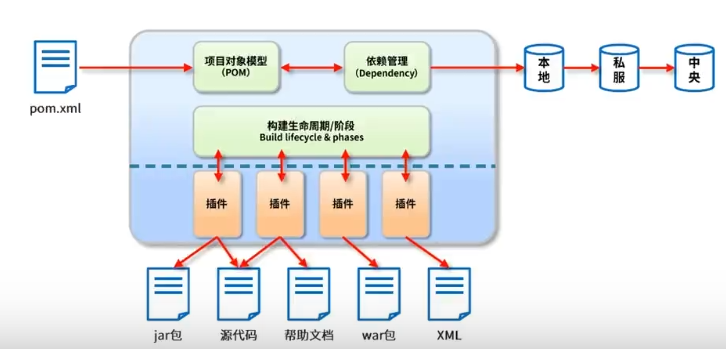
## 1 B/S, C/S 架构

B/S 更轻便

C/S 需要适配不同的操作系统

## 2 Maven

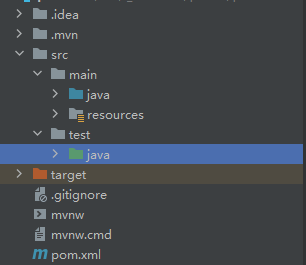
Maven是一个项目管理工具，对Java项目进行**自动化的构建**和**依赖管理。**



自动化构建：编译、运行、打包、解析xml；

依赖管理：下载数据库驱动等

目录结构：



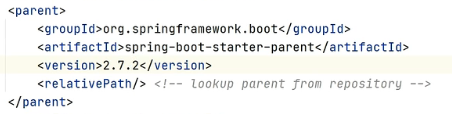
## 3 SpringBoot

### 3.1 创建项目

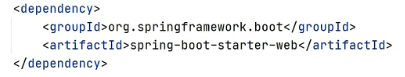
简化Spring的初始搭建和开发过程；

默认>配置

依赖于父级配置



Java web



Com.xxx package下建controller package；

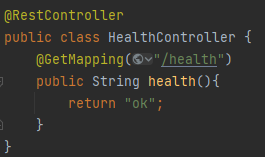
Controller package下建xxxController；

类前添加 @RestController 注解，使其变为控制器，可以接收浏览器请求；

类中加入@GetMapping注解，表明浏览器要发送http中的get请求，需要一个明确的地址；

[http://](http://www.baidu.com（域名）/health)**[www.baidu.com](http://www.baidu.com（域名）/health)**[（域名）/health](http://www.baidu.com（域名）/health)

<http://localhost:8080/health>



### 3.2 开发环境热部署

优化因频繁修改后台类文件导致的需要重新编译重新启动的效率问题；

提供spring-boot-devtools组件；



Application.properties中配置devtools



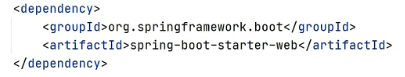
Setting/Build/compiler/build automatically

Setting/advanced setting/auto run

## 4 web开发

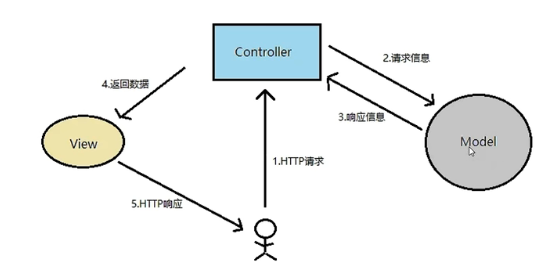
* Spring boot 整合了mvc, json, tomcat等，提供了spring-boot-starter-web组件，**简化web应用配置**；
* Webmvc为web开发基础框架；json为JSON数据解析组件；tomcat为自带的容器依赖；

启动时勾选即可；



### 4.1 控制器

* Spring boot提供了@controller和@RestController两种annotations**标识**此类负责接收和处理HTTP请求；
* MVC设计模式：
  + Model用于存储数据、封装数据；
  + Controller用于协调和控制；
  + View用于显示数据
  + 用户发送请求，请求控制器；控制器去取数据；最终把数据交给视图；视图再响应给用户；
  + 如果请求的是 **页面** 和 **数据** ，使用@controller即可；如果请求的只是数据，则可以使用@Restcontroller注解；



#### 4.1.1 @Controller的用法

返回的使一个hello页面和name数据，在前端页面中可以通过${name}参数获取后台返回的数据并显示



#### 4.1.2 @RestController的用法

返回的对象数据都会转换为JSON格式

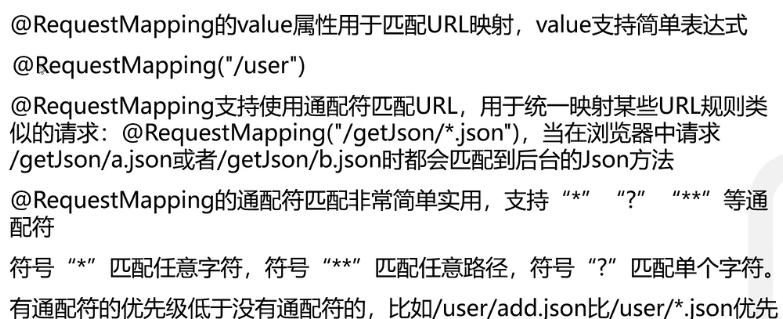
### 4.2 路由映射

如何接收请求？

@RequestMapping(“/ url”)负责URL的路由映射；

Parameters：

**Value - 请求URL的路径，支持URL模板、正则；**



**Method - HTTP请求方法，get or post；**



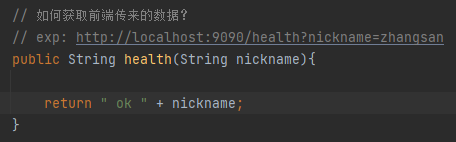
Consumes：请求类型；

Produces：响应的媒体类型；

Params，headers：请求的参数及请求头的值；

### 4.3 参数传递

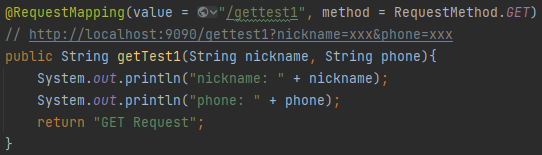
如何传递参数？





#### 4.3.1 多个参数的情况：

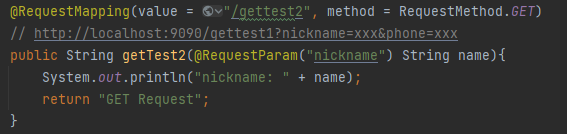
对于参数符不符合要求，可以在方法中做判断；



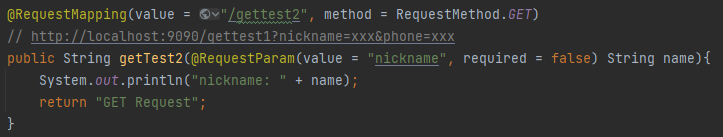


#### 4.3.2 传递的参数和后端不一致的情况

使用@RequestParam将前端传递的和后端的进行绑定，但这也使得该参数**必须**被传递



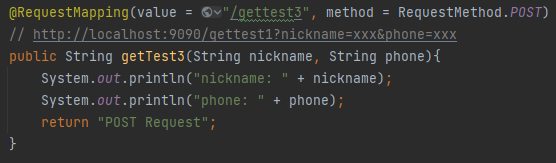
加required=false使其变为可选；



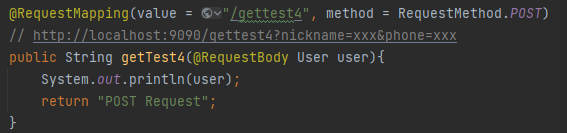
#### 4.3.3 post请求

安装apipost测试；地址栏中发送的使get请求





传递json类型数据，封装入一个对象中



* @RequestParam将请求参数绑定到控制器的方法参数上，接收的参数来自HTTP请求体或请求url的QueryString，当请求的参数名称与Controller的业务方法参数一致，@RequestParam可以省略；

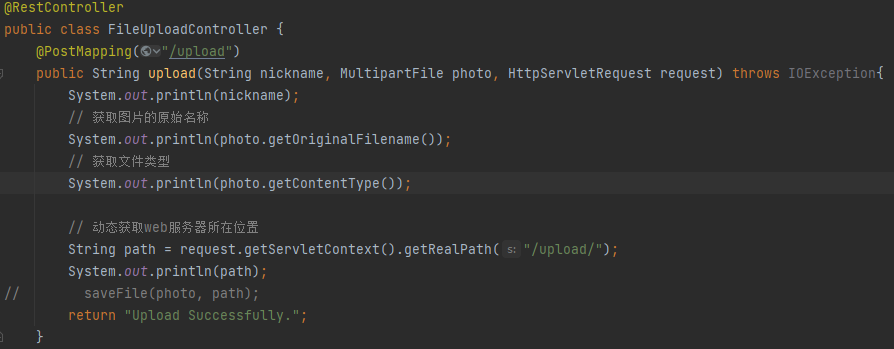
## 5 文件上传

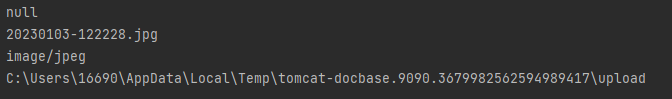
上传原理





参数名称要和表单的名称一致

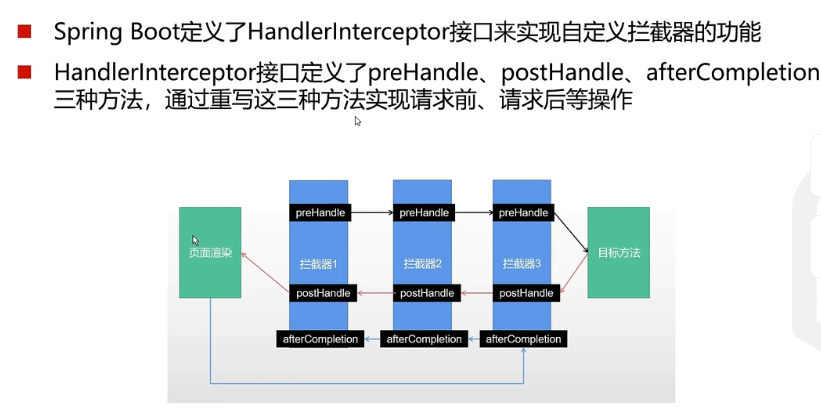




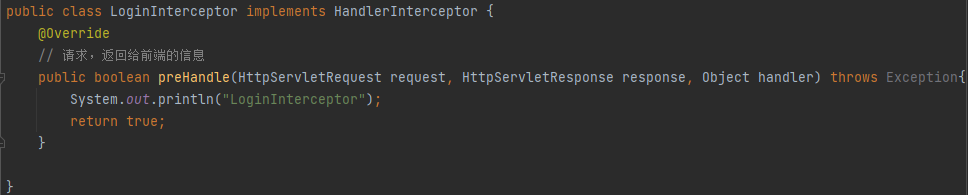
## 6 拦截器

用于拦截请求；用于对controller一些操作进行统一的处理；

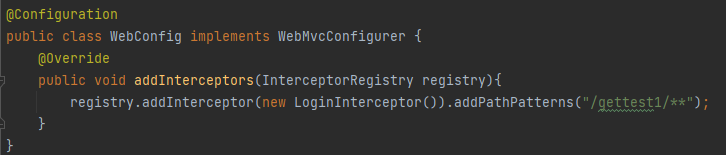
**Pre-handle：要访问，但没登陆**



继承系统的拦截器 extends HandleInterceptor



拦截器注册：设置拦截路径



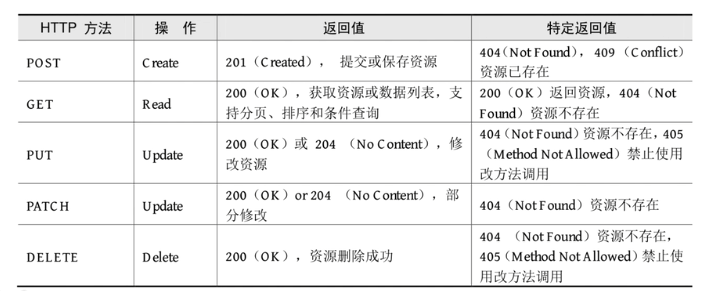
## 7 RESTful

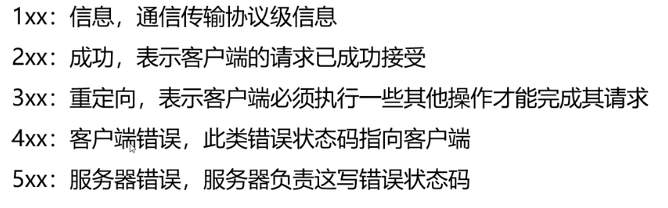
目前流行的架构设计风格；

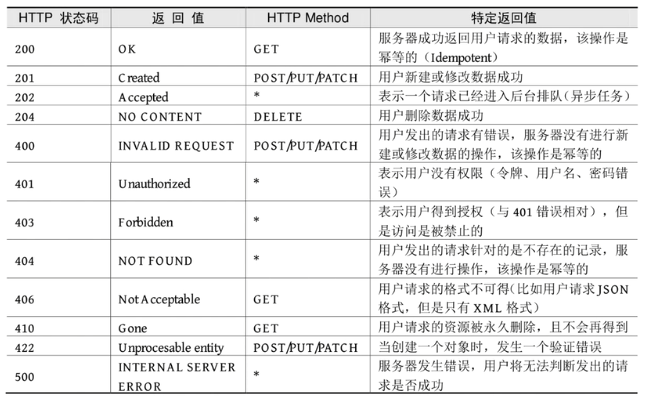
Representational State Transfer 表述性状态转移

每一个URI代表一种资源；客户端使用GET, POST, PUT, DELETE四种操作对服务端资源进行操作，获取、新建、更新、删除。

不应该在GET请求中进行修改、删除等操作。







传统方法：

Get <http://localhost/del?id=10>

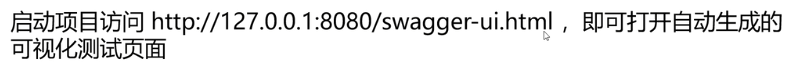
Restful:

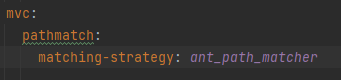
Delete <http://localhost/user/10> 不包含动词，名词往往与数据库的表名对应

配置Swagger：

|  |
| --- |
| <dependency>  <groupId>io.springfox</groupId>  <artifactId>springfox-swagger2</artifactId>  <version>2.9.2</version> </dependency> <dependency>  <groupId>io.springfox</groupId>  <artifactId>springfox-swagger-ui</artifactId>  <version>2.9.2</version> </dependency> |



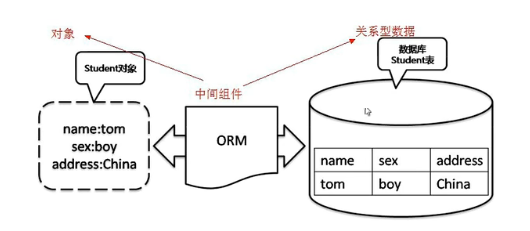




## 8 MybatisPlus

### ORM (Object Relational Mapping) 对象关系映射

完成对象和数据库之间的映射；



### MybatisPlus

一个ORM框架

### 步骤

添加依赖



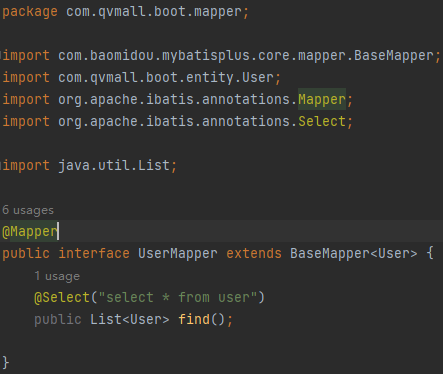
全局配置：

要告诉驱动去找哪个数据库；

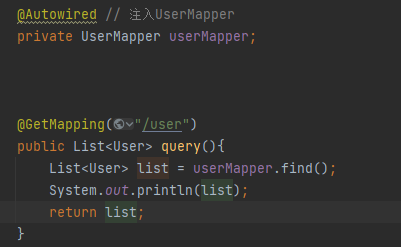
数据库相关的操作都放在mapper里



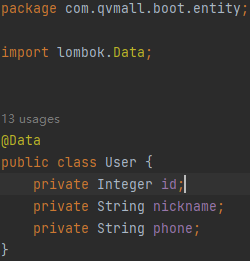
Mapper package - interface usermapper



Controller package - UserController



Entity package - User



## 9 Mybatis多表查询

动机：想知道某一个用户有哪些粉丝，有哪些订单。。。

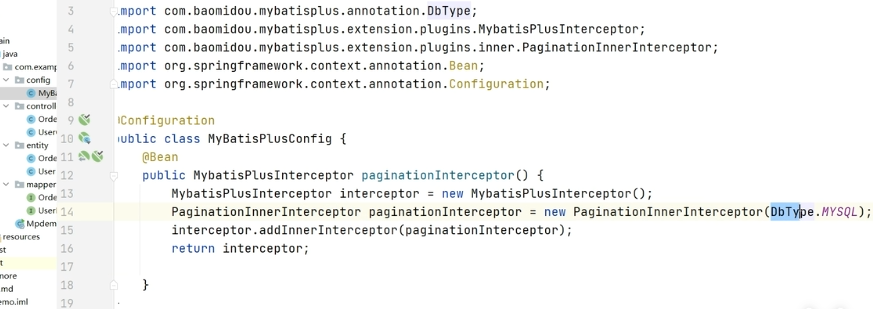
涉及多表的关联查询，如何完成复杂关系的映射



Querywarapper实现条件查询

分页查询

配置





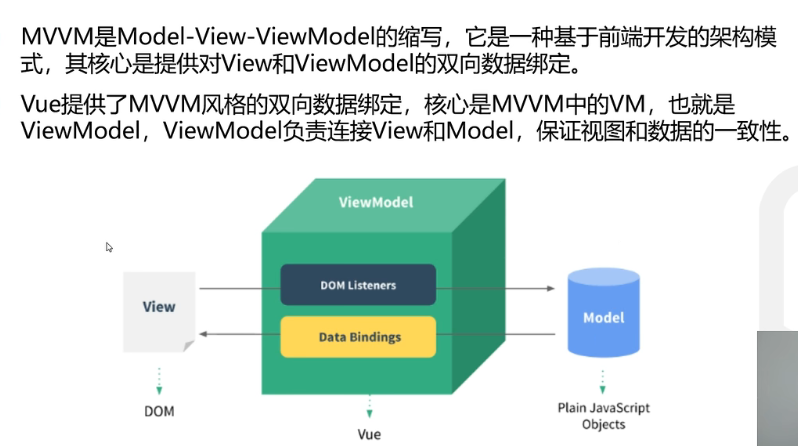
# 前端

## 1 环境准备

编码工具：webstorm

依赖管理：NPM

项目构建：Vuecli



Js中创建model和VM对象，将其挂载到html的view上；

模块化开发：import ... from ...

组件化开发：<selfdesignedPage></selfdesignedPage>

## 2 element-ui

## 3 axios网络请求

使用基于ajax框架的axios发送请求，XMLHttpRequest，可以自动解析json；

导入

### 3.1 发送网络请求：

GET请求：



Post请求。

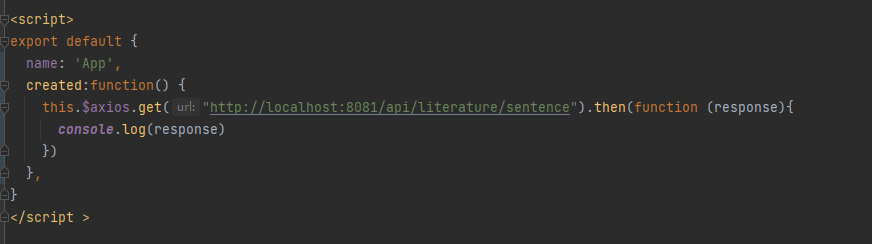
### 3.2 生命周期

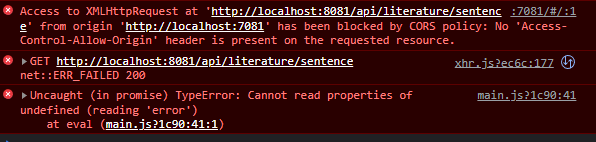
创建（created:function()) - 挂载（mounted:function) - 销毁

一般页面在挂载之前进行网络请求，测试后端正常使用后，要把后端的链接复制到axios

没有说明往哪个端口发，就会往当前的发。

直接这样写会跨域错误：





### 3.3 跨域问题

同源指两个主机具有相同的协议、主机号、端口号，不允许不同源的发送请求，

那么如何解决服务器的授权？

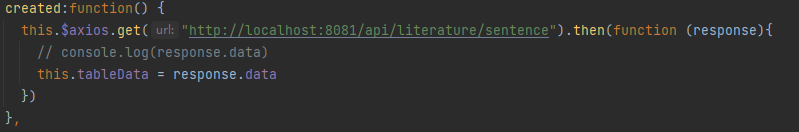
后端实现：CORS有两种请求，简单请求和非简单请求。

控制器前加@CrossOrigin 或 加一个配置类

### 3.4 无法设置undefined

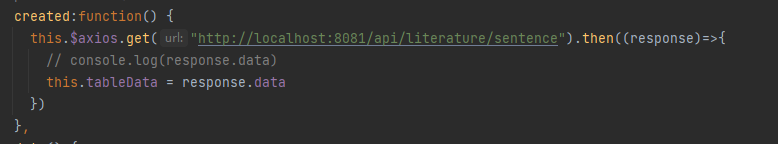
用then((response) => {})

好处是回调函数作用域会继承于父级





解决方案



## 4 路由

组件切换，单页面，用路由控制

动态路由



