因为我们接下来要换一个环境基于develop分支的代码在本地打包，打成一个jar包，然后将这个jar可以用scp从本地拷贝到集成测试环境的机器上去，然后在集成测试机器上，直接java -jar启动系统，就可以了853769620

在集成测试环境的机器上，其实是应该让这个系统去连接集成测试环境的mysql数据库的，所以这个配置呢就跟在开发环境连接的那个数据是不一样的。

所以这个问题需要用maven profile支持多环境的配置文件的适配

我本来是想用阿里云的机器，来演示多个环境的机器，集成测试机器，系统测试机器，验收测试机器，线上生产机器。项目阶段一采取的是最简单的一种技术架构，单块应用+3大框架整合+mysql。

部署，就是打一个jar包，在各个环境下启动，就ok了。而且初期，雏形版，其实就是给CEO和投资人看看的。就团队内部几个人，有几个账号，各个RD，各个前端，各个产品，运用，去试用一下。其实几乎可以认为就没什么用户。根本也没什么所谓的并发量和这个压力。

尽快结束项目阶段一，就是在我的本地windows机器，模仿成是各个环境的机器就可以了。然后各个环境的数据库，我就用本地mysql里的不同的库名就可以了。

spring boot也可以支持多profile。我个人不用spring boot profile，慎用。如果你以后系统变得很复杂了，我自己之前设计过一个系统，他就是要自己搞一个针对自己系统的复杂的xml文件，此时你用spring boot profile是不太合适的。

很多大公司里，是用公司统一的部署框架去部署的，要求的是你的系统针对不同的环境可以提供不同的指定的文件，META-INF/boot.properties。

所以还是用我之前讲解的maven profile多环境支持，可以解决所有的问题，扩展性很好

dev：开发环境

beta：集成测试/联调测试的环境

src/main/profiles/dev

src/main/profiles/beta

<profiles>

<profile>

<id>dev</id>

<build>

<resources>

<resource>

<directory>src/main/profiles/dev</directory>

<includes>

<include>\*\*/\*.xml</include>

<include>\*\*/\*.properties</include>

</includes>

<filtering>true</filtering>

</resource>

</resources>

</build>

</profile>

<profile>

<id>beta</id>

<build>

<resources>

<resource>

<directory>src/main/profiles/beta</directory>

<includes>

<include>\*\*/\*.xml</include>

<include>\*\*/\*.properties</include>

</includes>

<filtering>true</filtering>

</resource>

</resources>

</build>

</profile>

</profiles>

mvn clean pakcage -Pbeta

java -jar target\eshop-1.0.0.jar

修改windows上的mysql的my.ini配置文件

在C:\ProgramData\MySQL\MySQL Server 5.7中，有my.ini，里面可以各种配置

max\_connections = 1000

重启mysql服务