**一、Spring boot 的介绍：**

1. Springboot是springmvc的升级版，两者并没有必然的联系，
2. 特点：

①简化配置，简化配置

无需Xml的配置

②一下代框架有可能，微服务的入门级框架 springcloud---springboot

1. 第一个SpringBoot程序

主要学习：五个方面

包括自定义属性配置，Controller的使用，spring-data-jpa数据库的一个操作（重点），数据库的事务管理！

需要用到的知识：

* maven构建项目；了解spring注解；RESTful API
* 不需要学习springmvc
* Java maven等版本保持一致 mvn -version

选create new project--------spring Initializr----直选web------确定---下载好maven包之后----删除不需要的文件：.mvn文件夹，mvnw文件，mvnv.cmd文件

1. 默认maven配置会很慢 推荐使用阿里云镜像配件
2. 本地端口127.0.0.1:8080
3. 第二种方式启动springBoot

命令行进入d:

cd/stronger文件夹

mvn spring-boot:run

启动 了，启动了之后如何停止？Ctrl+c即可停止。

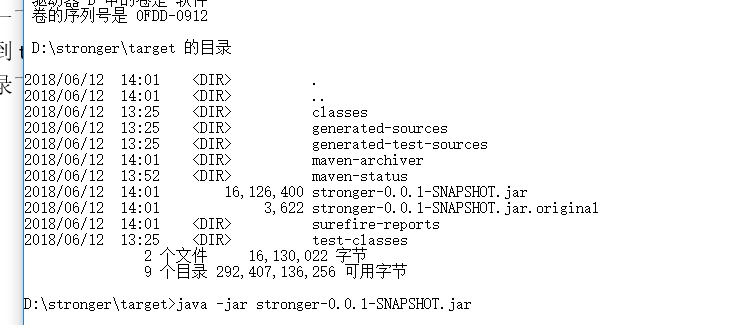
注意：查看stronger里边的内容 dir

第三种方式启动：

先编译一下 mvn install

再进入到target目录下cd target

发现目录下多了一个stronger-0.0.-SNAPSHOT.jar



用java命令来启动 java –jar stronger-0.0.-SNAPSHOT.jar 启动成功！

1. **Spring boot 的配置：**
2. 默认项目配置文件：application.properties里边进行添加端口号，以及context-path（默认url加了一个前缀）

sever.port=8081

server.servlet.context-path=/stronger

//url加了一个前缀

本来是127.0.0.1:8081/ hello运行之后：

此时8080端口不再能访问换成了127.0.0.1:8081/stronger/hello

但这种配置有点复杂，相同的直接竖下来写，故改为用application.yml文件（以后尽量用这种格式）

server:

port: 8082

servlet:

context-path: /stronger

注意：每个冒号必须有一个空格！！！

1. 小栗子，配置文件.yml里边属性

server:

port: 8082

cupSize: B //注意一定要和server中的s对其，不然出错！

age：20

content: “cupSize:${cupsize},age:${age}” //在配置里边直接用配置直接在后面返回content

HelloController里边应该是：

使用@Value(“${age}”)//通过一个注解把配置文件的内容注入到age这个属性变量里边，可以打印试一下：

return age；//只要在注进里边给他配置一个属性就行！

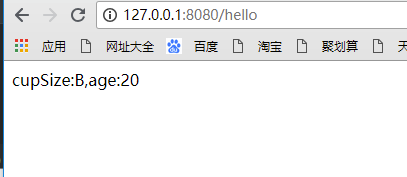
返回多个属性：

return cupSize+age;





content返回结果：



扩展一个人由于很多特质：

① 直接在application.yml文件里边写

girl：

beauty: old

age:18

//相当与给他加一个girl的前缀

② 新建一个类：StrongerProperties里边写：

@Component//注入配置还需要加这个注解

@ConfigurationProperties(prefix=”girl”)//获取girl为前缀的配置

Public class StrongerProperties{

Private String beauty;

Private Integer age;//getter和setter方法

}

④ HomeController中

@Autowired

private StrongerProperties strongerProperties;

@RequestMapping(value=”/say”,method=”RequestMethod.GET”)

public String a(){

return girlProtities.getBeauty();

}

1. 不同环境使用不同配置！
   1. application.yml copy两个application-dev.yml和application-prod.yml

application.yml内容改为：

spring：

profiles:

active:dev //随时切换！

* 1. 第三种方法的启动//切换至8081端口

mvn install

cd target

dir(多了一个)

java –jar stronger-0.0.1-SNAPSHOT.jar --spring.profiles.active=prod

即可

**三、控制器的使用**

1. @Controller注解，处理http请求，不过不加这个注解，浏览器是无法访问到的

@RestController @Controller加上@ResponseBody

1. 注解接着看：

如果单独用@Controller 的话（了解一下）前后分离，只传一些json数据（得再加上@ResponseBody），配合使用使用模板，导致性能降低!

* 1. 需要在pom.xml文件里边加入依赖

<dependency>加入spring官方模板thymeleaf 刷新一下右击 pom.xml 点击maven点击刷新reimport

* 1. 在resource底下新建templates文件夹，下边建立index.html
  2. HomeController返回 “index”;
  3. 成功了！

1. @RequestMapping 配置url映射，通过url访问到我们写的方法

①：如果想要两个，可以写成一个集合：value={“/hello”,”/hi”}

这样：<http://127.0.0.1:8080/hi>

<http://127.0.0.1:8080/hello>

②：给这个类加上映射在最上边加上 叠加起来吧

1. 如何处理url里边的参数
   1. @PathVariable获取url中的数据，直接加id型

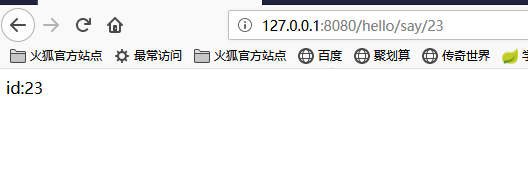
@RequestMapping(vaule=”/say/{id}”,method=”RequestMethod.GET”)

Public String say(@PathVariable(“id”) Integer id){

return “id:”+id;

}

结果：



* 1. @RequestParam获取请求参数的值，

比较传统的url 是 127.0.0.1:8080/hello/say?id=100

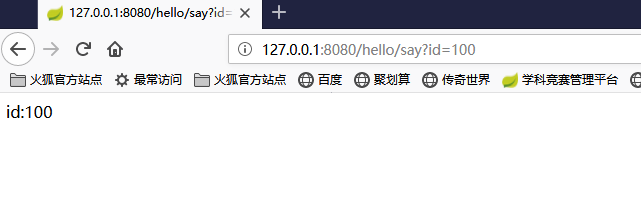
@RequestMapping(vaule=”/say”,method=”RequestMethod.GET”)

Public String say(@RequestParam(“id”) Integer id){

return “id:”+id;

}

结果：

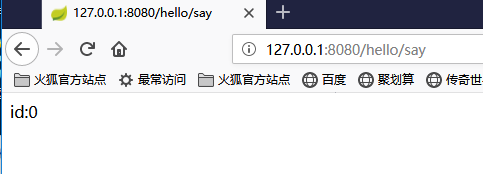


还可以设置@RequestParam的属性：

@RequestParam(value=”id”,required=false,defaultValue=”0”)//默认属性值为0，即使是访问

127.0.0.1:8080/hello/say

结果：



* 1. 组合注解:@GetMapping

@GetMapping(value=”/say”)可以替换

@RequestMapping(vaule=”/say”,method=”RequestMethod.GET”)

同理@PostMapping(value=””)

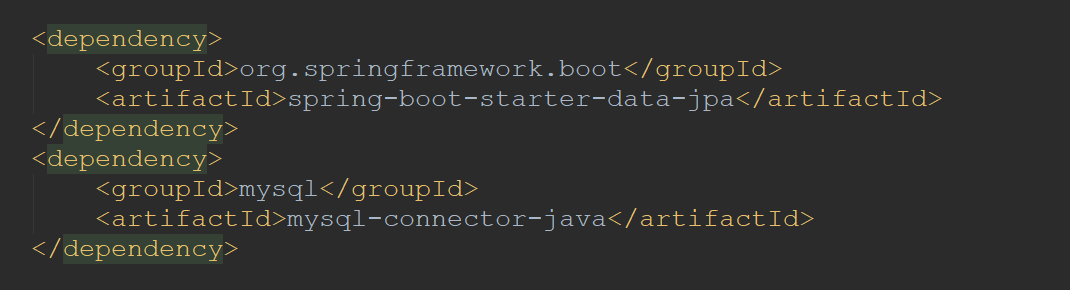
四、数据库操作！！

1. Java客户端使用spring-data-jpa组件，就是spring对hibernate的一个整合。

Jpa，定义了一系列对象持久化的标准，文本上的一个规范：Hibernate、TopLink

2.新建连接数据库

① 首先添加pom.xml两个组件： data-jpa；mysql

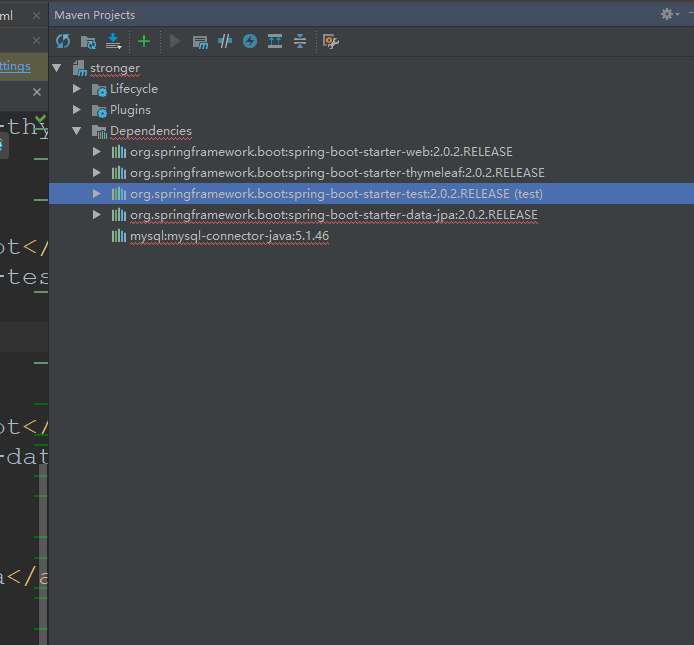


# 其中遇到的问题，红色的线：(Maven配置)Failed to read artifact descriptor for xxx:jar解决方法：

在本地maven包：c:/user/.m2/maven里边找到对应的刚刚加入的包，是否下载下来，

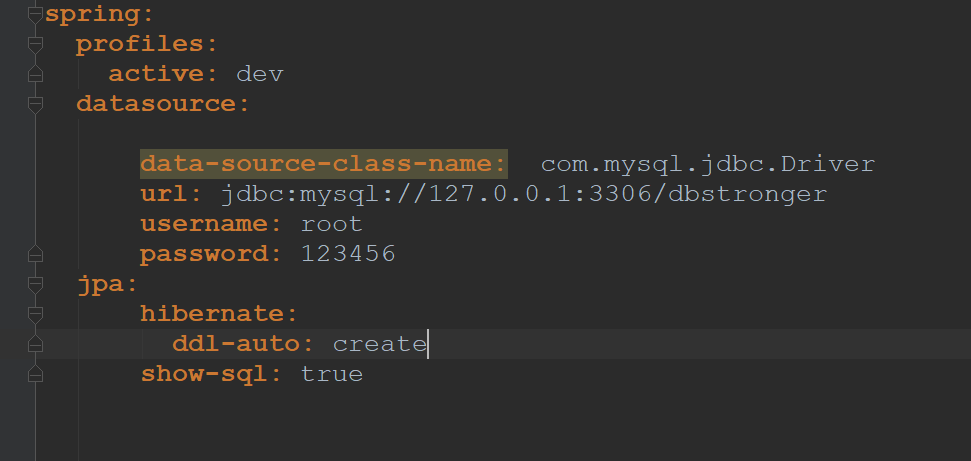
file--setting--maven--然后看一下user setting file和maven home directory

如果没有的话，就删掉 ，点击右侧刷新按钮！重新来



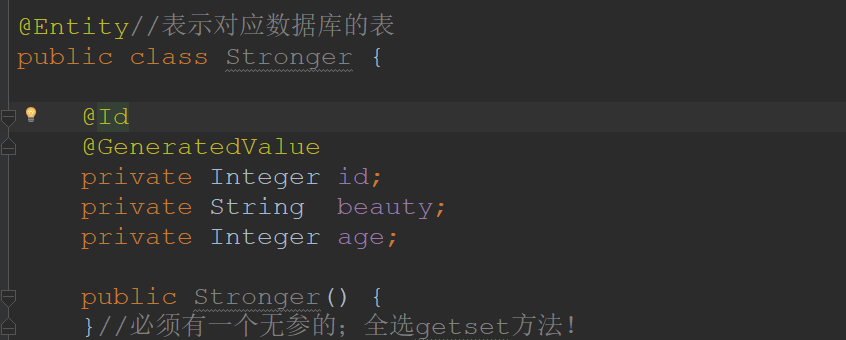
<https://blog.csdn.net/AinUser/article/details/71196256>

1. 在application.yml配置一下数据库；spring-jpa



Spring.datasource

1. 打开mysql新建数据库dbstronger



新建类Stronger

@Entity //对应数据库中的表

@Id//一般用来自增

Alt+Insert //添加一个construction构造方法（无参的！！否则会出错！！）；还有getter 和setter方法（全选）

重启 刷新表出来了！

**Jpa里边的ddl-auto:create**

**//create代表的是每回都新建一张表，之前有这个表会给你先删掉！Drop stable if exist stronger**

**//update第一次运行也会创建对应的表的结构，不同的是有数据不会给你删掉的**

**//create-drop：应用停下来的时候会把表给删掉**

**//none：默认什么都不做**

**//validate：验证与表内的属性是否一致，如果不一样就会报错！**

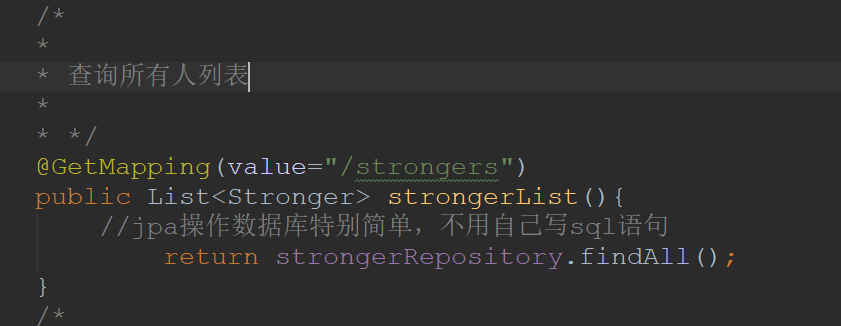
3. 实现以下



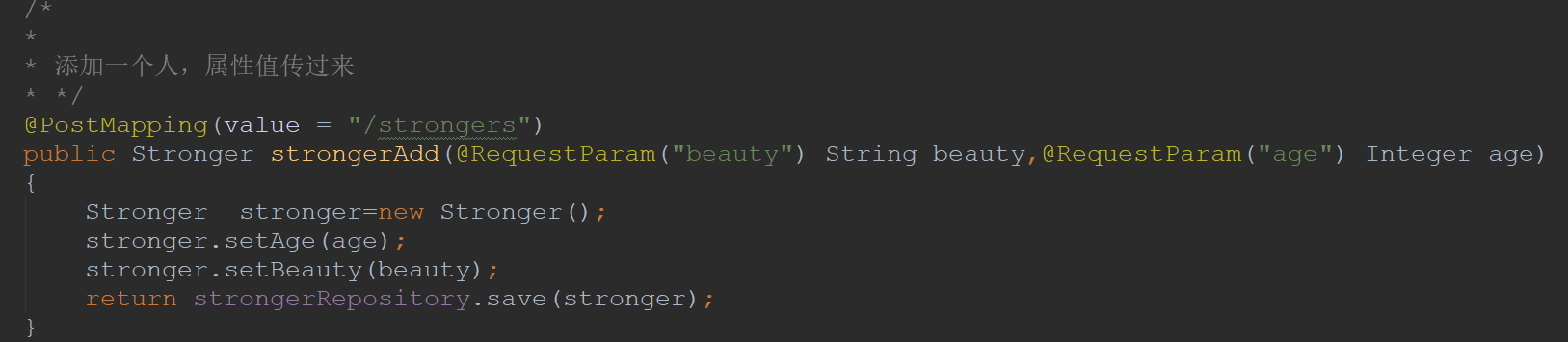
1. Spring-data-Jpa操作数据库特别简单！不用自己写sql语句

需要新建一个接口(interface)：StrongerRepository，继承jpaRepository<Stronger,Integer>

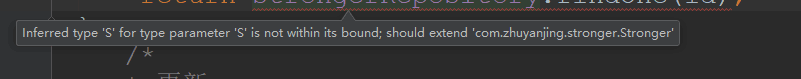
在StrongerController里边就可以直接调用



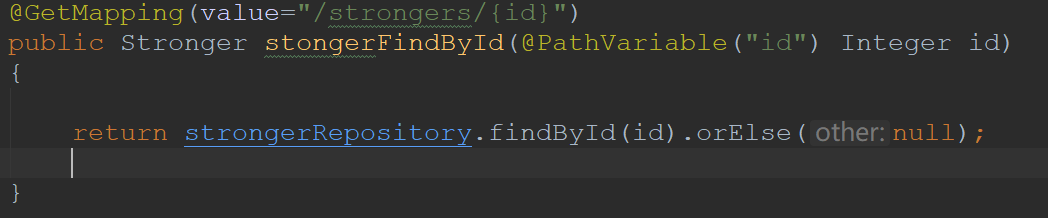
1. 添加一个人，save返回的是添加的对象

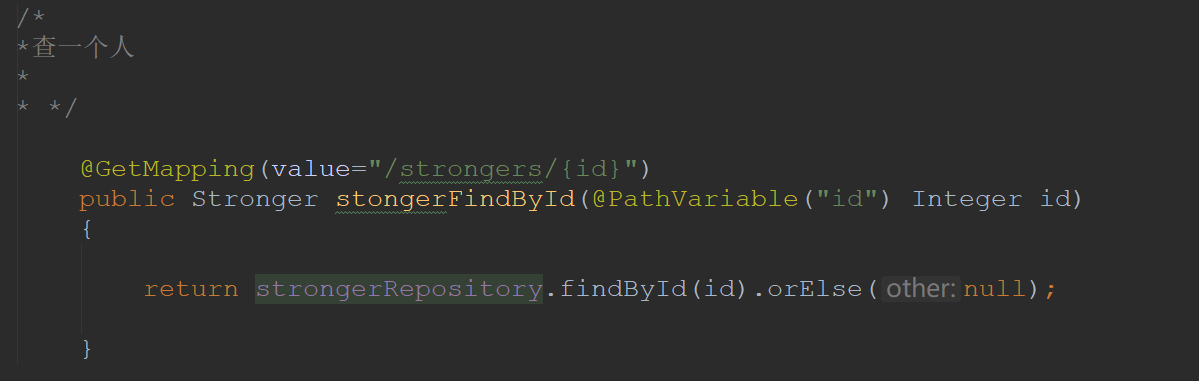


1. 查找一个人这是版本问题findOne()方法，有错

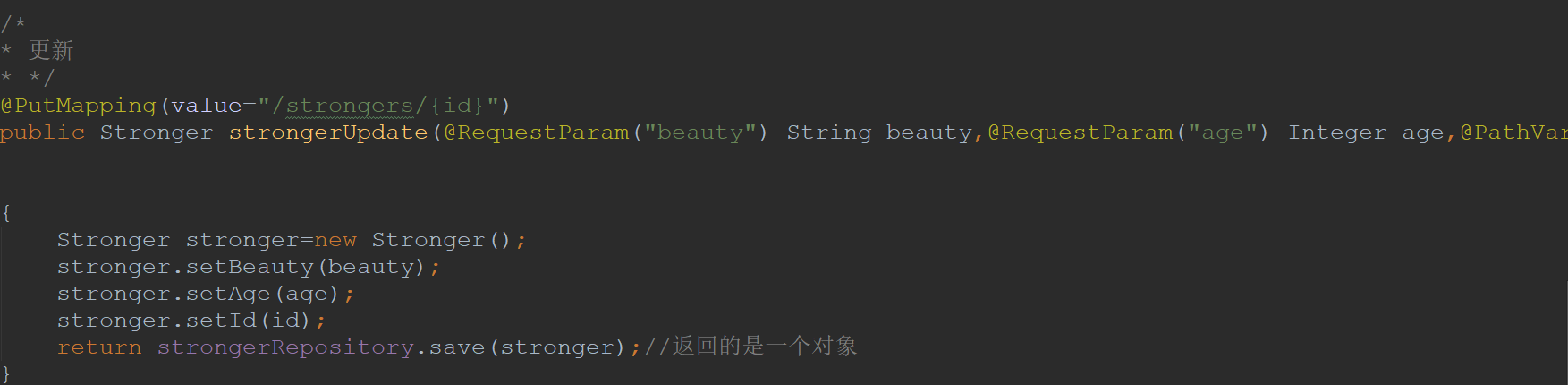


解决方法，可以改变springboot的版本，也可以改代码：

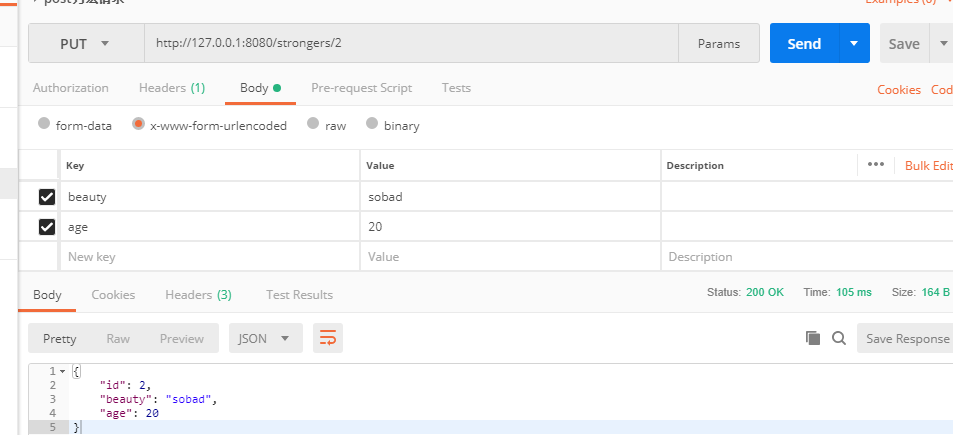




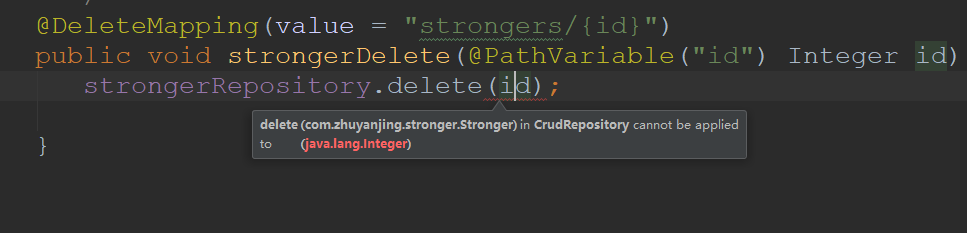
1. 更新数据库

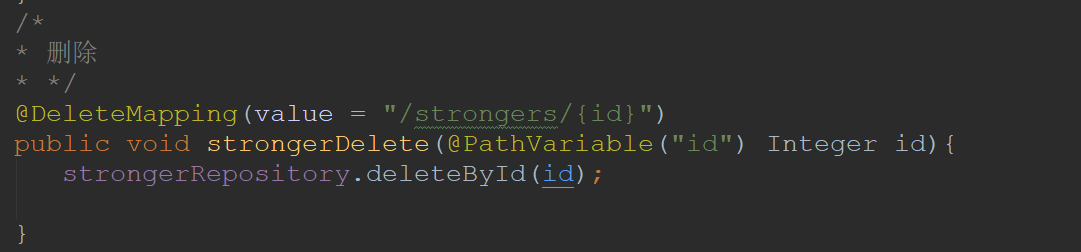


（选择第二个），更改属性



1. 删除（换成这个deleteById（）就解决了，返回为空）





1. 通过年龄查找首先在strongerRespository接口里边添加

public List<Stronger> findByAge(Interger age);

然后在StrongerController里边添加：

@GetMapping(value=”/strongers/age/{age}”)

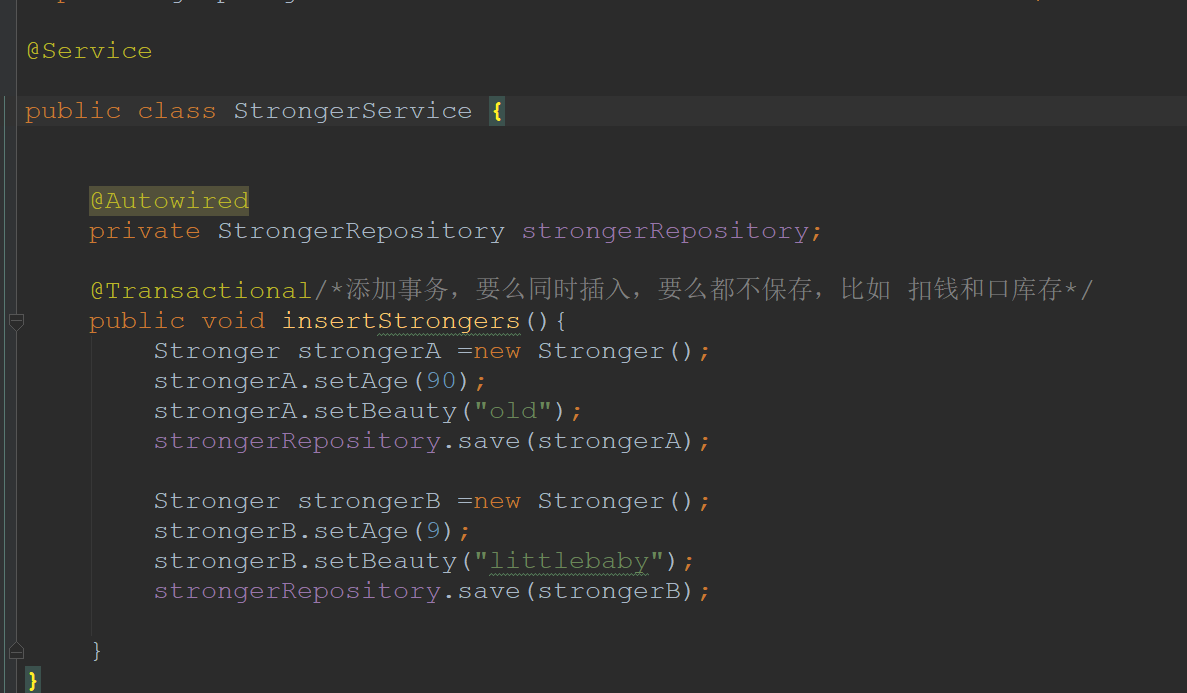
public List<Stronger> getListByAge(@PathVariable(“age”) Integer age){

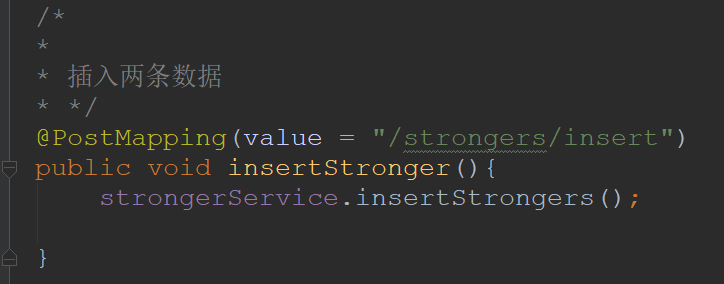
return strongerRepository.findByAge(age);

}

**五、事务管理，**

只有查询的时候不需要插入事物（插入两条数据要么同时，要么都不插入，比如扣钱和扣库存）

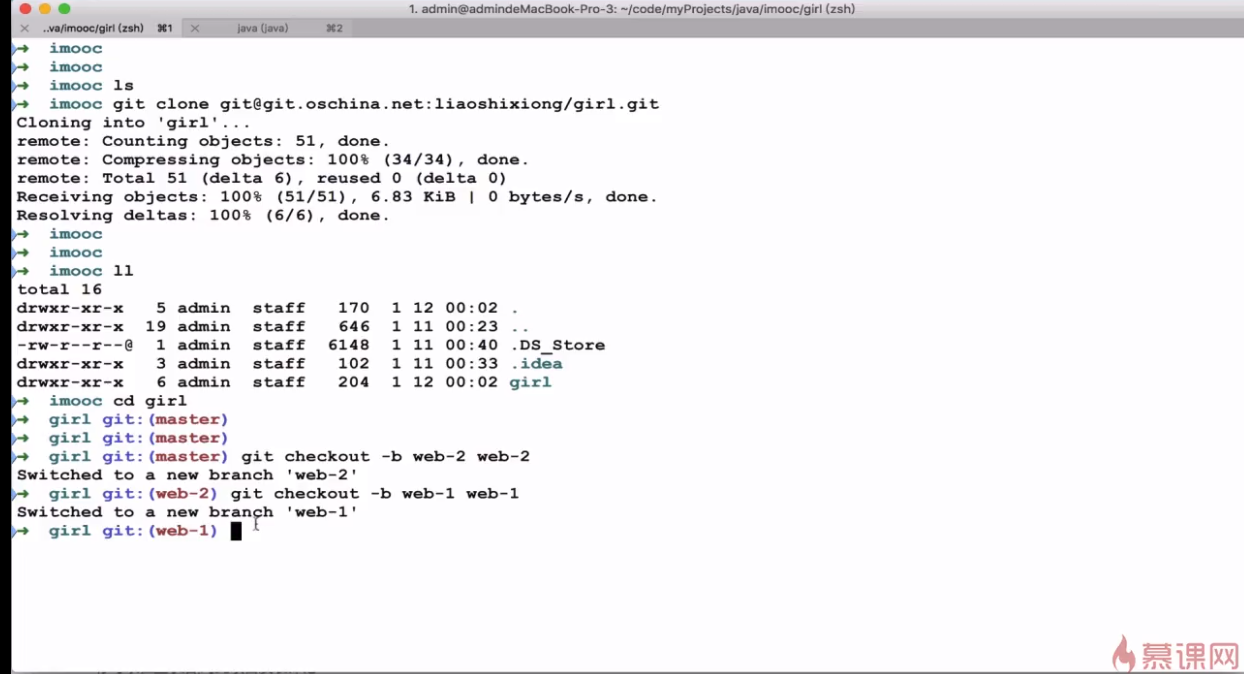




有一条出错的时候，另一条也存进去了



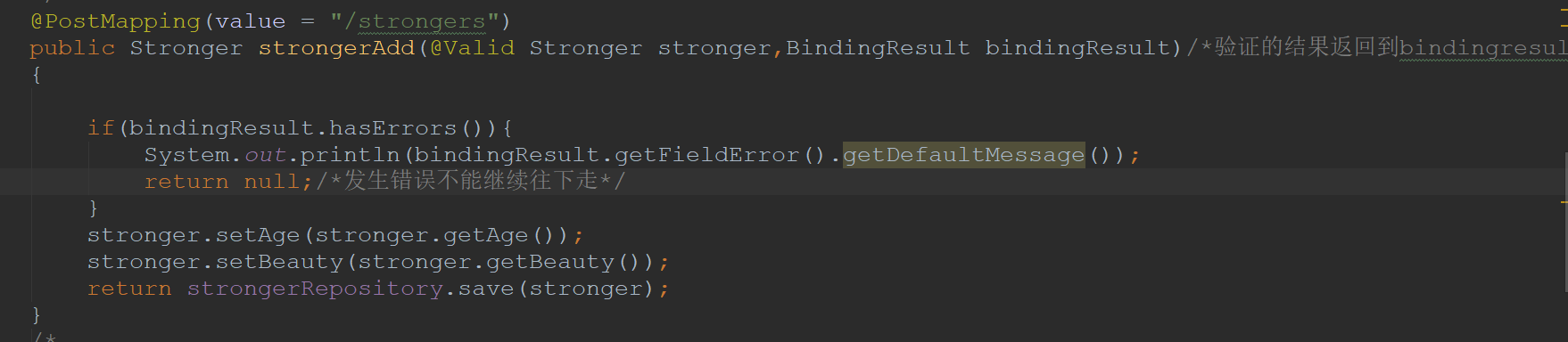
加入事务@Transactional适用于：同时操作多条数据，操作一条数据也会出现错误



Git下载代码！！！

**六、@Valid表单验证**

**1. @Min(value=18,message=”年龄不够”)，注意在后面添加@Valid**



**2.使用AOP:统一处理请求日志**

**AOP是一种编程范式（程序的设计思想，与语言无关）//面向切面**

**面向过程（C语言）：**

**面向对象（c#）：**

**如果登陆才能进行以下的操作每一个方法都加一个判断，太麻烦！！使用AOP解决这个问题**

* + 1. **Pom.xml文件里边加入AOP的依赖**
    2. **往常做法在启动类上边加一个注解，但是AOP不需要加**
    3. **建立我们的处理文件：aspect javaclass类文件，**

**@Aspect //aspect文件**

**@Component //将这个文件引入spring容器中**

**@Before //在http请求到这些方法之前就把他们记录下来**

**先定义方法//方法定义完之后就有提示**

@Before(“execution (piblic \*包名.controller.StrongerController.strongerList(..))”)

//..表示只要是这个方法任何参数都会被拦截

@Before(“execution (piblic \*包名.controller.StrongerController.\*(..))”)

//\*所有的方法都被拦截！

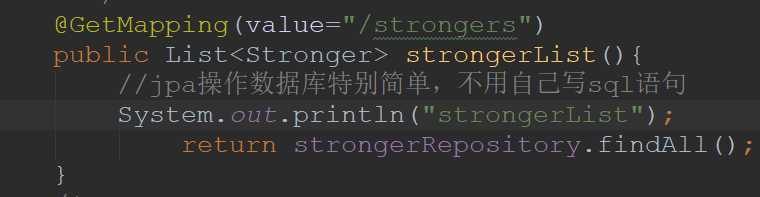
**Public void log()**

**{**

**System.out.println(1111111);**

**}**

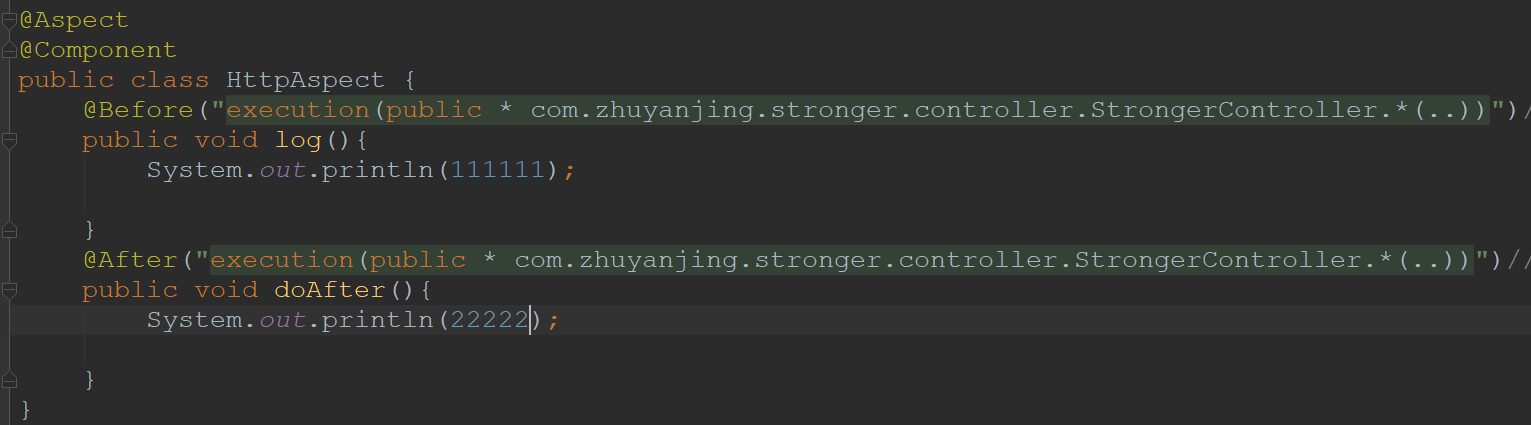
**检验：如何知道成功拦截可以在strongerList()方法里边打印**



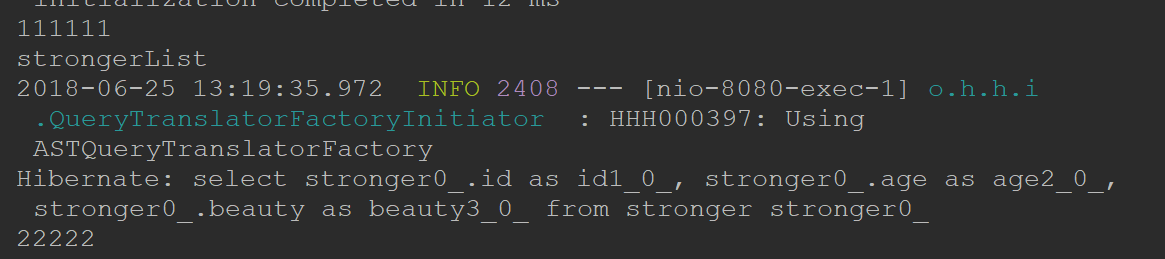
**结果：是先打印: 11111**

**Then : strongerList**

**联想：既然有before 就会有after，**



**三者的先后顺序，**

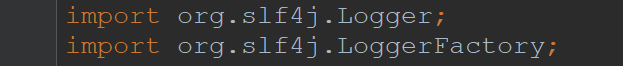


**但这样会不会很麻烦，代码的冗余！**

**改进：用到@Pointcut注解，结果一致**



**3.打印数据的时候采用日志打印：**



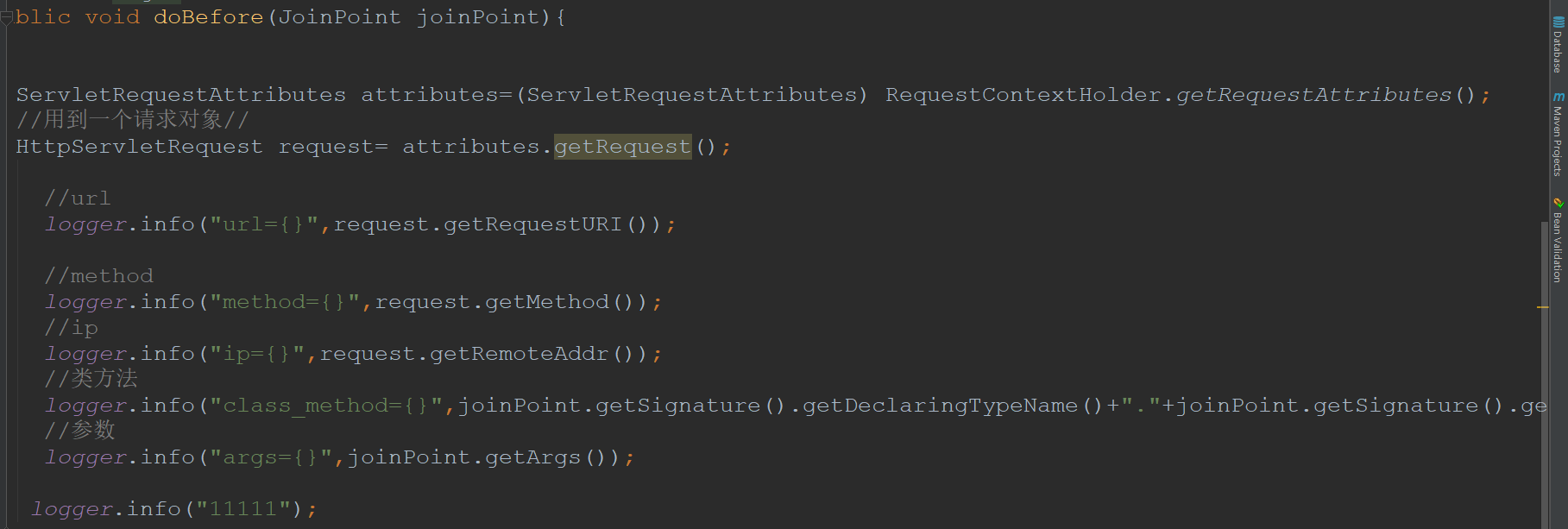
private final static Logger *logger*= LoggerFactory.*getLogger*(HttpAspect.class);注意，这儿得换成类的名称

**//加入日志的格式**

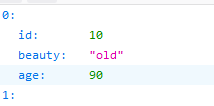
**logger.info(“1111111”)//打印数字**

**logger.error()//打印错误信息**

1. **①获取http的内容：请求url，method，**



**②如何获取以下内容**



* **利用**@AfterReturning注解

@AfterReturning(returning=””,pointcut=”log()”) Public void doAfterReturning(Object object)

{

Logger.info(“response={}”,object)

}

* **此时把对象打印出来了，单打印具体内容需要去实体里边加一个toString()方法,再将上述改为：**

@AfterReturning(returning=””,pointcut=”log()”) Public void doAfterReturning(Object object)

{

Logger.info(“response={}”,object.toString())

}

**六、统一异常处理**

**1.异常处理：人有悲欢离合，月有阴晴圆缺**

**2.为什么要统一异常处理：**

**<10 小学 16> >10初中**

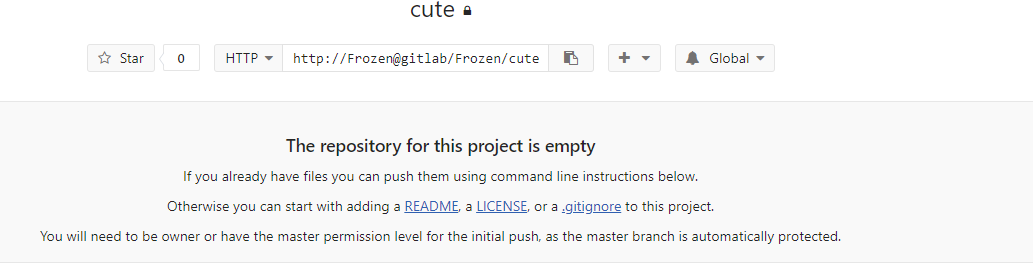
**七、Gitlab 有关知识**

**修改mysql密码：**

|  |  |
| --- | --- |
| 2 | MySQL -u root  　　mysql> **SET** **PASSWORD** **FOR** 'root'@'localhost' = **PASSWORD**('newpass'); |

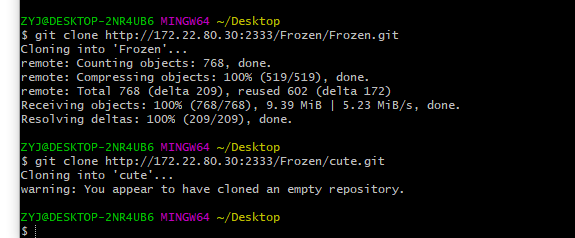
**1.克隆项目：账号：Frozen**

**密码：zyj12231**



**代码：**

**克隆：git clone** [**http://172.22.80.30:2333/Frozen/cute.git**](http://172.22.80.30:2333/Frozen/cute.git)



**2.还是得用ssh的形式：上传文件**

**教程：参照**[**https://blog.csdn.net/litianxiang\_kaola/article/details/74075151**](https://blog.csdn.net/litianxiang_kaola/article/details/74075151)

* 1. **先将创建密钥，key粘贴到即**把id\_rsa.pub里的内容粘贴到gitlab密钥中SSH keys中
* cd ~/.ssh/

如果提示 “ No such file or directory”，你可以手动的创建一个 .ssh文件夹即可

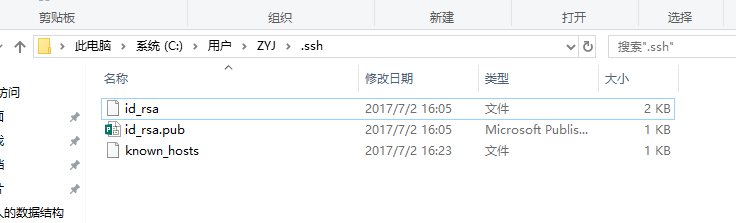
    mkdir ~/.ssh

* 配置全局的name和email，参照你创建的工程Git global setup

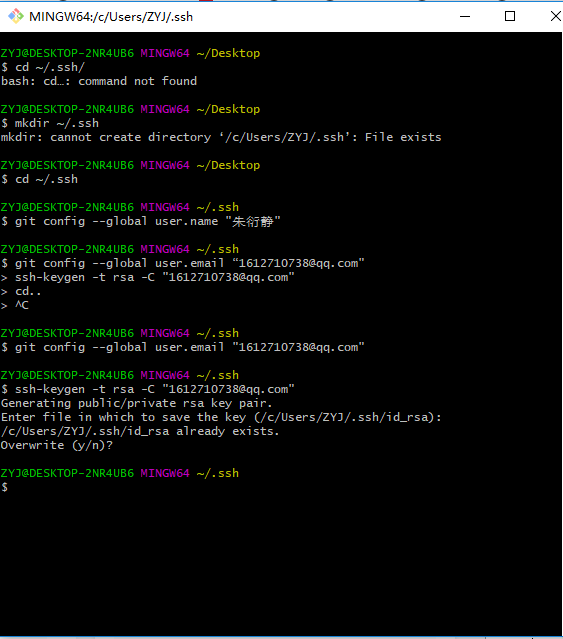
     git config --global user.name "朱衍静"  
     git config --global user.email"1612710738@qq.com"

* 生成key

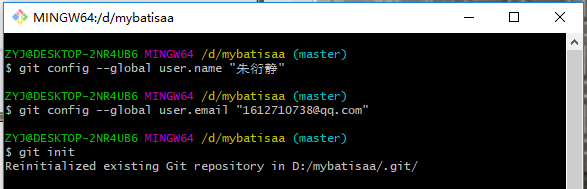
      ssh-keygen -t rsa -C "1612710738@qq.com "

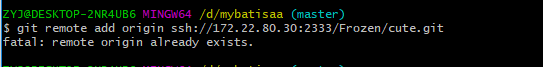
* **粘贴上去（**把id\_rsa.pub里的内容粘贴到gitlab密钥中**）**
* 

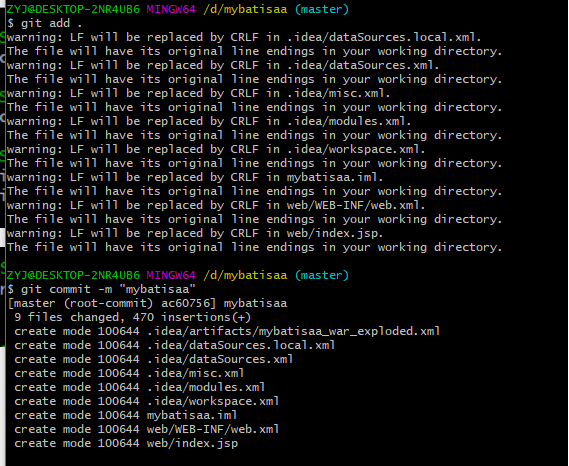
**以下是艰苦历程（早就存在了）：**

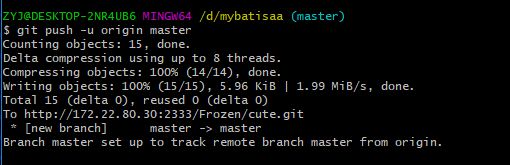


* 1. **开始上传，**右键要上传的项目的文件夹，选择Git Bash Here



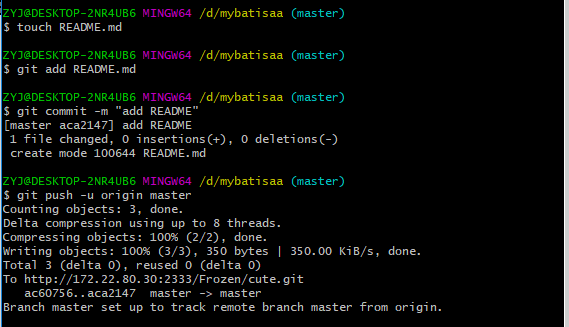






**上传成功！**

**添加README.md文件**



**结束了之后需要安全exit退出！**

**计算机原理**

[**https://wenku.baidu.com/view/dc7d26d29f3143323968011ca300a6c30c22f161.html**](https://wenku.baidu.com/view/dc7d26d29f3143323968011ca300a6c30c22f161.html)