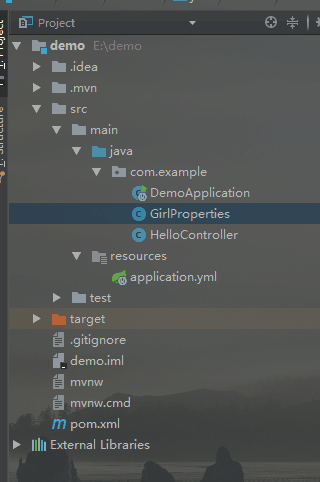
**SpringBoot学习手册**

**开发工具：IDEA**

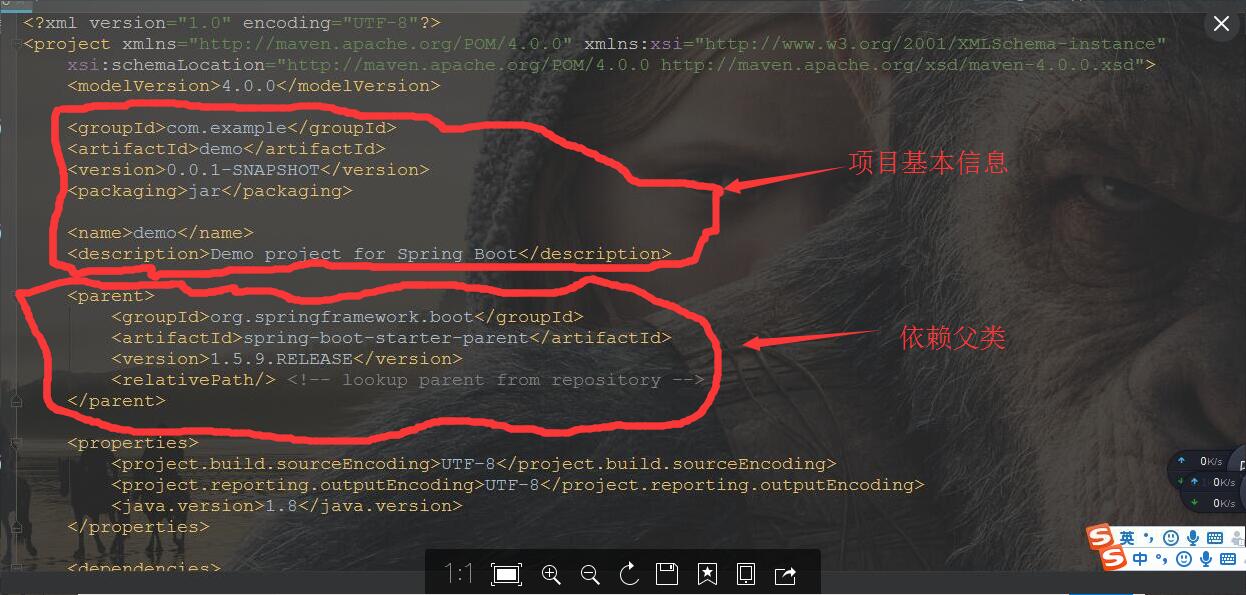
**项目构建：maven**

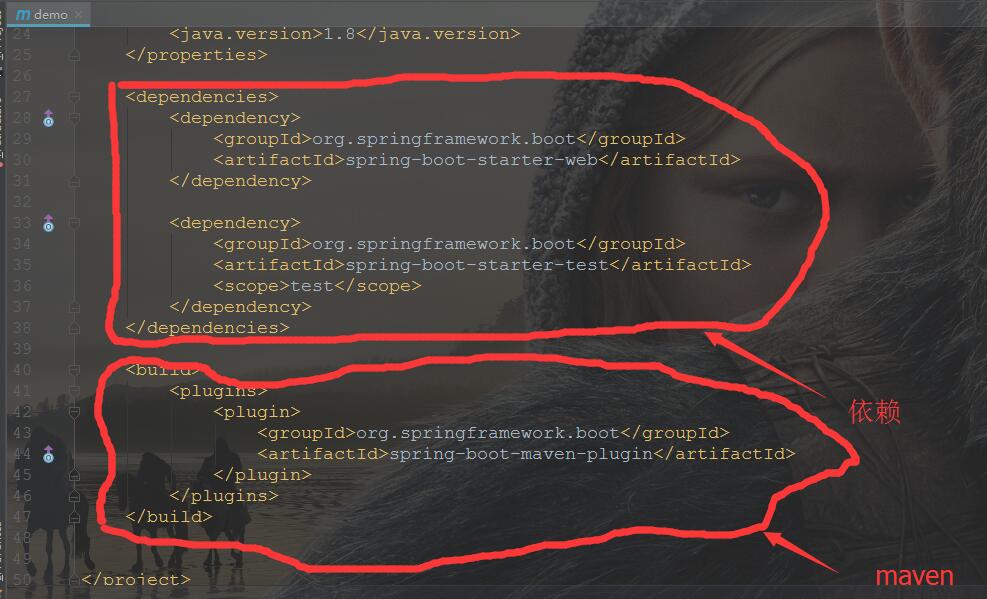
**一、创建maven项目**

项目结构：

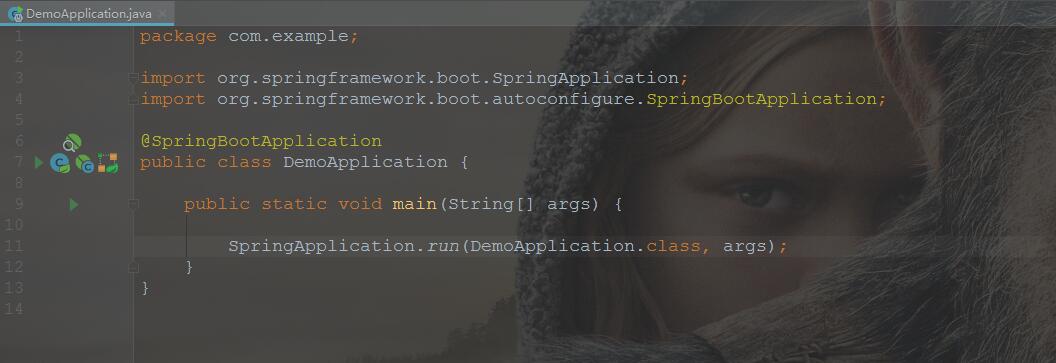


pom文件：

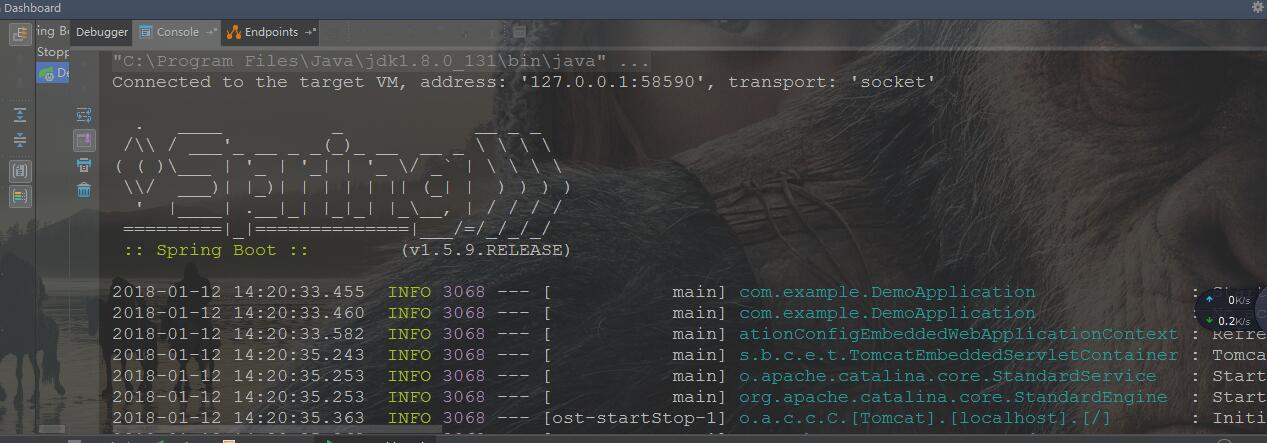




**二、启动项目**



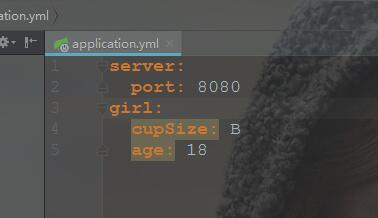
**上面这个类是SpringBoot项目程序主入口，直接运行这个类就成功启动项目。**



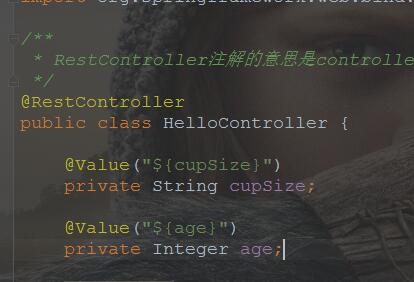
**属性配置**

**SpringBoot项目主配置文件application.properties,一般我们对项目的配置信息都设置在该文件中。将该文件重命名为application.yml**

**一般的配置有:**



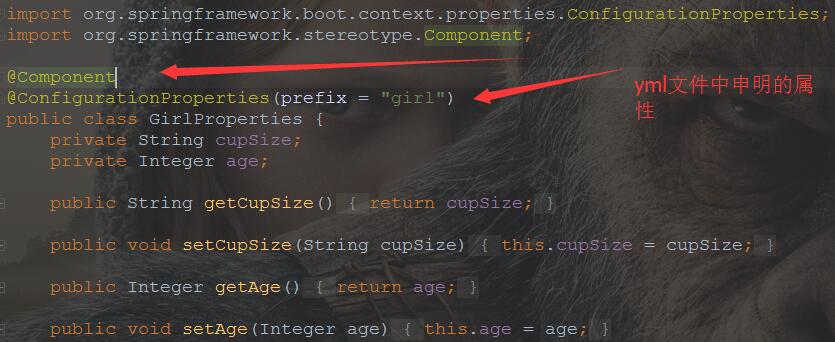
我们还可以再该文件中配置一些属性，类似上图中的属性。如何取到该属性呢？



**我们可以在yml文件中做如下配置**



那么我们可以创建一个类，包含cupSize和age这两个属性并提供get/set



**切记：注解的配置**

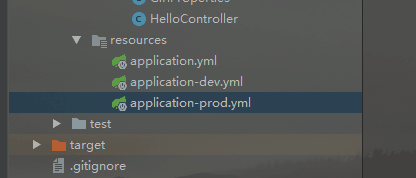
**@Component**

**@ConfigurationProperties(prefix = "girl")--------指明yml文件中的配置**

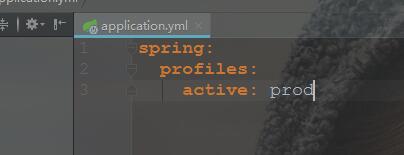
**在controller中如何拿到该类 ？**



**实际开发过程中：配置文件会有开发与测试与生产之分**



**application.yml--------------主配置文件**

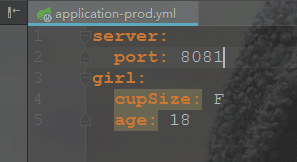


**spring.profiles.active:prod------主配置文件使用生产环境的主配置文件**

**application-dev.yml--------------开发环境使用的主配置文件**



**application-prod.yml-------生产环境使用的主配置文件**



**总结：**

**属性配置：@Value**

**@Component**

**@ConfigurationProperties(value="xxx")**

**Controller的使用**

**@Controller**

**处理http请求**

**@RestController**

**Spring4之后新加的注解，相当于@Controller与@ResponseBody配合使用**

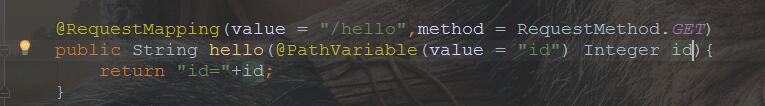
**@RequestMapping**

**配置url映射**

**接收参数时候用到的注解**

**@PathVariable**

**获取url中的数据**



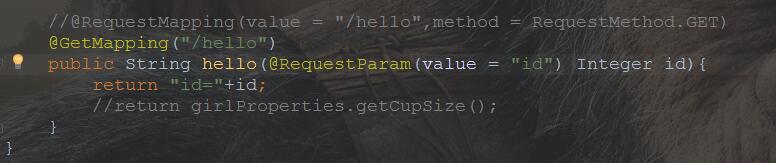
**@RequestParam**

**获取请求参数的值**



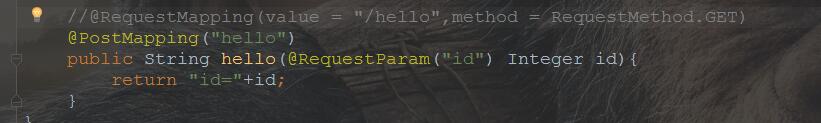
**@GetMapping(value="/hello")**

**组合注解（@RequestMapping(value="/hello"),method=RequestMethod.GET）**

****

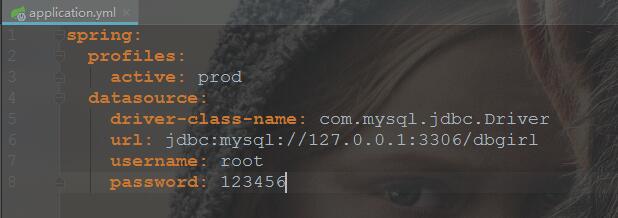
**@PostMapping(value="/hello")**

**组合注解（@RequestMapping(value="/hello"),method=RequestMethod.POST）**

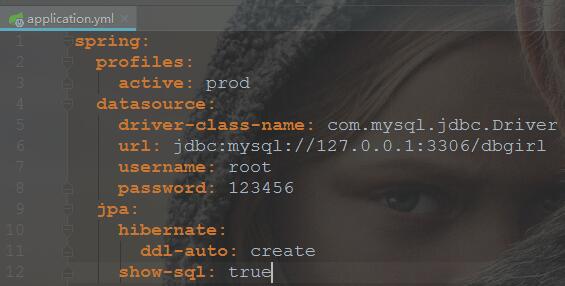


**数据库操作**

**在主配置文件中配置数据库的相关链接：**



**对于操作数据库，我们采用spring-data-jpa**



**pom文件添加两个构建**



**事务管理**

**我们在操作数据库时经常会遇到事务管理的问题：**

**业务场景：用户在天猫商城购买商品**

**步骤1：下单并且生成订单**

**步骤2：库存减少**

**步骤3：扣款**

**问题：如果库存减少了但是扣款时候失败了怎么办？**

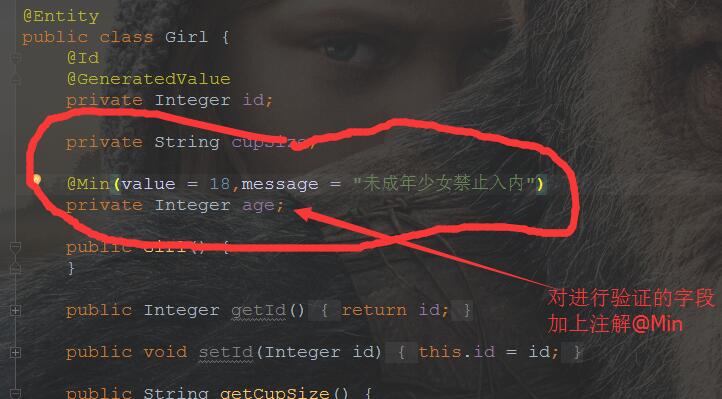
**这就涉及到了数据库的事务管理**

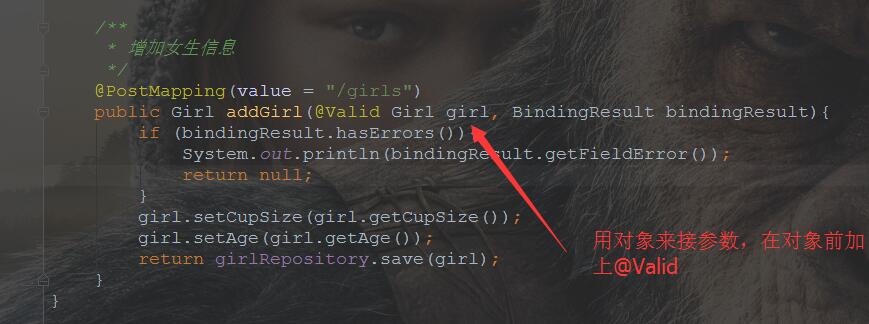
**在方法上加上@Transactional这样的话就涉及到事务的管理了**

**@Valid的使用**

**@Valid注解是用来做表单验证的，在表单提交时，对表单数据进行验证。**

**具体如何使用看下图：**

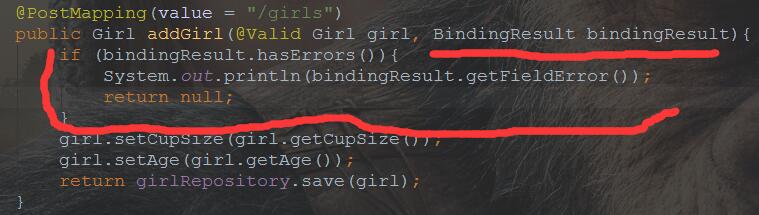




**如何请求参数age小于18，那么控制台会报错：**

Field error in object 'girl' on field 'age': rejected value [12]; codes [Min.girl.age,Min.age,Min.java.lang.Integer,Min]; arguments [org.springframework.context.support.DefaultMessageSourceResolvable: codes [girl.age,age]; arguments []; default message [age],18]; default message [未成年少女禁止入内]

内部类实现如下图：



**AOP**

**面向对象：将需求功能垂直划分为不同的并且相对独立的，它会封装成良好的类，并且让它们有属于自己的行为。**

**AOP：它利用的是一种叫横切的技术，将面向对象构建的庞大的类的体系进行水平的切割并且将那些影响到了多个类的公共行为封装成为一个可重用的模块，这个模块成为切面。所以AOP翻译成中文叫做面向切面编程。**

**AOP的观念思想就是将通用逻辑从业务逻辑中分离出来。**

**AOP统一处理请求日志**

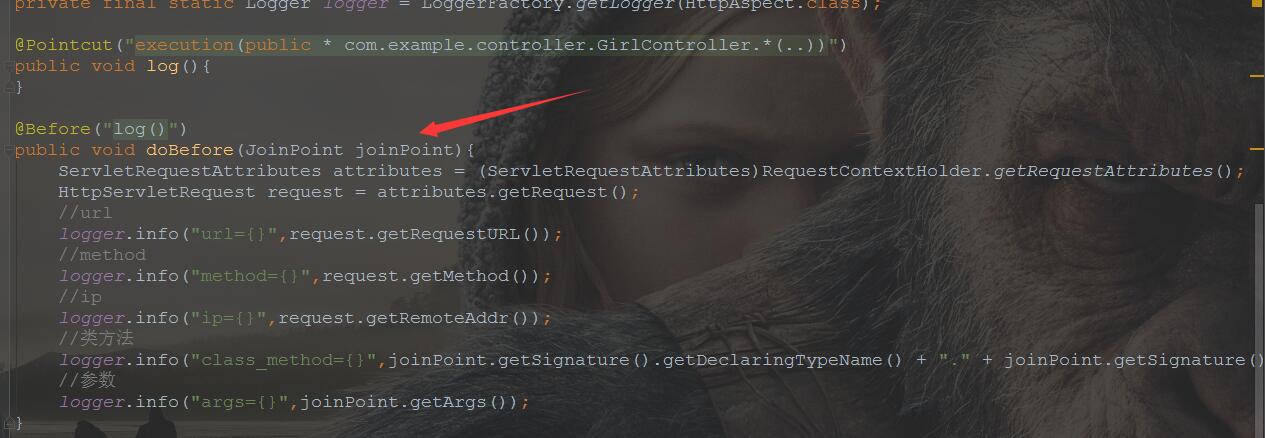
![C:\Users\lenovo\Documents\Tencent Files\295873321\Image\C2C\AU7B4HQ)L](02(CEBGO7EHY.png](data:image/png;base64,)

pom文件中先添加AOP的构建

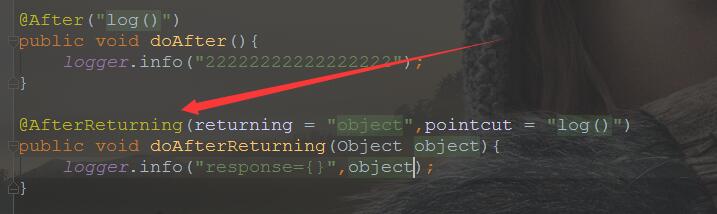




**日志获取http请求内容：**



日志获取请求返回的内容：

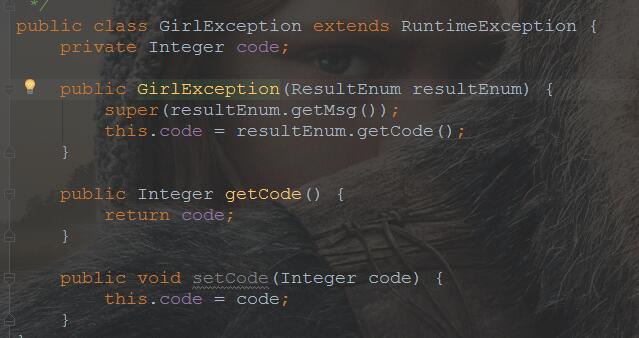


**统一异常处理**

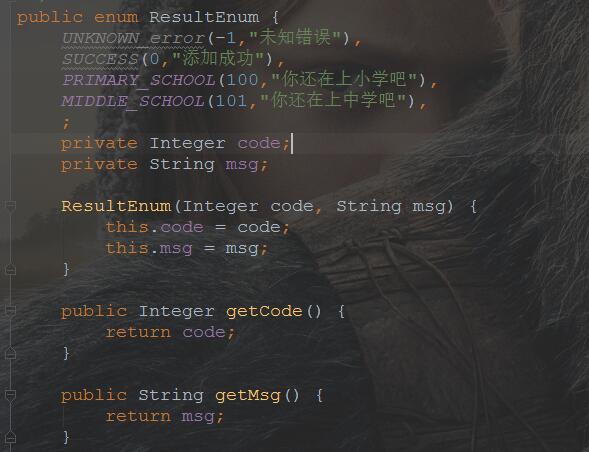
**Spring框架只对你抛出的异常RuntimeException才会进行事务回滚。如果抛出的是Exception，是不会进行回滚的。**

**对抛出的异常进行抓获：**

![C:\Users\lenovo\Documents\Tencent Files\295873321\Image\C2C\C`RV$LD$F{KB6QW7](D0SUI.jpg](data:image/jpeg;base64,)

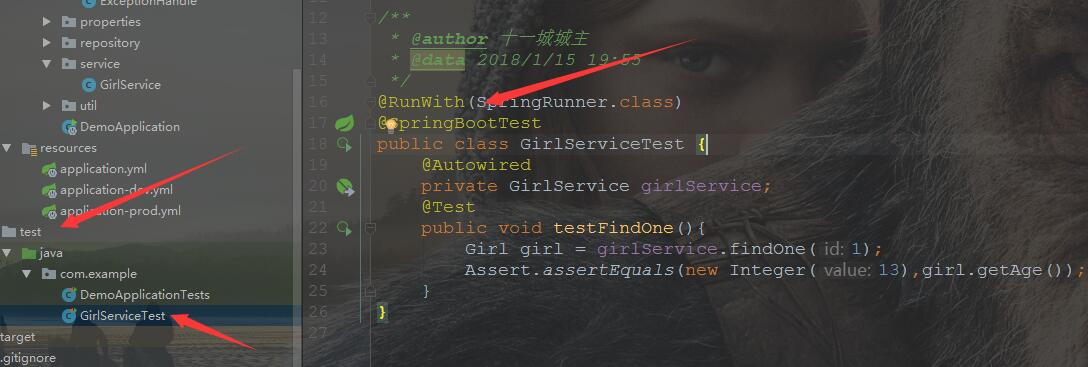


**枚举类：将全局定义的code以及msg全部写到枚举类中：**

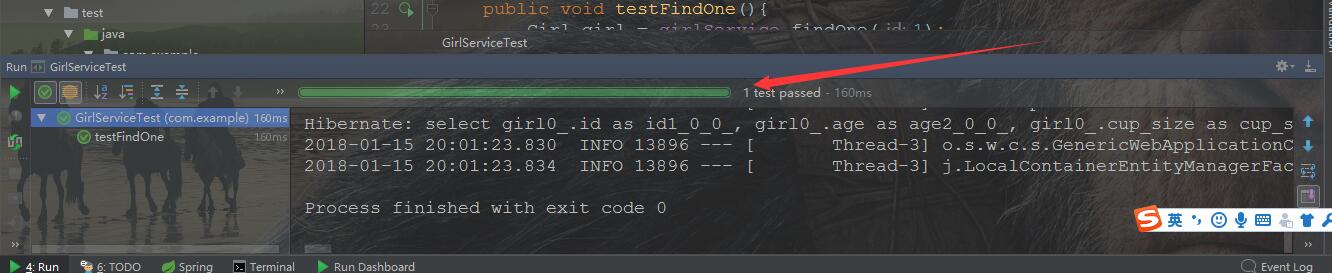


**单元测试**

**对service测试**



**测试成功的话如下图：**



**也可以直接在类方法上右键：**

