**1. Java EE开发环境的配置**

**本文中Java EE开发环境的配置以32位Windows XP操作系统为例。**

**1.1 JDK 7安装与配置**

1. 在Oracle官方网站下载JDK 7u45（网址为<http://www.oracle.com/technetwork/java/javase/downloads/index.html>）。选择“Java Platform (JDK) 7u45”上面图标后，从弹出的列表选择Windows x86所对应的jdk文件jdk-7u25-windows-i586.exe下载。

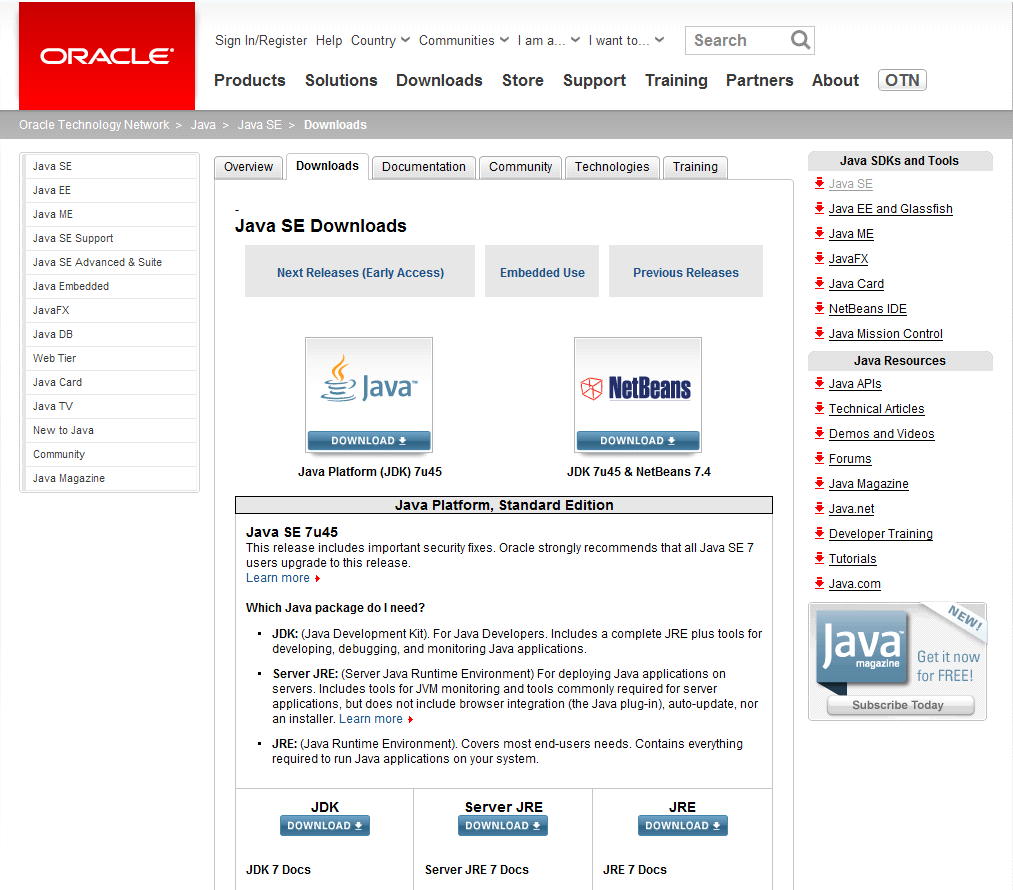


图1-2 下载JDK 步骤1

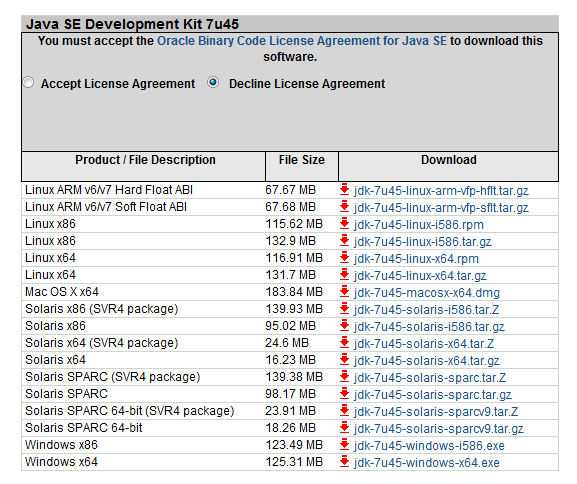


图1-3 下载JDK 步骤2

2. 双击安装文件jdk-7u25-windows-i586.exe，JDK 7安装程序运行，单击“更改”选项转到“更改当前目标文件夹”窗口，把“文件夹名称”改为D:\Java\jdk7，即把JDK 7安装在D盘上。单击“确定”，返回再单击“下一步”开始安装JDK。



图1-4 JDK设置

安装jdk后，会跳出一个窗口，为“Java SE Runtime Environment 7自定义安装”窗口，它将安装JRE（Java运行时环境），和前一步类似，单击“更改”选项转到“更改当前目的地文件夹”窗口，把“文件夹名称”改为D:\Java\jre7。单击“确定”，返回再单击“下一步”后开始安装JRE。安装完成后显示“安装完成”窗口，单击“完成”，这样JDK 和JRE都安装在D盘上了。



图1-5 JRE设置

3. 设置系统变量。右击“我的电脑”>>>“属性”>>>“高级”>>>“环境变量”。现在出现的界面如下所示

1) 点击“系统变量”的“新建”，如下图：



图1-6 设置JAVA\_HOME环境变量

在“变量名”中填入JAVA\_HOME,在“变量值”中填入D:\Java\jdk7 (JDK安装的路径),然后点击确定。

2) 再点击“系统变量”下面的“新建”，在“变量名”中填入CLASSPATH，在“变量值”中填入.;%JAVA\_HOME%\lib;(注意：在JAVA\_HOME前面有.;)，然后点击确定。



图1-7设置CLASSPATH环境变量

3) 设置PATH变量，你可以直接在“系统变量”中找到它后双击，没有的话就新建一个。如下图所示，将D:\Java\jdk7\bin目录添加到PATH变量中。

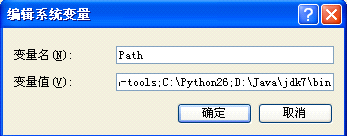


图1-8 设置PATH环境变量

**1.2 Eclipse IDE安装**

Eclipse是一个免费的Java开发平台，Eclipse以其代码开源、使用免费、界面美观、功能强大、插件丰富等特性成为Java开发中使用最为广泛的开发平台。Eclipse是一个典型的绿色软件，不需要安装，直接解压到任意文件夹并启动Eclipse.exe文件就可以运行Eclipse。本书中使用的是Eclipse Jave EE Kepler版本。下载网址为<http://www.eclipse.org/downloads/>，在此页面上选择Eclipse IDE for Java EE Developers的32位版本下载，下载后文件名为eclipse-jee-kepler-SR1-win32.zip，将其解压到路径D:\Eclipse。

**1.3 Mysql安装与配置**

1. 在官网上下载Mysql，下载网址为<http://dev.mysql.com/downloads/>，选择”MySQL Community Server (GPL)”下面的download，进入下载页面如图1-15所示，现在Mysql的版本是5.6。默认选择是Windows平台，然后选择“Windows (x86, 32-bit), ZIP Archive”。

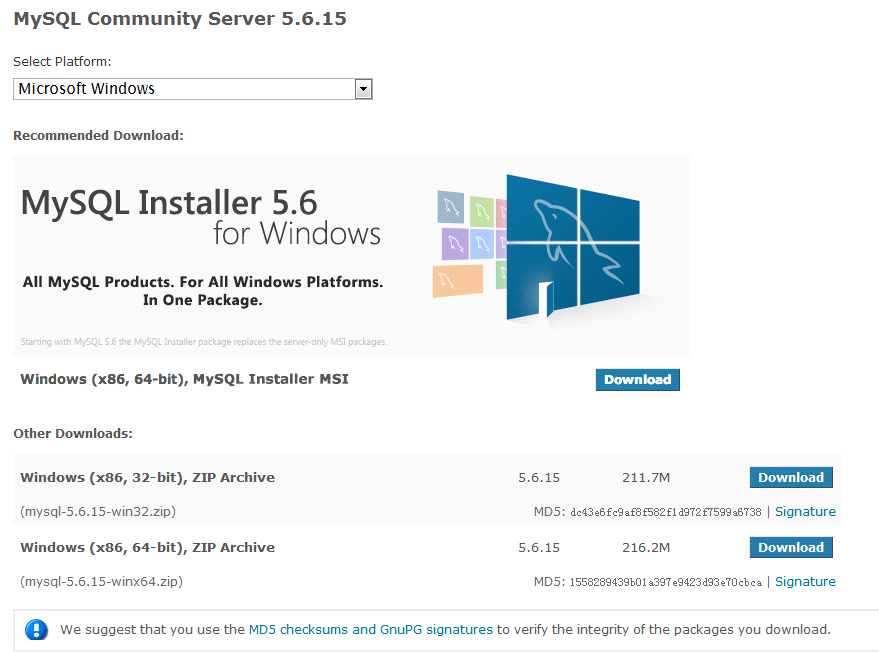


图1-15 MySQL 下载页面

2. 双击下载后的文件mysql-installer-community-5.6.14.0.msi安装Mysql。按照提示进行安装时，会进入服务器配置页面，按照图1-16所示的进行配置。

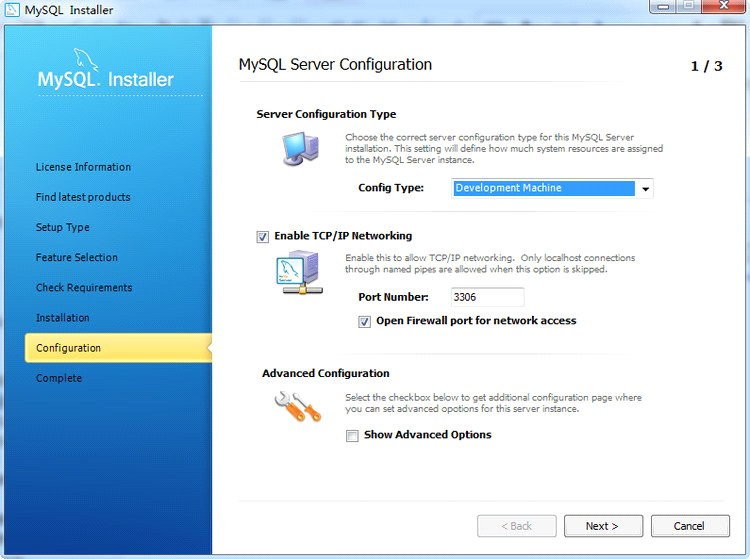


图1-16 安装Mysql过程中对Mysql服务器进行配置界面

3. 单击next按钮后进入图1-17所示界面。需要设置root用户的密码，在“MySQL Root password”(输入新密码)和“Repeat Password”（确认）两个编辑框内输入期望的密码。然后一直单击next，直至安装完成。

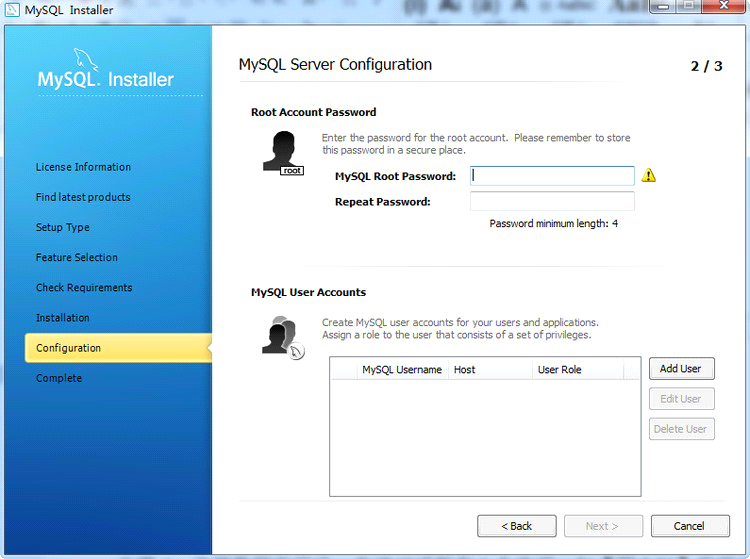


图1-17 安装Mysql过程中对Mysql服务器root用户进行配置界面

**1.4 JBoss AS 7.1.1.Final安装**

JBoss是目前Java市场上应用比较广泛、得到Sun认证的JavaEE服务器之一，它开源和免费的性质得倒了全球大批专业人员开发的青睐。本书使用的JBoss版本为7.1.1版本，它支持Java EE 6的全部功能。

* 在Red Hat官网上下载JBoss AS 7.1.1.Final，下载网址为 <http://www.jboss.org/jbossas/downloads/>，找到JBoss AS 7.1.1.Final版本下载，下载后的文件为jboss-as-7.1.1.Final.zip。
* 双击下载后的文件jboss-as-7.1.1.Final.zip，将其解压到目录“D:\jboss7.1.1”，就完成了安装。
* 设置JBOSS\_HOME系统变量，如下图所示。



图1-9 设置JBOSS\_HOME环境变量

* JBOSS默认使用JBOSS\_HOME\standalone\configuration文件夹下的standalone.xml文件作为配置文件，为了能够启动JBoss对JMS、MDB、JacORB、CMP、JAXR等功能的支持，我们需要通过名为standalone-full.xml的模板来创建该文件，具体如下：首先将其备份，然后把standalone-full.xml的名字改为standalone.xml。
* **启动和测试JBoss。**

可以采用两种方法来进行启动和测试JBoss。方法一是在cmd命令行下运行standalone.bat脚本文件来启动JBoss进行测试。方法二是在Eclipse中配置并启动JBoss来进行测试。

* **方法一：脚本启动**

运行脚本D:\jboss7.1.1\bin\standalone.bat完成启动。

访问<http://127.0.0.1:8080/>，出现 Welcome to AS 7访问界面，说明JBoss启动成功。

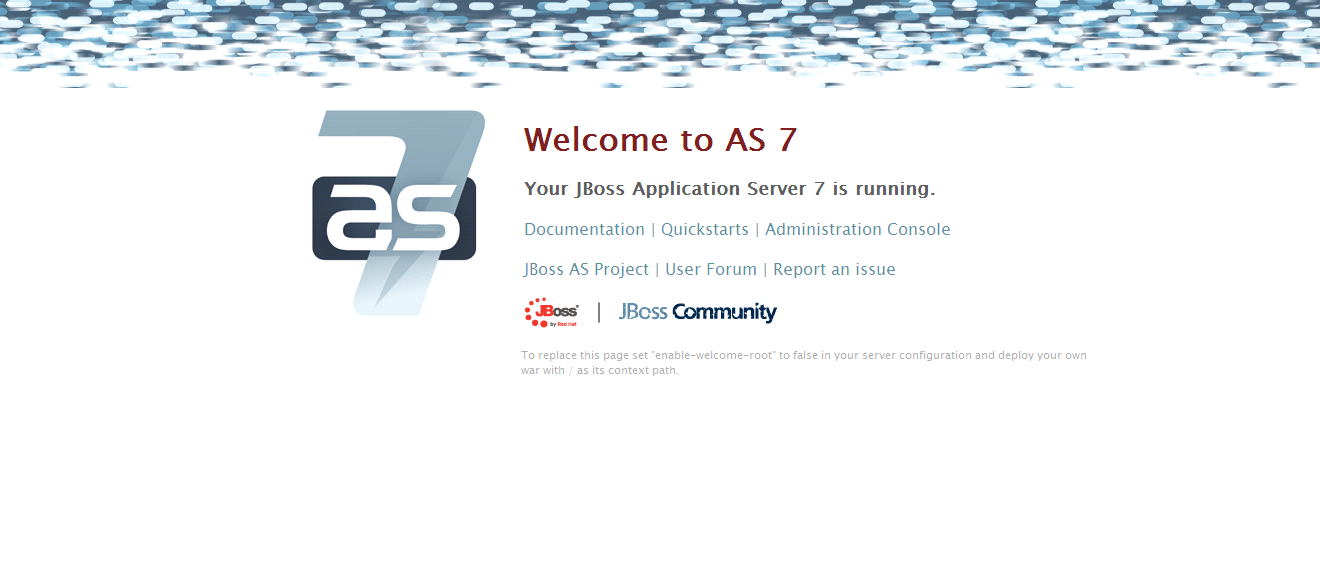


图1-10 JBoss启动界面

* **方法二：在Eclipse中配置并启动JBoss**

为了实现在Eclipse中对JBoss的控制，可以将JBoss的启动和停止添加到Eclipse中。启动Eclipse后，进入菜单Window-> Preferences-> Server -> Runtime Environments，如图1-12所示。

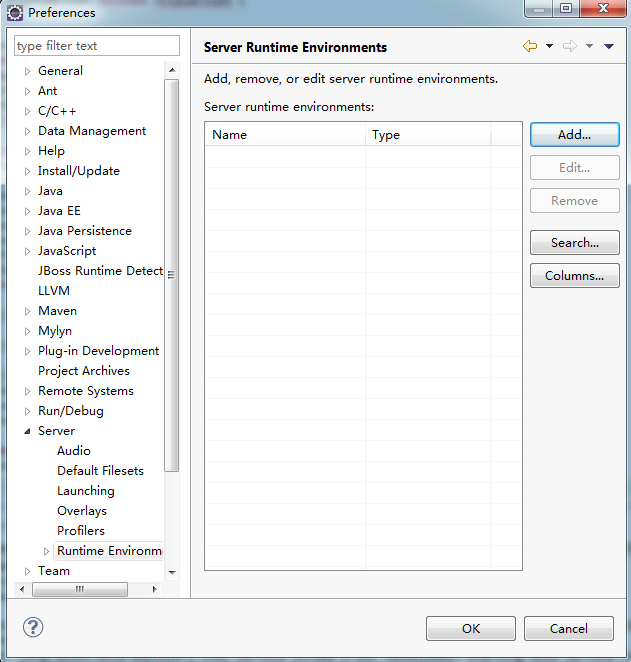


图1-12 进入JBoss配置界面

点击“Add”按钮，添加一个新的应用服务器，如图所示，选择JBoss7.1，并将jre和Application Server Directory进行正确配置，然后点击finish按钮。

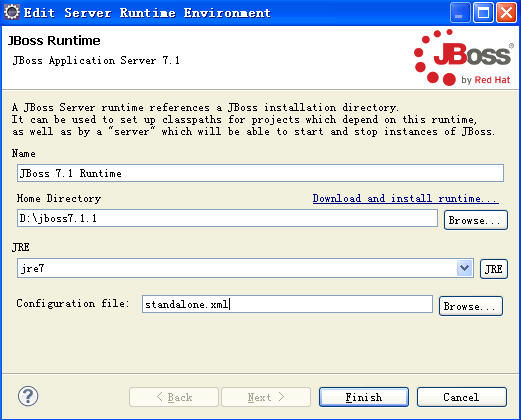


图1-13 JBoss配置界面

以上步骤完毕后，在Servers视图内会有JBoss服务器出现，如图1-14所示。选中JBoss7.1Runtime Server，在右键菜单中点击Start，JBoss服务器开始启动。若浏览器可以访问<http://127.0.0.1:8080/>，则配置成功。

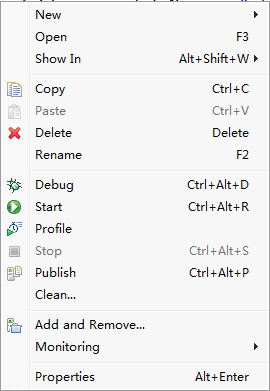
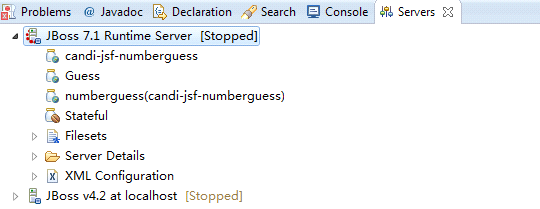


图1-14 Eclipse中添加JBoss控制

* **添加用户**

在JBoss中用户分为管理员用户（Management User）和应用用户（Application User）两类。如果需要对JBoss服务器进行配置和管理需要添加和使用管理员用户。如果是需要使用JBoss服务器提供的JNDI、JMS、EJB、web等服务，则需要添加和使用应用用户。

* **管理员用户的添加**

访问该Web服务器时，提示新增用户。可在服务端执行add-user.bat(在Jboss主目录下的bin目录中) 来添加管理员用户。

|  |
| --- |
| What type of user do you wish to add?  a) Management User (mgmt-users.properties)  b) Application User (application-users.properties)  (a): a  Enter the details of the new user to add.  Realm (ManagementRealm) : //回车，选用默认  Username : admin //填写管理员用户名admin  Password : //填写管理员密码123456(输入后不会被显示出来)  Re-enter Password :  The username 'admin' is easy to guess  Are you sure you want to add user 'admin' yes/no? yes  About to add user 'admin' for realm 'ManagementRealm'  Is this correct yes/no? yes  Added user 'admin' to file 'C:\Users\Jack\Desktop\jboss-as-7.1.1.Final\standalone\configuration\  mgmt-users.properties'  Added user 'admin' to file 'C:\Users\Jack\Desktop\jboss-as-7.1.1.Final\domain\configuration\  mgmt-users.properties'  请按任意键继续. . . |

添加管理员后，重新启动Jboss后访问<http://127.0.0.1:9990/console>，会弹出要求输入用户名和密码的页面，如图1-11所示。

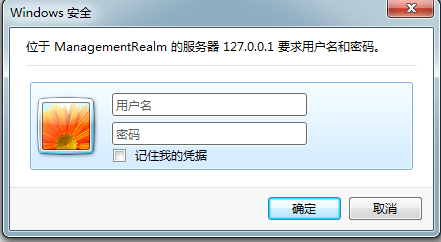


图1-11 JBoss中进入Console的用户名和密码页面

* **应用用户的添加**

为了能够通远远程访问的方式来使用EJB、JNDI、Web和JMS等服务，还需要在JBoss中添加应用用户。

|  |
| --- |
| What type of user do you wish to add?  a) Management User (mgmt-users.properties)  b) Application User (application-users.properties)  (a): b  Enter the details of the new user to add.  Realm (ApplicationRealm) : //回车，选用默认  Username : testJNDI //填写应用用户的用户名testJNDI  Password :  Re-enter Password :  What roles do you want this user to belong to? (Please enter a comma separated list, or leave blank for none) : testrole //用户的角色，在使用JMS时需要用到。  About to add user 'testJNDI' for realm 'ApplicationRealm'  Is this correct yes/no? y  Added user 'testJNDI' to file 'C:\Users\Jack\Desktop\jboss-as-7.1.1.Final\standalone\  configuration\application-users.properties'  Added user 'testJNDI' to file 'C:\Users\Jack\Desktop\jboss-as-7.1.1.Final\domain\  configuration\application-users.properties'  Added user 'testJNDI' with roles testrole to file 'C:\Users\Jack\Desktop\jboss-as-7.1.1.Final\standalone\  configuration\application-roles.properties'  Added user 'testJNDI' with roles testrole to file 'C:\Users\Jack\Desktop\jboss-as-7.1.1.Final\domain\  configuration\application-roles.properties'  请按任意键继续. . . |

* **设置外网访问**

因为JBoss安装完成后，默认只能本地访问（即：只使用<http://127.0.0.1/:8080>或[http://localhost:8080](http://localhost:8080/)进行访问，不能采用[http://192.168.10.1](http://192.168.10.1/)：8080等非本地地址进行访问），如果想让其他人也可以访问你的JBoss服务器，需要修改JBoss的配置文件，即修改standalone.xml，增加本机Web地址的内容。

(1). 打开D:\jboss7.1.1\standalone\configuration\standalone.xml

(2). 在<interfaces>与</interfaces>之间，增加一个interface节点，节点内容如下:

<interface name="any">

<any-ipv4-address/>

</interface>

接着，修改socket-binding-group节点的default-interface属性的为any

<socket-binding-group name="standard-sockets" default-interface="**any**"

port-offset="${jboss.socket.binding.port-offset:0}">

修改完后，重启JBoss7.1就可以用外网的IP来访问了。

* **在JBoss中配置**MySql**数据源**

1)为JBoss添加MySql数据驱动

首先，在JBOSS\_HOME\modules\com路径下，新建文件夹mysql，在mysql文件夹下再新建文件夹main，并将mysql-connector-java-5.1.26-bin.jar（在MYSQL\_HOME\Connector J 5.1.26目录下） 复制到main文件夹下。同时在main目录中新建一个文本文件，并改名为module.xml，文件内容如配置文件1-1所示。

配置文件 1-1 module.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <module xmlns="urn:jboss:module:1.0" name="**com.mysql**">  <resources>  <resource-root path="mysql-connector-java-5.1.26-bin.jar"/>  </resources>  <dependencies>  <module name="javax.api"/>  <module name="javax.transaction.api"/>  <module name="javax.servlet.api" optional="true"/>  </dependencies>  </module> |

接下来，修改standalone.xml配置文件，其路径为JBOSS\_HOME\standalone\configuration。打开standalone.xml文件，找到在 <datasources>...<drivers>...</drivers>...</datasources>标签。在drivers标签中添加如下的配置。

配置文件 1-2 在standalone.xml中添加的driver标签

|  |
| --- |
| <driver name="**mysql**" module="**com.mysql**">  <driver-class>  com.mysql.jdbc.Driver  </driver-class>  <xa-datasource-class>  com.mysql.jdbc.jdbc2.optional.MysqlXADataSource  </xa-datasource-class>  </driver> |

2)为JBoss添加数据MySql源

首先，修改standalone.xml配置文件，其路径为JBOSS\_HOME\standalone\configuration。打开standalone.xml文件，找到 <datasources>标签，里面应该已经有一个默认的datasource：ExampleDS，现在加入我们的mysql的datasource：

配置文件 1-3 在standalone.xml中添加的datasource标签

|  |
| --- |
| <datasource jndi-name="*java:jboss/datasources/MySqlDS* "  pool-name="*MySqlDS*" enabled="true" use-java-context="true">  <connection-url>  jdbc:mysql://localhost:3306/*database-name*  </connection-url>  <driver>**mysql**</driver>  <security>  <user-name>*root*</user-name>  <password>*123456*</password>  </security>  </datasource>  <!-- *database-name*是数据库的名字，该数据库必须是已存在的，否则JBoss启动时报错--> |

还可以添加一个支持分布式数据库的xa-datasource

配置文件 1-4 在standalone.xml中添加的xa-datasource标签

|  |
| --- |
| <xa-datasource jndi-name="*java:jboss/datasources/ MySqlXADS* "  pool-name=" *MySqlXADS* " enabled="true" use-java-context="true">  <xa-datasource-property name="URL">  jdbc:mysql://localhost:3306/xa-database-name  </xa-datasource-property>  <driver>**mysql**</driver>  <security>  <user-name>*root*</user-name>  <password>*123456*</password>  </security>  </xa-datasource>  <!-- xa-database-name是数据库的名字，该数据库必须是已存在的，否则JBoss启动时报错--> |

**注意黑色字体部分要与module.xml和standalone.xml中datasource里面的黑色字体的名字一致**。

<https://dev.mysql.com/doc/connector-j/8.0/en/connector-j-api-changes.html>

* **在JBoss中配置使用JMS的权限**

由于JBoss对安全性的限制，在使用JMS时，为了使用户能够发送和接收消息，必须为用户添加相应的权限。在JBoss的消息系统中，权限是以角色(而非单个用户)的形式进行赋予的。因此需要将使用JMS的用户的角色添加到相应的权限允许配置中，如下所示。默认情况下仅对guest角色赋予了接收和发送消息的权限，现在我们将testrole角色也赋予了接收和发送消息的权限。用户的角色在创建应用型用户(application user)时，在创建过程中进行指定。

|  |
| --- |
| ……  <subsystem xmlns="urn:jboss:domain:**messaging**:1.1">  ……  <security-settings>  <security-setting match="#">  <permission type="send" roles="guest testrole"/>  <permission type="consume" roles="guest testrole"/>  </security-setting>  </security-settings>  ……  </subsystem >  …… |