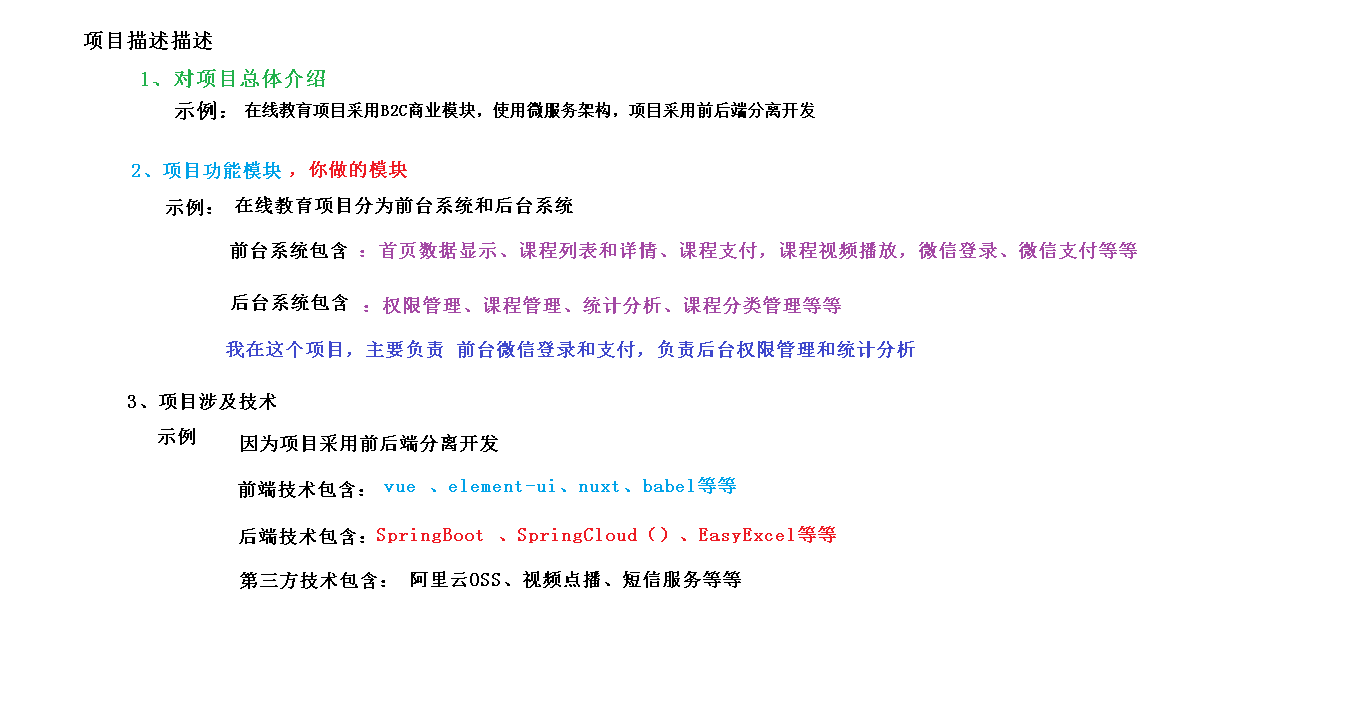
1. **项目描述**



（1）

在线教育系统，分为前台网站系统和后台运营平台，B2C模式。

前台用户系统包括课程、讲师、问答、文章几大大部分，使用了微服务技术架构，前后端分离开发。

后端的主要技术架构是：SpringBoot + SpringCloud + MyBatis-Plus + HttpClient + MySQL + Maven+EasyExcel+ nginx

前端的架构是：Node.js + Vue.js +element-ui+NUXT+ECharts

其他涉及到的中间件包括Redis、阿里云OSS、阿里云视频点播

业务中使用了ECharts做图表展示，使用EasyExcel完成分类批量添加、注册分布式单点登录使用了JWT

（2）

项目前后端分离开发，后端采用SpringCloud微服务架构，持久层用的是MyBatis-Plus，微服务分库设计，使用Swagger生成接口文档

接入了阿里云视频点播、阿里云OSS。

系统分为前台用户系统和后台管理系统两部分。

前台用户系统包括：首页、课程、名师、问答、文章。

后台管理系统包括：讲师管理、课程分类管理、课程管理、统计分析、Banner管理、订单管理、权限管理等功能。

在线教育计费案例：

小A是一名杭州的创业者，带领团队研发了一个在线教育平台。他希望把视频托管在阿里云上，存量视频大约1000个，占用存储空间近1T，每月预计新增视频100个，并新增存储约100G，课程视频的时长集中在20-40分钟，并且按照不同课程进行分类管理。为了保障各端的观看效果，计划为用户提供“标清480P”和“高清720P”两种清晰度。目前已有用户400人左右，每日平均视频观看次数1000次，在移动端和PC端观看次数比例大致为3:1。

**2、这是一个项目还是一个产品**

这是一个产品

1.0版本是单体应用：SSM

2.0版本加入了SpringCloud，将一些关键业务和访问量比较大的部分分离了出去

目前独立出来的服务有教学服务、视频点播服务、用户服务、统计分析服务、网关服务

**3、测试要求**

首页和视频详情页qps单机qps要求 2000+

经常用每秒查询率来衡量域名系统服务器的机器的性能，其即为QPS

QPS = 并发量 / 平均响应时间

**4、企业中的项目（产品）开发流程**

一个中大型项目的开发流程

1、需求调研（产品经理）

2、需求评审（产品/设计/前端/后端/测试/运营）

3、立项（项目经理、品管）

4、UI设计

5、开发

* 架构、数据库设计、API文档、MOCK数据、开发、单元测试
* 前端
* 后端

6、前端后端联调

7、项目提测：黑盒白盒、压力测试（qps）  loadrunner

8、bug修改

9、回归测试

10、运维和部署上线

11、灰度发布

12、全量发布

13、维护和运营

**5、系统中都有那些角色？数据库是怎么设计的？**

前台：会员（学员）

后台：系统管理员、运营人员

后台分库，每个微服务一个独立的数据库，使用了分布式id生成器

**6、视频点播是怎么实现的（流媒体你们是怎么实现的）**

我们直接接入了阿里云的云视频点播。云平台上的功能包括视频上传、转码、加密、智能审核、监控统计等。

还包括视频播放功能，阿里云还提供了一个视频播放器。

**7、前后端联调经常遇到的问题：**

      1、请求方式post、get

      2、json、x-wwww-form-urlencoded混乱的错误

      3、后台必要的参数，前端省略了

      4、数据类型不匹配

      5、空指针异常

      6、分布式系统中分布式id生成器生成的id 长度过大（19个字符长度的整数），js无法解析（js智能解析16个长度：2的53次幂）

          id策略改成 ID\_WORKER\_STR

**8、前后端分离项目中的跨域问题是如何解决的**

后端服务器配置：我们的项目中是通过Spring注解解决跨域的 @CrossOrigin

也可以使用nginx反向代理、httpClient、网关

**9、说说你做了哪个部分、遇到了什么问题、怎么解决的**

问题1：

分布式id生成器在前端无法处理，总是在后三位进行四舍五入。

分布式id生成器生成的id是19个字符的长度，前端javascript脚本对整数的处理能力只有2的53次方，也就是最多只能处理16个字符

解决的方案是把id在程序中设置成了字符串的性质

问题2：

项目迁移到Spring-Cloud的时候，经过网关时，前端传递的cookie后端一只获取不了，看了cloud中zuul的源码，发现向下游传递数据的时候，zull默认过滤了敏感信息，将cookie过滤掉了

解决的方案是在配置文件中将请求头的过滤清除掉，使cookie可以向下游传递

问题3.......

**10、分布式系统的id生成策略**

<https://www.cnblogs.com/haoxinyue/p/5208136.html>

**11、项目组有多少人，人员如何组成？**

**12、分布式系统的CAP原理**

**CAP定理：**

指的是在一个分布式系统中，Consistency（一致性）、 Availability（可用性）、Partition tolerance（分区容错性），三者不可同时获得。

 一致性（C）：在分布式系统中的所有数据备份，在同一时刻是否同样的值。（所有节点在同一时间的数据完全一致，越多节点，数据同步越耗时）

可用性（A）：负载过大后，集群整体是否还能响应客户端的读写请求。（服务一直可用，而且是正常响应时间）

分区容错性（P）：分区容错性，就是高可用性，一个节点崩了，并不影响其它的节点（100个节点，挂了几个，不影响服务，越多机器越好）

**CA 满足的情况下，P不能满足的原因：**

数据同步(C)需要时间，也要正常的时间内响应(A)，那么机器数量就要少，所以P就不满足

**CP 满足的情况下，A不能满足的原因：**

数据同步(C)需要时间, 机器数量也多(P)，但是同步数据需要时间，所以不能再正常时间内响应，所以A就不满足

**AP 满足的情况下，C不能满足的原因：**

机器数量也多(P)，正常的时间内响应(A)，那么数据就不能及时同步到其他节点，所以C不满足

**注册中心选择的原则：**

Zookeeper：CP设计，保证了一致性，集群搭建的时候，某个节点失效，则会进行选举行的leader，或者半数以上节点不可用，则无法提供服务，因此可用性没法满足

Eureka：AP原则，无主从节点，一个节点挂了，自动切换其他节点可以使用，去中心化

**结论：**

分布式系统中P,肯定要满足，所以我们只能在一致性和可用性之间进行权衡

如果要求一致性，则选择zookeeper，如金融行业

如果要求可用性，则Eureka，如教育、电商系统

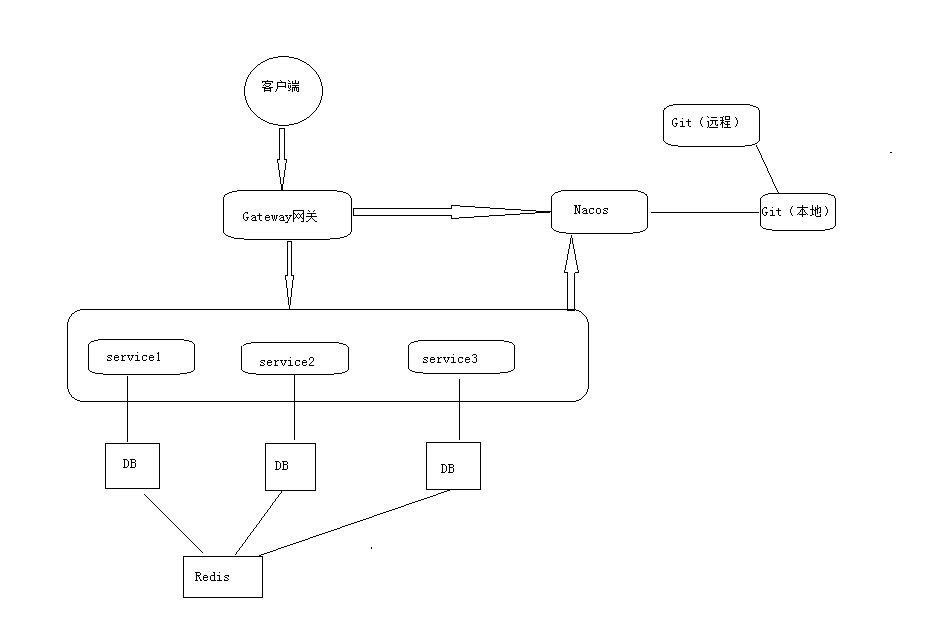
没有最好的选择，最好的选择是根据业务场景来进行架构设计

**13、前端渲染和后端渲染有什么区别**

前端渲染是返回json给前端，通过javascript将数据绑定到页面上

后端渲染是在服务器端将页面生成直接发送给服务器，有利于SEO的优化

**14、能画一下系统架构图吗**



# 课堂总结：

## 一、准备

1、把后端接口启动起来.

2、启动前端项目（前台系统和后台系统）.

## 二、项目后台管理系统功能

### 1、登录功能（SpringSecurity框架）..

### 2、权限管理功能

（1）菜单管理：列表、添加、修改、删除

（2）角色管理

\*列表、添加、修改、删除、批量删除

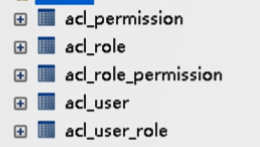
\*为角色分配菜单

（3）用户管理

\*列表、添加、修改、删除、批量删除

\*为用户分配角色

（4）权限管理表和关系

5张表

### 3、讲师管理模块.

（1）条件查询分页列表、添加、修改、删除

### 4、课程分类模块

（1）添加课程分类.

\*读取Excel里面课程分类数据，添加到数据库中.

（2）课程分类列表.

\*使用树形结构显示课程分类列表

### 5、课程管理模块

（1）课程列表功能

（2）添加课程

\*课程发布流程：

第一步填写课程基本信息，

第二步添加课程大纲（章节和小节），

第三步课程信息确认，最终课程发布.  
\*课程如何判断是否已经被发布了？使用status字段.

\*课程添加过程中，中途把课程停止添加，重新去添加新的课程，如何找到之前没有发布完成课程，继续进行发布？到课程列表中根据课程状态查询未发布的课程，点击课程右边超链接把课程继续发布完成

（3）添加小节上传课程视频.



### 6、统计分析模块

（1）生成统计数据.

（2）统计数据图表显示

## 三、项目前台用户系统

### 1、首页数据显示.

(1)显示幻灯片功能：后台系统控制显示的图片

(2）显热门课程：排序 可以手动置顶【8个】

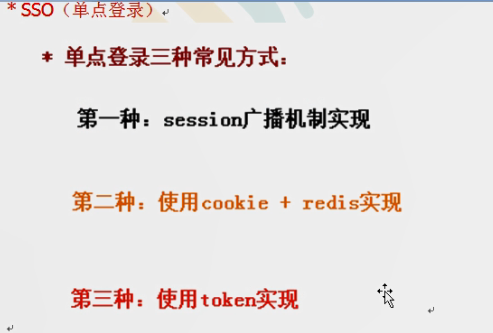
（3）显示名师：课程点击最多的讲师 可以手动置顶【8】

### 2、注册功能

（1）获取手机验证码.

### 3、登录功能.

（1）普通登录.



JWT

使用JWT生成token字符串

JWT有三部分组成. 头部 载荷 签名

**登陆流程：**

登录调用登录接口返回token字符串，前端把返回token字符串放到cookie里面，创建前端拦截器进行判断，如果cookie里面包含token字符串，把token字符串放到header里面。调用接口根据token获取用户信息，把用户信息放到cookie里面，进行显示

（2）微信扫码登陆.

OAuth2

是针对特定问题解决方案.

主要有两个问题：开放系统间授权，分布式访问

如何获取扫描人信息过程？.

扫码之后返回code （临时票据）拿着code请求微信固定地址，得到两个值 access\_token（访问凭证）、openid（微信唯一标识），拿着这两个值再去请求微信固定地址，得到微信扫描人信息（昵称、头像等待）

4、名师列表功能

5、名师详情功能

6、课程列表功能

（1）条件查询分页列表功能.

7、课程详情页.

（1）课程信息显示（包含课程基本信息，分类，讲师，i程大纲）.

（2）判断课程是否需要购买

8、课程视频在线播放

9、课程支付功能（微信支付）.

（1）生成课程订单

（2）生成微信支付二维码

（3）微信最终支付

\*微信支付实现流程：.

\*如果课程是收费课程，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，每隔3秒查询支付状态（是否支付成功），如果没有支付成功等待，如果支付成功之后，更新订单状态（已经支付状态），向支付记录表添加支付成功记录.

## 四、总结在线教育项目技术点.

### 1、在线教育项目采用前后端分离开发

### 2、项目使用前端技术.

1. VUE
   1. 基本语法：
   2. 常见指令：v-bind v-model v-for v-if v-show v-html【标签进行渲染】
   3. 绑定事件 ：v-on
   4. 声明周期：created()：页面渲染之前执行 mounted().页面渲染之后执行
   5. ES6语法规范
2. Element-ui
3. Nodejs：是JavaScript运行环境，不需要浏览器直接运行is代码，模拟服务器效果。
4. NPm：
   1. 包管理工具，类似于Maven
   2. Npm命令: npm init、 npm install 依赖名称 、npm run dev
5. Babel：转码器，把ES6的代码转换成ES5代码
6. 前端模块化
   1. 通过一个页面或者js文件，调用另外一个js文件里面的方法
   2. 问题：ES6的模块化无法在Node.js中执行，需要用Babel编辑成ES5后再执行
7. 后台系统使用vue-admin-template.：基于vue+Element-ui
8. 前台系统使用Nuxt：
   1. 基于Vue
   2. 服务端渲染技术SEO【搜索引擎缓存】
9. Echarts
   1. 图表工具

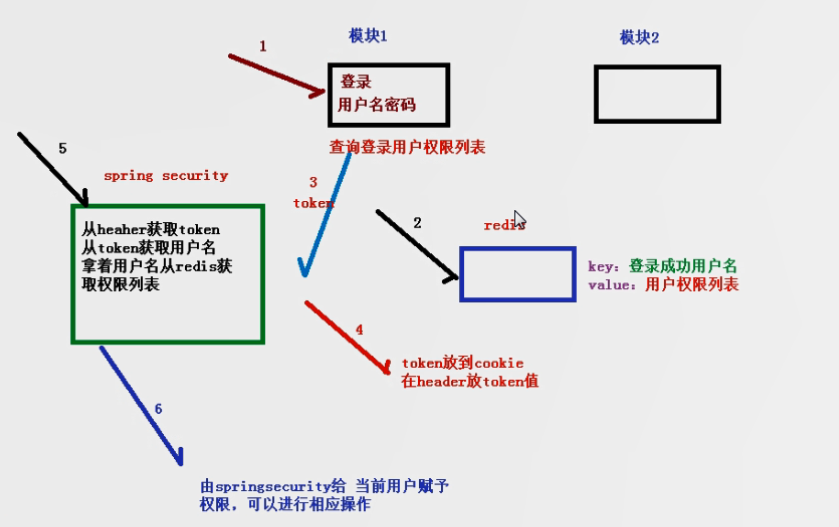
### 3、项目使用后端技术一

1. 项目采用微服务架构
2. SpringBoot
   1. Springboot本质上就是spring，只是快速构建spring工程的脚手架
   2. 细节：
      1. 启动类包扫描机制
      2. 设置扫描规则：@ComponentScan(basePackages = {"com.atguigu"})
      3. 配置类
   3. 配置文件：
      1. 配置文件类型：properties、yml
      2. 配置文件加载顺序：bootstrap、application、application-dev
3. SpringCloud
   1. 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
   2. 组成框架有哪些：
   3. 项目中，使用阿里巴巴Nacos替代SpringCloud一部分组件
   4. Nacos
      1. 作为注册中心
      2. 作为配置中心
   5. Feign
      1. 服务调用，一个微服务调用另外一个微服务，实现远程调用
   6. 熔断器
      1. 容错降级处理
   7. 网段Gatway网关
      1. Springcloud之前zuul网关，目前gateway网关
   8. 版本：
      1. H版本



1. MyBatisPlus
   1. MyBatisPlus就是对Mybaits做增强
   2. 自动填充
   3. 乐观锁
   4. 逻辑删除
   5. 代码生成器
   6. 主键策略：手动插入，自增，雪花算法
2. EasyExcle
   1. 阿里巴巴提供操作excel工具，代码简洁，效率很高
   2. EasyExcel对poi进行封装，采用SAX方式解析.【一行一行解析，Poi采用Dom模式一次全部读取到内存中】
   3. 项目应用在添加课程分类，读取excel数据

### 4、项目使用后端技术二

1. SpringSecurity
   1. 在项目中整合实现权限管理功能
   2. SpringSecurity框架组成：认证和授权
   3. SpringSecurity登陆认证过程：
   4. SpringSecurity代码执行过程
2. Redis
   1. 首页数据通过Redis进行缓存.
   2. 数据类型：字符串 Set Hash List zSet
   3. 使用Redis作为缓存，不太重要或者不经常改变数据适合放到Redis作为缓存。
3. Nginx
   1. 反向代理服务器
   2. 请求转发、负载均衡、动静分离
4. OAuth+JWT
   1. OAuth2针对特定问题解决方案
   2. JWT包含三部分
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
   2. 每次发送请求带着cookie值发送
   3. Cookie有会话级别长：关闭浏览器cookie默认不存在，可以设置cookie有效时长setMaxAge
7. 微信登录
8. 微信支付
9. 阿里云OSS
   1. 文件存储服务器.
   2. 添加讲师时候上传讲师头像。
10. 阿里云视频点播服务
    1. 视频上传、删除、播放.
    2. 整合阿里云视频播放器进行视频播放.
       1. 使用视频播放凭证
11. 阿里云短信服务
    1. 注册时候，发送短信验证码
12. Git
    1. 代码提交到远程git仓库
13. Docker+Jenkins：自动化部署
    1. 手动打包运行
    2. Idea打包
    3. Docker+Jenkins自动化部署过程

## 五、在线教育项目问题

### 前端问题-路由切换问题

* 1. 多次路由跳转到同一个vue页面，页面中created方法只会执行一次.
  2. 解决方案：使用Vue监听.路由变化 初始化页面数据

### 前端问题-ES6模块化运行问题

* 1. Nodejs.不能直接运行ES6模块化代码，使用Babel把ES6模块化代码转换ES5代码执行

### MP生成19位id值

* 1. mp生成id值是19位，JavaScript处理数字类型值时候，只会处理到16位
  2. javaBean采用String类型

### 跨域问题

* 1. 访问协议，ip地址，端口号，这三个如果有任何一个不一样，产生跨域
  2. 跨域解决：

1. @CrossOrigin
2. 通过网关解决

### 413问题：Nginx大小限制

* 1. 上传视频时候，因为Nginx有上传文件大小限制，如果超过Nginx大小，出现413
  2. 413错误：请求体过大
  3. 在Nginx配置客户端大小
  4. 响应状态码：413【请求体过大】 403【跨域会出现】 302【重定向】

### Maven加载问题

* 1. 位于java目录下的xml在构建项目时maven默认不会加载该违建，只会加载。Java文件
  2. 解决方式：
     1. 直接复制xml文件到target
     2. 通过配置



