**JAVA学习记录**

**001： IDEA历史版本下载（一般使用2017.1.5）：https://www.jetbrains.com/idea/download/previous.html**

**002： IDEA破解：https://www.jianshu.com/p/ad3830095fb3**

**003： MAVEN集成SSM：** [**http://blog.csdn.net/chengtanyong4777/article/details/79025753**](http://blog.csdn.net/chengtanyong4777/article/details/79025753)

**https://www.cnblogs.com/hackyo/p/6646051.html**

**004： 标注文件夹为Source文件夹，只有标注为Source的文件夹才能在其下建立JAVA类文件和Java包。Source的意思就是需要编译的。**

**005： Intellij IDEA中的资源文件中在一个包中建立多个包的方法是：先建立一个包com.gym.x1，然后再建立一个包com.gym.x2。**

**006： Intellij IDEA集成lombok：**[**http://blog.csdn.net/zhglance/article/details/54931430。lombok**](http://blog.csdn.net/zhglance/article/details/54931430。lombok)**的作用是让bean文件不用写getter,setter,toString,equals,hashCode等方法的代码。**

**007：如果还不生效，则需要重新生成一下MAVEN依赖，方法是：右键pom.xml然后再右键菜单中选择maven然后再选在菜单中选择Reimport。然后重新编译一下项目即可。另外其他的如springframework,mybatis等找不到也是同样的操作方法，具体原因是虽然改了pom.xml但是maven还并没有加载（或者下载？）对应的依赖，所以要重新引入一下（Reimport）。**

**008：在Intellij IDE中，如果在IMPL文件里面的Mapper依赖，类似于下图所示：**

**C:\Users\SYSWIN\AppData\Local\Temp\1519627629(1).png**

**出现下划线错误，但是编译的时候又不报错。这时的解决方案是：File->Settings->Editor->Inspections->Spring->Spring Core->Code->Autowiring for Bean Class。并将右边的Severity后面对应的第一个选项从error改成warning然后apply。**

**009：maven本地仓库的配置：**[**https://www.cnblogs.com/eagle6688/p/7838224.html**](https://www.cnblogs.com/eagle6688/p/7838224.html)

**Maven需要配置的文件有：conf/settings.xml 与IDEA的配置Idea**

**File->Settings->Build,Execution,Deployment->Build Tools->Maven：**

1. **maven home directory 改成maven本地路径 D:/apache-maven-3.5.2**
2. **User settings file 改成maven本地路径/conf/settings.xml D:\apache-maven-3.5.2\conf\settings.xml override勾选**
3. **Local Repository 改成maven本地仓库 E:\MavenRepository override勾选**

**010：Tomcat配置JNDI**

**配置context.xml并增加<Resource 标签，具体内容网上搜索**

**将mysql-connector-java-5.1.45-bin.jar文件拷贝到tomcat的lib目录下**

**011：本地host的作用**

**Hosts文件位置：C:\Windows\System32\drivers\etc**

**我们知道在网络上访问网站，要首先通过DNS服务器把网络域名（**[**www.XXXX.com**](http://www.xxxx.com/)**）解析成61.XXX.XXX.XXX的IP地址后，我们的计算机才能访问。要是对于每个域名请求我们都要等待域名服务器解析后返回IP信息，这样访问网络的效率就会降低，而**[**Hosts文件**](https://www.baidu.com/s?wd=Hosts%E6%96%87%E4%BB%B6&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3rHn3uynYrj6LP1-hmvmv0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWnYPHnYP10)**就能提高解析效率。根据Windows系统规定，在进行DNS请求以前，Windows系统会先检查自己的Hosts文件中是否有这个地址映射关系，如果有则调用这个 IP地址映射，如果没有再向已知的DNS服务器提出域名解析。也就是说Hosts的请求级别比DNS高。  
  
知道了Hosts文件的工作方式，那在具体使用中它有哪些作用呢?  
1、加快域名解析  
对于要经常访问的网站，我们可以通过在Hosts中配置域名和IP的映射关系，这样当我们输入域名计算机就能很快解析出IP，而不用请求网络上的DNS服务器。  
2、方便局域网用户  
在很多单位的局域网中，会有服务器提供给用户使用。但由于局域网中一般很少架设DNS服务器，访问这些服务要输入难记的IP地址，对不少人来说相当麻烦。现在可以分别给这些服务器取个容易记住的名字，然后在Hosts中建立IP映射，这样以后访问的时候我们输入这个服务器的名字就行了。  
3、屏蔽网站  
现在有很多网站不经过用户同意就将各种各样的插件安装到你的计算机中，有些说不定就是木马或病毒。对于这些网站我们可以利用Hosts把该网站的域名映射到错误的IP或自己计算机的IP，这样就不用访问了。我们在Hosts写上以下内容：  
127.0.0.1 #屏蔽的网站 0.0.0.0 #屏蔽的网站  
这样计算机解析域名就解析到本机或错误的IP，达到了屏蔽的目的。  
4、顺利连接系统  
对于Lotus的服务器和一些数据库服务器，在访问时如果直接输入IP地址那是不能访问的，只能输入服务器名才能访问。那么我们配置好Hosts文件，这样输入服务器名就能顺利连接了。  
最后要指出的是，Hosts文件配置的映射是静态的，如果网络上的计算机更改了请及时更新IP地址，否则将不能访问。**

**012：disconf与spring集成**

**disconf的主要作用就是从disconf配置文件服务器上将配置文件下载到本地项目**

**上。**

[**http://lipengjavablog.iteye.com/blog/2400981**](http://lipengjavablog.iteye.com/blog/2400981)

**013：spring集成swagger springfox等**

**http://blog.csdn.net/qq\_25615395/article/details/70229139**

**它主要是通过EnableSwagger2注解，向srping context注入了一系列bean，并在系**

**统启动的时候自动扫描系统的Controller类，生成相应的api信息并缓存起来**

**014：分布式数据库框架搭建**

[**http://www.zuidaima.com/blog/3523693477465088.htm**](http://www.zuidaima.com/blog/3523693477465088.htm)

**015：redis缓存**

[**http://blog.csdn.net/acmman/article/details/53167917**](http://blog.csdn.net/acmman/article/details/53167917) **016:** **Jenkins 内部项目管理，和GIT结合使用，用于将GIT的更新自动部署到服务器上，**

**并且可以查看部署日志。**

**017：jmeter性能测试工具 网上下载jmeter的ZIP包，解压到相应路径，设置环境变量**

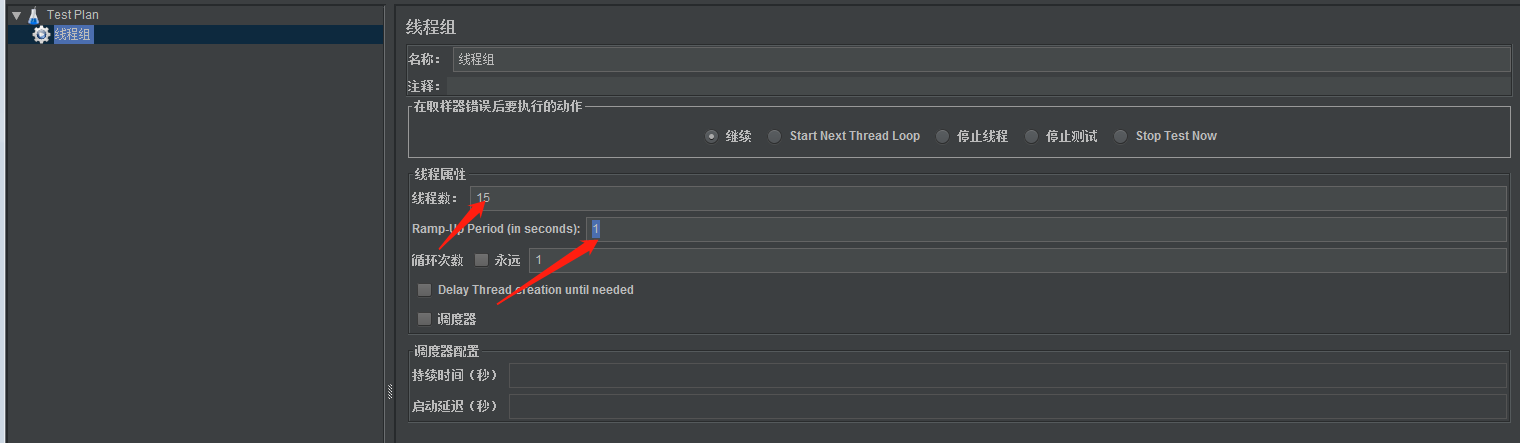
**JMETER\_HOME:D:\apache-jmeter-4.0**

**Path:%JMETER\_HOME%\bin; CLASSPATH:%JMETER\_HOME%\lib\ext\ApacheJMeter\_core.jar;%JMETER\_HOME%\lib\jorphan.jar;**

**然后打开D:\apache-jmeter-4.0\bin\ jmeter.bat 运行jmeter**

1. **Options🡪choose language🡪chinese(simplified)**
2. **默认出来在左边应该有一个TestPlan（测试计划）单击该测试计划可以修改名称**
3. **右键测试计划添加Threads(Users)🡪线程组，设置**

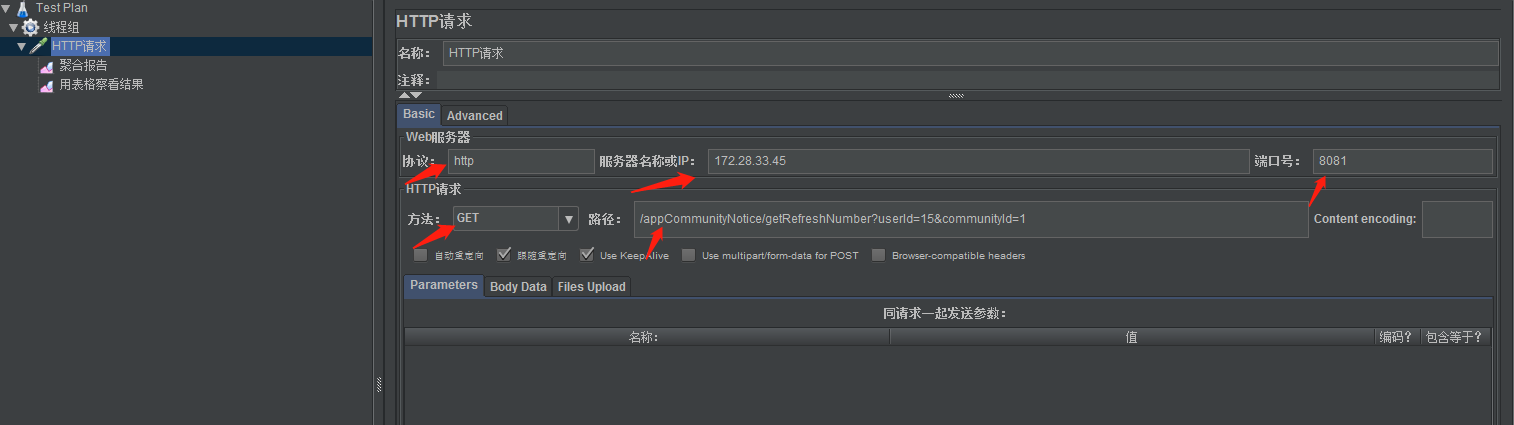
**线程的个数和在多少时间内将这些线程执行完，循环次数填1，如下图所示：**

****

**D）：右键线程组选择添加🡪Sampler🡪HTTP请求**

**新建HTTP请求，设置HTTP请求的IP地址，端口，访问路径等，参数就放到访问**

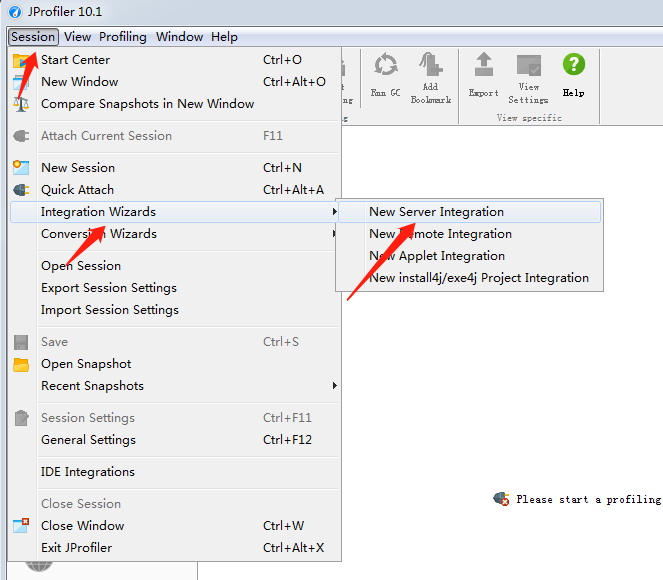
**路径里边。如下图所示：**

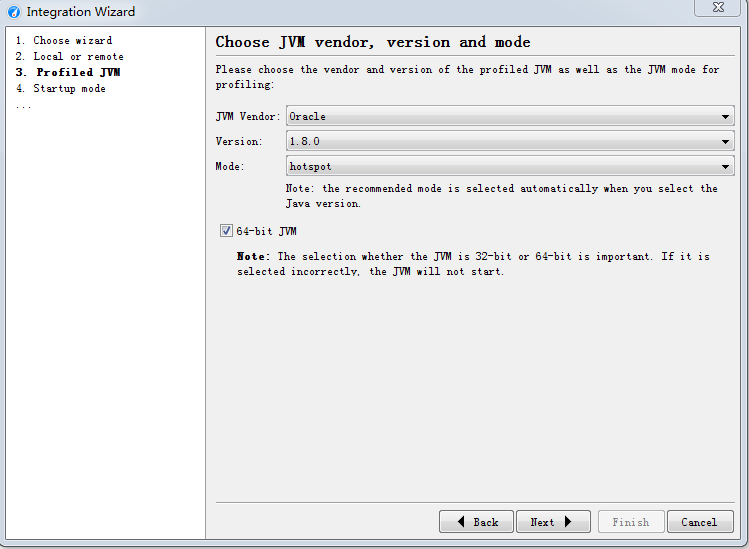
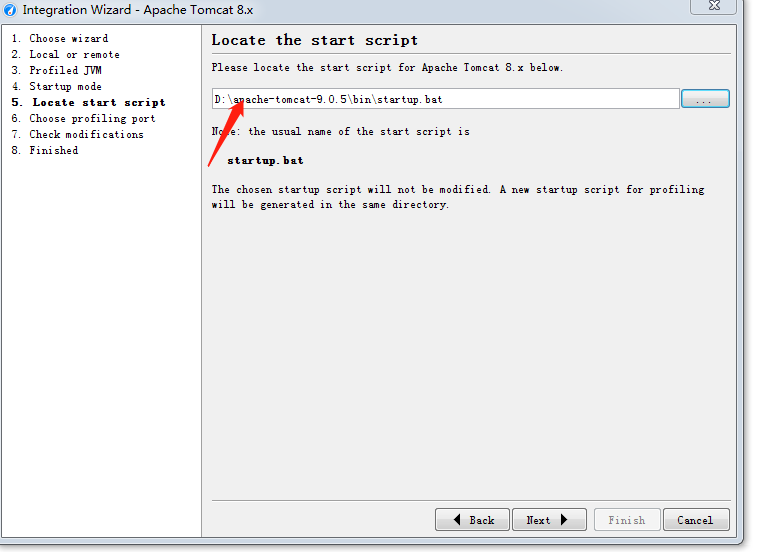
****

**E)：右键HTTP请求分别添加聚合报告和用表格查看结果来看看测试结果**

**018：jprofiler性能监控工具，可以监控tomcat服务器的CPU使用量，内存使用量，**

**线程数，等等。**

****

** **

**启动的时候要确保tomcat处于关闭状态，因为jprofiler会新起一个tomcat。**

**019：Oauth2.0身份验证协议** [**https://www.cnblogs.com/flashsun/p/7424071.html**](https://www.cnblogs.com/flashsun/p/7424071.html)

**020：十年架构师经验分享：**[**https://blog.csdn.net/zhangcanyan/article/details/79123707**](https://blog.csdn.net/zhangcanyan/article/details/79123707)

**021：MYSQL有自动行锁，针对一个数据的并发操作，如果并发量不是很大，可以不用在代码层面做加锁操作，但是MYSQL的行锁默认是锁住整个表的，必须要在操作的列上加一个索引来实现行锁住功能。比如点击量、阅读数等等这些操作。就在数据库对应的列上加一个索引。**

**022：springboot 搭建SSM框架并集成分页插件**

**https://www.cnblogs.com/qlqwjy/p/8463822.html**

**023：springboot API和REFERENCE**

[**https://docs.spring.io/spring-boot/docs/**](https://docs.spring.io/spring-boot/docs/)

[**https://docs.spring.io/spring-boot/docs/2.0.1.RELEASE/reference/htmlsingle/**](https://docs.spring.io/spring-boot/docs/2.0.1.RELEASE/reference/htmlsingle/)

**024：线程的状态有：新建状态：new创建线程。就绪状态start()方法使处于新建状态处于就绪状态。运行状态run()方法使线程从就绪状态进入到运行状态。阻塞状态（阻塞状态分为等待阻塞：运行中的线程执行wait()方法使线程处于等待阻塞状态、同步阻塞：线程在获取synchronized同步锁失败后处于同步阻塞状态，这种状态下一一般可以给程序设置一个等待锁的时间，超过该时间后如果此线程还未获得该锁，那么就继续执行不再等待锁。其他阻塞：线程调用了sleep()或join()发出了I/O请求时，线程就会处于阻塞状态）。死亡状态，当线程执行完任务后会自然终止或者调用终止方法。JAVA创建线程有三种方式：实现runnable接口、继承thread类本身、通过callable和future创建线程。**

**025：异步方法，同步的意思是在一个线程从顺序调用ABC三个方法时，这三个方法是顺序被调用的，就是说B方法必须在A方法执行完了才能被执行，C方法必须等B方法执行完了才能被执行。异步的意思就是假设B方法是异步方法，那么线程在调用B方法之后，不用等到B方法执行完，而是在调用B方法之后立刻执行C方法。一般比如I/O操作、点击量统计、耗时操作都可以设置成异步方法，在SPING中使用@Async注解来标识一个方式是异步方法。**

**026：菜鸟教程** [**http://www.runoob.com/git/git-create-repository.html**](http://www.runoob.com/git/git-create-repository.html)

**027：JAVA的三层架构是指页面展示层、业务逻辑层、数据访问层。而通常所说的MVC只存在于页面展示层（M-DOMAIN、V-HTML、C-CONTROLLER）而业务逻辑层指的是SERVICE、数据访问层指的是MYBATIS（MAPPER、DAO）**

**028：有时候WEB项目不能启动是因为各版本的JAR包版本冲突，例如spring-boot-starter-redis的版本是1.0.0，但是jedis的版本是2.9.0显然是不行的，可以在maven仓库里看spring-boot-starter-redis支持的最高的jedis版本**

**029：GIT配置多个KEY和账户：如果电脑上已经配置了全局的user.name和user.email那么则使用以下两条命令取消git config --global --unset user.name和git config --global --unset user.email.**

**1：在电脑的任意位置右键Git Bash Here 在命令行输入ssh-keygen -t rsa -C "464293746@qq.com"🡪点击enter键。然后在Enter file in which to save the key:后面输入/c/Users/HP/.ssh/id\_rsa\_gihub（第一个必须要指定全路径，因为有的GIT安装好以后并没有创建.ssh文件夹）。然后一路ENTER直到KEY创建成功为止。然后定位到/c/Users/HP/.ssh/目录下打开GIT命令行输入ssh-keygen -t rsa -C "muxue@syswin.com",然后在Enter file in which to save the key:后面输入id\_rsa\_syswin然后一直ENTER创建第二个KEY。然后在**

**/c/Users/HP/.ssh/目录下创建config.txt内容如下：**

**#github**

**Host git@github.com**

**HostName https://github.com**

**User git-mx**

**IdentityFile ~/.ssh/id\_rsa\_github**

**#syswin**

**Host git@toongitlab.syswin.com**

**HostName http://toongitlab.syswin.com/**

**User MuXue**

**IdentityFile ~/.ssh/id\_rsa\_syswin**

**注意host和HostName一定要写对，然后还有对应的KEY。**

**然后分别去**[**https://www.github.com上用用户登录然后点击头像Settings**](https://www.github.com上用用户登录然后点击头像Settings)**然后SSH and GPG keys然后New SSH key然后title填写之前创建的KEY的名称，位置在/c/Users/HP/.ssh/（这里是id\_rsa\_github）然后右击该目录下的id\_rsa\_github.pub编辑。将里面的内容复制到key里面，然后点Add SSH key。第二个KEY也是一样的。弄完之后，本地的KEY就和GITHUB或者GITLAB远程仓库建立了对应关系。弄完之后去任意位置，Git bash here然后依次输入一下命令来验证与远程仓库的连接是否建立成功：ssh-agent bash,ssh-add ~/.ssh/id\_rsa\_github,ssh-add ~/.ssh/id\_rsa\_syswin。将两个KEY添加到AGENT中。然后依次执行ssh -T** [**git@github.com**](mailto:git@github.com) **（ssh -T git@toongitlab.syswin.com）如果出现HI MX。。。(welcome牟雪)就表示建立连接成功。**[**连接成功之后就可以去GITLAB上将代码的地址拷贝过来例如git@toongitlab.syswin.com:gov\_lztoon/lztoon-cms.git（config.txt文件里的Host一定要和这里的git@toongitlab.syswin.com这一部分相同）然后只用git**](mailto:连接成功之后就可以去GITLAB上将代码的地址拷贝过来例如git@toongitlab.syswin.com:gov_lztoon/lztoon-cms.git（config.txt文件里的Host一定要和这里的git@toongitlab.syswin.com这一部分相同）然后只用git) **clone** [**git@toongitlab.syswin.com:gov\_lztoon/lztoon-cms.git**](mailto:git@toongitlab.syswin.com:gov_lztoon/lztoon-cms.git) **命令来拉取代码(下一次拉取代码，或者电脑重启之后需要执行命令ssh-agent bash和ssh-add ~/.ssh/id\_rsa\_github来建立远程与本地的连接,猜测应该是有一个超时机制)。然后cd到某个项目的根目录执行git branch -a 查看当前项目的所有分支和当前自己所在的分支，绿色那个表示自己所在的分支，使用 git checkout -b branchA origin/branchA 来切换分支。git commit将代码提交到本地仓库， git push将自己本地仓库的代码push到远程仓库**

**030：GIT将本地项目上传到GITHUB远程仓库**[**https://blog.csdn.net/lucky\_lxg/article/details/77849212**](https://blog.csdn.net/lucky_lxg/article/details/77849212)

**031：当本地有多个GIT账户时要把一个新项目整个上传到GIT上的时候**

1. **首先在项目文件夹下git bash here**
2. **ssh-agent bash**
3. **ssh-add ~/.ssh/id\_rsa\_github**
4. **Git init**
5. **touch .gitignore（将那些不需要上传的GIT的文件过滤掉）**
6. **git add .**
7. **git config user.name git-mx**
8. **git config user.email 464293746@qq.com**
9. **git commit -m “commit description”**
10. **git remote add mx** [**https://github.com/git-mx/elies-base.git**](https://github.com/git-mx/elies-base.git)
11. **git pull –rebase mx master**
12. **git push -u mx master -f (不加-f始终报错，-f是强制的意思)**

**032：当本地有多个GIT账户时要提交修改的代码到远程仓库时:**

1. **首先在项目文件夹下git bash here**
2. **ssh-agent bash**
3. **ssh-add ~/.ssh/id-rsa\_github**
4. **git add .**
5. **git config user.name git-mx**
6. **git config user.email** [**464293746@qq.com**](mailto:464293746@qq.com)
7. **git commit -m “commit description”**
8. **git push**

**033：首次从公司gitlab上拉取代码。在远程仓库建了自己的分支feature-mx,但是在**

**本地命令行用git branch -a查看分支的时候却看不到刚刚自己建立的分支，那是**

**因为分支和代码一样，也分远程分支和本地分支，本地的分支只是远程仓库分支的一个“拷贝”，而本地的分支是首次拉取代码的时候从远程仓库的分支拷贝过来的。所以在远程仓库建立了新的分支的时候要执行git fetch命令将远程的分支同步到本地，然后再执行git branch -a就可以看见刚才建立的分支了。**

**034：git commit操作的时候如果遇到Changes not staged for commit错误那么**

**需要先执行一下命令git add .**

**035: mybaits的update操作返回的都是操作的行数**

**update post\_info set pi\_status=8 where pi\_id=#{piId} and**

**6>pi\_status**

**return postInfoMapper.updateStatus >= 1;**

**036：搭建前后端分离框架需要做的事情**

**<1>：HTML、CSS、JS。**

**<2>：AJAX或Fetch。**

**<3>：学习一个前端的框架， React或者Vue或者Angularjs2都可以。**

**<4>：学会一个前端的路由框架， 如React-Router或者Vue-Router。**

**<5>：在学会3的基础上你肯定已经搭建好前端的开发环境了，所有和后端的交互走**

**AJAX或者Fetch。**

**<6>：SpringMVC 不再返回一个视图， 直接返回JSON。**

**<7>：前端从后端获取的所有数据都是JSON，至于怎么去更新页面， 你学会3后你**

**肯定已经知道了。**

**<8>：页面跳转相关的东西你学会4后你肯定已经知道了。**

**037：git reset –hard HEAD^ HEAD^^ HEAD^3 倒数第四次提交**

**038：在浏览器地址栏直接输入chrome://apps/然后再点击postman打开**

**Postman，**

**039：关系型数据库遵循的ACID规则**

**事务在英文中是transaction，和现实世界中的交易很类似，它有如下四个特性：**

**A (Atomicity) 原子性**

**原子性很容易理解，也就是说事务里的所有操作要么全部做完，要么都不做，事务成功的条件是事务里的所有操作都成功，只要有一个操作失败，整个事务就失败，需要回滚。**

**比如银行转账，从A账户转100元至B账户，分为两个步骤：1）从A账户取100元；2）存入100元至B账户。这两步要么一起完成，要么一起不完成，如果只完成第一步，第二步失败，钱会莫名其妙少了100元。**

**C (Consistency) 一致性**

**一致性也比较容易理解，也就是说数据库要一直处于一致的状态，事务的运行不会改变数据库原本的一致性约束。**

**例如现有完整性约束a+b=10，如果一个事务改变了a，那么必须得改变b，使得事务结束后依然满足a+b=10，否则事务失败。**

**I (Isolation) 独立性**

**所谓的独立性是指并发的事务之间不会互相影响，如果一个事务要访问的数据正在被另外一个事务修改，只要另外一个事务未提交，它所访问的数据就不受未提交事务的影响。**

**比如现在有个交易是从A账户转100元至B账户，在这个交易还未完成的情况下，如果此时B查询自己的账户，是看不到新增加的100元的。**

**D (Durability) 持久性**

**持久性是指一旦事务提交后，它所做的修改将会永久的保存在数据库上，即使出现宕机也不会丢失。**

**040：mongodb历史版本下载**[**https://www.mongodb.org/dl/linux/**](https://www.mongodb.org/dl/linux/)

**041：linux新建目录命令mkdir**

**Linux从命名命令mv dirA dirB**

**Linux临时关闭防火墙命令 systemctl stop firewalld**

**Linux查看线程命令 ps aux | grep mongodb**

**Linux查看本机IP命令ip addr**

**Linux查看占用某个端口的进程信息netstat -lnp|grep 8000**

**Linux杀死某个进程kill -9 154564**

**042：reactjs入门教程**[**https://blog.csdn.net/column/details/14545.html**](https://blog.csdn.net/column/details/14545.html)

**043：starUML画时序图的步骤**

1. **在新建FILE之后**
2. **在右边的树结构右键main节点选择Add Diagram🡪Sequence Digagram**
3. **在左下方interactions菜单里选择Lifeline**
4. **在左下方interactions菜单里选择Message**
5. **在左下方interactions菜单里选择Reply Message**

**044：starUML破解教程**

**A） 将以下JS代码拷贝到C:\Program Files (x86)\StarUML\www\license\node\LicenseManagerDomain.js第24行的var pk, decrypted;后面**

**B）打开starUML然后help选择Enter License然后**

**C）用户名输入StarUML密码输入8888-000-8888**

**045：starUML各个diagram的作用**

**Class:类的关系图**

**046：SpringBoot中mybatis的mapper文件坑**

**Mybatis mapper.xml里面的resultMap里的column一定要**

**<include refid对应的<sql>标签里的as 后面的一致是因为在resultMap里使用ai\_id,app\_id来和aiId,appId进行映射的，但是在查询的时候加include把本来的ai\_id改成了aiId、把app\_id改成了appId,然后再在map中映射的时候就找不到ai\_id,app\_id.**

**047：SpringBoot高版本配置disconf**

**Springboot在配置disconf的时候，高版本的springboot会报**

**NoSuchMethod fond:parseStringValue,直接把配置文件里引用com.baidu.disconf.addons.propertis.ReloadingPropertyPlaceholderConfigurer的这个bean给干掉**

**048：SptingBoot配置外部启动的tomcat**

**Springboot在配置外部tomcat的时候的步骤如下：**

**<1>jar改成war**

**<2>exclusions tomcat**

**<3>javax.servlet-api不用写版本号**

**<4>maven-war-plugin不用写版本号**

**<5>写一个与启动类ApplicationStarter.java同级别的文件继承SpringBootServletInitializer，并把启动类名ApplicationStarter.java放进去**

**<6>也可以不要启动类ApplicationStarter.java直接用5的类**

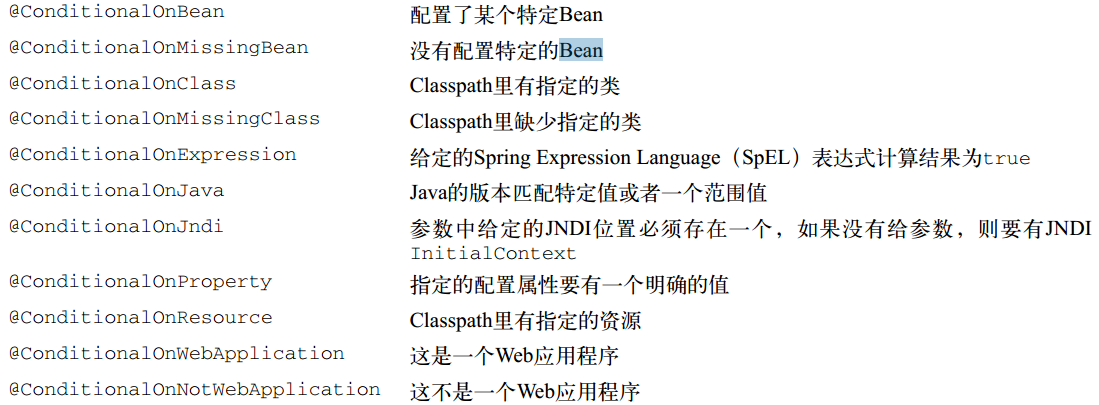
**049：SpringBoot的某些starter并不一定是完美的，有时候你配置的starter相关的功能无法使用，就将starter干掉，用maven依赖注入，例如1.5.4版本的spring-boot-starter-aop就可能会导致你写的切面代码无法使用（编译并不会报错）这时候你就需要换成aop的manve依赖。**

**050：SpringBoot中disconf下载的.yml配置文件，是无法注入到容器中的，这目前是disconf的一个BUG**

**051：Spring条件配置，即在满足某个条件的情况下在进行某一项Spring配**

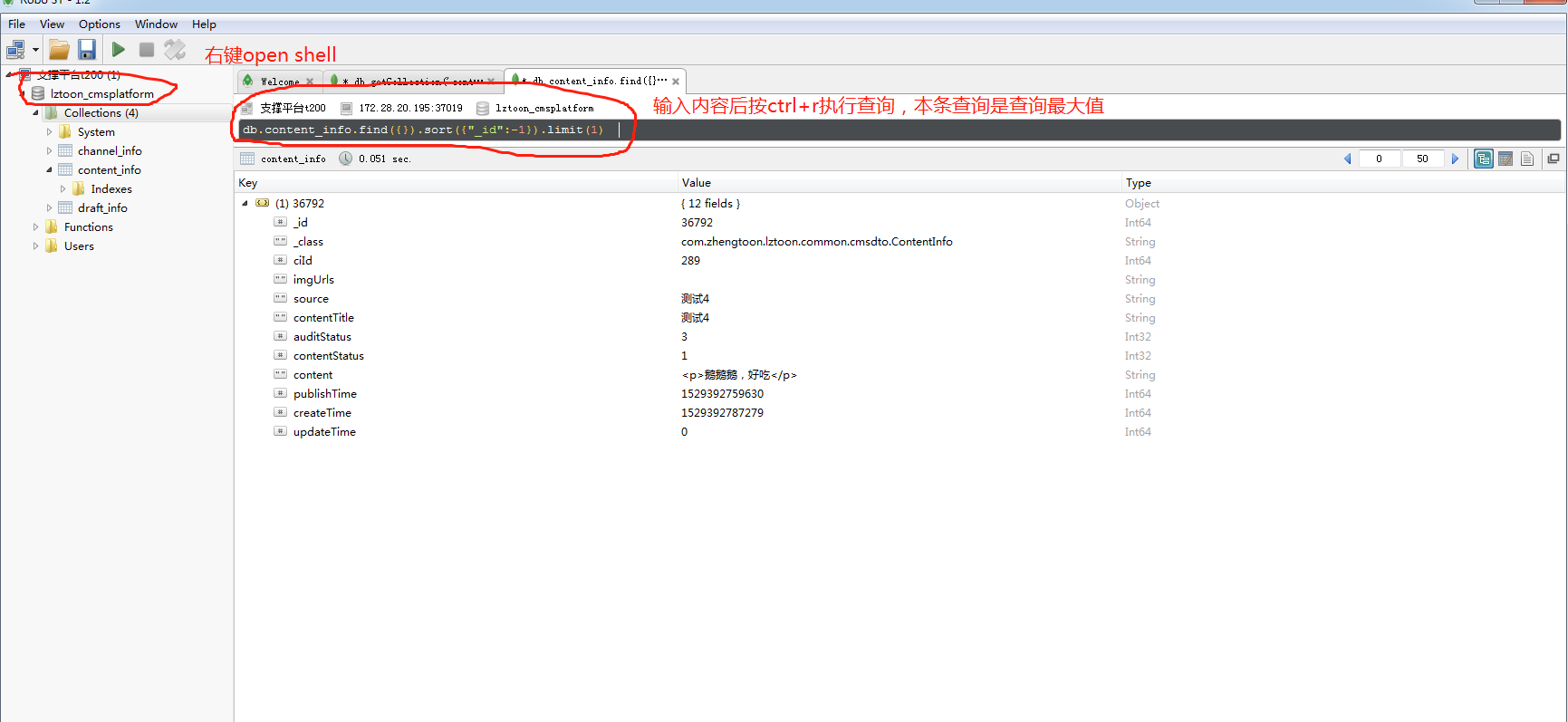
**置，例如下面的例子是classpath存在JdbcTemplate时才配置**



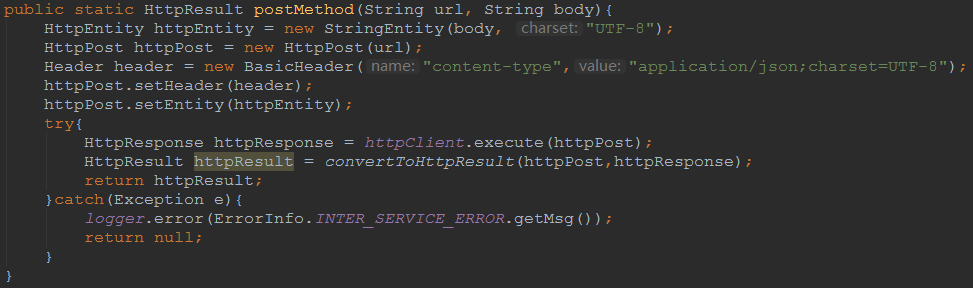


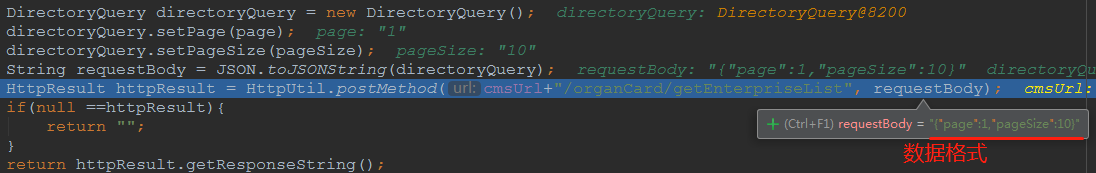
**052：PAAS Platform as a Service**

**053：mongodb查询最大值**

****

**054：利用HttpPost发送post请求**



****

**055：AND org\_name like CONCAT(‘%’, REPLACE(REPLACE(#{orgName}, ‘%’, ‘/%’), ‘\_’, ‘/\_’), ‘%’) ESCAPE ‘/’**

**056：git reset –hard head^这个操作会把本地和远程仓库的代码都还原**

**了，恢复的方式是:首先git reflog ,然后再用**

**Git reset –hard b7057a9**

**057: org.apache.commons.lang3.StringUtils**

**String str1 = null;**

**String str2 = “ ”;**

**StringUtils.isNotBlank(str1) = false; isBlank = true**

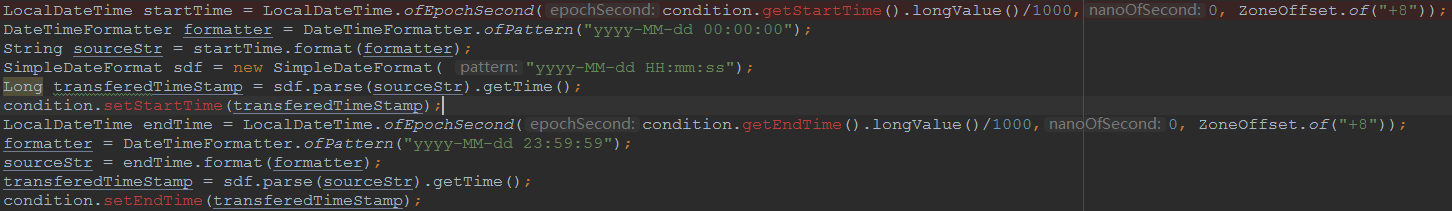
**StringUtils.isNotBlank(str2) = false; isBlank = true**

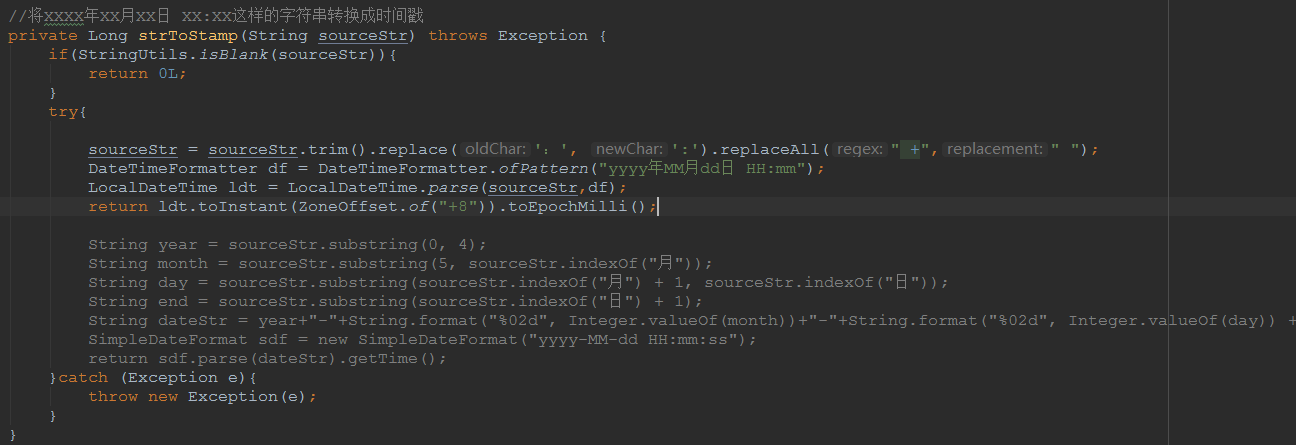
**058：Calendar.getInstance().get(Calendar.HOUR)12小时制**

**Calendar.getInstance().get(Calendar.HOUR\_OF\_DAY) 24小时制**

**059：时间戳转换成XX年XX月XX日 反过来也可以**







**060：在MySql中有使用全模糊查询的列不用建立索引，因为全模糊查询是不**

**会使用索引的**

**061：@RestController无法返回ModelAndView这类Servlet容器，它只**

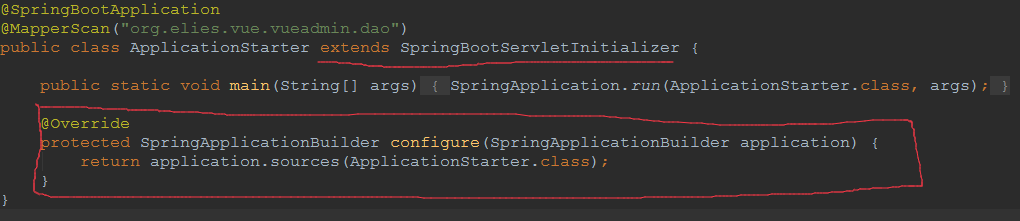
**能用于返回String或者JSON格式的数据。@Contrller注解不能用于**

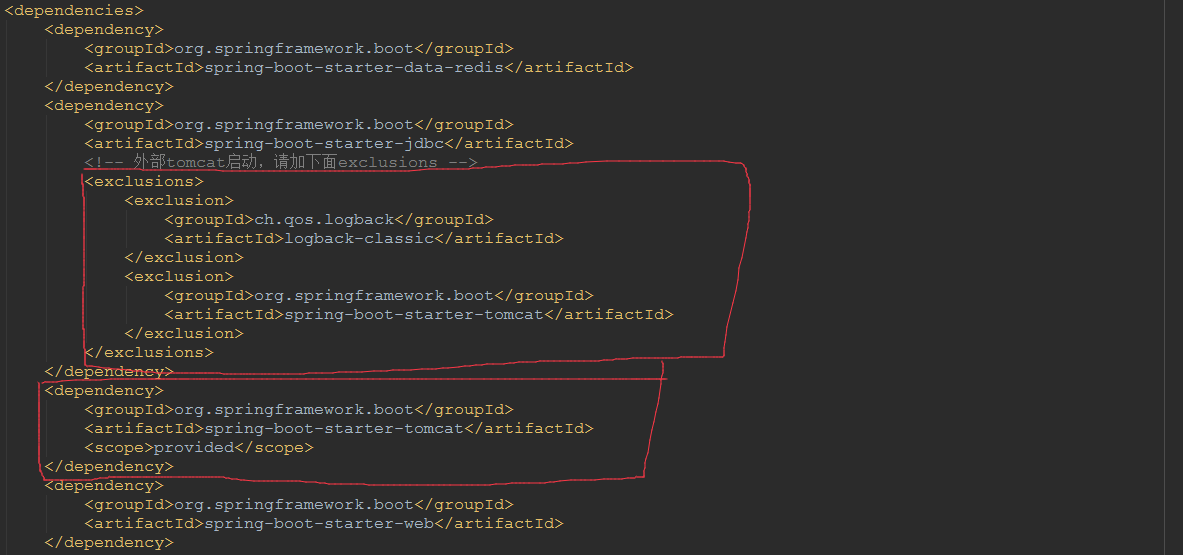
**返回自定义格式的数据，只能返回String或者ModelAndView这类**

**Servlet容器。**

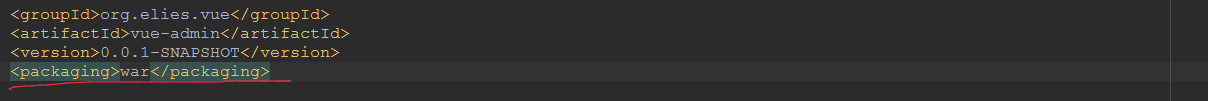
**062：使用IDEA生成WAR包并部署到TOMCAT中**

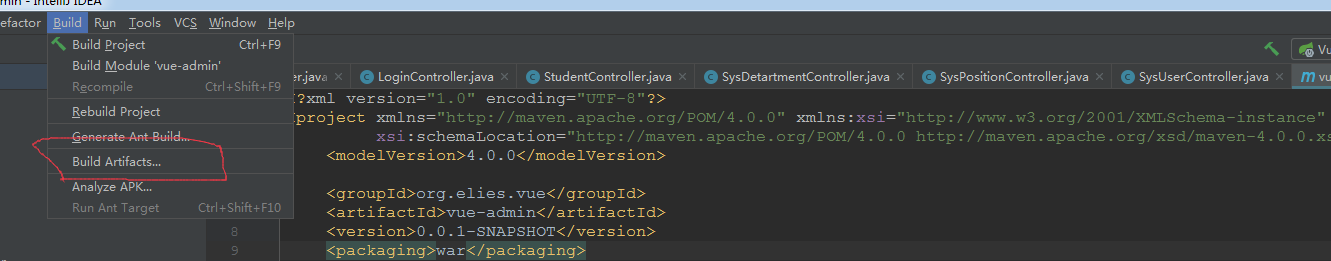
**首先SpringBoot项目要做如下修改**

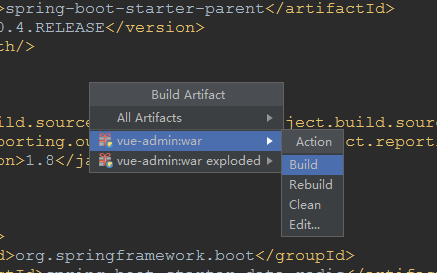












**Build成功过后把target根目录下的xxxx.war拷贝到tomact的webap**

**目录下面，在bin里面启动startup.bat运行tomcat,然后在浏览器里面**

**输入Locahost:8080/xxx/swagger-ui.html 就可以了XXX就是war包**

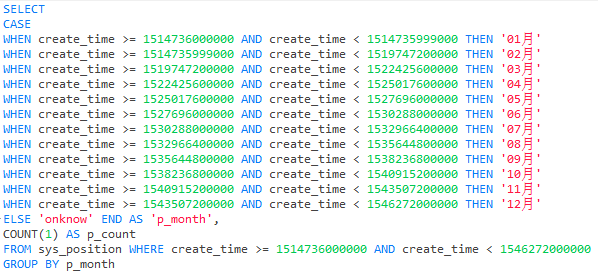
**的名称**

**063 在Spring框架的Controller如果是有输出流的控制器，就不需要再**

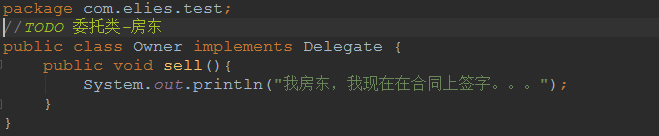
**返回结构类型的数据，如果返回了那么JAVA后台就会抛出Could not find**

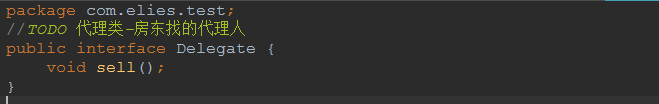
**acceptable representation异常。**

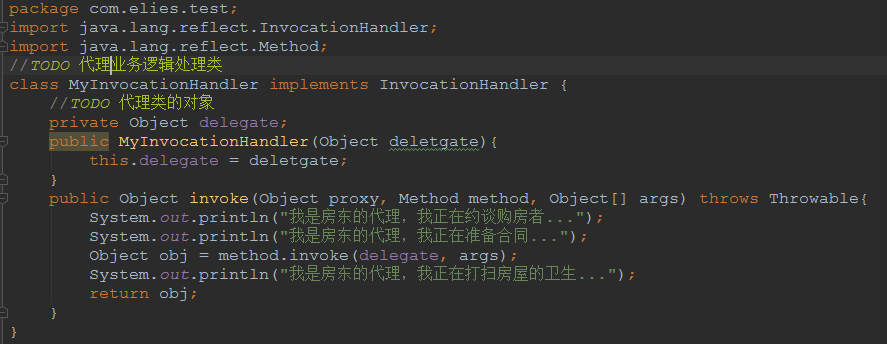
**064 MYSQL按月查询语句**

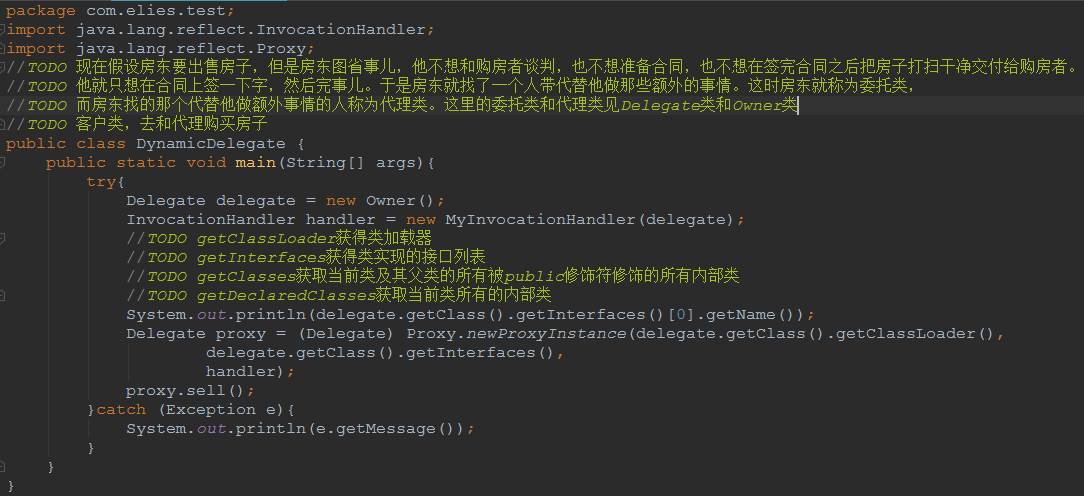


**065 powerdesigner将表导出到word，在已经存在下载模板的前提下，在菜单中选择report-Generate Report就可以导出架构文档中需要的字段了。**

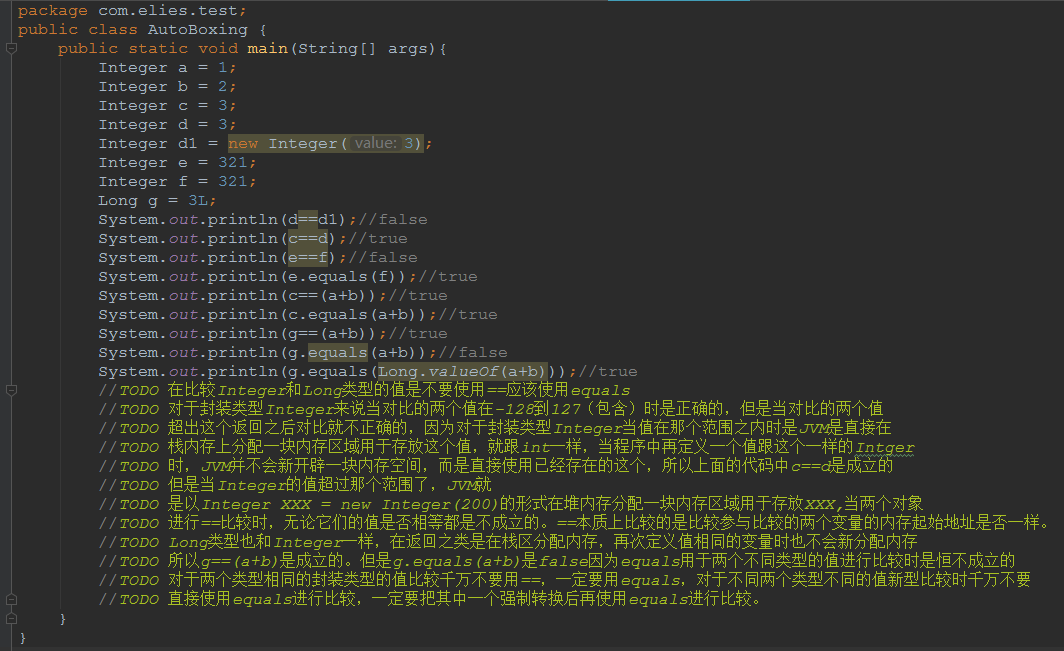
**066 JAVA动态代理**







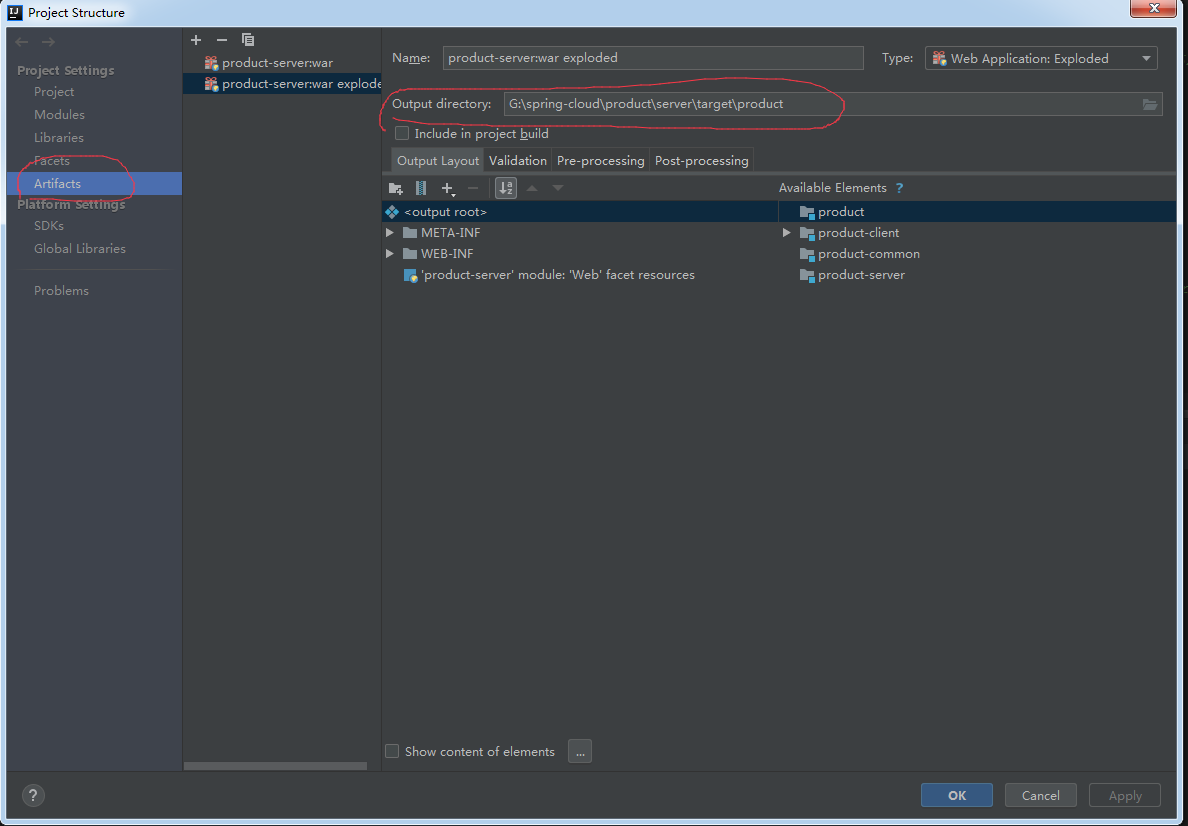
**067 自动封箱类型的比较陷阱**



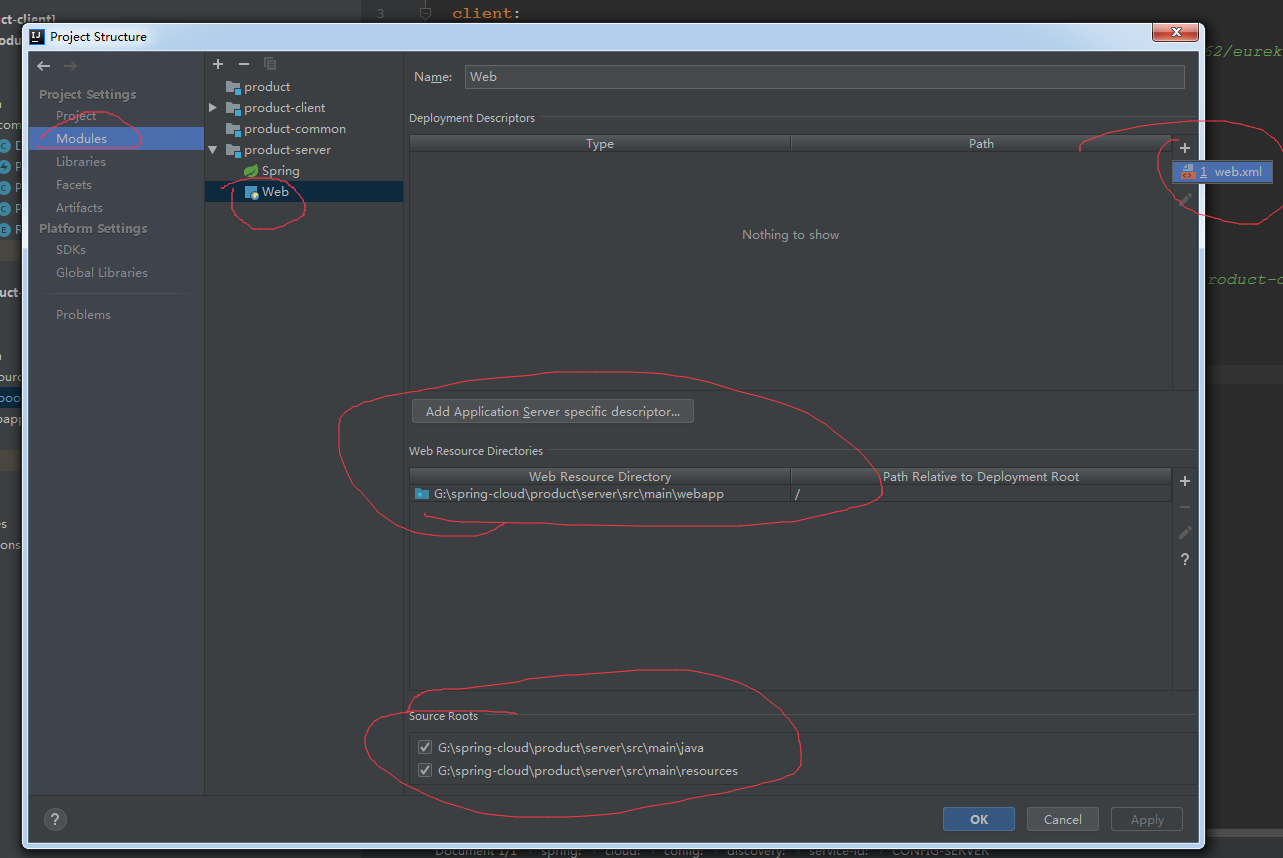
**068 代码检查工具CheckStyle、FindBug 、Klocwork**

**069 IntellijIDEA如果使用Ctrl+Shift+r全局替换了内容，但是又不小心替换错了，则可以通过local history来回复具体做法是右键项目名称点击右键菜单里面的local history再选择Show History然后按住Shift键多选，选好要恢复的文件后再点击右键，在右键菜单中选择Revert即可恢复到指定版本。**

**070 IntellijIDEA生成的目标文件中，xxx.war代表的就是war包，xxx.war exploded 对应的就是和war包路径同级目录的xxx文件夹，即一个是war包一个是文件夹，可以在Project Structure里面指定目标war包和目标项目文件，指定方式如下图**



**071 某些特殊情况下可能需要SpringBoot项目写需要web.xml文件，可以像下面这样指定**



**072 使用ctrl+shift+alt+n来查找类，查找的类可以是jar包里面的**

**073 在调试时，可以在Debugger那一栏的右边Variables那里点击+号和-号来动态设置某个变量的值例如当前断点处有一个变量叫做path，那么就可以写成 path=”AAAA”**