2小时学会SpringBoot初学者

SpringBoot是SpringMVC的升级版并没有必然的联系

特点、

1. 化繁为简，简化配置
2. 备受关注，下一代框架
3. 微服务的入门级框架 微服务 SpringCloud

注意:

1.具备必要的前置知识（①利用maven构建项目 ②Spring注解 ③RESTul API）

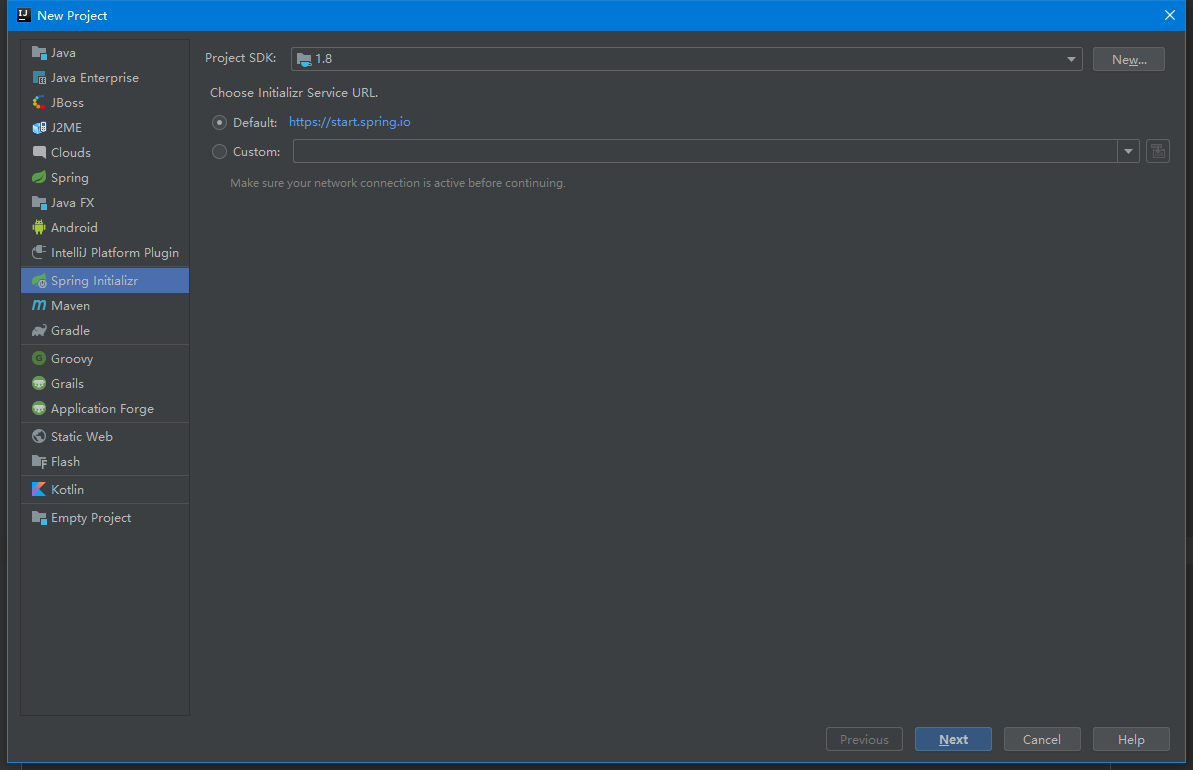
2.不需要学习SpringMVC

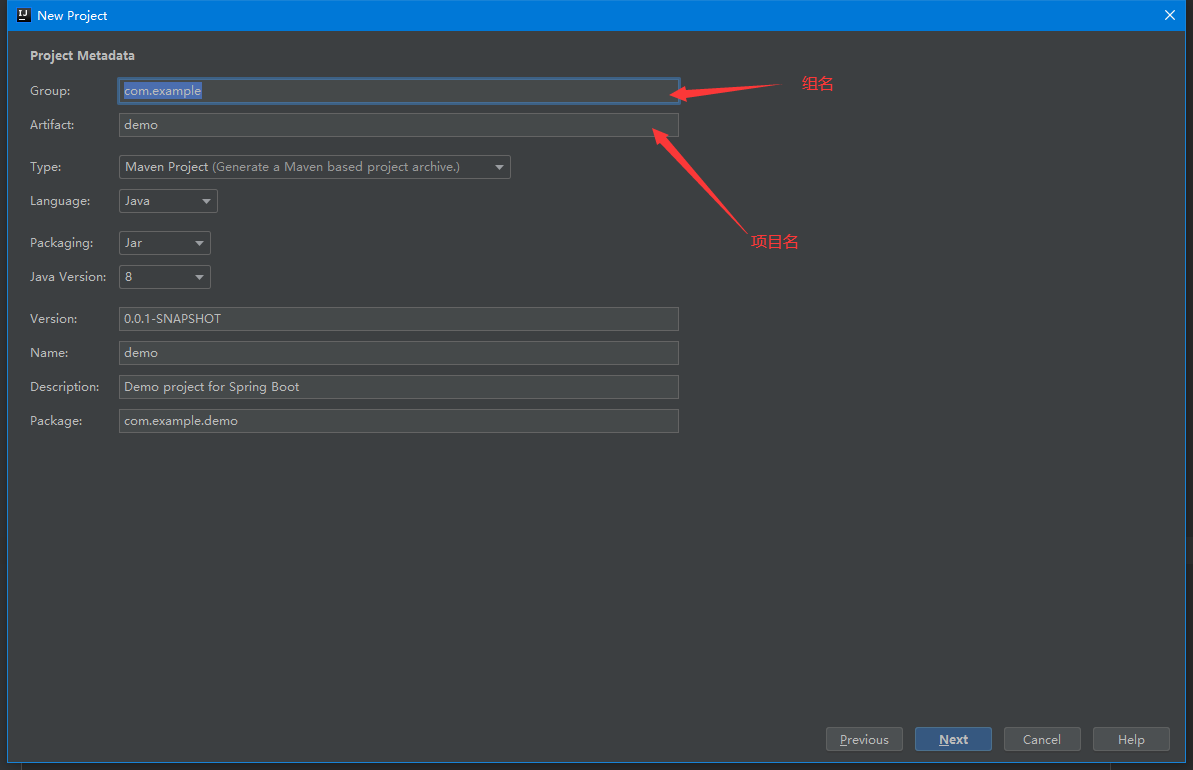
3.JAVA Maven版本保持一致

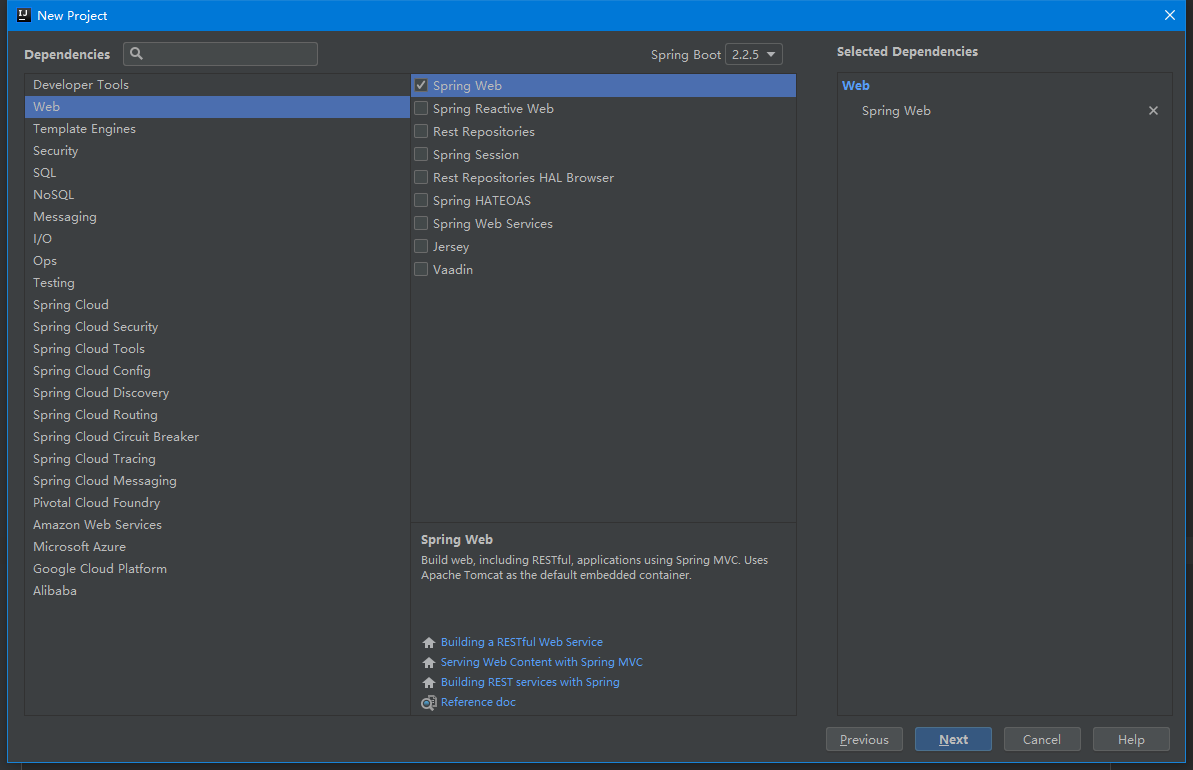
JDK 和 Maven 事前安装

Java -version mvn -version

Idea创建新项目





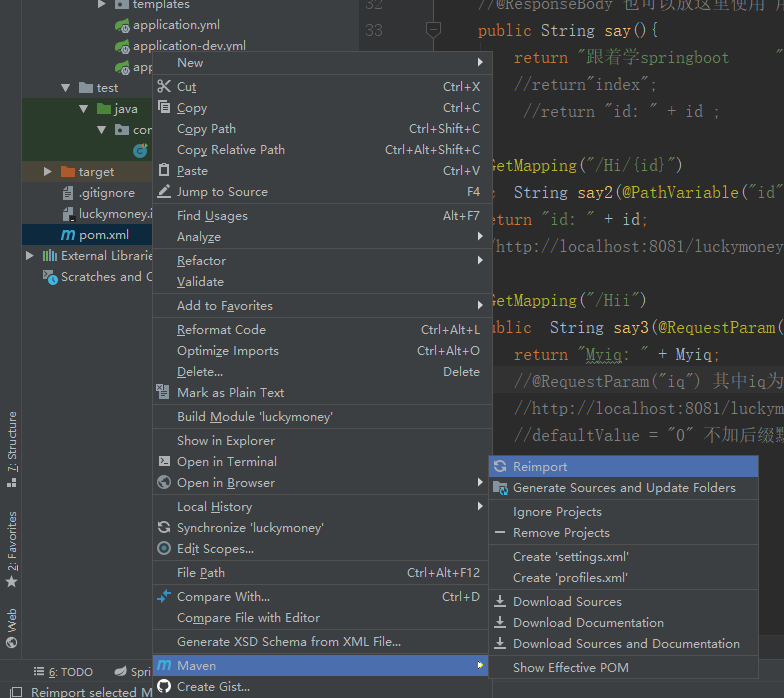


一定选正常版本不要用测试版本

一路next即可

右下角下载各种jar包 如果缓慢则在设置maven时设置阿里云地址

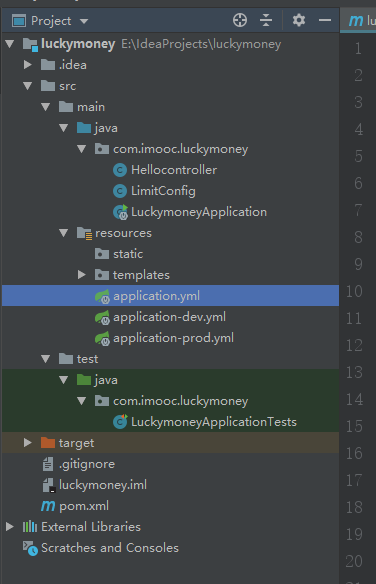
需要在项目上pom.xml >> maven >> Reimport



若果使用其他编辑器

<https://start.spring.io/> 访问这个下载即可

可以删去.mvn/ HELP.md /mvnw /mvnw.cmd四个文件或文件夹



pom.xml 是maven的配置文件

src文件夹即是源码文件夹

src >> main >> java >> 包名 >> xxxxApplication 启动文件

@SpringBootApplication启动注解

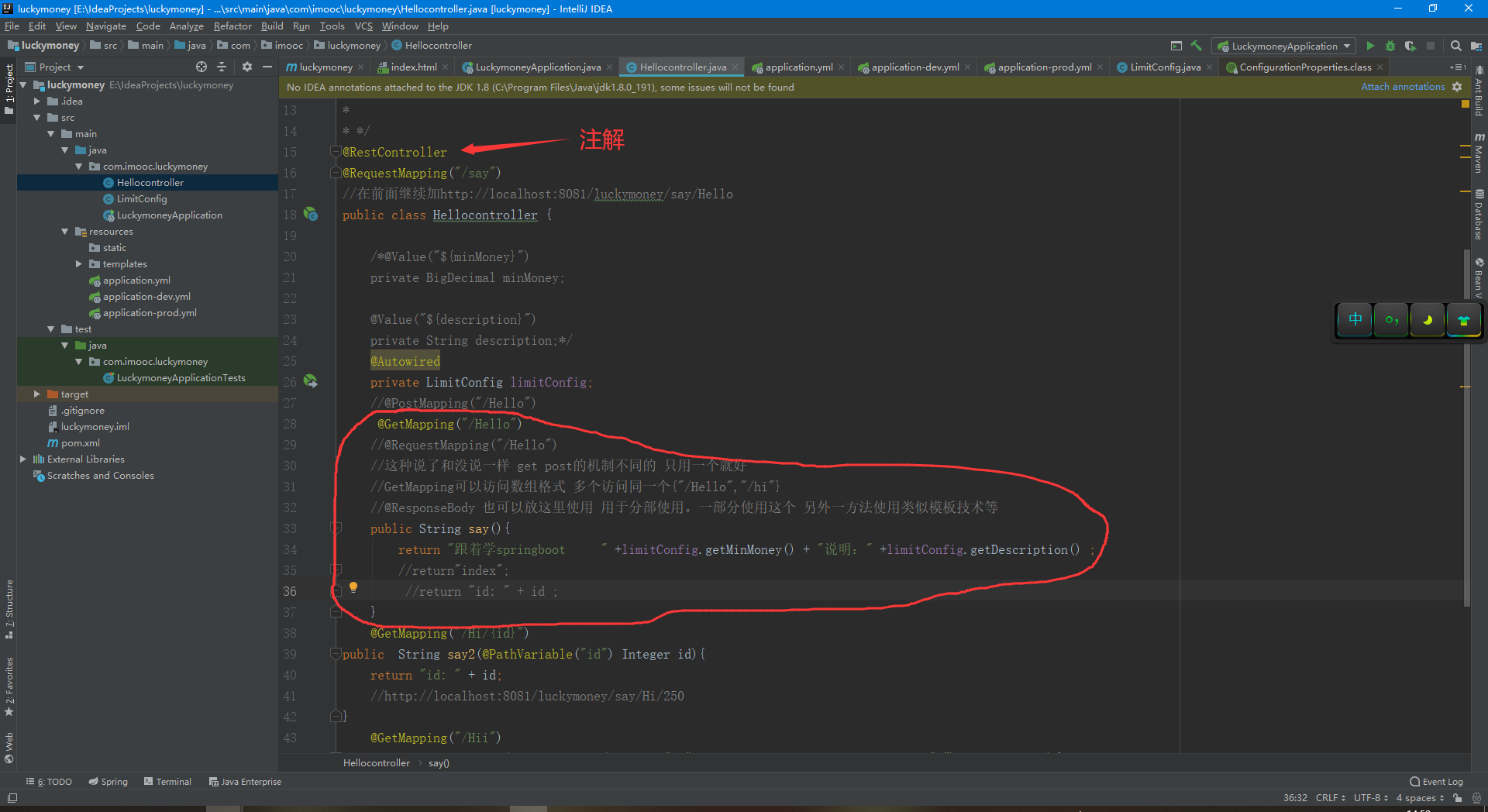
src >> main >> resources >> application.properties 配置文件

src >> test >> java >> 测试文件夹 注意注解

启动文件 src >> main >> java >> 包名 >> xxxxApplication 右键run即可启动

红色正方形停止

新建一文件



启动方式

1. idea启动
2. 命令行启动
3. Maven打包成jar包 启动

项目的配置

修改application.properties中

添加

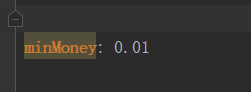
server.port =8081 修改端口

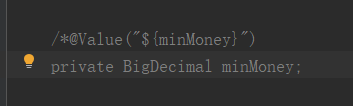
server.servlet.context-path = /xxxxx url访问需要在地址栏前添加

application.yml 功能同上 相同部分折叠

也可设置其他的属性配置并在程序中使用

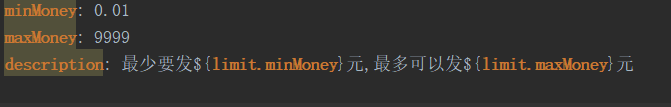
例：





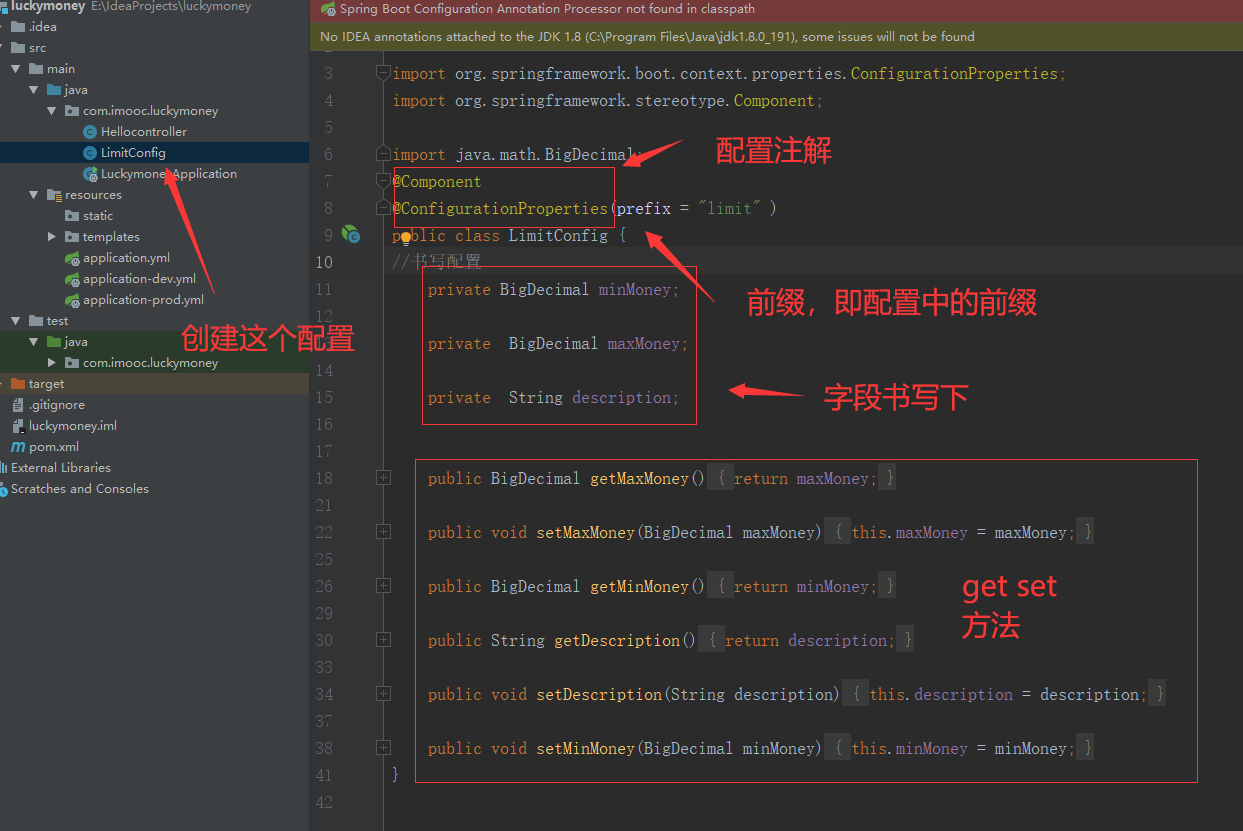
在程序中直接使用minMoney即可

如何在配置中使用配置？如下

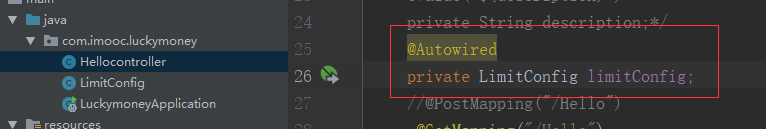




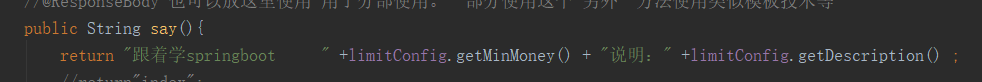
可以理解为对象有三个字段



使用上图的配置



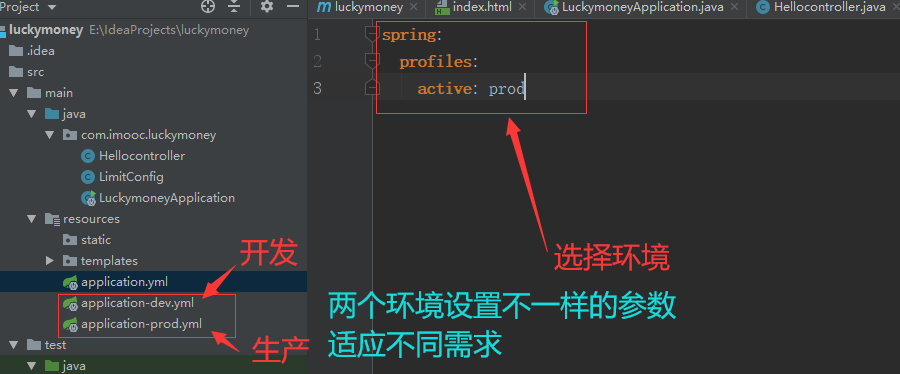
在方法中使用



什么时候用单个配置，什么时候用对象配置呢？

看需求 若像图上的两个属性为限制条件，可用配置类

多个环境的使用、



如果打成jar包

在服务器中怎么修改配置

Java -jar -Dspring.profiles.active=dev xxxxx.jar

即可修改

配置总结即注解

@Value 单个配置

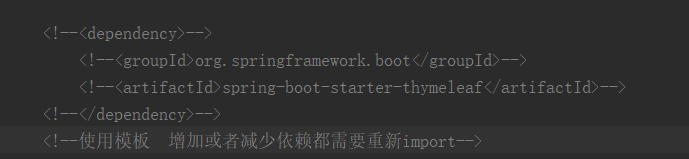
@Component

@ConfigurationProperties 这俩用来配置对象配置

多环境配置

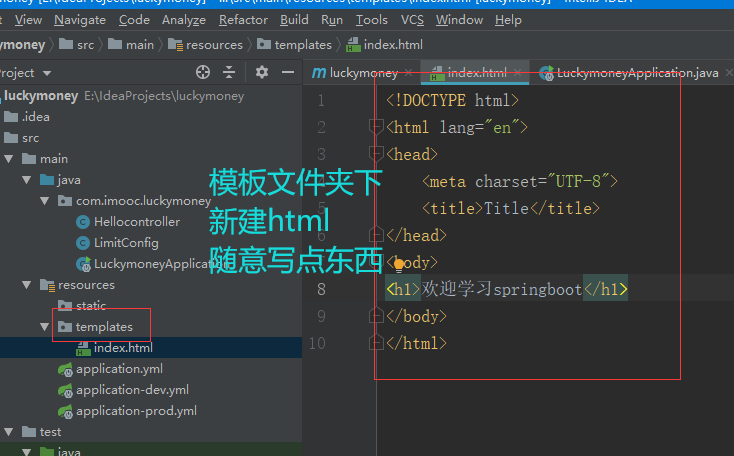
Controller的使用

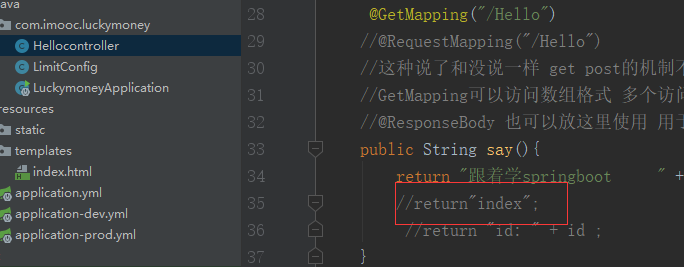
如果仅用到@Controller又想返回信息的话需要用到模板并在pom.xml加一个依赖



需要重新import

看是否部署好 Ctrl键按住单击看是否能点进去





重新启动 模板渲染用的比较少 大多用前后端渲染

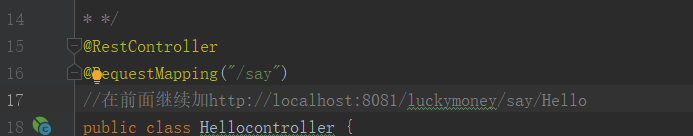
删掉依赖并重新import

如果还是想返回之前的string

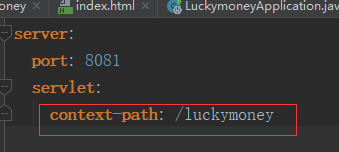


@ResponseBody 也可以放方法前使用 用于分部使用。一部分使用这个 另外一方法使用类似模板技术等

GetMapping可以访问数组格式 多个访问同一个方法可以使用{"/Hello","/hi"}



在类前加入@RequestMapping("/say") 类似配置文件中的



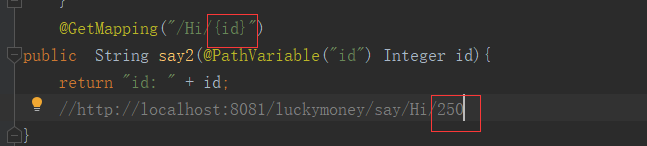
@GetMapping可以改成@PostMapping 改成post请求 使用软件postman 可以

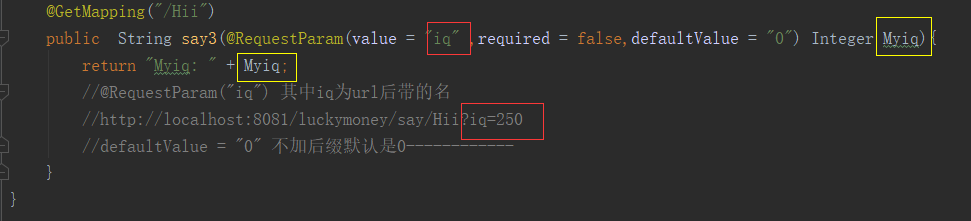
也可以使用@RequestMapping("/Hello")

但是这种说了和没说一样 get post的机制不同的 只用一个就好

@PathVariable 获取url中的数据

@RequestParam 获取请求参数的值





Post传递参数方式多样

可以写在url后面也可以使用postman中的body 框架使用了兼容

Spring-Data-Jpa

JPA定义了一系列对象持久化的标准

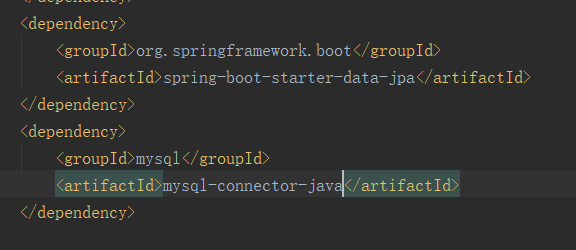
RESTful API设计

REST，即Representational State Transfer的缩写。直接翻译的意思是"表现层状态转化"。

它是一种互联网应用程序的API设计理念：URL定位资源，用HTTP动词（GET,POST,DELETE,DETC）描述操作

例：

|  |  |  |
| --- | --- | --- |
| 请求类型 | 请求路径 | 功能 |
| GET | /luckymoneys | 获取列表 |
| POST | /luckymoneys | 创建 |
| GET | /luckymoneys/id | Id查询 |
| PUT | /luckymoneys/id | 更新 |
|  |  |  |

1.pom.xml引入两个依赖并import

2.application.yml写入数据库配置



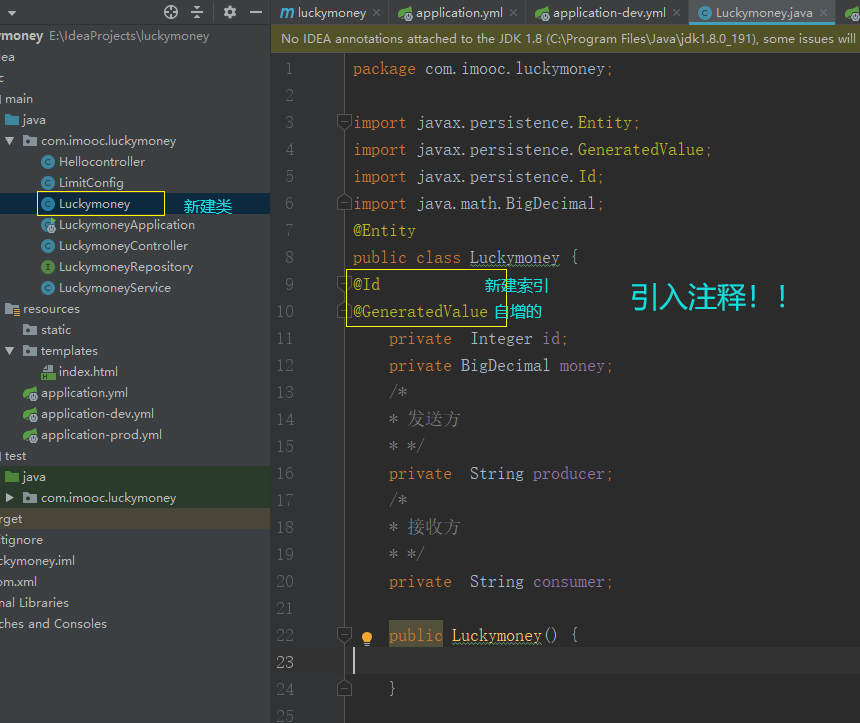
ddl-auto:create----每次运行该程序，没有表格会新建表格，表内有数据会清空

ddl-auto:create-drop----每次程序结束的时候会清空表

ddl-auto:update----每次运行程序，没有表格会新建表格，表内有数据不会清空，只会更新

ddl-auto:validate----运行程序会校验数据与数据库的字段类型是否相同，不同会报错

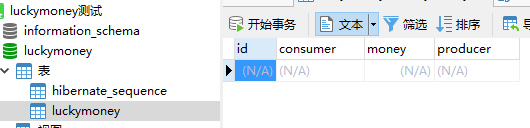
3.新建类



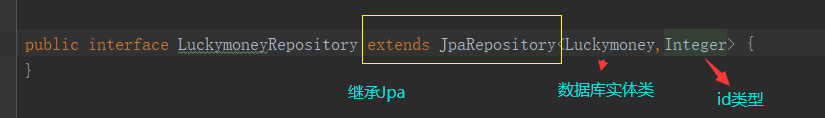
并

为变量添加getset方法

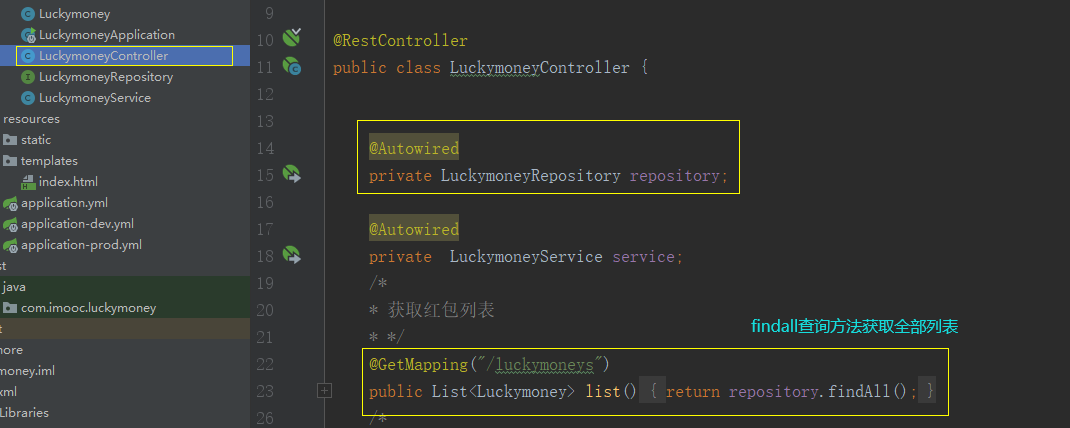
运行并查看数据库



1. 新建Repository接口继承

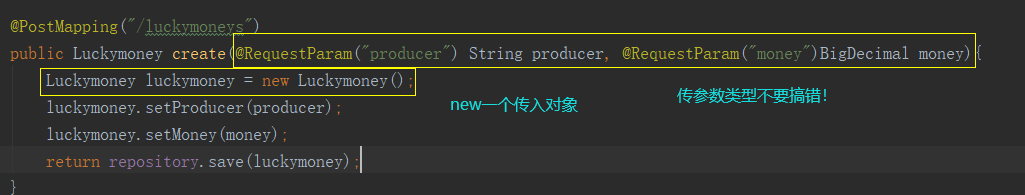


1. 新建Cotroller类

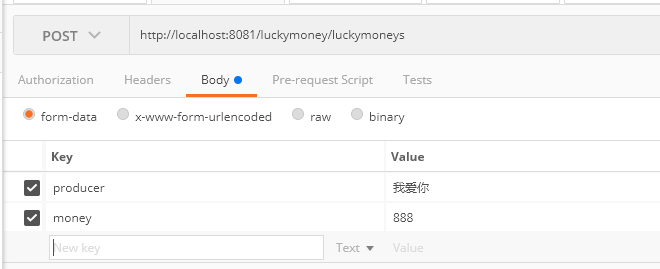


并有sql语句打印出来

并把其他几个语句写完



手动写的数据和Jpa管理的组件不一样。可能导致重复id不能写进数据从而报错

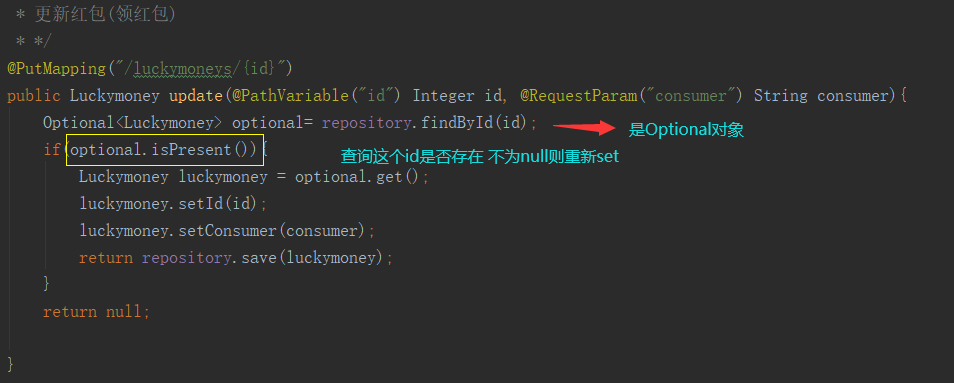
可用postman写入

如果数据库中文乱码则需要在数据库url后加

通过id查询



更新id

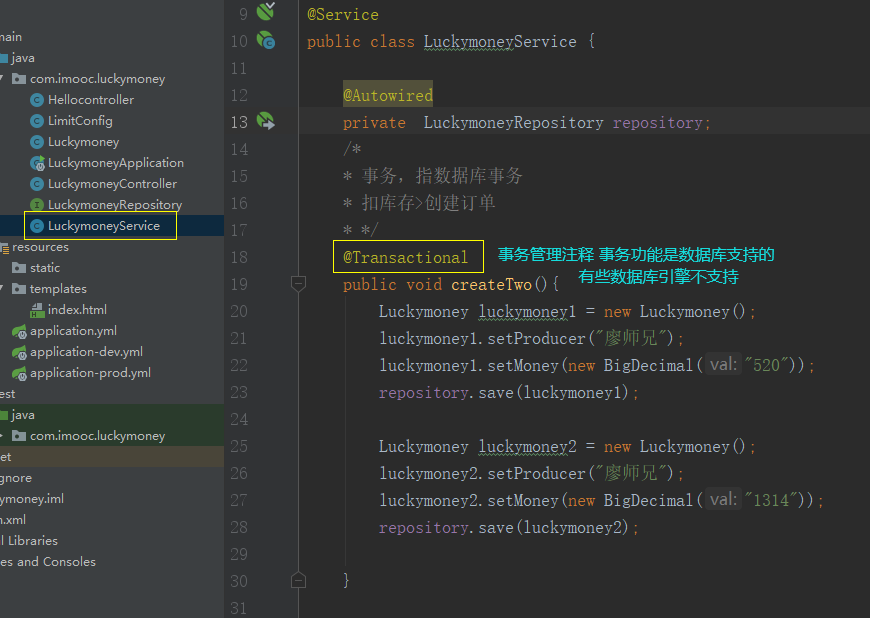


数据库又问题则需要查控制台中sql语句

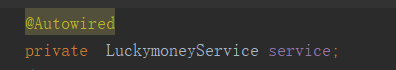
配置文件中的 show-sql为是否显示的开关

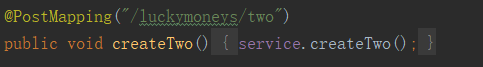
数据库事务，是指作为单个逻辑工作单元执行的一系列操作，要么完全地执行，要么完全地不执行。例如扣除库存和提交订单需要同时执行

修改数据库条件使得两条数据一条不符合标准



InnoDB这个引擎支持事务管理





在Controller中使用即可

总结

1. Springboot介绍（是什么，为什么要学）

2. 第一个SpringBoot应用（idea，官方布置）

3. 项目配置（①单个配置@Value ②对象配置@ConfigurationProperties ③区分环境）

4. Controller的使用（①方式 @Controller + @ResponseBody = @RestController

② 获取参数url@PathVariable 参数@RequestParam）

5.数据库操作 Spring-Data-Jpa 定义接口extends JpaRepository

（①save创建更新②findById id查找③findAll查询）

1. 事务 特指数据库事务 @Transactional

学完仅仅是入门 >> 进阶>> springboot微信点餐实战 >> 微服务实战