## 2-1 第一个springboot项目（对应项目——gril）

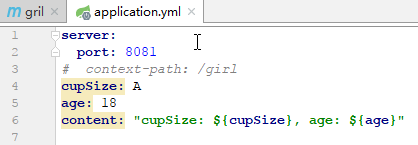
1. 使用idea新建springboot项目，选择1.5.9版本，jdk使用1.8，maven使用3.5
2. 直接启动GrilApplication类，访问localhost:8080,出现错误页面为正常。
3. 创建HelloController类，定义@RestController，定义访问路径@RequestMapping("/hello")，定义返回json。再启动访问<http://localhost:8080/hello>。
4. 其他启动方式：

命令行进入到项目目录：girl，mvn spring-boot:run

命令行进入到项目目录：girl，mvn install,然后进入到target目录下，java -jar 运行jar文件

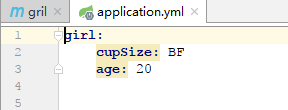
## 3-1 项目属性配置

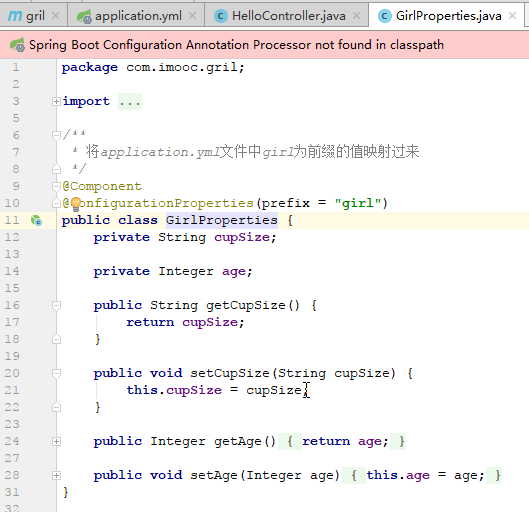
1. 推荐使用yml
2. 配置端口号，配置项目名
3. 在配置文件中配置属性值，在类中使用@Value("${cupSize}")进行赋值
4. 在配置中使用配置，类似content。





1. 当属性很多的时候，一个一个@Value太麻烦了，所以用类来注入值，新建GrilProperties类，@ConfigurationProperties(prefix = "girl")，它会自动将配置文件中的值注入到这个类的属性中，同时要让这个类成为@Component是Spring管理的类，才可以@Autowired使用。





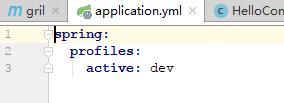
在HelloController类中使用



1. 修改生产环境或者开发环境

新建application-dev.yml，application-prod.yml

在application.yml文件中定义运行环境

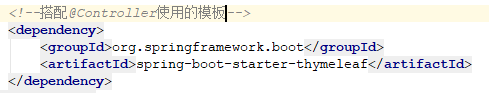


可以使用java -jar girl-0.0.1-SNAPSHOT.jar --spring.profiles.active=prod选择运行环境运行。

## 4-1 Controller的使用

1、@Controller 处理http请求

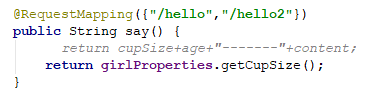
当只使用@Controller时候，必须配合模板使用，添加pom依赖



然后在resources目录下新建templates目录，然后新建index.html

2、@RestController Spring4之后新加的注解，原来返回json需要@ResponseBody配合@Controller使用

3、@RequestMapping 配置url映射，可以是在类上，也可以是在方法上。

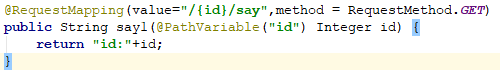


关于使用方式，可以使用method = RequestMethod.POST

可以使用POSTMAN工具测试

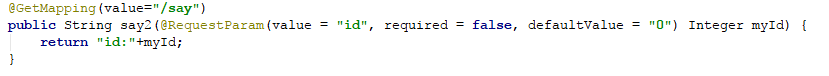
不同的请求应该用不同的方式，所以建议在请求上写明请求方式

4、@PathVariable 获取url中的数据



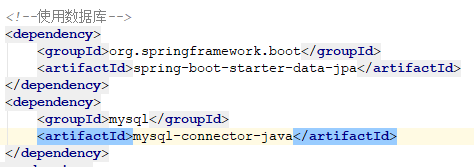
5、@RequestParam 获取请求参数的值

6、@GetMapping 组合注解



## 5-1数据库操作（上）

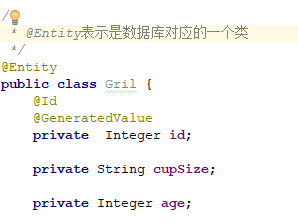
1. Spring-Data-Jpa：
   1. JPA（Java Persistence API）定义了一系列对象持久化的标准，目前实现这一规范的产品有Hibernate、TopLink等
2. 添加数据库相关依赖：



3、配置数据库

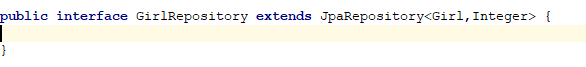


需要事先创建数据库dbgirl，不需要建表，直接建类

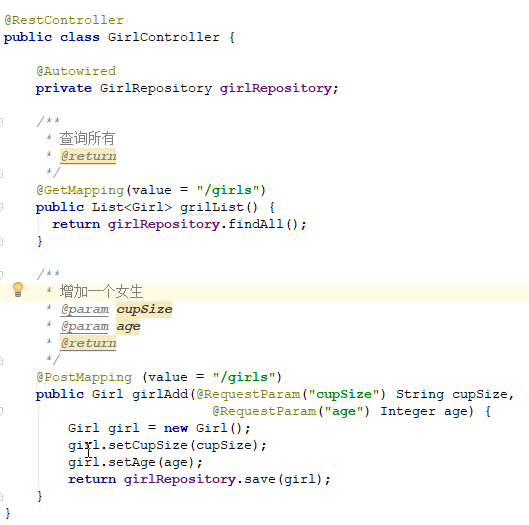


4、增删改查

（1）新建GirlRepository类



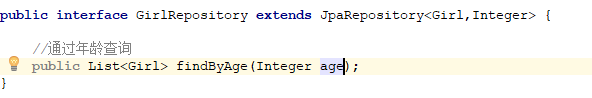
（2）新建GirlController类

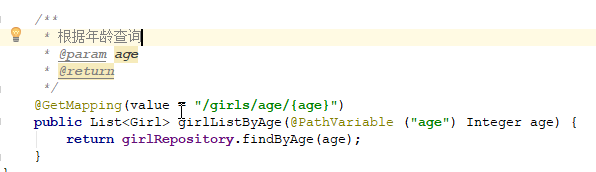


***使用postman put方式，不能使用form-data，要使用x-www-form-urlencoded***



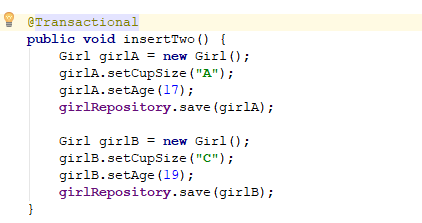
5、自定义方法，方法名有讲究





6-1 事务管理

在Service层加一个注解即可



**只有查询的时候不需要加事务**