Atitit 数据库抽象层jdbc pdo ado.net等比较与异常点

目录

[1. 应该具有的功能 1](#_Toc14573)

[1.1. 元数据 API 1](#_Toc19486)

[1.2. 分布式事务 vs事务中使用 Savepoint 1](#_Toc8639)

[1.3. 离线数据api 1](#_Toc11885)

[1.4. Ui数据绑定 2](#_Toc28193)

[1.5. 多数据集的支持 2](#_Toc4731)

[1.6. 分页访问数据的支持 2](#_Toc31991)

[1.7. 数据源的获取方式式多样 2](#_Toc28954)

[2. 离线数据集合的填充方式 2](#_Toc2910)

[2.1. （1）从数据库直接获取数据 3](#_Toc25652)

[2.2. （2）用ResultSet填充 3](#_Toc32699)

[2.3. （3）用XML填充 3](#_Toc9277)

[2.4. （4）用其他方法填充 csv、excel、text、http等格式 4](#_Toc21988)

[3. 4、操作离线数据集的操作RowSet中的数据及元数据 4](#_Toc5658)

[3.1. 1）更新数据 4](#_Toc32236)

[3.2. （2）插入数据 4](#_Toc5449)

[3.3. （3）删除数据 4](#_Toc18984)

[3.4. （4）查询 4](#_Toc17494)

[3.5. （5）元数据 5](#_Toc11865)

[4. 类与方法对应表 5](#_Toc15836)

[5. Rf 5](#_Toc24863)

# 应该具有的功能

## 元数据 API

## 分布式事务 vs事务中使用 Savepoint

ADO .NET不包括分布式事务的接口，ADO .NET的分布式事务是通过MS DTC统一管理的。JDBC本身就提供了对分布式事务支持的接口，不同的JDBC Driver实现了这一个接口，就支持分布式事务了。

## 离线数据api

ADO .NET最大的特性是对断开连接方式的全方位支持，其引入了DataSet、DataTable、DataRow等等对象，构建了一个“内存数据库”简化版本

DataAdapter把DataReader的数据填充到DataTable或者DataSet中，给使用者一个直观的使用方式

ADO .NET最大的优点是对断开连接访问数据库方式的强有力支持。相比起来，JDBC也引入类似的功能，RowSet，但是比起ADO .NET来，还是不够。

Php pdo好像没有离线数据api

## Ui数据绑定

ADO .NET还能够和ASP .NET和Windows Form的控件进行数据邦定，使得编写一些小程序很简单。

## 多数据集的支持

## 分页访问数据的支持

## 数据源的获取方式式多样

JDBC的数据源的获取方式多样：  
  a、直接通过DriverManager.getConnection获得连接  
  b、通过应用服务器的数据源获得连接。例如：

IMG_256  // Get the initial JNDI naming context  
IMG_257  Context ctx = new InitialContext();  
IMG_258  // Get the DataSource object associated with the logical name  
IMG_259  // "jdbc/AcmeDB" and use it to obtain a database connection  
IMG_260  DataSource ds = (DataSource)ctx.lookup("jdbc/AcmeDB");  
IMG_261  Connection con = ds.getConnection("user", "pwd");

 .NET中，没有类似的咚咚。

# 离线数据集合的填充方式

3、填充RowSet

　　前面说过，应该倾向于把RowSet看成是与数据库无关而只代表一行行数据的对象，因此就涉及到数据从哪里来的问题。

## （1）从数据库直接获取数据

　　由于大部分情况下，与数据打交道也就是与数据库打交道，因此RowSet接口提供了通过JDBC直接从数据库获取数据的方法，以参考实现JdbcRowSetImpl为例，就是这样：

[IMG_256](https://www.cnblogs.com/wxgblogs/p/javascript:void(0);)

　RowSet rs = new JdbcRowSetImpl();//也可以是 CachedRowSetImpl， WebRowSetImpl，FilteredRowSetImpl，JoinRowSetImpl。

　　rs.setUrl("jdbc:mysql:///test");

　　rs.setUsername("root");

　　rs.setPassword("");

　　rs.setCommand("SELECT \* FROM EMPLOYEES");

　　rs.execute();

[IMG_257](https://www.cnblogs.com/wxgblogs/p/javascript:void(0);)

　　设置好相关属性，运行execute()方法后，EMPLOYEES表中的数据就被填充到rs对象中了。

　　除了通过设置JDBC连接URL、用户名和密码外，RowSet也可以使用数据源名称属性的值来查找已经在命名服务中注册的DataSource对象。完成检索后，可以使用DataSource对象创建到它所表示的数据源的连接，设置数据源名称可以使用setDataSourceName()方法。

## （2）用ResultSet填充

　　在有现成ResultSet的情况下，如果想将其作为RowSet使用；或者当DBMS不提供对滚动和更新的完全支持时，如果想使不可滚动和只读的 ResultSet对象变得可滚动和可更新，可以创建一个使用该ResultSet 对象的数据所填充的 CachedRowSet 对象。

　　ResultSet rs = stmt.executeQuery("SELECT \* FROM EMPLOYEES");

　　CachedRowSet crs = newCachedRowSetImpl();//也可以是WebRowSetImpl，FilteredRowSetImpl，JoinRowSetImpl，因为他们均继承自CachedRowSetImpl

　　crs.populate(rs);

　　运行populate()方法后，ResultSet对象rs中的数据就被填充到crs对象中了。

## （3）用XML填充

　　如果您打算将XML作为数据交换格式在客户端和你的服务器之间传输数据并且向实现数据离线编辑、或者向使用XML格式的数据的话，可以使用WebRowSet接口来用XML填充数据。

　WebRowSet wrs = new WebRowSetImpl();

wrs.readXml(new FileReader(new File("D:\\employees.xml")));

　　运行readXml()方法后，employees.xml文件的数据就被填充到wrs对象中了。employees.xml文件的格式参见附录。

## （4）用其他方法填充 csv、excel、text、http等格式

　　如果形用其他方式填充，比如csv、excel、text、http等格式或方法填充数据，那么就需要自己编写代码实现RowSet。

# 4、操作离线数据集的操作RowSet中的数据及元数据

　　除了ResultSet提供的操作数据和元数据方法外，RowSet接口没有提供太多额外的方法。

## 1）更新数据

　 rs.absolute(5);

　　rs.updateInt(1, 10);

　　rs.updateInt(2, 1000);

　　rs.updateString(3, "John");

　　rs.updateRow();

## （2）插入数据

　　rs.moveToInsertRow();

　　rs.updateInt(1, 10);

　　rs.updateInt(2, 1000);

　　rs.updateString(3, "John");

　　rs.insertRow();

## （3）删除数据

rs.absolute(5);

rs.deleteRow();

## （4）查询

IMG_256

rs.setCommand("select id, salary, name from employees where id=?");

　　rs.setInt(1, 1);

　　rs.execute();

## （5）元数据

RowSetMetaData rsmd = (RowSetMetaData)rs.getMetaData();

　　int count = rsmd.getColumnCount();

　　int type = rsmd.getColumnType(2);

# 类与方法对应表

|  |  |  |  |
| --- | --- | --- | --- |
| 概念，类，方法 | Net | Java | Php pdo |
| 脱机数据集 | DataSet、DataTable | RowSet |  |
| conn | connection | Connection | pdo |
| **statement** | **command** | **statement** | Statement |
|  |  |  |  |
| 执行curd | command.ExecuteNonQuery | Statement。executeUpdate | $pdo->exec |
| 执行查询 | command.ExecuteReader | statement.executeQuery | Statement.query（） |
| 数据集转换提取批量 | ataAdapter把DataReader的数据填充到DataTable或者DataSet中， | ResultSet填充 | Statement->fetchAll(); |
| 数据集转换提取 | 遍历结果 | 遍历结果 | Statement->fetch(); |
|  |  |  |  |
|  |  |  |  |

# Rf

Atitit jdbc3.0 4.0 4.1新特性

再谈ADO .NET vs JDBC - 温少 - 博客园.html

RowSet的使用 - 简单爱\_wxg - 博客园.html

RowSet的使用 - 简单爱\_wxg - 博客园.html

JDBC RowSet接口.html

PHP PDO \_ w3cschool菜鸟教程.html