Basic java.sql Package Interface References

C1: CallableStatement Interface Methods

Most CallableStatement Interface Methods are shown in Table C.1.

Table C.1 Methods defined in the CallableStatement interface

Method	Function
getBigDecimal(int index, int scale)	Returns the value of parameter specified by the parameter index as a BigDecimal
getBoolean(int index)	Returns the value of parameter specified by the parameter index as a Boolean
getByte(int index)	Returns the value of parameter specified by the parameter index as a Byte
getBytes(int index)	Returns the value of parameter specified by the parameter index as an array of Byte – byte[]
getDouble(int index)	Returns the value of parameter specified by the parameter index as a Double
getShort(int index)	Returns the value of parameter specified by the parameter index as a Short Integer
getFloat(int index)	Returns the value of parameter specified by the parameter index as a Floating point number
getInt(int index)	Returns the value of parameter specified by the parameter index as a Integer
getLong(int index)	Returns the value of parameter specified by the parameter index as a Long
getObject(int index)	Returns the value of parameter specified by the parameter index as an Object object. The object type is determined by the default mapping of the SQL data type to Java data type
getString(int index)	Returns the value of parameter specified by the parameter index as a String object
registerOutParameter(int index, int sqlType)	Register specified output parameter to receive the SQL data type indictaed by the argument passed. If the output is registered as either DECIMAL or NUMERIC, the scale value maybe specified
wasNull()	Determines if the last value read by a getter method was a SQL null value

C2: Connection Interface Methods

Most Connection Interface Methods are shown in Table C.2.

Table C.2 Methods defined in the Connection interface

Method	Function
Close()	Close a connection to the database
Commit()	Immediately commits all transactions to the database
createStatement()	Create a Statement object for the execution of static SQL statements
getMetaData()	Returns a Database MetaData object for the current connection
isClosed()	Returns a Boolean value to indicate whether a database connection has been closed, true – closed
prepareCall(String sqlString)	Create a CallableStatement object for use with SQL statement requiring IN and OUT parameters
prepareStatement(String sqlString)	Create a PreparedStatement object for use with SQL statements requiring IN parameters
rollback()	Immediately perform a transaction rollback on the database and cancel all pending transactions
setAutoCommit(Boolean autoCommit)	Toggles the automatic commit feature on or off
setCatalog(String catalog)	Select which catalog within the database is to be used. If the database does not support catalog, this method is ignored
setReadOnly(Boolean readOnly)	Set the current connection to read-only mode
setTransactionIsolation(int level)	Set the transaction isolation level for all subsequent transactions

C3: DatabaseMetaData Interface Methods

Most DatabaseMetaData Interface Methods are shown in Table C.3.

Table C.3 Popular methods defined in the DatabaseMetaData interface

Method	Function
getCatalogs()	Return a ResultSet containing a list of all catalogs available in the database
getCatalogTerm()	Determine what the database specific name for Catalog is
getDatabaseProductName()	Return the name of the database product
getDatabaseProductVersion()	Return the database revision number
getDriverName()	Return the name of the driver
getDriverVersion()	Return the revision number of the driver
getPrimaryKeys(String catalog, String schema, String table)	Return a ResultSet describing all of the primary keys within a table
getProcedures(string catalog, String schPatt, String proPatt)	Return a ResultSet describing all stored procedures available in the catalog
getProcedureTerm()	Determine the database specific term for procedure
getSchemas()	Return a ResultSet containing a list of all schemas available in the database
getSchemaTerm()	Determine the database specific term for schema
getTables(String catalog, String schePatt, String tablePatt, String[] types)	Return a ResultSet containing a list of all tables available matching the catalog, schema and table type selection critaria
getTableTypes()	Return a ResultSet listing the table types available
getTypeInfo()	Return a ResultSet describing all of the standard SQL types supported by the database
getURL()	Return the current URL for the database
getUserName()	Return the current user name used by the database
getColumns(String catalog, String schema, String table, String pattern)	Return a ResultSet object containing information for each column of a table

C4: Driver Interface Methods

Most Driver Interface Methods are shown in Table C.4.

Table C.4 Methods defined in the Driver interface

Method	Function
acceptsURL(String url)	Return a true if the driver is able to open a connection to the database given by the URL
connect(String url, Properties login)	Check the syntax of the URL and the matched drivers in the driver list. Attempt to make a database connection to the given URL
getMajorVersion()	Determine the minor revision number of the driver
getMinorVersion()	Determine the major revision number of the driver
getPropertyInfo(String url, Properties login)	Return an array of DriverPropertyInfo objects describing login properties accepted by the database
jdbcCompliant()	Determine if the driver is JDBC COMPLIANT

C5: PreparedStatement Interface Methods

Most PreparedStatement Interface Methods are shown in Table C.5.

Table C.5 Methods defined in the PreparedStatement interface

Method	Function
clearParameters()	Clear all parameters associated with a PreparedStatement. After execution of this method, all parameters have the value null
execute()	Execute the associated SQL Statement when the number of results returned is unknown. A False is returned if the returned result is null
executeQuery()	Execute an SQL Select statement. A ResultSet object that contained the query results from the database will be returned
executeUpdate()	Execute an SQL Update, Insert or Delete statement. An integer will be returned to indicate the number of rows that have been affected
getMetaData()	Return a set of meta data for the returned ResultSet object
getParameterMetaData()	Return the number, types and properties of this PreparedStatement object's parameters
setBoolean(int index, Boolean value)	Bind a Boolean value to an input parameter
setByte(int index, Byte value)	Bind a byte value to an input parameter
setDouble(int index, double value)	Bind a double value to an input parameter
setFloat(int index, float value)	Bind a floating point value to an input parameter
setInt(int index, int value)	Bind an integer value to an input parameter
setLong(int index, long value)	Bind a long value to an input parameter
setNull(int index, int sqlType)	Bind a null value to an input parameter
setObject(int index, Object obj)	Bind an Object to an input parameter. The Object will be converted to an SQL data type before being sent to the database
setShort(int index, short value)	Bind a short value to an input parameter
setString(int index, String value)	Bind a String value to an input parameter
setTime(int index, Time value)	Bind a Time value to an input parameter

C6: ResultSet Interface Methods

Most ResultSet Interface Methods are shown in Table C.6.

Table C.6 Methods defined in the ResultSet interface

Method	Function
close()	Close the ResultSet and release all resources associated with it
findColumn(String colName)	Return the column index number corresponding to the column name argument
getAsciiStream(int index)	Retrieve the value of the specified column from the current row as an ASCII stream. The column can be represented by either the column index or the column name
getBigDecimal(int index)	Return the value of the referenced column from the current row as a BigDecimal object
getBoolean(int index)	Return the value of the referenced column from the current row as a Boolean
getByte(int index)	Return the value of the referenced column from the current row as a byte
getBytes(int index)	Return the value of the referenced column from the current row as an array of bytes
getDouble(int index)	Return the value of the referenced column from the current row as a double
getFloat(int index)	Return the value of the referenced column from the current row as a floating point number
getInt(int index)	Return the value of the referenced column from the current row as an integer
getLong(int index)	Return the value of the referenced column from the current row as a long integer
getObject(int index)	Return the value of the referenced column from the current row as an Object. The object type is determined by the default mapping of the SQL data type
getShort(int index)	Return the value of the referenced column from the current row as a short integer
getString(int index)	Return the value of the referenced column from the current row as a String object
getTime(int index)	Return the value of the referenced column from the current row as a java.sql.Time object
getMetaData()	Return a meta data object from the ResultSet object
next()	Move the ResultSet row cursor to the next row
wasNull()	Determine if the last value read by a getXXX() method was a SQL null value. A True is returned if the last read value contained a null value

C7: ResultSetMetaData Interface Methods

Most ResultSetMetaData Interface Methods are shown in Table C.7.

Table C.7 Methods defined in the ResultSetMetaData interface

Method	Function
getCatalogName(int index)	Determine the name of the catalog that contains the referenced column
getColumnCount()	Return the total number of columns contained in the ResultSet object
getColumnDisplaySize(int index)	Return the maximum display width for the selected column
getColumnLabel(int index)	Return the preferred display name for the selected column
getColumnName(int index)	Return the name of the column for the selected column
getColumnType(int index)	Return the SQL data type for the selected column
getPrecision(int index)	Return the precision used for the selected column
getScale(int index)	Return the scale used for the selected column
getSchemaName(int index)	Return the name of the schema that contains the selected column
getTableName(int index)	Return the name of the table that contains the selected column
isAutoIncrement(int index)	Determine if the column is automatically numbered by the database (auto-number)
isCurrency(int index)	Determine if the column represents currency
isNullable(int index)	Determine if the column is able to accept null values
isSigned(int index)	Determine if the column contains signed numbers
isWritable(int index)	Determine if the column is writable by the user
isReadOnly(int index)	Determine if the column is read-only

C8: Statement Interface Methods

Most Statement Interface Methods are shown in Table C.8.

Table C.8 Methods defined in the Statement interface

Method	Function
close()	Close the Statement and release all resources including the ResultSet associated with it
execute(String sqlString)	Execute an SQL statement that may have an unknown number of results. Returned a True means that the first set of results from the sqlString execution is a ResultSet. If the execution resulted in either no results or an update count, a False is returned
executeQuery(String sqlString)	Execute an SQL Select statement. A ResultSet object that contained the query results from the database will be returned
executeUpdate(String sqlString)	Execute an SQL Update, Insert or Delete statement. An integer will be returned to indicate the number of rows that have been affected
getMaxRows()	Determine the maximum number of rows that can be returned in a ResultSet object
getMoreResults()	Move to the Statements next result. Only in conjunction with the execute statement and where multiple results are returned by the SQL statement. A False is returned if the next result is null or the results are an update count
getResultSet()	Return the current result set for the Statement. Only used in conjunction with execute() method. The current ResultSet object will be returned
getUpdateCount()	Return the number of rows affected by the last SQL statement. Is only meaningful for INSERT, UPDATE or DELETE statements
setCursorName(String name)	Set the cursor name to be used by the Statement. Only useful for databases that support positional updates and deletes
setMaxRows(int rows)	Set the maximum number of rows that can be returned in a ResultSet. If more results are returned by the query, they are truncated