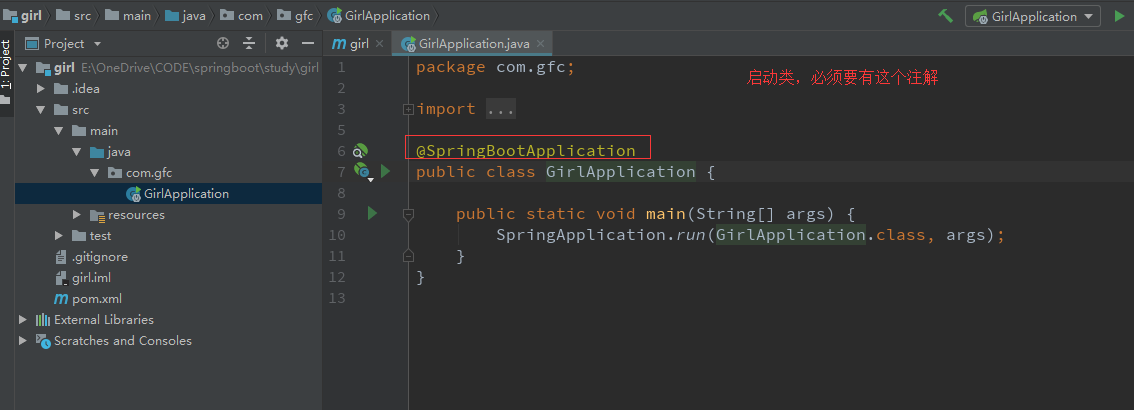
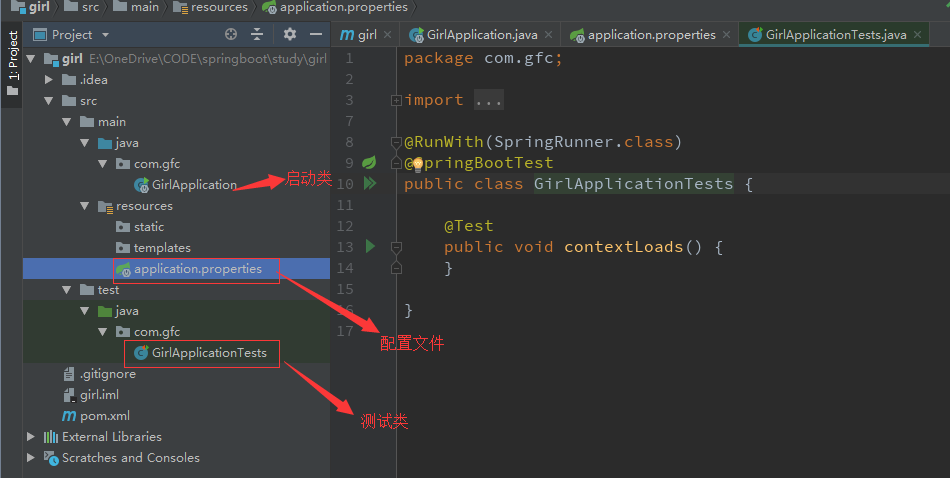


修改maven镜像仓库

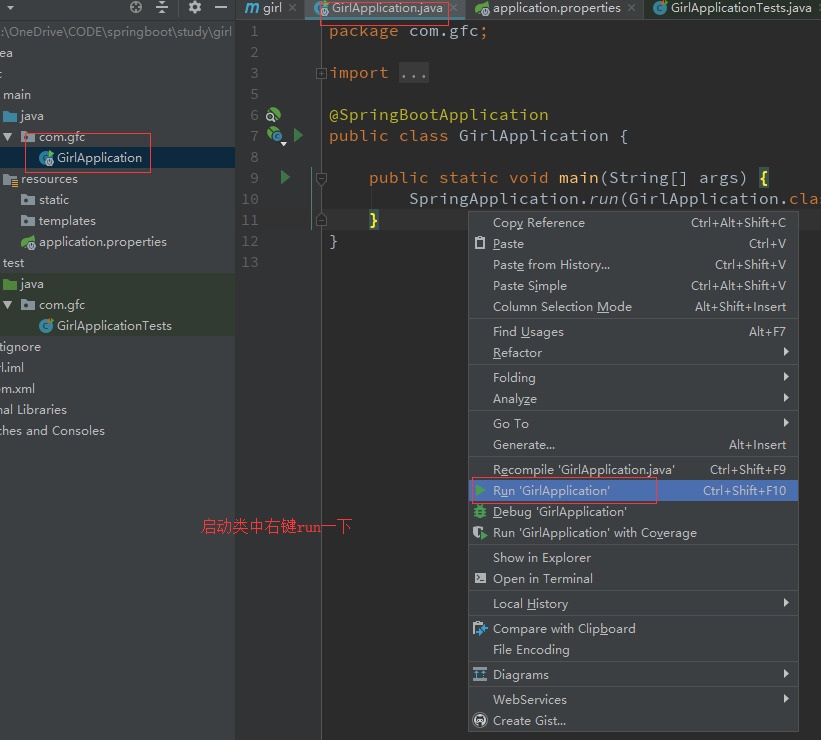
<https://yq.aliyun.com/articles/78124>

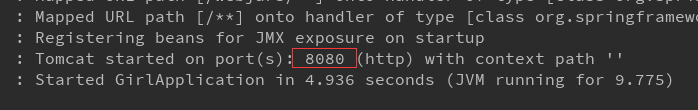


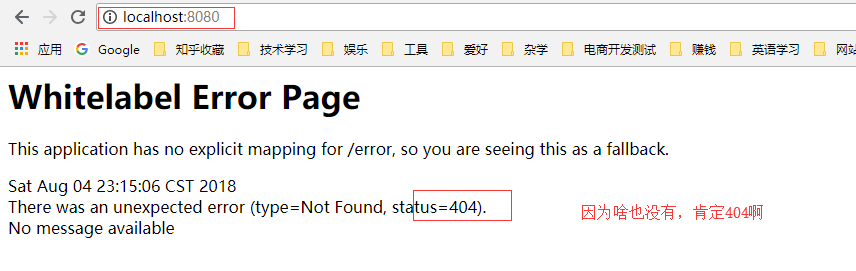


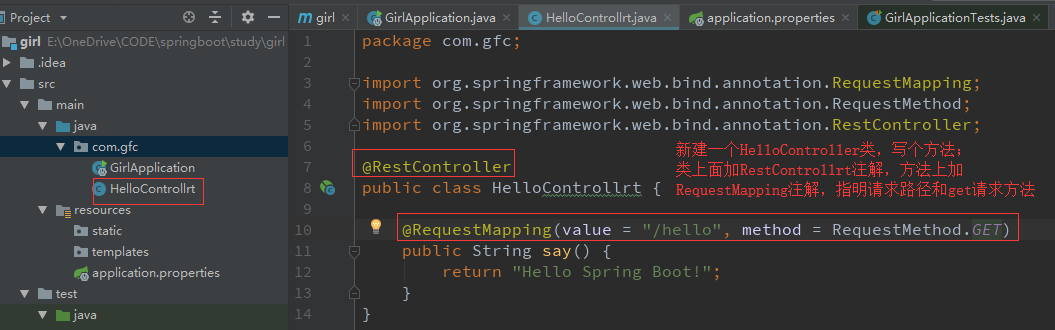


启动方式：

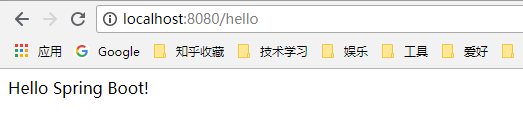
1、







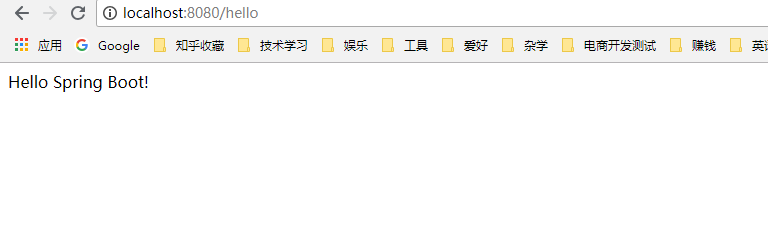
重新到启动类里启动springboot项目，在浏览器中访问一下：



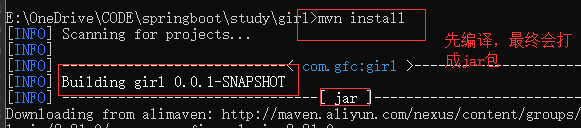
2

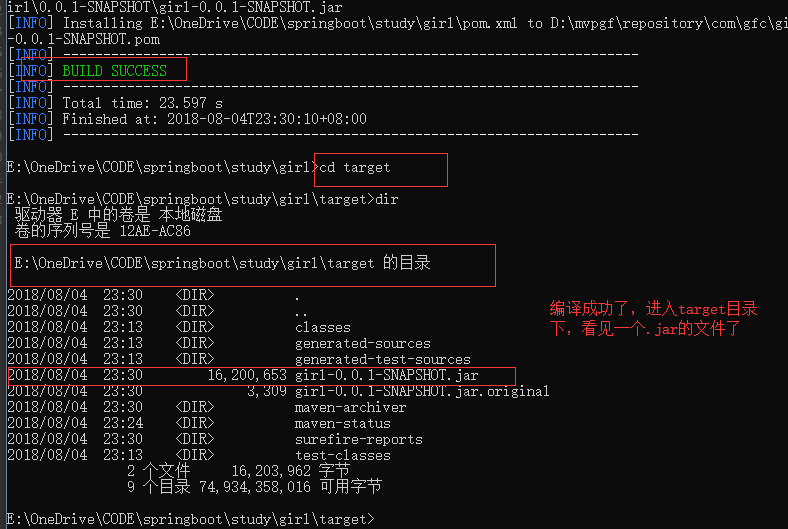
命令行方式

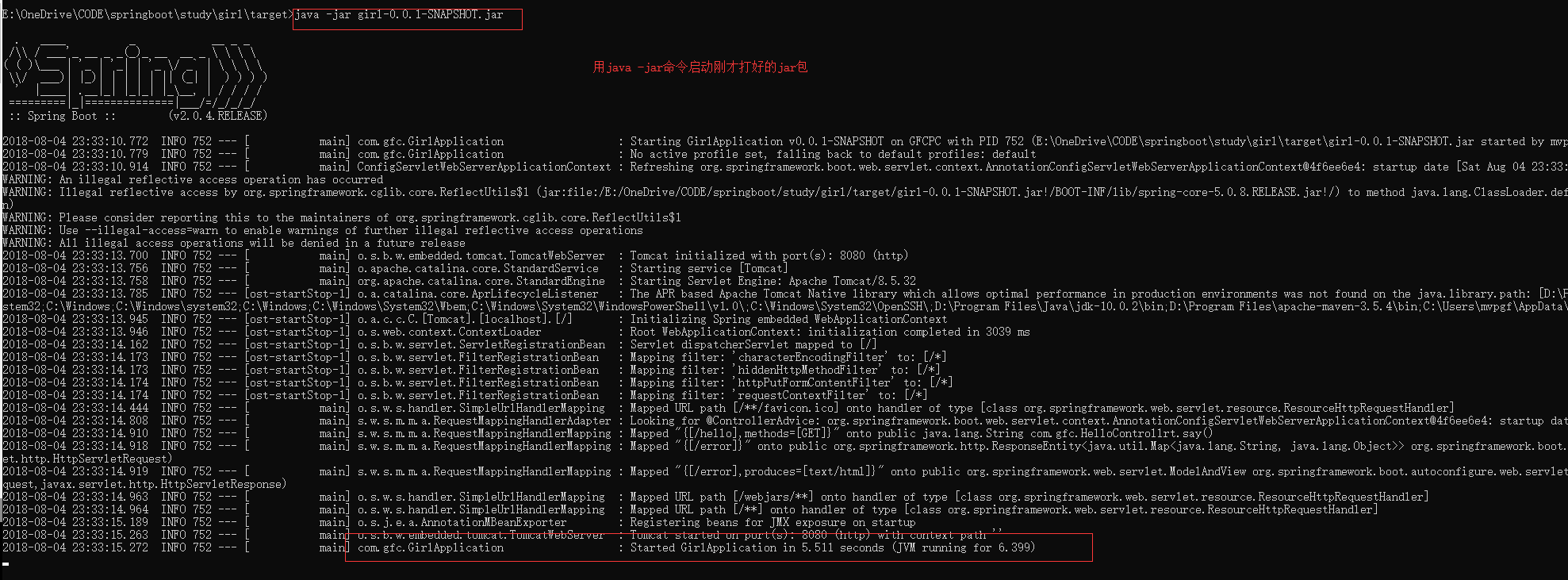


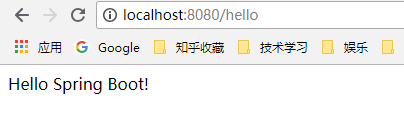


3

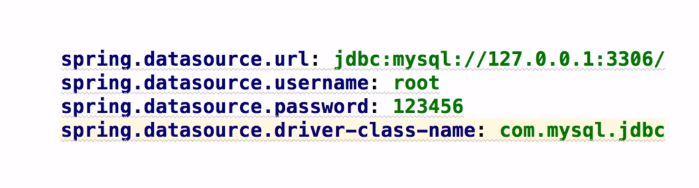


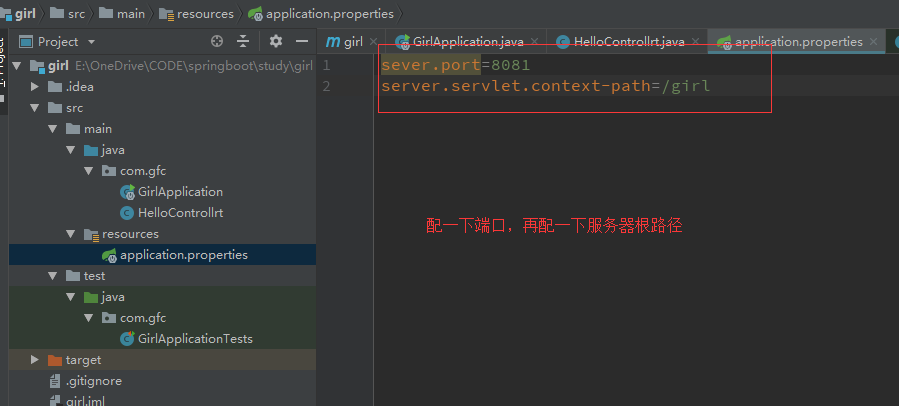


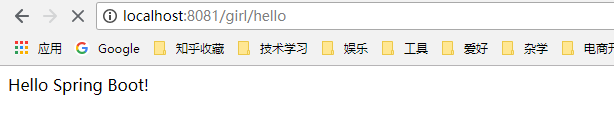


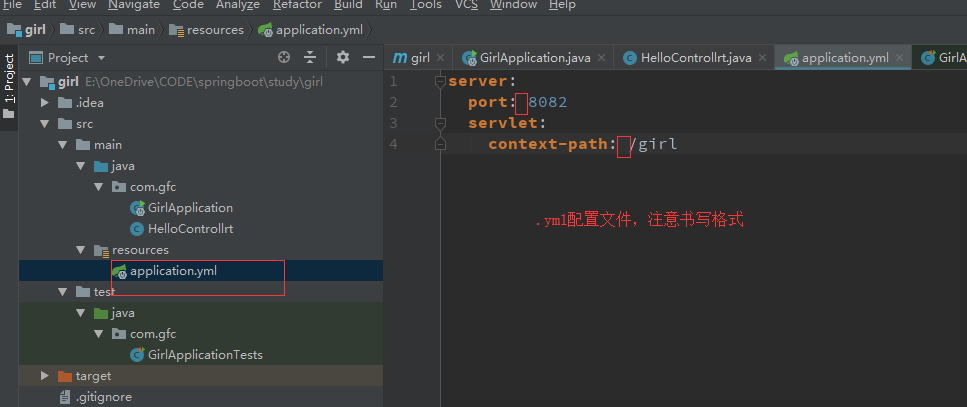


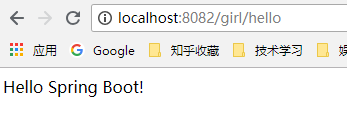
属性配置



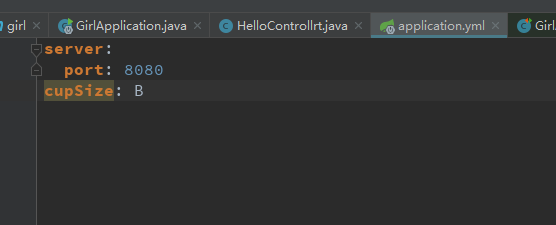


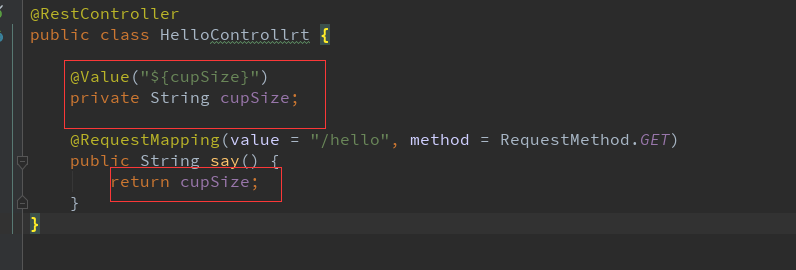


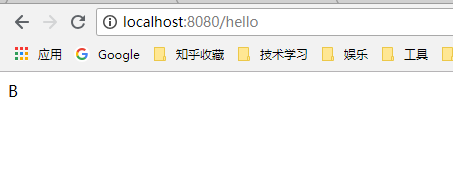


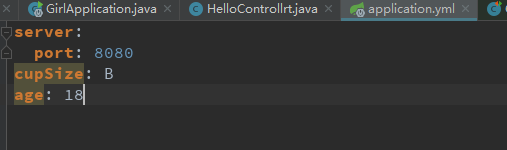


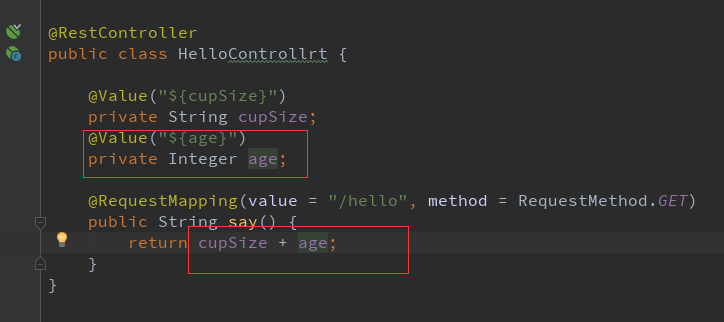
配置文件：

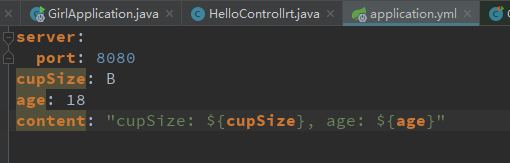


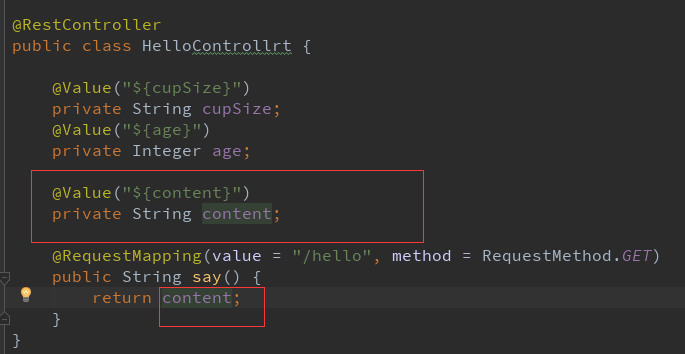




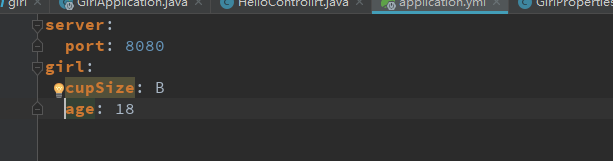


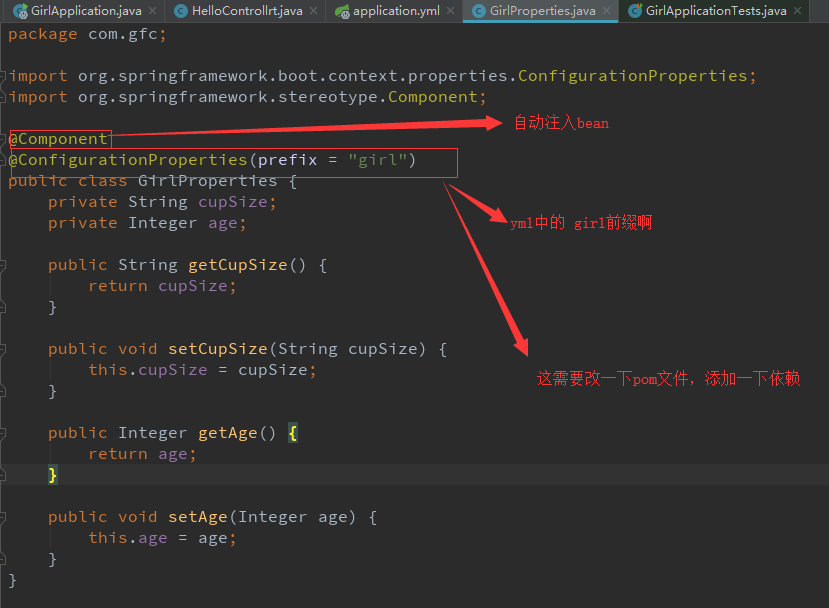






这么写真累！

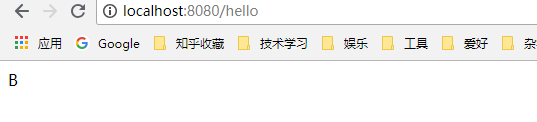




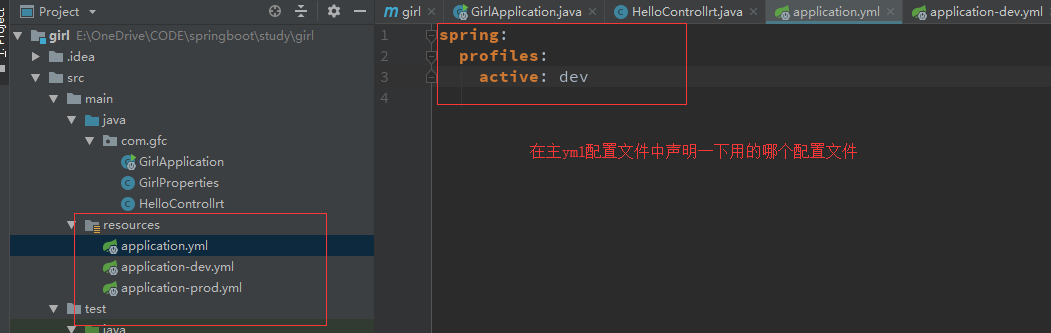


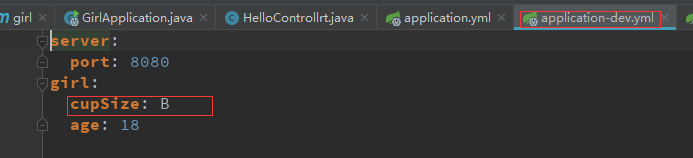


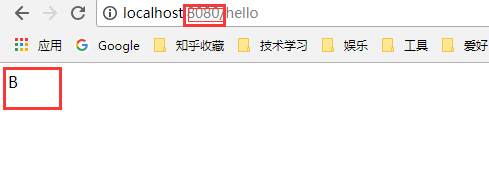
测试一下：

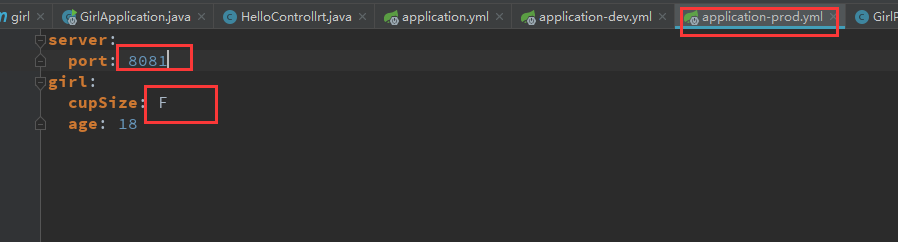


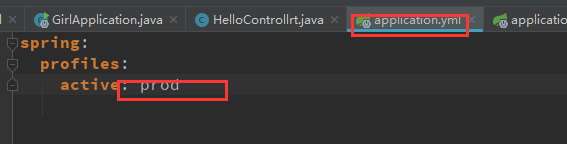
配置文件隔离化（生产开发测试。。。。。）

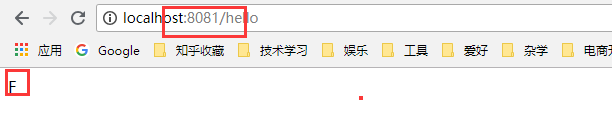






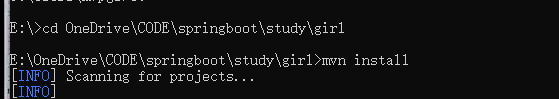






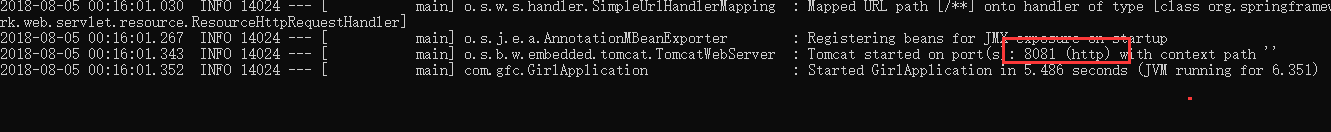
如何同时启动两种环境下的项目呢？之前介绍了三种启动spring boot的方式：

首先用打jar包的方式：



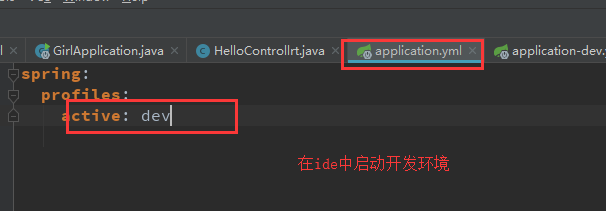
进入taget目录，启动命令有点变化了：

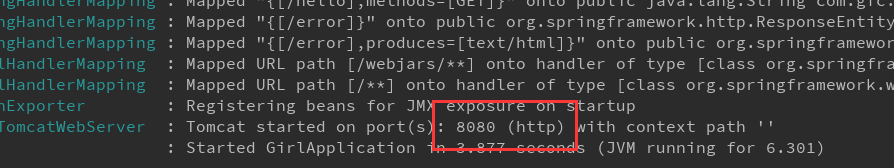




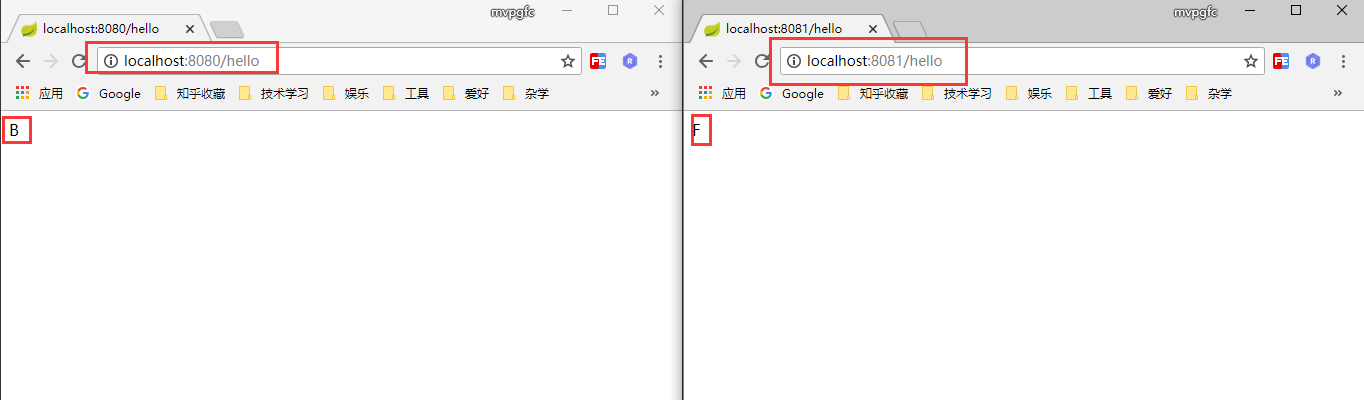
此时生产环境8081端口启动成功

然后





同时启动成功

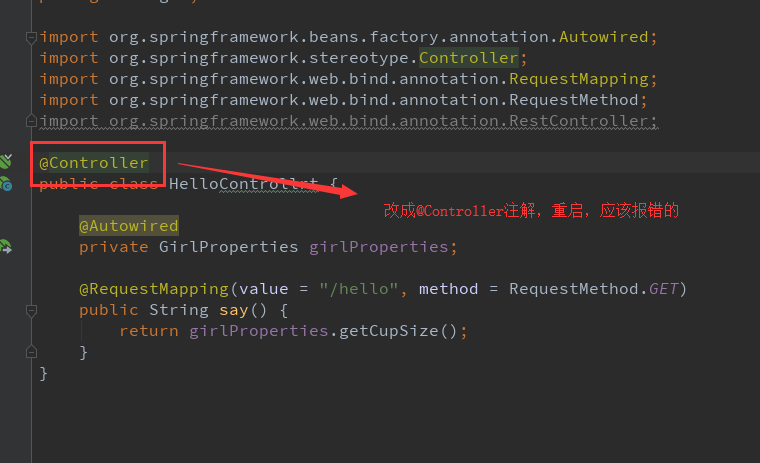


总结



Controller的使用

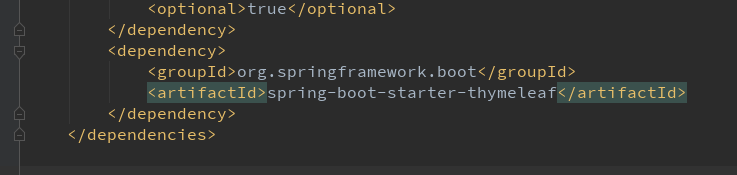




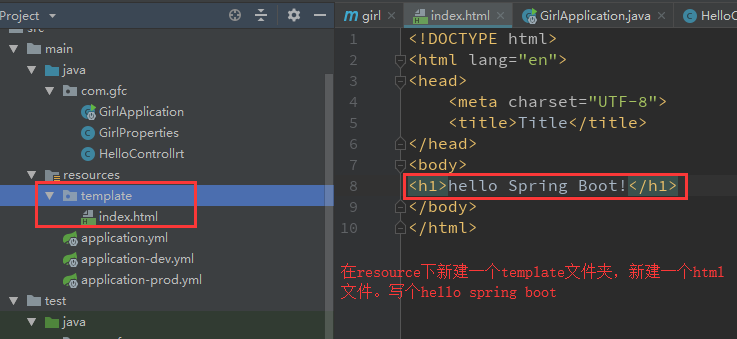
因为Controller的使用需要模板，

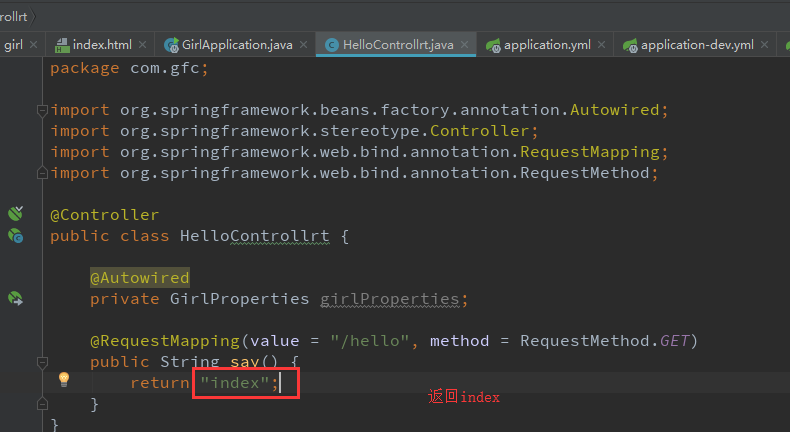
解决方法：

Pom.xml

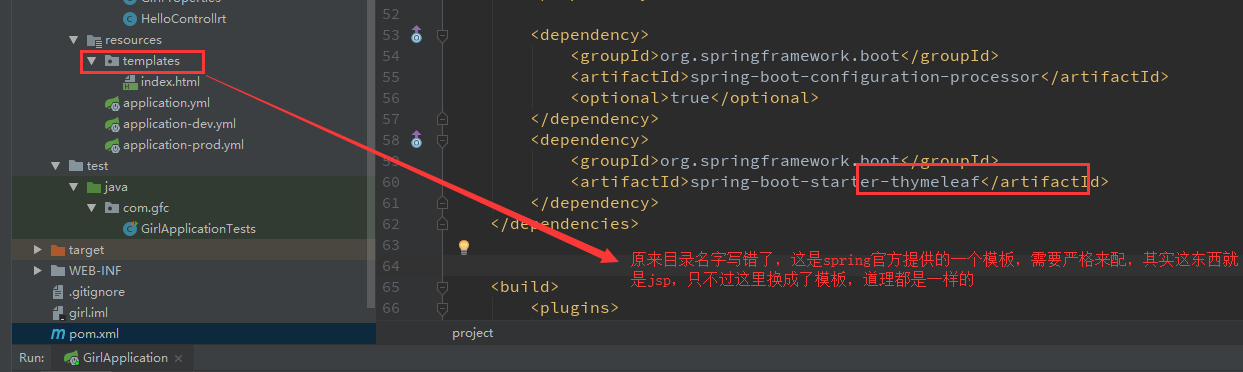


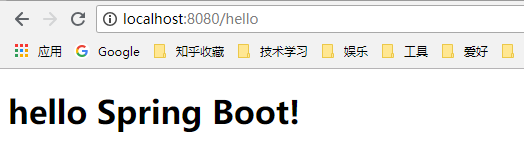
来个HTML文件









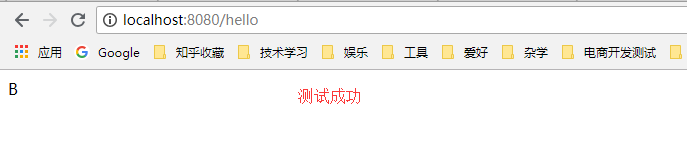


就是演示一下，模板现在都不用，现在都是前后端分离开发，这里就是为了介绍一下controller注解。



那么不用模板咋办呢？



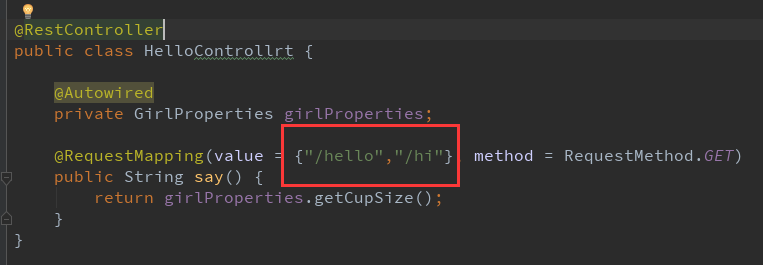


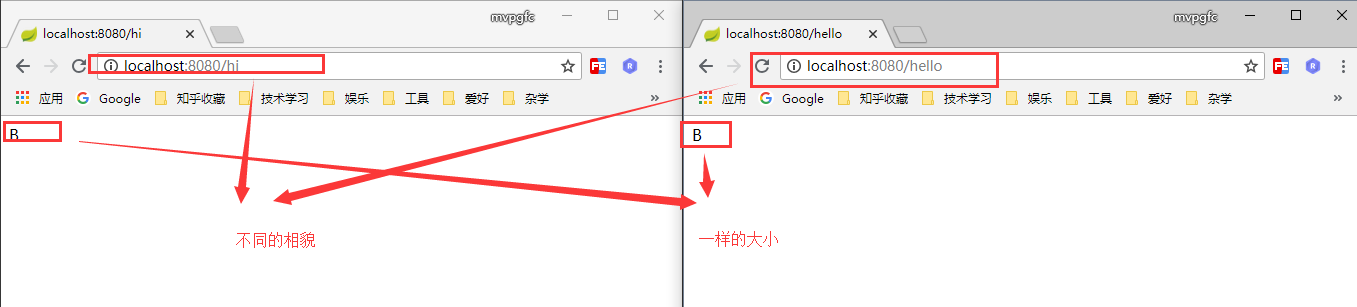
以上内容可以被淘汰了，直接用一个注解就行了，@RestController

@RestMapping

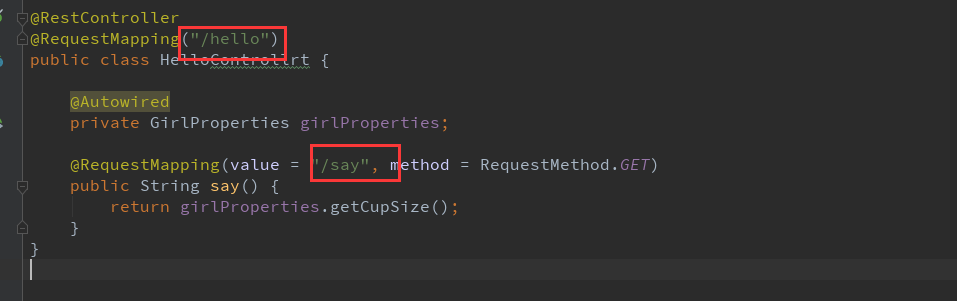
需求来了，我想8080/hello和8080/hi都显示这个B咋办啊。

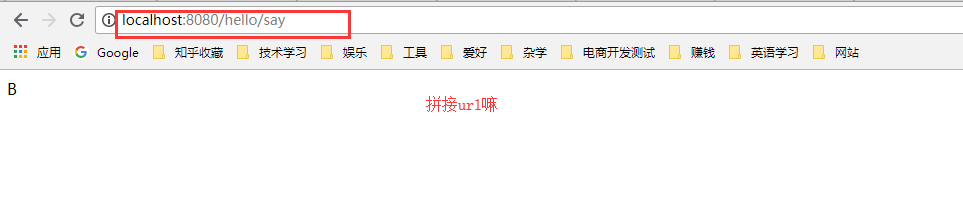
@RestMapping注解可以写集合的





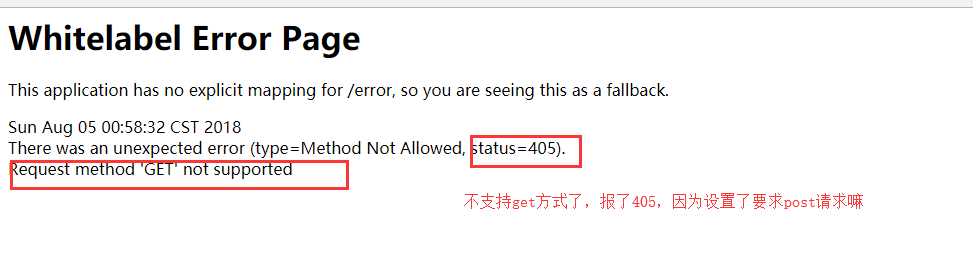
还能干啥呢，设置根路径

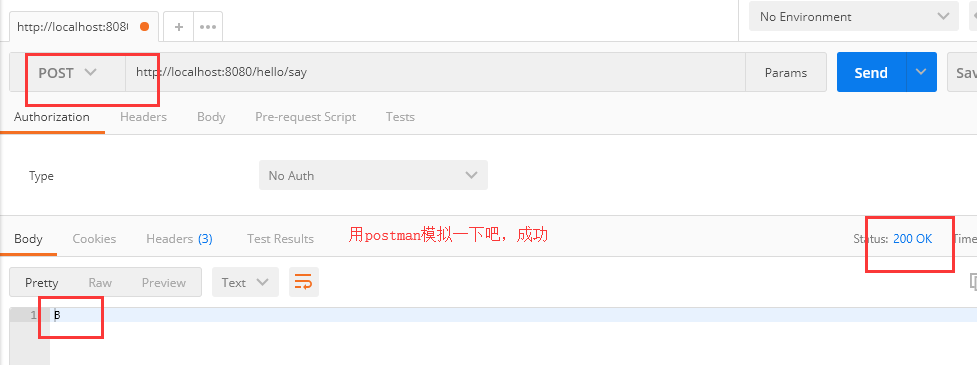




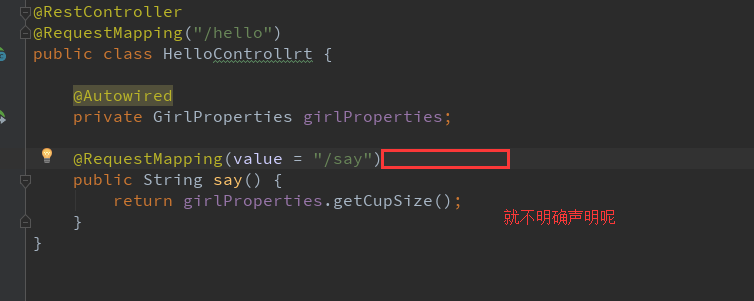
试试请求方式的改变

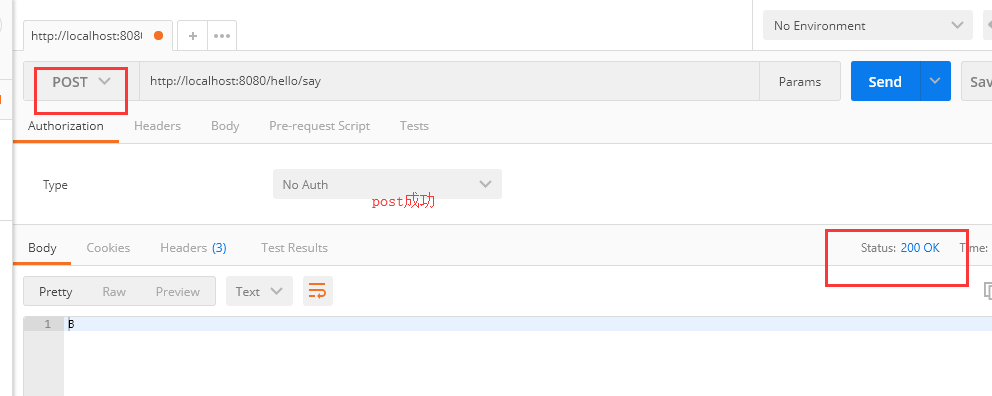


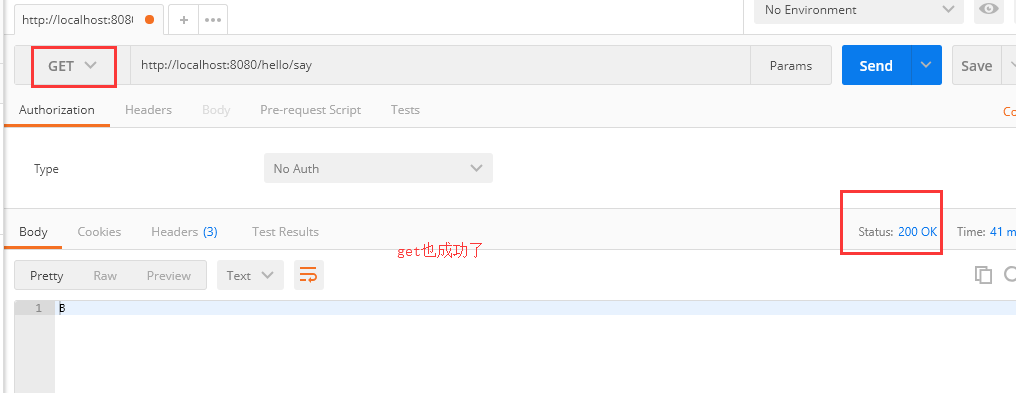




那么：

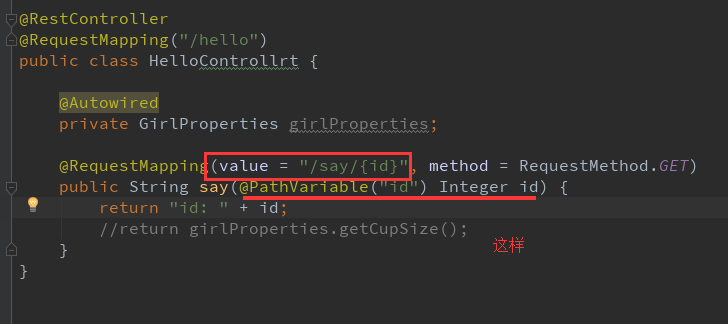


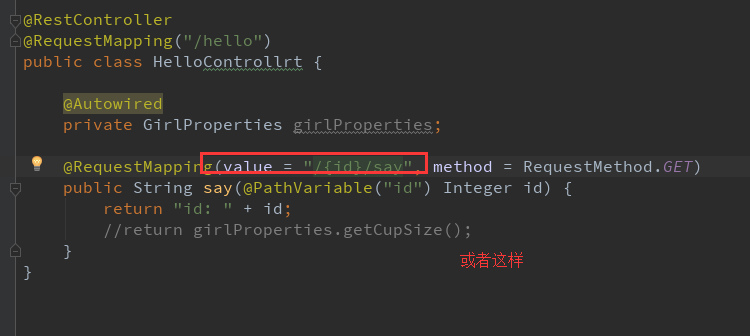


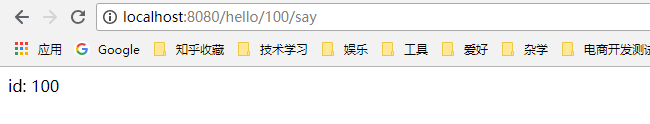


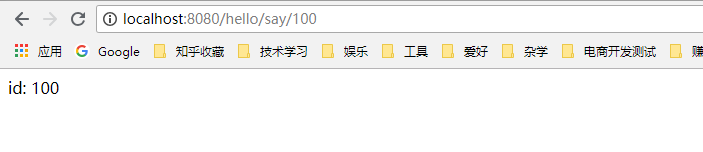
但是最好别这样.



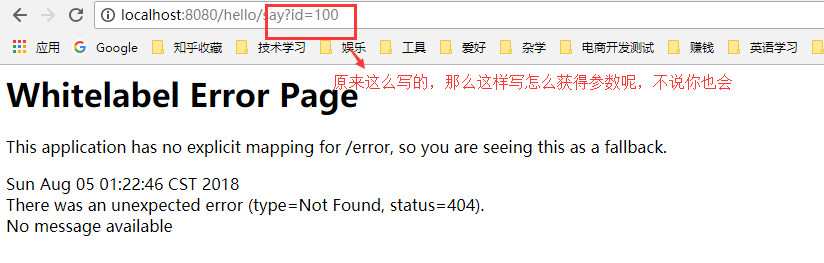




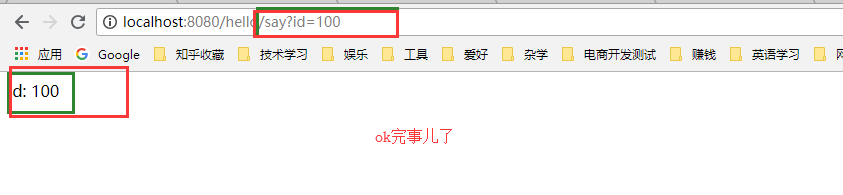




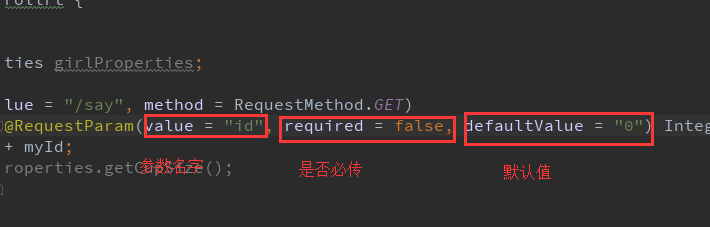
这样写多么酷啊，原先怎么干的

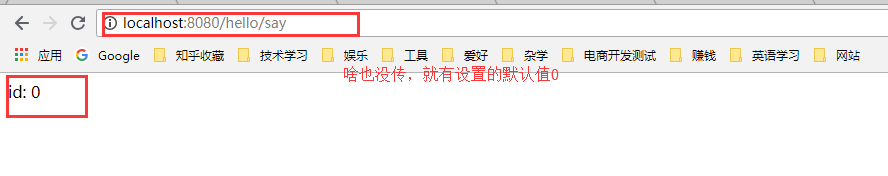






那么就是不传id，但是就要一个缺省值。看看@RequestParam中的配置属性吧





@RequestMapping里又写url又要写请求方法，真费劲啊，能不能功效二合一，当然可以

@GetMapping了解一下



同理，还有@PostMapping啥啥的

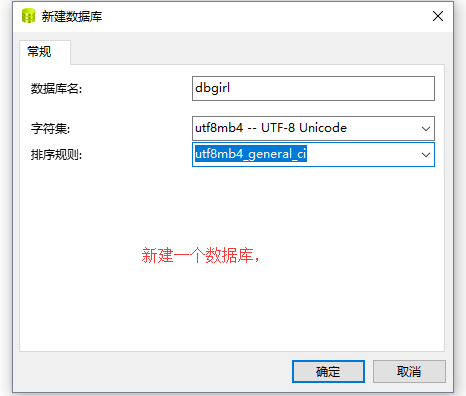
数据库操作



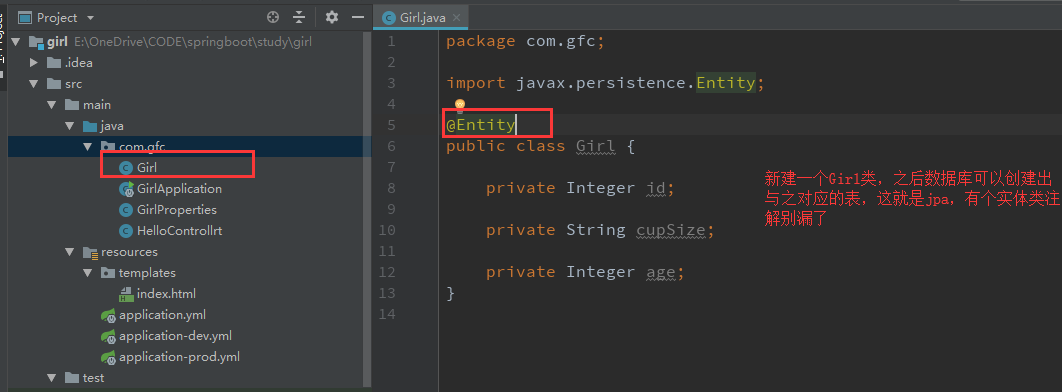




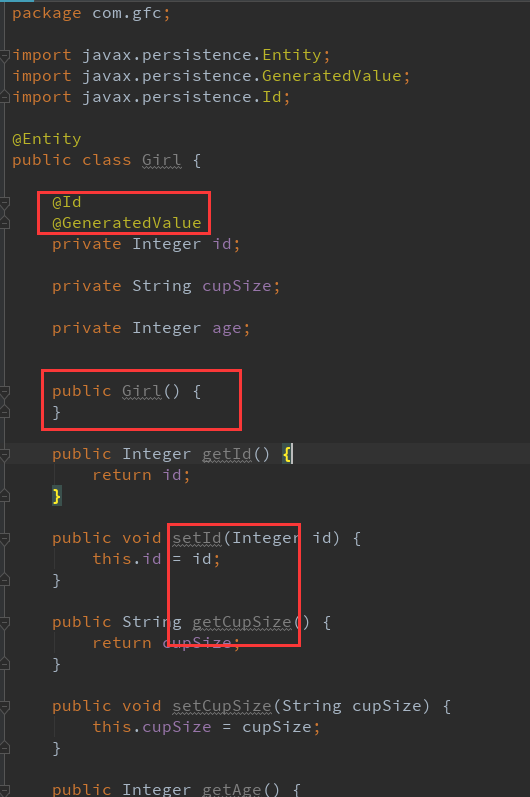




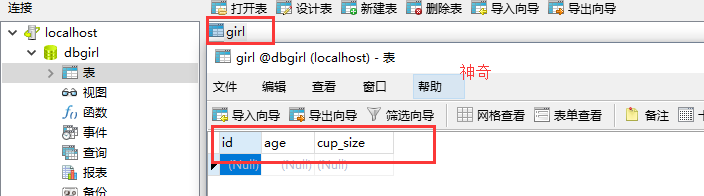
跑一下项目，不报错就行



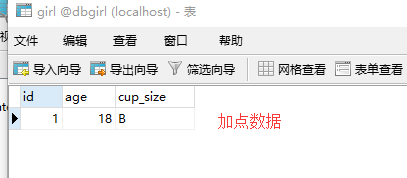
还得再添点东西，构造方法，id递增，还有getter setter方法



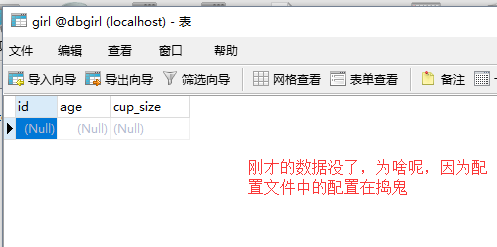
Run一下，去数据库看看



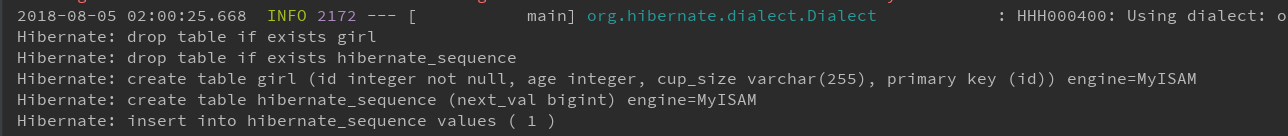




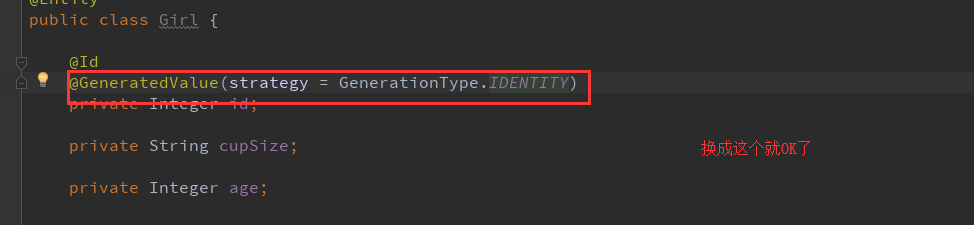
重新run一下项目







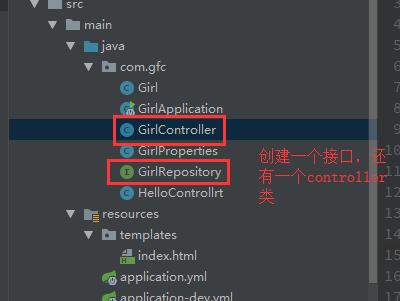
那个hibernate\_sequence表是怎么回事儿呢，因为主键生成策略的设置问题。

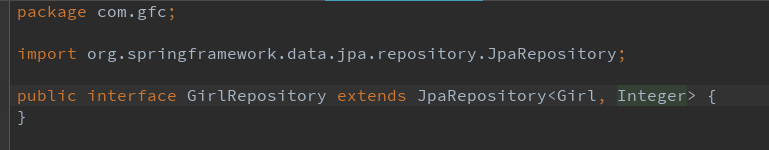






（下）写接口

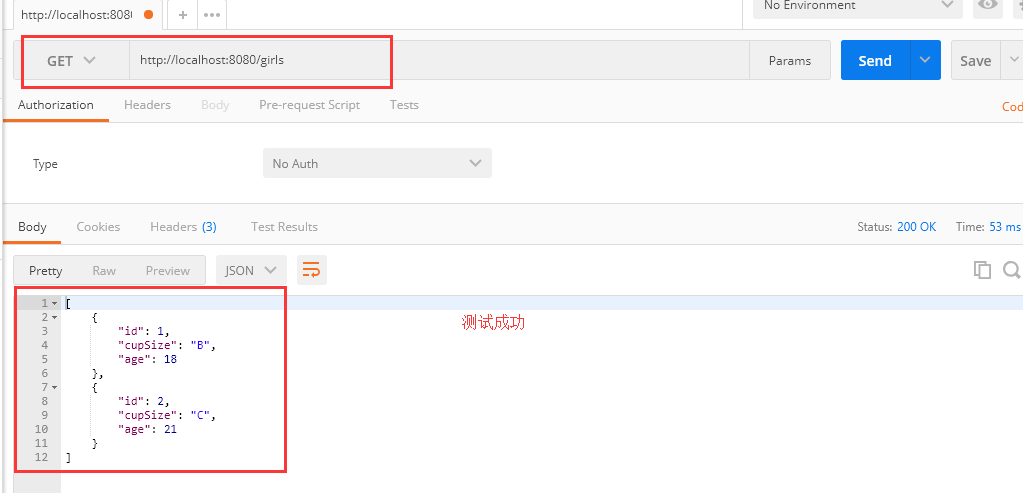


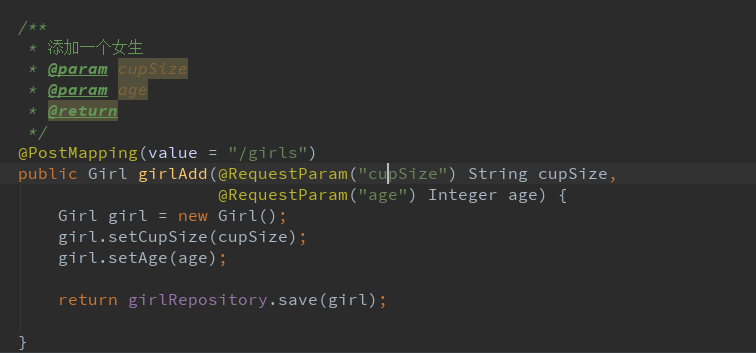


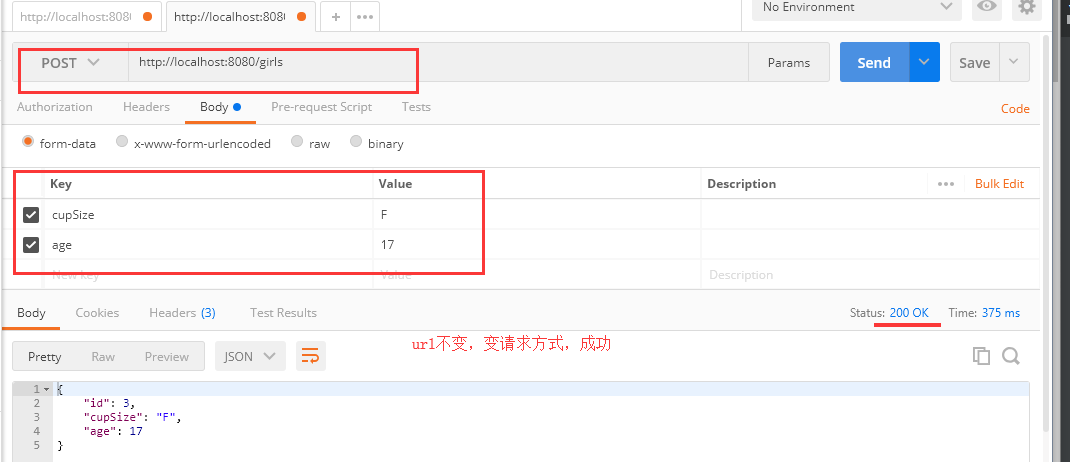


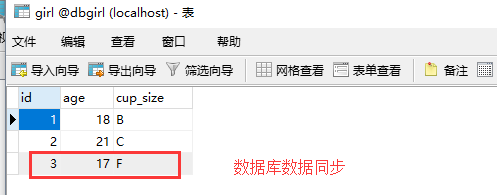
不用写sql，简单的很



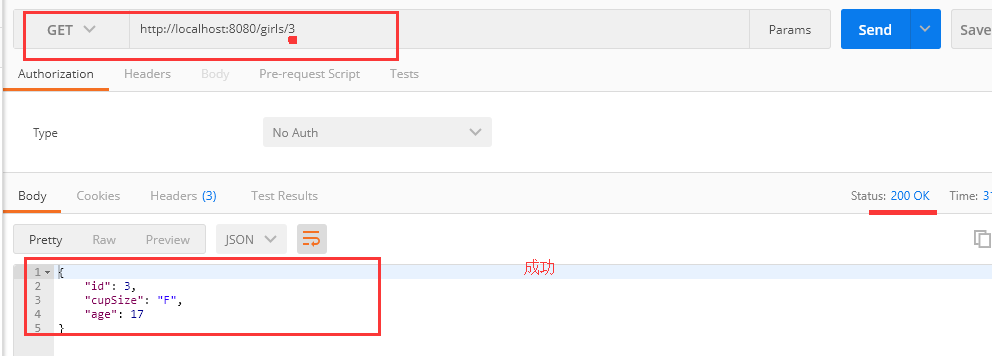




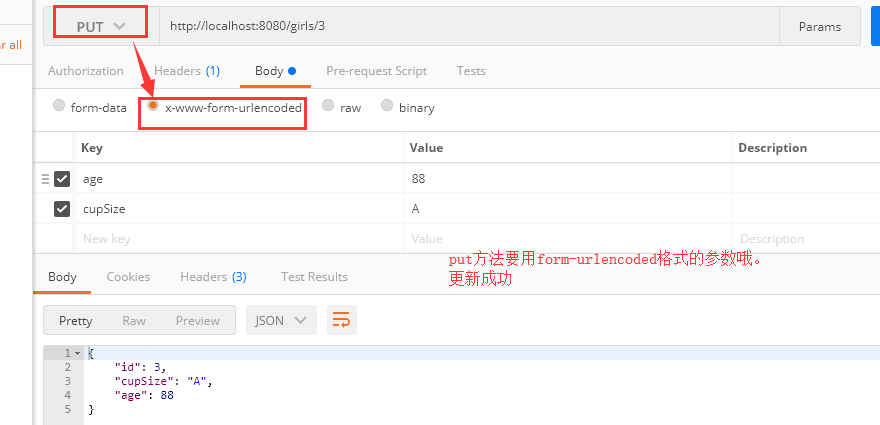


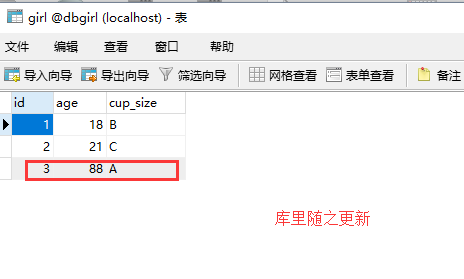
、

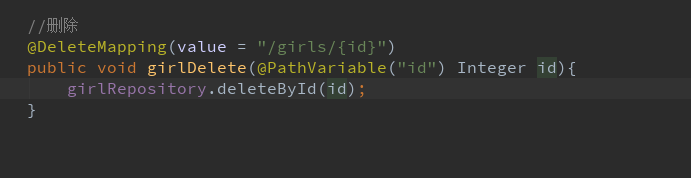


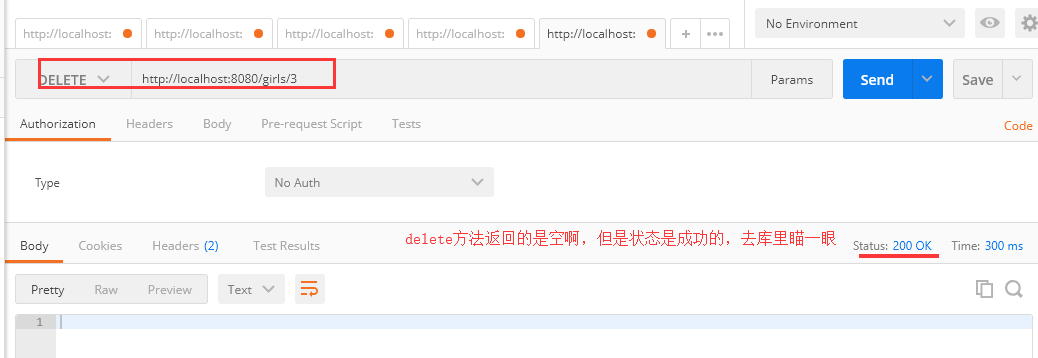


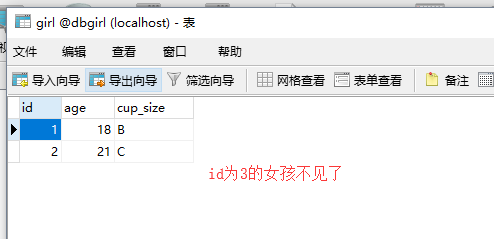






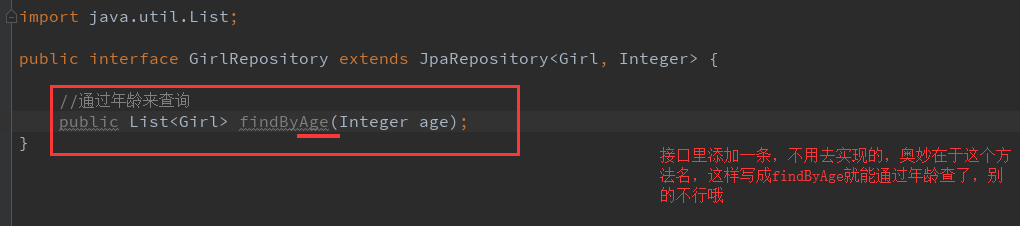




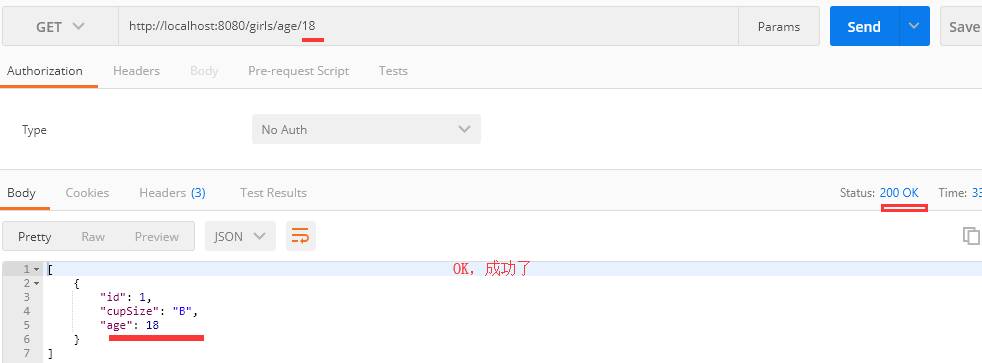


Jpa真方便啊，扩展一下

用年龄条件查一下



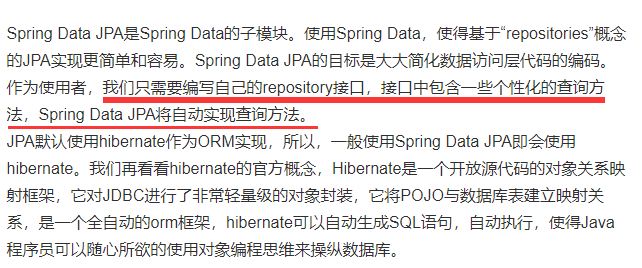




科普一下，spring data jpa和mybatis的对比

<https://www.jianshu.com/p/3927c2b6acc0>

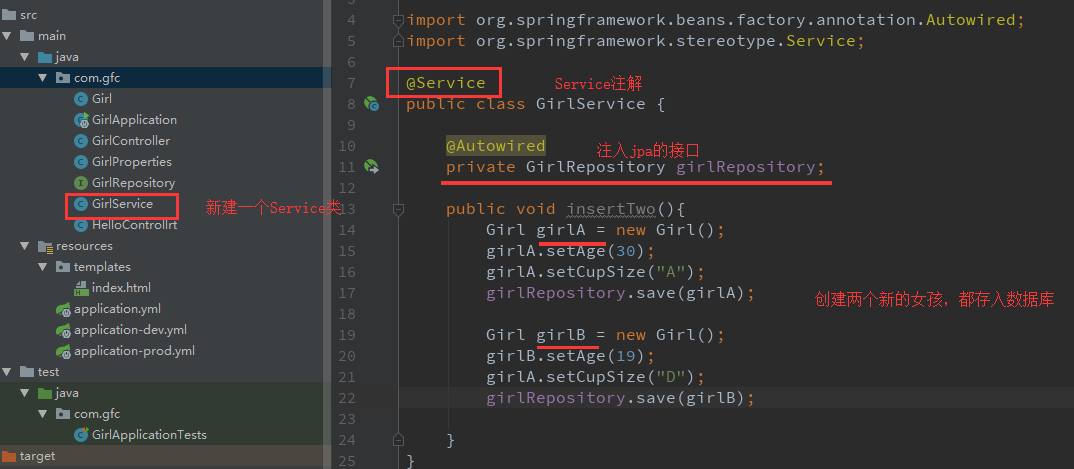
其中刚才的通过年龄查询：



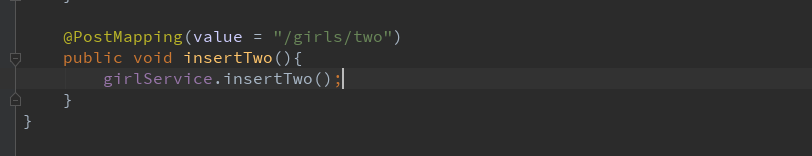
事务管理

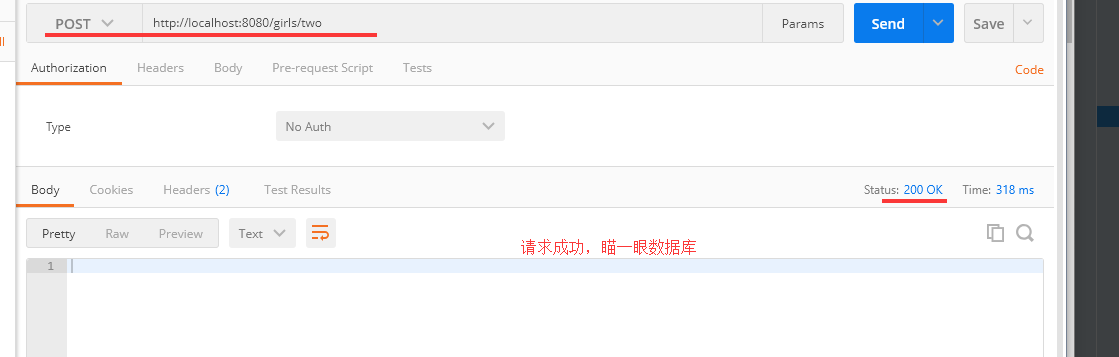
就是要不都成功，要不都回滚

官方点：事务管理是对于一系列[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93)操作进行管理，一个事务包含一个或多个[SQL](https://baike.baidu.com/item/SQL)语句，是逻辑管理的工作单元（原子单元）。





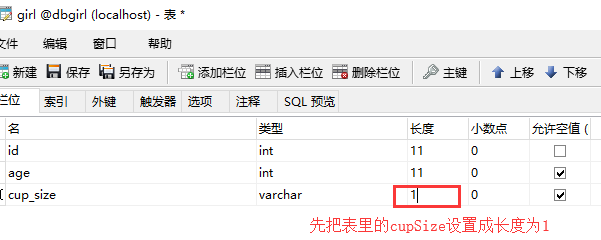




上面service中写的有点问题，B女孩的cupSize没 set，注意一下

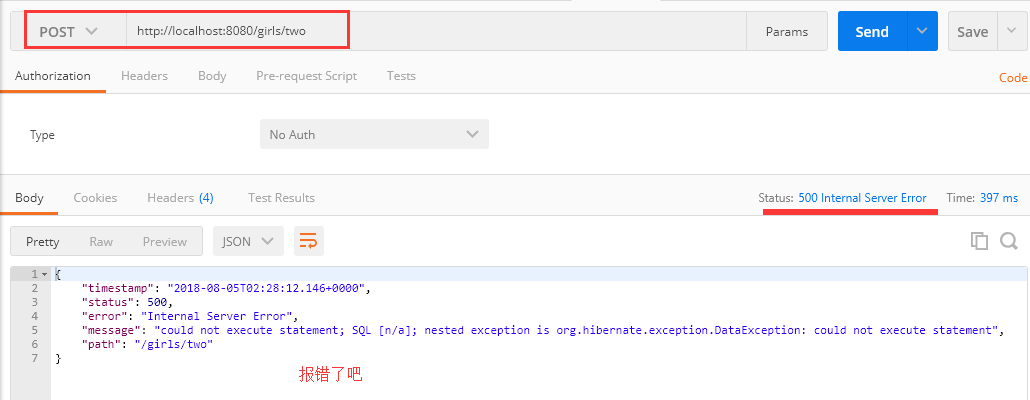
数据库中没问题

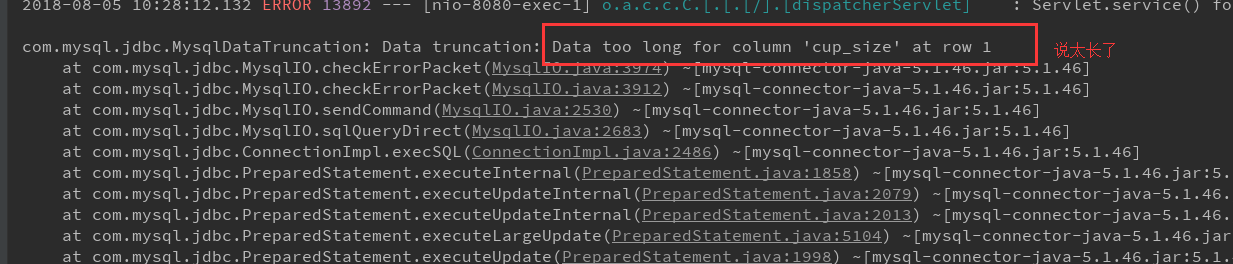
现在让B女孩插入失败，看看会怎么样

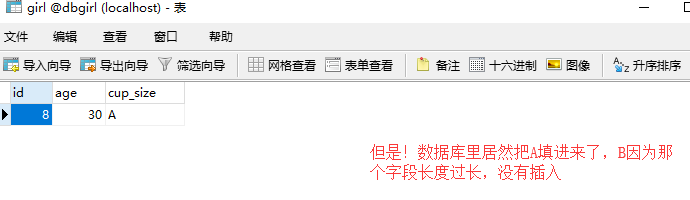




清空一下表，便于观察

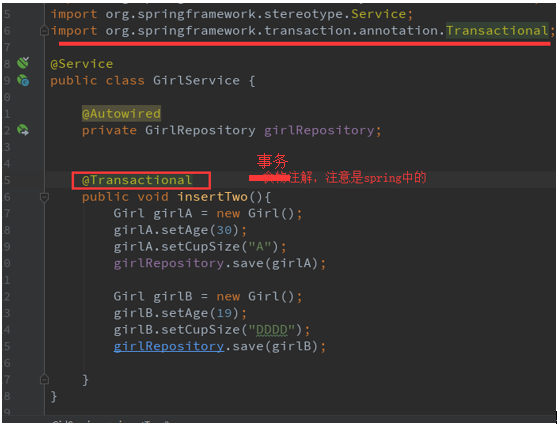


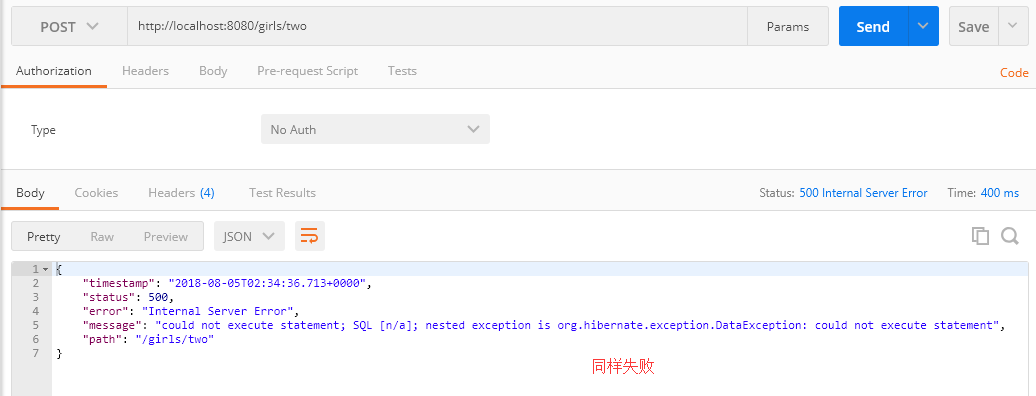


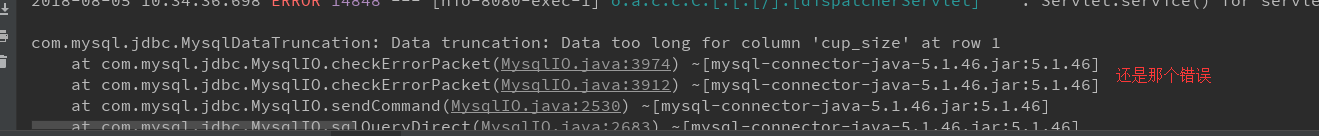


可我们不想这样，我们需要同时更新，毕竟线下业务场景，比如银行，可能由于你身份信息填写错误，身份表无法更新，但与此同时，额度表里的钱却给扣完了，当然多了倒是可以，那么怎么办呢。用注解@Transactional

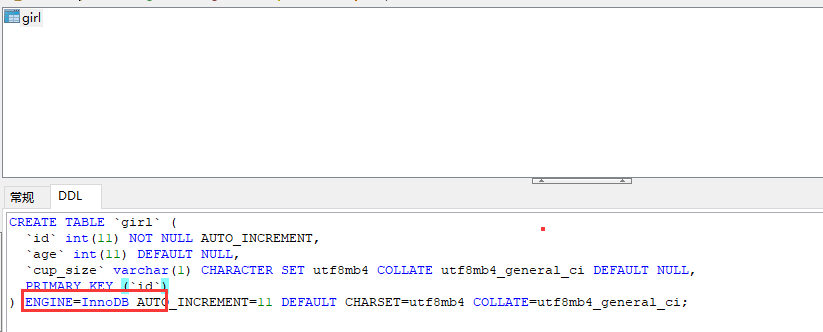
清空一下表，便于观察



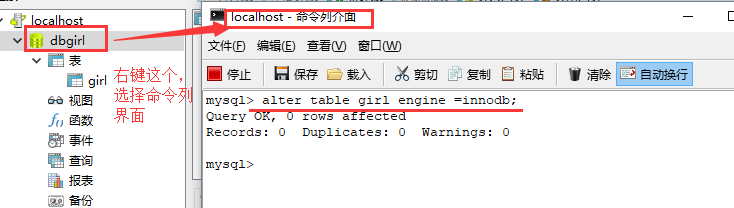


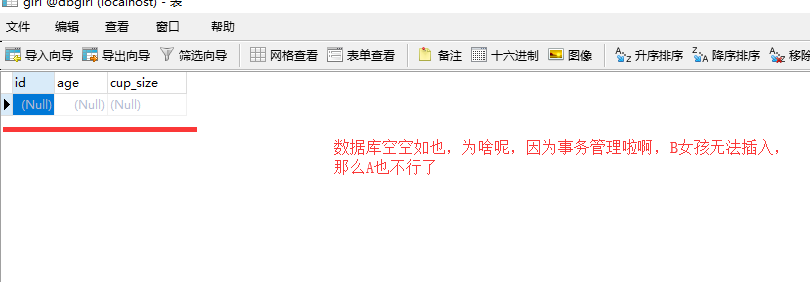


说一下数据库需要设置成InnoDB引擎才行，要不然事务注解也白搭



如果你的不是，那就改一下





只有查询的时候不用事务，别的时候都加上事务没毛病

总结

