

在OracleJDBC访问中加入Spring特性（1）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/143/2021_2022_E5_9C_A8

OracleJ_c102_143578.htm 字符编码转换:\$utf8 =

iconv(gb2312,utf-8,\$str). 获得字符的编码形

式:mb_detect_encoding. JDBC 是一个常用于访问关系数据库的标准数据访问协议。JDBC 的一个显著优势是其标准化的 API

，为基于 Java 的数据访问应用程序的可移植性奠定了基础

。 JDBC 是标准 Java (J2SE) 和企业 Java (J2EE) 中一个不可或缺的部分，在 Java 早期阶段就已推出。 JDBC 有许多优势，使得它能够在许多 J2SE 和 J2EE 应用程序中发挥重要作用。但它

也有一些不足之处，使得我们不能称心如意的使用它们。这些麻烦（有时候让人厌恶）的 JDBC 特性催生出了许多公开的 JDBC 抽象框架（例如 SQLExecutor 和 Apache Jakarta Commons DBUtil）以及更多得多的自主开发的 JDBC 应用程序框架。 Spring 框架的 JDBC 抽象就是一个公开的 JDBC 抽象框架。 Spring 框架是一个在 Apache 许可下发布的 Java/J2EE 应用程序框架，它支持 J2EE 应用程序中的多个层次。 Spring 框

架的一个突出特性是支持更易于维护和更强健的 JDBC 数据访问。在本文中，您将了解到 Spring 框架 它可以和 Oracle TopLink 对象/关系映射工具结合使用 如何大大减少与编写 JDBC 代码相关的繁琐工作和风险。 使用 Spring 框架，开发人

员编写的 Oracle 数据库访问 JDBC 代码可以更为简洁、更不易出错以及更加灵活。 正确关闭数据库资源 JDBC 代码中的一个常见错误是没有正确关闭连接。 这将导致数据库资源的不合理分配。 类似地，关闭结果集和语句也是有用并通常推

荐的操作。为了确保即使在异常的运行条件下也能正确执行这些关闭操作，一般将采用代码清单 1 中 finally 子句中的代码。代码清单 17 演示了本文中第一次在基于 Spring 的代码中使用 PreparedStatement，并显示了对 SQLException 的另一种引用。正如代码清单 16 的情况一样，SQLException 主要用于引用 Spring 框架的 JdbcTemplate 类，后者将处理它并将任何异常作为非强制 Spring 异常提供。代码清单 16 和 17 演示了 Spring 的 RowCallbackHandler 和 PreparedStatementSetter 回调接口的用法。在这些代码清单中使用匿名内部类实现了这些接口。虽然与前面的代码清单中显示的 JdbcTemplate 的更简单的用法相比，开发人员编写的内部类必须知道关于 ResultSet 和 PreparedStatement 以及它们的各个 API 的更多信息，但您仍然无需关心 SQLException 的处理；JdbcTemplate 将执行异常处理。前面的基于 Spring 的代码清单（例如代码清单 3 和 6 中使用的 JdbcTemplate）甚至没有提到 ResultSet、Statement、PreparedStatement 或 SQLException。这些高度抽象的方法对于不想关心 JDBC 的具体用法的开发人员特别有用。不过，这些极其方便的方法没有代码清单 16 和 17 所演示的内部类方法灵活。代码清单 16 和 17 中显示的更灵活的方法可以在需要时使用（只需稍微了解基本的 JDBC API）。在所有情况下，异常处理都由 Spring 异常层次结构来一致地执行，您不需要关心 SQLException。其他好处 代码清单 1

```
try{// JDBC Connection/Statement/Result Set}catch (SQLException sqlEx){// Handle the exception}finally{try{ // Closing connection *should* close statement and result setif (stmt != null) stmt.close().if (conn != null) conn.close(). }catch (SQLException sqlEx)
```

{System.err.println("SQLException NOT handled"). } }finally 子句通常被用来确保关闭数据库连接和语句。但即使当开发人员的确用这种方法确保成功关闭连接，代码也是冗长、膨胀和重复的。Spring 框架对连接处理和相关资源管理进行了抽象，开发人员不用直接处理上述事项，从而实现更一致的资源关闭并编写更易于理解的代码。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