

如何2小时以内实现个人相册 Lck

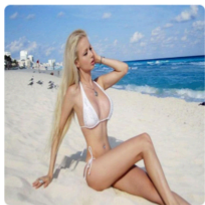
Lck——前大型电商架构师，9年从业经验，有多次主导从0到1的实战经验。熟悉云计算平台技术架构，有丰富的分布式系统、高并发、高可用、混合云系统的设计开发经验，负责和参与过IaaS、PaaS平台的设计实现，对分布式数据库、中间件、消息队列、缓存、虚拟化、容器化等技术有深入的研究。

一、课程介绍

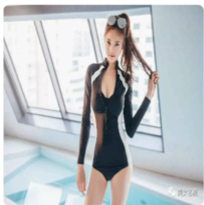
相册

操作


上传




美




漂亮




美女与马




瀑布



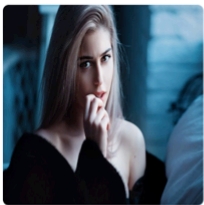
椰树



暖阳



山谷



小姐姐

本课程通过目前市面主流的技术(SpringBoot+JPA+Mysql+前端)实现个人相册系统，包括拖拽上传图片、图片预览、相册展示与管理等功能，除了教会学员如何实现本项目外也可以作为后台管理系统等毕设课题的基础。

后台管理系统 CRUD增删改查 增删查，文件上传

二、课程前置准备

本课程要运行使用需要一定的前置环境需要准备，如果没有接触过的小伙伴们也无需担心

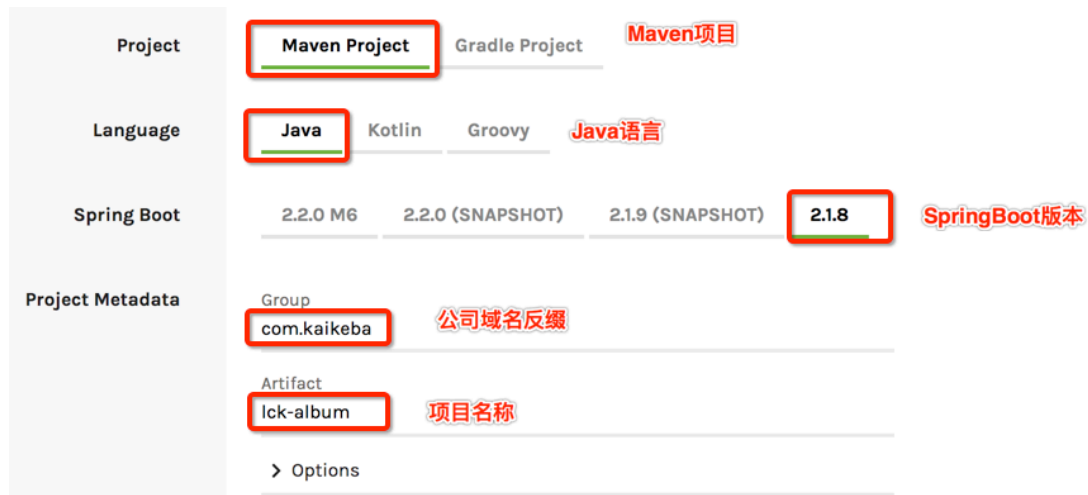
- 1.Java开发环境：JDK与环境变量，编译器IDE
- 2.Maven配置
- 3.Mysql数据库安装与可视化工具
- 4.谷歌浏览器

三、后端实现

1.SpringBoot项目搭建

- 从<https://start.spring.io/>将所需使用的技术的对应依赖选好并生成初始项目

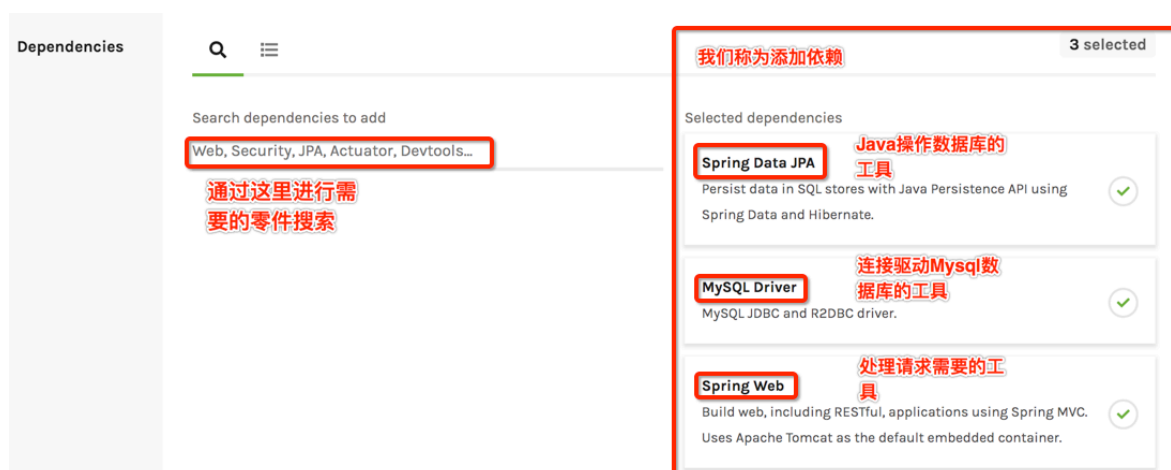
-



The image shows the Spring Boot project configuration form. The 'Project' section has 'Maven Project' selected. The 'Language' section has 'Java' selected. The 'Spring Boot' section has '2.1.8' selected. The 'Project Metadata' section has 'com.kaikeba' for the Group and 'lck-album' for the Artifact. The 'Options' section is expanded.

Section	Configuration	Notes
Project	Maven Project	Maven项目
Language	Java	Java语言
Spring Boot	2.1.8	SpringBoot版本
Project Metadata	Group: com.kaikeba	公司域名反缀
Project Metadata	Artifact: lck-album	项目名称
Options	> Options	

-



The image shows the Spring Boot Dependencies selection screen. The search bar contains 'Web, Security, JPA, Actuator, Devtools...'. The 'Selected dependencies' section shows three selected dependencies: Spring Data JPA, MySQL Driver, and Spring Web. The 'Add dependencies' section shows a search bar and a list of dependencies.

Section	Configuration	Notes
Search dependencies to add	Web, Security, JPA, Actuator, Devtools...	通过这里进行需要的零件搜索
Selected dependencies	Spring Data JPA	Java操作数据库的工具
Selected dependencies	MySQL Driver	连接驱动Mysql数据库的工具
Selected dependencies	Spring Web	处理请求需要的工具

- 解压缩、使用编译器打开项目
 - 项目结构介绍



- 搭建Controller控制层并测试

- 创建一个请求，验证项目能正常工作

- 什么是请求?
 - 创建请求
 - 处理请求
 - 使用SpringBoot处理请求的步骤:
 - 1、创建一个controller文件-》Java专业名词叫做创建一个类
 - 2、给这个类上加上@RestController注解，这样这个文件才能处理请求并返回文字

```
//注解
@RestController
public class HelloController {
    // JPA配置好
    // Java方法
    @RequestMapping("hello")
    public String hellotest(){

        return "Hello wolrd";
    }
}
```

- 3、创建一个方法

这个是方法名

```
public String hellotest(){  
  
    return "Hello wolrd";  
}
```

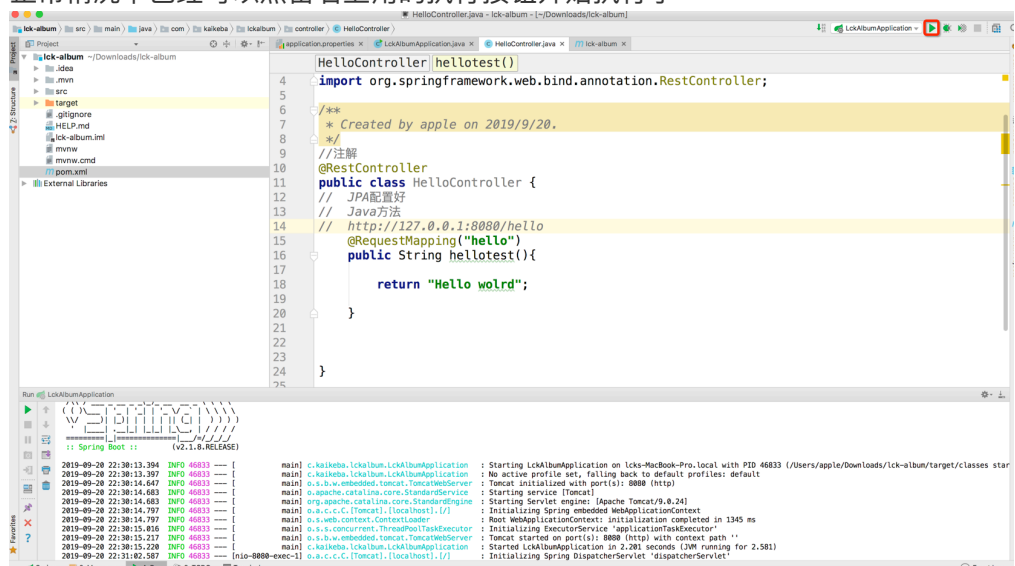
想返回什么内容在""双引号之间写就好

- 让你创建的HelloController这个类知道你发送<http://127.0.0.1:8080/hello>这个请求的时候要帮你执行hellotest这个方法，需要通过@RequestMapping注解来对应

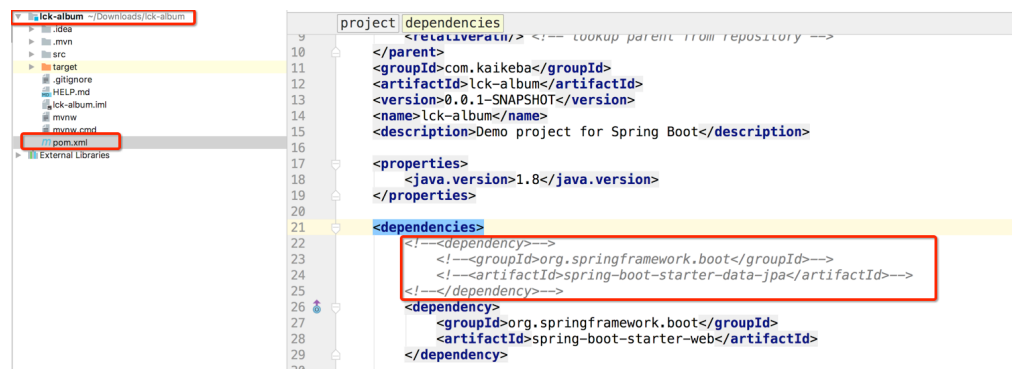
```
http://127.0.0.1:8080/hello  
@RequestMapping("hello")  
public String hellotest(){  
  
    return "Hello wolrd";  
}
```

请求地址和双引号内的内容一致就知道哪个请求需要执行哪个方法了

- 正常情况下已经可以点击右上角的执行按钮开始执行了



- 但是因为我在项目中增加了JPA,但是还没到配置他的时候，所以需要先把他忽略掉才能正常启动项目查看效果，操作是找到项目最外层目录下的pom.xml文件点开，找到jpa对应的dependency依赖注释掉，就可以正常启动项目了



- <http://127.0.0.1:8080/findAll>把数据库中的所有相册数据都查出来
- 类似登录不仅要访问请求，还要把希望给服务器的数据(用户名、密码)传递给服务器(前端需要做的事情)，并且服务器需要进行接收(后端)

- 前端会通过127.0.0.1:8080/add?name=lck1的方式用(?接收的名字=要传输的数据)的形式传给服务器
- 后端只需要在对应的方法的()中间用相同的名字接收下数据就好了

```

@RequestMapping("add")
// http://127.0.0.1:8080/add?name=lck1
public String addAlbum(String name){
    System.out.println("接收到的name是: "+name);
    return "我增加了";
}

```

- 如果要传递多个参数怎么办？
 - 方法上用多个参数接收,万一10个，100个咋整？
 - 使用类来接收
 - 1.创建一个Java Class
 - 2.创建对应的属性，属性名跟传递过来的参数名相同
 - 3.给属性增加get和set方法
 - 4.在请求方法上用类来接收
- 实体类概念讲解与创建
 - 什么是类，什么是对象
 - 人类 能知道有什么行为和特征，但是没法知道是哪个具体的人，无法具体化
 - 对象是类其中一员，周杰伦属于人类，类中某一个可以具体化的个体叫做对象
 - 万事万物皆对象
 - 相册项目中有一个类叫做图片：共同的特征每张图片都需要访问地址和图片名称，编程语言当中叫做属性，每张具体的图片就是一个图片类的对象
 - 人类是什么？
 - 现在在看我讲课的你是什么？
 - 实体类的作用：承载数据
 - 人为什么要有身份证
 - 两个张三怎么分辨
 - 唯一标识 主键id
 - 数据库是啥，怎么拿数据
 - 数据库是软件，当成QQ，驱动 Mysql
 - 数据库当中的数据怎么用Java给到网页，Java实体类来接受成一个个的对象，将结果返

回

- Dao层搭建与JPA使用以及增加、查询功能实现
 - 连接数据库配置

```
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://127.0.0.1:3306/firstjpa?
characterEncoding=utf-8
spring.datasource.username=root
spring.datasource.password=123456
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

- 创建Dao层接口

- 切记不是创建class
- 继承JpaRepository <实体类名,主键类型>
- 在要使用对应dao方法的类中创建对应的属性并在上面增加@Autowired注解就可以在想要使用的方法中使用这个对象了

```
@RestController
public class AlbumController {
    @Autowired
    private PhotoDao photoDao;

    @RequestMapping("findAll")
    public List<Photo> findAlbums(){
        // 使用list集合把一堆图片装下
        List<Photo> all = photoDao.findAll();
        return all;
    }
}
```

- 记得把查询的结果和添加的结果通过return关键字返回，需要修改方法的返回值

```
@RequestMapping("findAll")
public String findAlbums(){
    // 使用list集合把一堆图片装下
    return "我查询了";
}

@RequestMapping("findAll")
public List<Photo> findAlbums(){
    // 使用list集合把一堆图片装下
    List<Photo> all = photoDao.findAll();
    return all;
}
```

参照查询结果的返回类型将方法的类型修改掉

要把从数据库查询到的结果返回给前端

- 通过@RestController注解的作用会把结果以JSON字符串的形式返回出去来配合前端
{ "id":2,"photoName":"lck1","photoUrl":"baidu.com"},
{ "id":3,"photoName":"lck1","photoUrl":"baidu.com"} }

四、前后端结合

- 相册展示和删除功能实现
 - 前面的后台功能实现了，只需要将静态页面拷贝到static目录下就可以使用
- 文件上传功能实现
 - 后台使用MultipartFile类型来接收文件
 - 使用对象的transferTo(file)方法进行文件的存储
 - 将最终的文件访问路径放入存储的对象中，将对象存入数据库
- 相册上传功能实现
 - 结合文件上传和数据添加功能实现

五、结课总结

掌握的技能：

- SpringBoot+JPA+Mysql的Maven项目的搭建与配置 SpringBoot技术应用、Mysql数据库、Maven工具使用
- 什么是请求，请求怎么处理-》SpringMVC
- 什么是类，什么是对象，创建了实体类 -》JavaSE基础，面向对象
- JPA在SpringBoot中的应用-》JPA Hibernate
- JSON字符串
- @Autowired -> Spring

// todo

1.MVC模式

controller -》 service -》 dao

2.文件不是存放在服务器代码上，而是有专门的位置存放，如：文件服务器

3.文件如何访问，使用nginx来访问文件 -》 nginx要如何使用

4.数据返回格式太过粗糙，不会把数据库查询结果直接返回的，需要进行包装，以规定的格式进行返回