

JDBC基础教程之驱动设置 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/253/2021\\_2022\\_JDBC\\_E5\\_9F\\_BA\\_E7\\_A1\\_80\\_c97\\_253055.htm](https://www.100test.com/kao_ti2020/253/2021_2022_JDBC_E5_9F_BA_E7_A1_80_c97_253055.htm) 1、概述 DriverManager 类是

JDBC 的管理层，作用于用户和驱动程序之间。它跟踪可用的驱动程序，并在数据库和相应驱动程序之间建立连接。另外，DriverManager 类也处理诸如驱动程序登录时间限制及登录和跟踪消息的显示等事务。对于简单的应用程序，一般程序员需要在此类中直接使用的唯一方法是

DriverManager.getConnection。正如名称所示，该方法将建立与数据库的连接。JDBC 允许用户调用 DriverManager 的方法 getDriver、getDrivers 和 registerDriver 及 Driver 的方法 connect。但多数情况下，让 DriverManager 类管理建立连接的细节为上策。

1、跟踪可用驱动程序 DriverManager 类包含一系列 Driver 类，它们已通过调用方法 DriverManager.registerDriver 对自己进行了注册。所有 Driver 类都必须包含有一个静态部分。它创建该类的实例，然后在加载该实例时 DriverManager 类进行注册。这样，用户正常情况下将不会直接调用

DriverManager.registerDriver；而是在加载驱动程序时由驱动程序自动调用。加载 Driver 类，然后自动在 DriverManager 中注册的方式有两种：通过调用方法 Class.forName。这将显式地加载驱动程序类。由于这与外部设置无关，因此推荐使用这种加载驱动程序的方法。以下代码加载类 acme.db.Driver：

```
Class.forName("acme.db.Driver").
```

如果将 acme.db.Driver 编写为加载时创建实例，并调用以该实例为参数的

DriverManager.registerDriver（本该如此），则它在

DriverManager 的驱动程序列表中，并可用于创建连接。通过将驱动程序添加到 java.lang.System 的属性 jdbc.drivers 中。这是一个由 DriverManager 类加载的驱动程序类名的列表，由冒号分隔：初始化 DriverManager 类时，它搜索系统属性 jdbc.drivers，如果用户已输入了一个或多个驱动程序，则 DriverManager 类将试图加载它们。以下代码说明程序员如何在 ~/.hotjava/properties 中输入三个驱动程序类（启动时，HotJava 将把它加载到系统属性列表中）：

```
jdbc.drivers=foo.bah.Driver:wombat.sql.Driver:bad.test.ourDriver.
```

对 DriverManager 方法的第一次调用将自动加载这些驱动程序类。注意：加载驱动程序的第二种方法需要持久的预设环境。如果对这一点不能保证，则调用方法 Class.forName 显式地加载每个驱动程序就显得更为安全。这也是引入特定驱动程序的方法，因为一旦 DriverManager 类被初始化，它将不再检查 jdbc.drivers 属性列表。在以上两种情况中，新加载的 Driver 类都要通过调用 DriverManager.registerDriver 类进行自我注册。如上所述，加载类时将自动执行这一过程。由于安全方面的原因，JDBC 管理层将跟踪哪个类加载器提供哪个驱动程序。这样，当 DriverManager 类打开连接时，它仅使用本地文件系统或与发出连接请求的代码相同的类加载器提供的驱动程序。

## 2、建立连接 加载 Driver 类并在 DriverManager 类中注册后，它们即可用来与数据库建立连接。当调用 DriverManager.getConnection 方法发出连接请求时

，DriverManager 将检查每个驱动程序，查看它是否可以建立连接。有时可能有多个 JDBC 驱动程序可以与给定的 URL 连接。例如，与给定远程数据库连接时，可以使用

JDBC-ODBC 桥驱动程序、JDBC 到通用网络协议驱动程序或数据库厂商提供的驱动程序。在这种情况下，测试驱动程序的顺序至关重要，因为 DriverManager 将使用它所找到的第一个可以成功连接到给定 URL 的驱动程序。首先 DriverManager 试图按注册的顺序使用每个驱动程序（jdbc.drivers 中列出的驱动程序总是先注册）。它将跳过代码不可信任的驱动程序，除非加载它们的源与试图打开连接的代码的源相同。它通过轮流在每个驱动程序上调用方法 Driver.connect，并向它们传递用户开始传递给方法 DriverManager.getConnection 的 URL 来对驱动程序进行测试，然后连接第一个认出该 URL 的驱动程序。这种方法初看起来效率不高，但由于不可能同时加载数十个驱动程序，因此每次连接实际只需几个过程调用和字符串比较。以下代码是通常情况下用驱动程序（例如 JDBC-ODBC 桥驱动程序）建立连接所需所有步骤的示例：

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"). //加载驱动程序
String url = "jdbc:odbc:fred". DriverManager.getConnection(url,
"userID", "passwd").
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)