

WebSphere应用服务器保护电子商务应用 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/242/2021\\_2022\\_WebSphere\\_E5\\_c40\\_242491.htm](https://www.100test.com/kao_ti2020/242/2021_2022_WebSphere_E5_c40_242491.htm) 企业应用一般都需要连接数据库，自从有了应用服务器之后，应用连接数据库就可以利用数据库连接池的方式来提高性能。基于J2EE的WEB应用一般是通过JDBC的方式来访问数据库，而且大部分都是通过数据源的方式来得到数据库连接的。WebSphere应用服务器（WAS）自从3.5版本便开始支持JDBC2.0，并支持用JNDI的方式来查找数据源然后生成数据库连接。我们在工作的过程中发现有相当多的用户在配置和使用数据源时曾遇到各种各样的问题。本文将介绍WAS 5.x版本中数据源的配置和使用方法，并结合实际客户遇到的常见问题进行一些讨论，以使更多用户能在WAS上顺利的配置和使用数据源。

### 1. 简单原理和背景介绍

#### 1.1 什么是数据源？为什么要使用数据源？

我们先来看一下使用JDBC1.0的时候，一般是怎样来连接数据库的。代码片断

```
示例1： java.sql.Connection con = null. try {
Class.forName("COM.ibm.db2.jdbc.app.DB2Driver"). con =
java.sql.DriverManager.getConnection("jdbc:db2:DBName"). }
catch (Exception e) { System.err.println("Exception: "
e.getMessage()). }
```

由于建立一个数据库连接是一种比较耗时的操作，在使用JDBC1.0的时候，如果要提高性能就需要应用程序自己对连接进行缓存处理。但是在JDBC2.0 以一个附加的API的方式引入了标准连接池的特性，这个附加的API叫做JDBC 2.0 可选包（也称作JDBC 2.0 标准扩展）。这样对于支持JDBC2.0的数据库（一般是指JDBC驱动程序支持）

，WebSphere应用服务器提供了连接池的支持。JDBC2.0 API为连接池提供了一个客户端和一个服务器端的接口。客户端接口是 `javax.sql.DataSource`，通常就是应用代码用来请求一个缓冲了的数据库连接的东西。这个数据源就是生成数据库连接对象的工厂。服务器接口是 `javax.sql.ConnectionPoolDataSource`，通常是大多数应用服务器和数据库JDBC驱动打交道的接口。连接池可改进任何需要连接的应用程序（特别是基于Web的应用程序，Web用户的连接和断开更为频繁）的响应时间。因为连接池的实现，在客户端调用 `close()` 方法的时候实际上并不关闭连接，而是把连接返回到一个可重用连接的连接池中给其它客户端使用。这样就避免了任何重复打开和关闭数据库连接造成的开销，并且允许大量的客户端分享相对较少的数据库连接，从而提高数据库操作的性能。为了使用应用服务器提供的连接池功能，我们需要先定义相应的数据源。数据源在WebSphere应用服务器中作为资源存在，因此可以统一通过JNDI来查询它们。

使用数据源的代码片断示例2：

```
try{
    javax.naming.InitialContext ctx = new
    javax.naming.InitialContext(). javax.sql.DataSource ds =
    (javax.sql.DataSource)ctx.lookup("jdbc/DSName").
    java.sql.Connection con = ds.getConnection().
}catch(javax.naming.NamingException e1){
    System.err.println("Naming-Exception: " e1.getMessage()). }catch
    (java.sql.SQLException e2) {
    System.err.println("Naming-Exception: " e2.getMessage()). }
```

使用数据源的另一个好处是安全性的提高。在不使用数据源的情

况下，我们一般通过下面的代码来连接Oracle数据库。代码

片断3：//装载JDBC驱动程序

```
java.sql.DriverManager.registerDriver(new
```

```
oracle.jdbc.driver.OracleDriver()). //设置数据库连接字符串
```

```
java.lang.String strConn =
```

```
"jdbc:oracle:thin:@hostname/IP:1521:SIDName". //连接数据库
```

```
java.sql.Connection sqlConn =
```

```
java.sql.DriverManager.getConnection(strConn,"userName","pass
```

```
Word"). 这种方式需要把数据库的用户名和口令放在程序的源
```

```
代码中。而在WebSphere应用服务器中我们可以在定义数据源的
```

```
时候通过J2C认证的方式配置数据库访问别名，从而提升安
```

```
全性。另外，这种方法也更灵活，如果数据库用户名或口令
```

```
发生变化不需要更改应用程序代码。 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
```