

一、在创建JPA之前看看什么是JPA

- 1、Java Persistence API：用于对象持久化的 API
- 2、Java EE 5.0 平台标准的 ORM 规范，使得应用程序以统一的方式访问持久层
- 3、结合下面两幅图片进行理解：

JPA功能示意图：

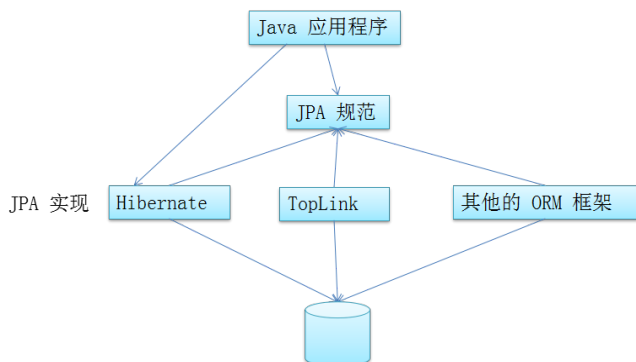


Figure 1.1

JDBC功能示意图：

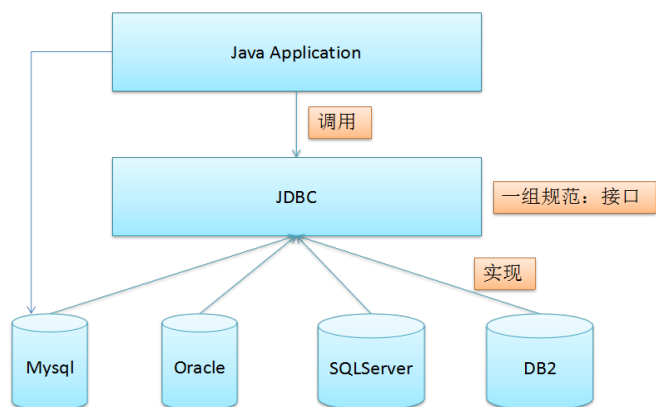


Figure 1.2

4、JPA和Hibernate的关系

JPA 是 hibernate 的一个抽象（就像JDBC和JDBC驱动的关系）：

——**JPA 是规范**：JPA 本质上就是一种 ORM 规范，不是ORM 框架 —— 因为 JPA 并未提供 ORM 实现，它只是制订了一些规范，提供了一些编程的 API 接口，但具体实现则由 ORM 厂商提供实现

——**Hibernate 是实现**：Hibernate 除了作为 ORM 框架之外，它也是一种 JPA 实现

从功能上来说，**JPA 是 Hibernate 功能的一个子集**

二、创建JPA工程之前需要解决一个问题：at least one user library must be selected

问题描述：在Workspace中右键，new → JPA Project，选择JAP version为2.0。此时，向导中的Finish是灰色的，不能够点击完成。没办法，只好点击Next看看，一路Next下去看到如下图Figure 2.1所示这样一个问题：at least one user library must be selected。

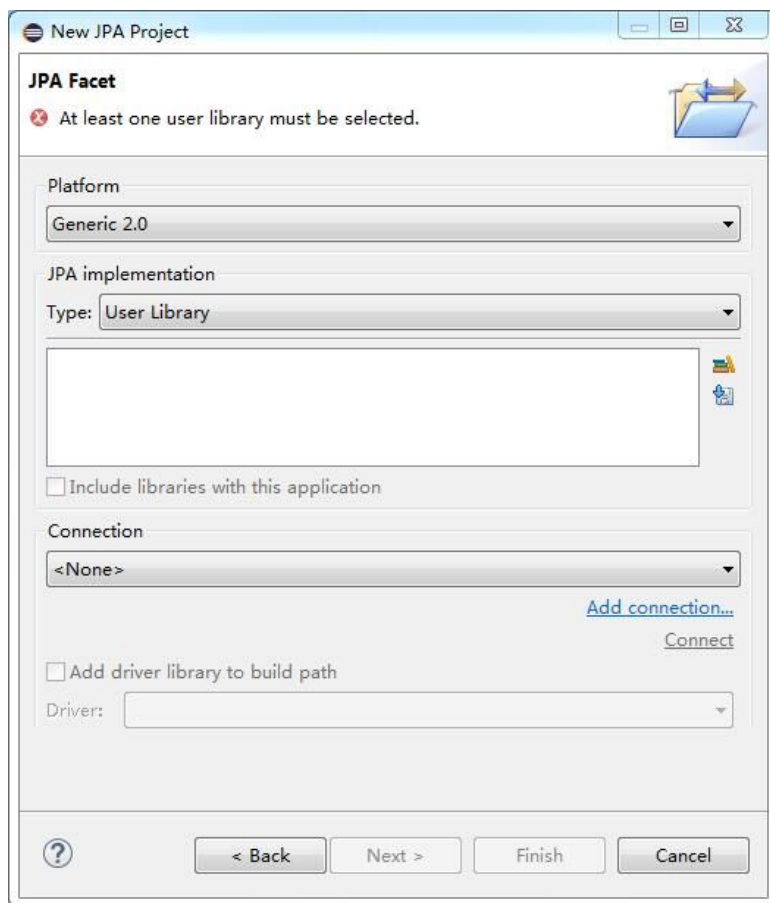


Figure 2.1

没有办法创建JAP工程，只好网上搜索，找到一篇歪果仁写的一篇博文，很好的解决了我这个问题：

<http://www.javabeat.net/eclipselink-jpa-installation-configuration/>

下面将博文要点整理成如下中文：

这篇文章将向你介绍如何安装和使用EclipseLink-JPA：

1、安装EclipseLink JAR 文件

- 从 <https://www.eclipse.org/eclipselink/downloads/> 处下载 **EclipseLink 2.5.2 Installer Zip (38 MB)** 资源（也可以选择其它版本）。EclipseLink中提供了所有持久化服务的的实现。
- 持久化服务将会用到下面三个jar包：
 - ①、eclipselink.jar
 - ②、javax.persistence.source_2.1.0.v201304241213.jar
 - ③、javax.persistence_2.1.0.v201304241213.jar
- 将下载好的EclipseLink 2.5.2压缩包解压到任何你想用于完成安装的文件夹中。我将文件解压到“.\workspace\libraries\EclipseLink 2.5.2”文件夹下。

2、在Eclipse IDE中设置 JPA Facet

首先在Eclipse中创建一个动态web工程，为我们后续添加 JPA facet 作准备。步骤如下：

- ①、打开 Eclipse IDE
- ②、创建一个动态的Web工程
- ③、在刚刚创建好的工程上点击右键，选择properties
- ④、在左侧那一列中点击“Project Facets”
- ⑤、然后在右侧选中 JPA 选项，同时选择 JPA version 为2.0。如下图Figure 2.2所示：

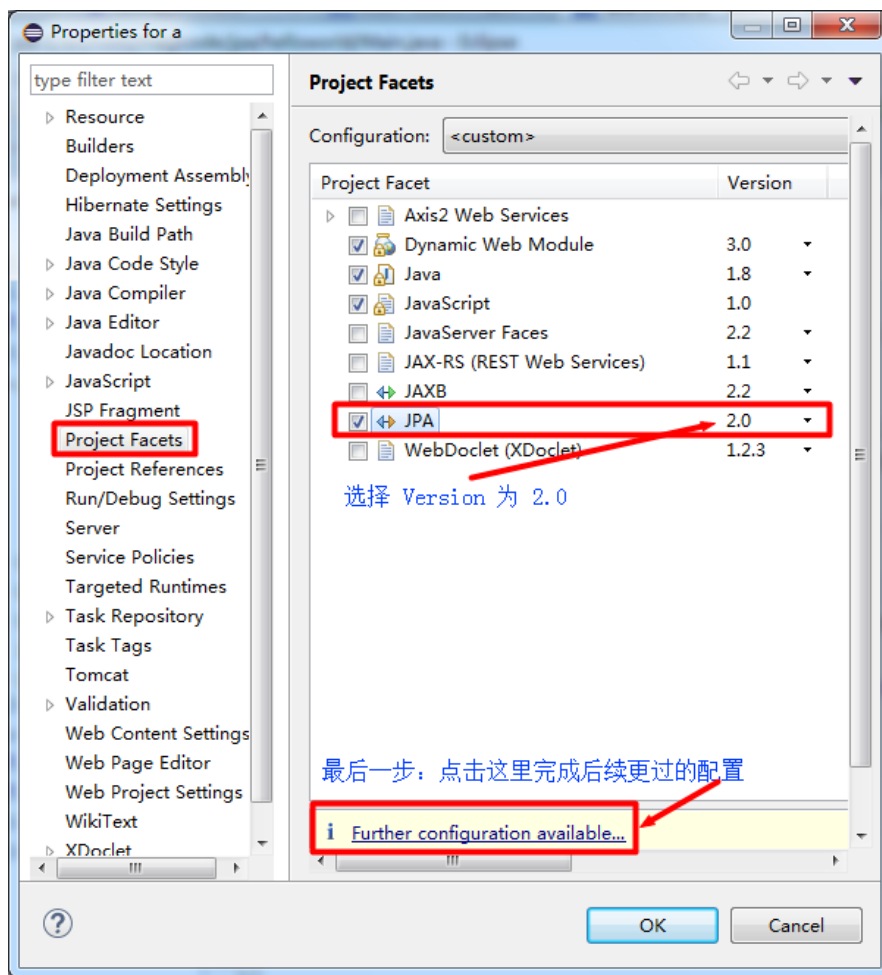


Figure 2.2

这里有一个要注意的，因为有后续的很多设置要完成，所以在选择JPA和Version以后不能直接点击OK，而是要点击上图中下面部分标注的“Further configuration ...”链接来进行后续配置。这个操作将会打开一个新的对话框用于配置JPA Facet。

3、添加 JPA Facet

新弹出的对话框如下图Figure 2.3所示。用户可以在这个对话框中添加必须的库来实现JPA。

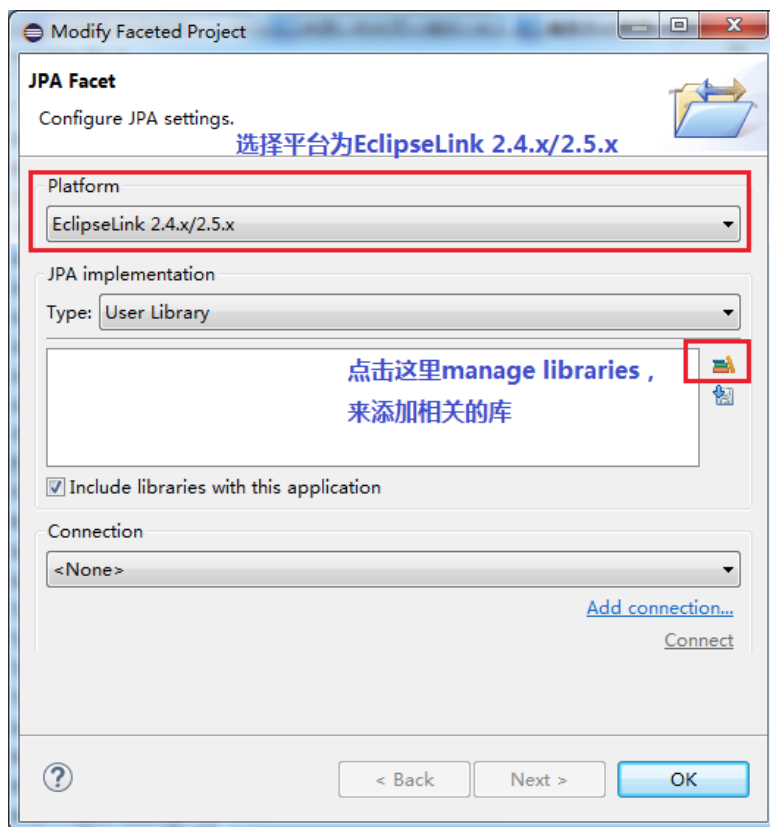


Figure 2.3

①、对话框上端的platform默认情况下是“Generic 2.0”，我们可以依据我们自己下载的EclipseLink版本来进行修改。我下载的是2.5.2版本，所以如Figure 2.3所示选择platform为EclipseLink 2.4.x/2.5.x版本

②、JPA implementation菜单不需要做任何修改。“User Library” type将会允许用户添加JPA库

③、点击右侧的“Manager Libraries”。除此之外，另外一个实现方法就是点击“Download Library”来进行下载库，但是这里下载实在是太慢，所以不使用它。

④、点击“Manager Libraries”以后会弹出一个对话框，在新的对话框中点击“New”。

⑤、“New”以后又会弹出一个对话框，要我们填写库的名字，我填写为“EclipseLink 2.5.2”。点击OK。

到目前为止，我们已经做好了添加jar包的前期准备。接下来就是添加上面“1、安装EclipseLink JAR 文件”这一步中提到的那三个jar包了。到目前的效果图如下Figure 2.4:

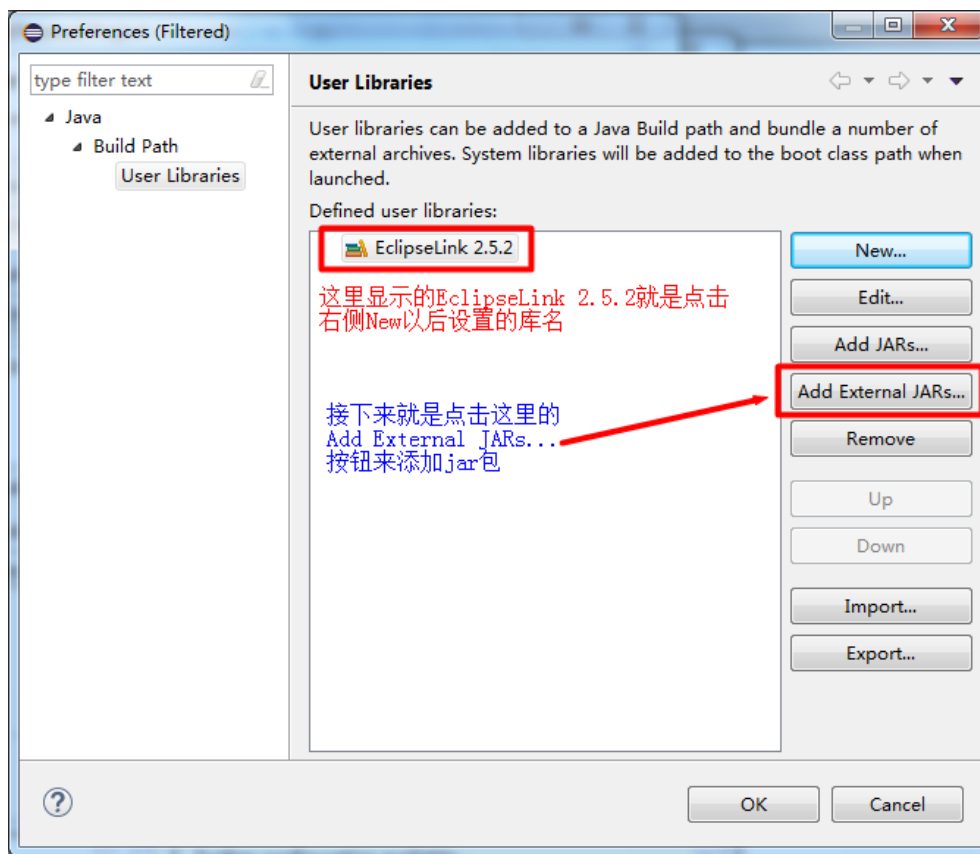


Figure 2.4

4、配置Library

配置Library需要我们添加JPA实现所必须的jar包。按照下面的步骤来完成配置：

- ①、点击“Add External JARs...”（也就是Figure 2.4图中红色框中的按钮）
- ②、添加 eclipselink.jar，它位于“.\workspace\libraries\EclipseLink 2.5.2\lib”文件夹下面
- ③、添加 “.\workspace\libraries\EclipseLink 2.5.2\lib\jpa”文件夹下 javax.persistence.* 开头的jar包（也就是前面说的后两个jar包）。如Figure 2.5所示

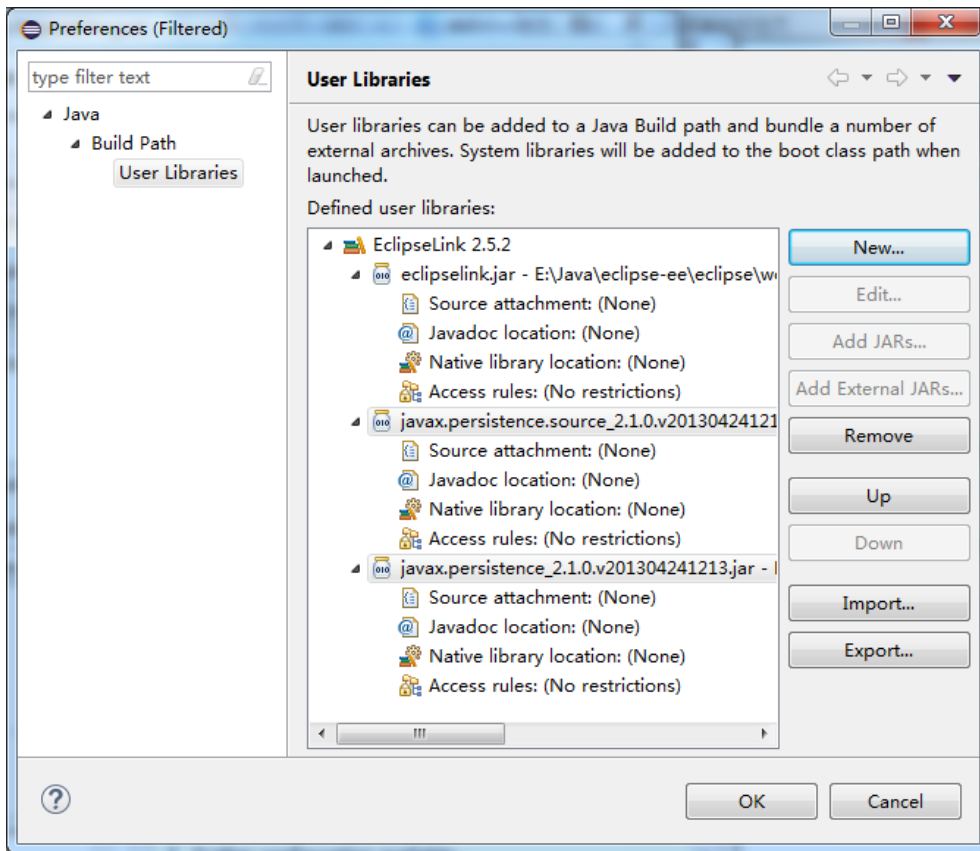


Figure 2.5

④、点击下端的OK按钮，会关闭添加library的对话框。

⑤、但是，为了避免JPA Facet报“`At least one user library must be selected`”这个错误。我们选择刚刚生成的library

⑥、点击OK。

到现在为止，前面提到的`At least one user library must be selected`错误已经得到了解决。到这里我们再回头去创建一个JPA工程就没有什么问题了。但是，歪果仁的这篇文章的目的是为了在动态Web工程中创建一个JPA环境。

这个是很有必要的，想想也是，现在单纯为了学习建了一个JPA工程，以后做Web工程的时候就需要创建一个动态Web工程了。

而且，后面的两个步骤还是挺有意思的，所以，这里是插叙，下面继续翻译歪果仁的文章。

到现在为止我们创建的动态web工程已经能够使用JPA了。但是，JPA是一个和关系型数据库相联系的持久化服务。所以，接下来我们就需要创建一个数据库连接

5、创建数据库链接（新意在于可以在配置的同时完成链接的测试工作）

一旦你创建完成了一个user library，接下来的工作就是创建一个数据库连接。最后一个对话框应该如Figure 2.6所示：

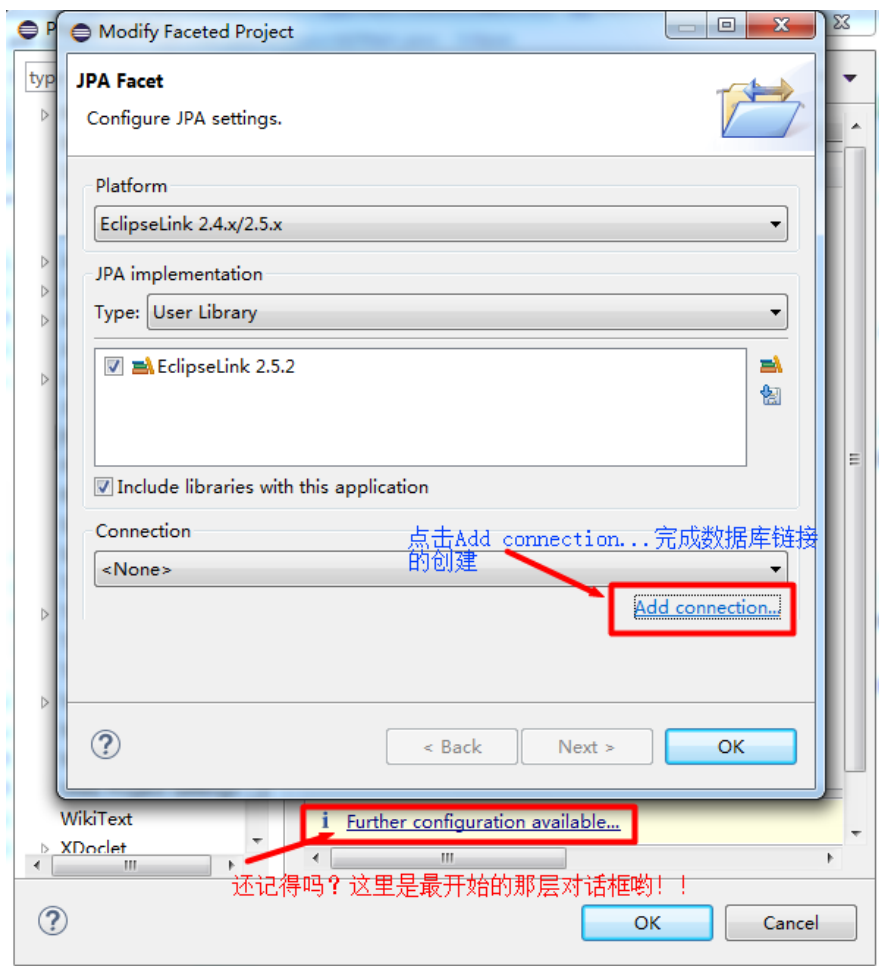


Figure 2.6

为了建立数据库的链接，按照以下步骤进行：

- ①、点击“Add Connection...”超链接（Figure 2.6中有标注）
- ②、Connection Profile 对话框将会弹出
- ③、选择工程中在对话框中选择工程使用的数据库种类。本教程中假设你使用的是MySQL数据库。
- ④、修改链接的名字，点击Next

⑤、填写“Specify a Driver and Connection Details”表单。在最初的时候可能弹出的对话框中没有驱动可以选择，这个时候我们需要手动的添加相关驱动，步骤如Figure 2.7和Figure2.8所示：

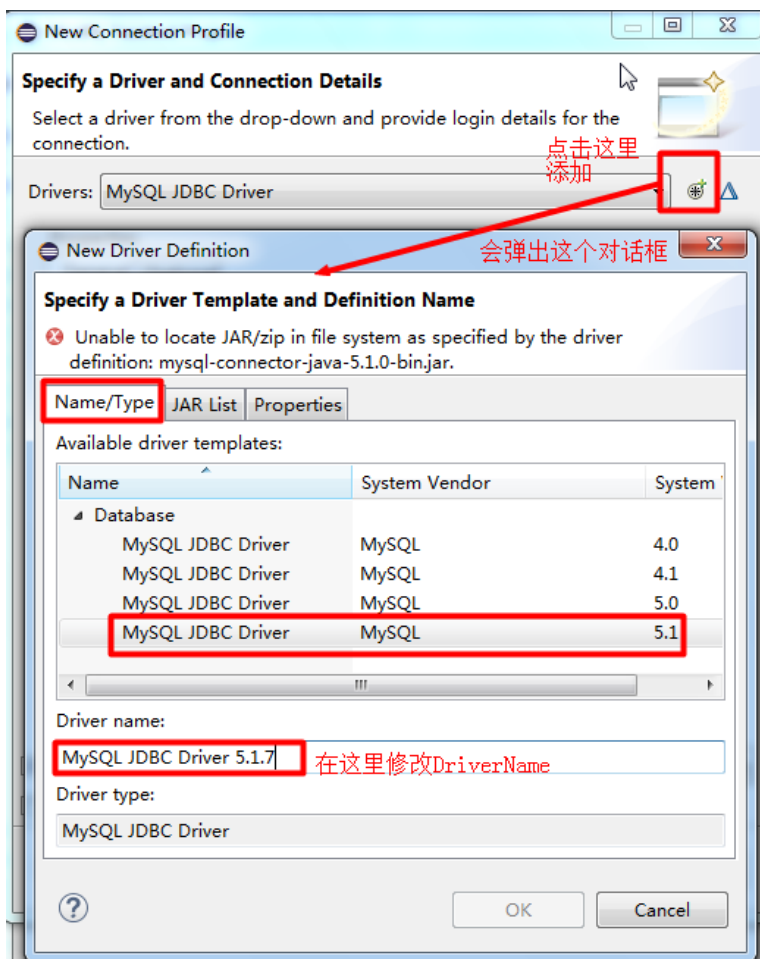


Figure 2.7

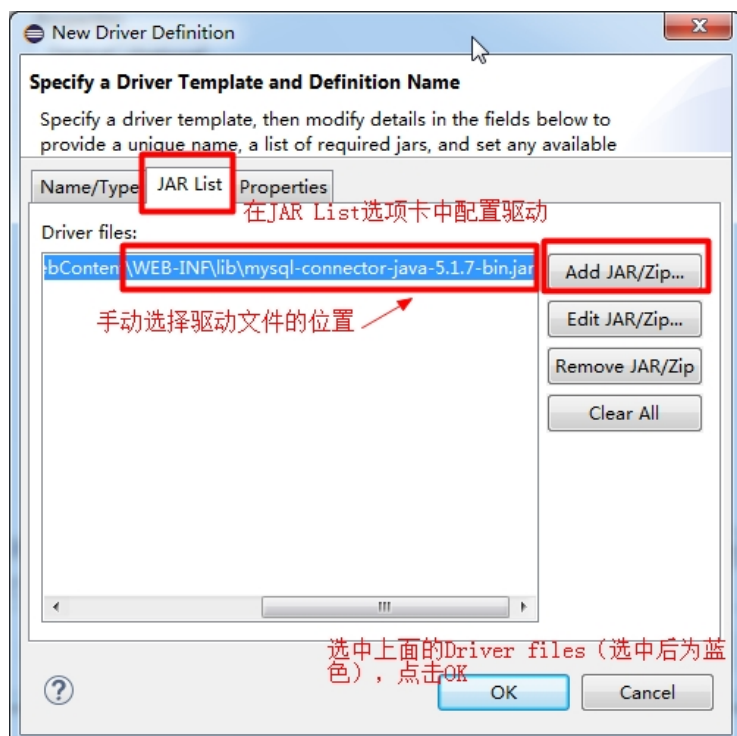


Figure2.8

点击上面的Figure 2.8中的OK以后，新的driver就添加完成了。随后就是在Figure 2.9中修改数据库名，填写用户名和密码，然后点击“Test Connection”测试数据库连接是否成功：

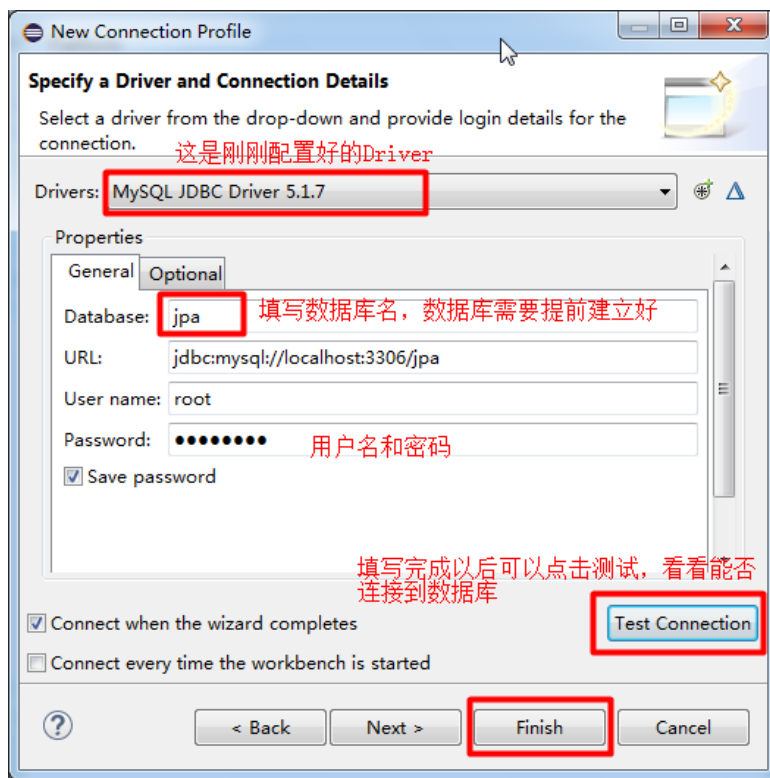


Figure 2.9

有意思的是，测试通过会弹出一个 ping succeeded! 提示框，如Figure 2.10所示：

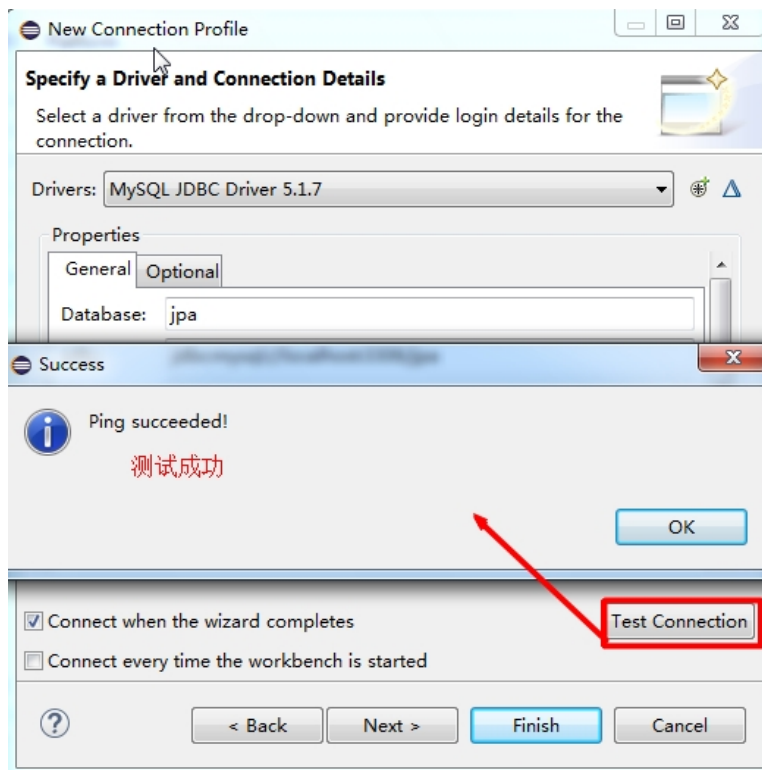


Figure 2.10

- ⑥、在第⑤步的末尾，我们已经为JPA工程准备好了数据库连接
- ⑦、点击Figure 2.9中的finish按钮
- ⑧、点击finish以后对话框将会关闭，所以要确定连接已经被选中
- ⑨、点击，OK关闭Further configuration dialog。继续点击OK关闭project facets

现在，你已经创建完成了一个新的动态web工程，同时为它配置好了JPA环境。在eclipse左侧的导航栏里面你可以在src下看见一个META-INF文件夹了，该文件夹下面有一个persistence.xml文件。

6、配置persistence.xml文件

如果你按照步骤进行到这里了，而且所有步骤都成功操作了。现在是时候“编辑”persistence.xml文件的时候了，这个文件将会帮助我们连接我们的数据库。

为了配置persistence.xml文件，你需要双击打开它。它会在单独的一个xml文件编辑器中打开，在这个编辑器的底端会有很多的标签。现在我们按照以下的步骤完成persistence.xml文件的配置工作：

- ①、导航到Connection标签
- ②、将transaction type选择为“Resource Local”。这个改变将会激活EclipseLink connection pool。
- ③、点击“populate from connection”。见Figure 2.11
- ④、选择之前配置好的链接。见Figure 2.12
- ⑥、用 Ctrl + s 来保存文件。见Figure 2.13
- ⑦、导航到Source标签。你将会看到用于数据库连接的最小配置。

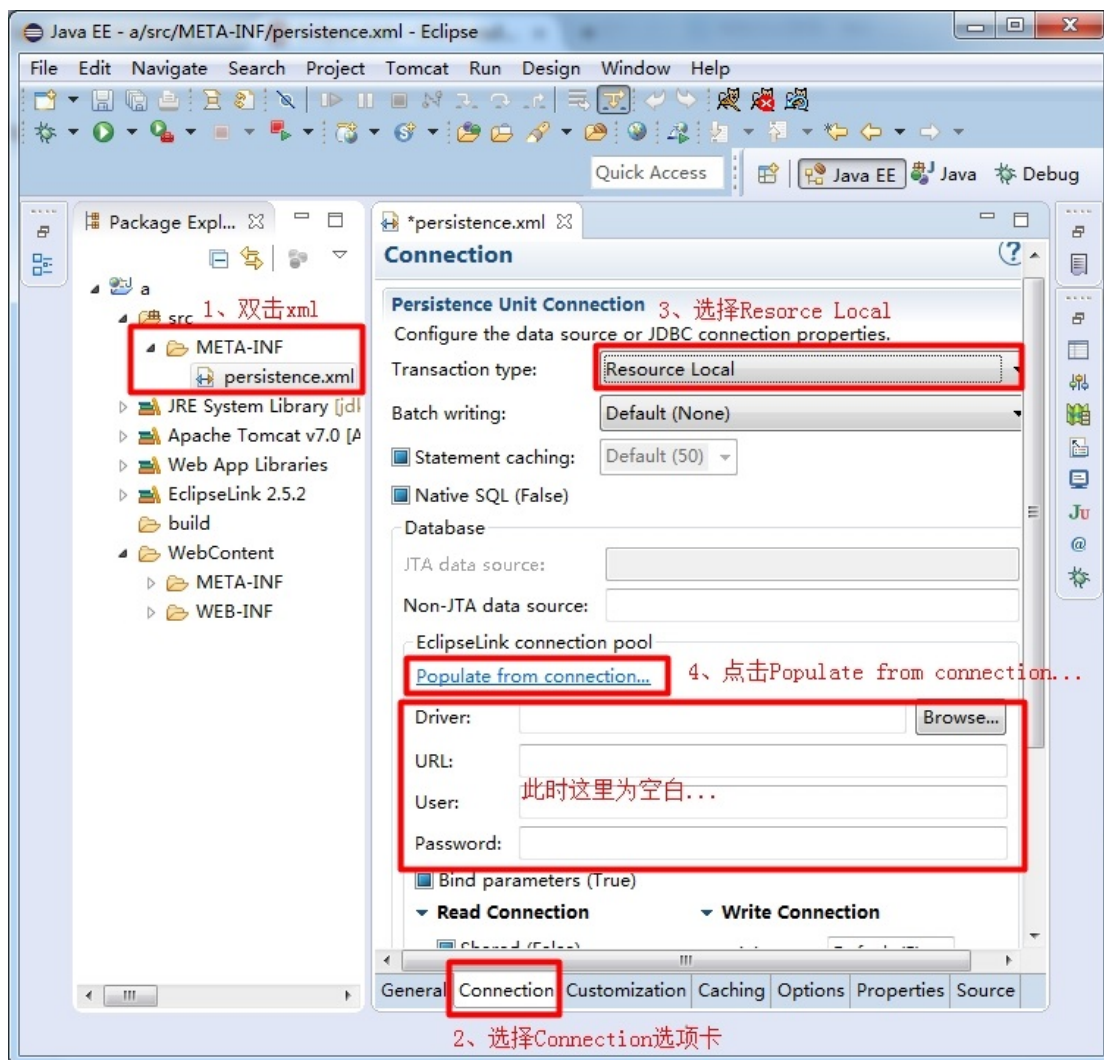


Figure 2.11

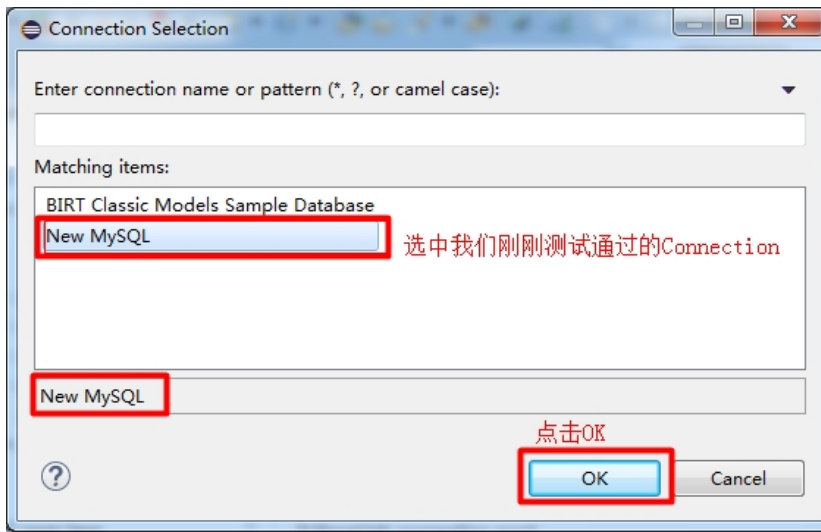


Figure 2.12

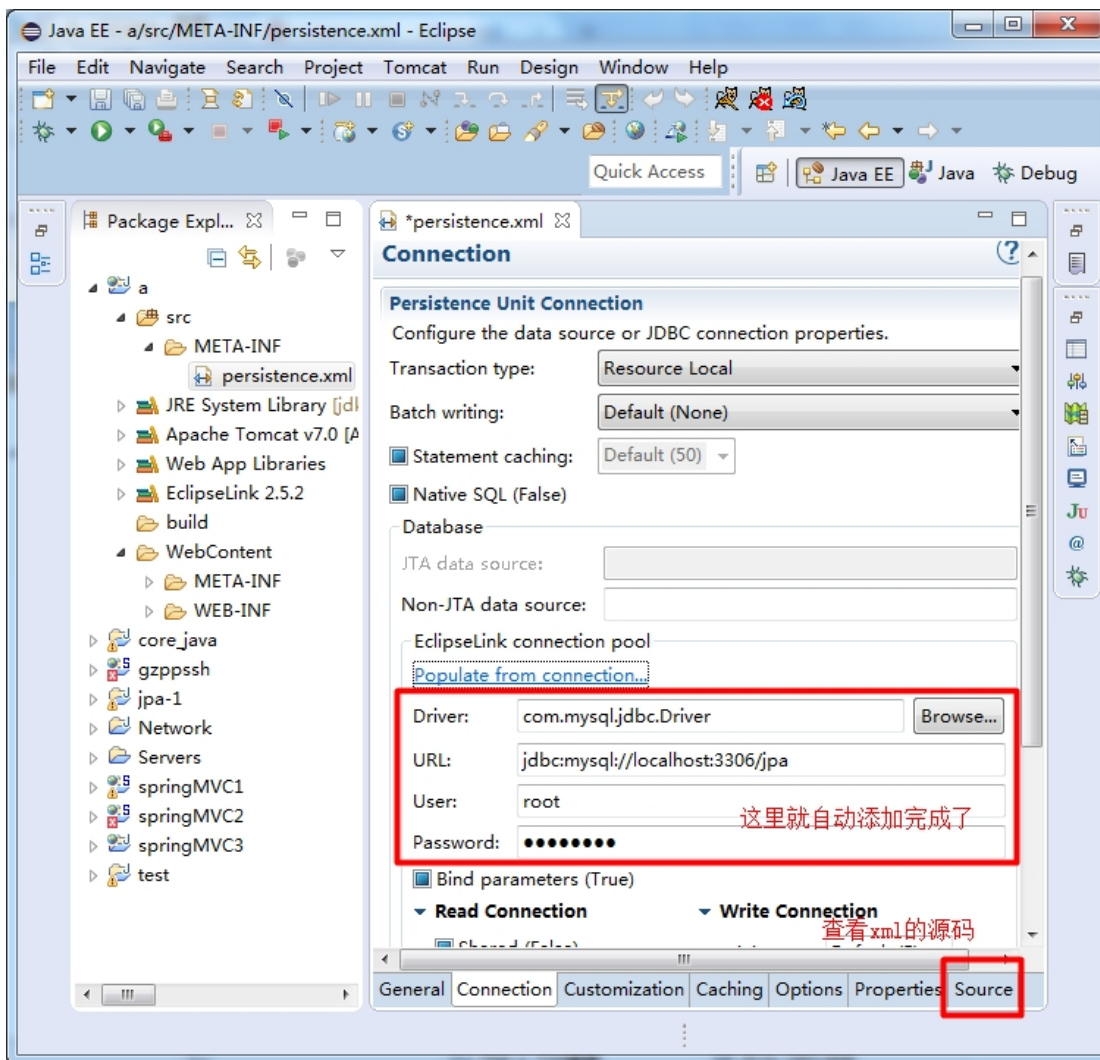


Figure 2.13

看看xml中的源码如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0"
  xmlns="http://java.sun.com/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">
  <persistence-unit name="a" transaction-type="RESOURCE_LOCAL">
    <properties>
```

```
        <property name="javax.persistence.jdbc.url"
value="jdbc:mysql://localhost:3306/jpa"/>
        <property name="javax.persistence.jdbc.user" value="root"/>
        <property name="javax.persistence.jdbc.password" value="tiger123"/>
        <property name="javax.persistence.jdbc.driver"
value="com.mysql.jdbc.Driver"/>
    </properties>
</persistence-unit>
</persistence>
```

