| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/WebRowSet.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/sql/rowset/RowSetWarning.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?javax/sql/rowset/WebRowSet.html)    [**NO FRAMES**](http://docs.google.com/WebRowSet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#26in1rg) | CONSTR | [METHOD](#1ksv4uv) |

## **javax.sql.rowset**

Interface WebRowSet

**All Superinterfaces:** [CachedRowSet](http://docs.google.com/javax/sql/rowset/CachedRowSet.html), [Joinable](http://docs.google.com/javax/sql/rowset/Joinable.html), [ResultSet](http://docs.google.com/java/sql/ResultSet.html), [RowSet](http://docs.google.com/javax/sql/RowSet.html), [Wrapper](http://docs.google.com/java/sql/Wrapper.html) **All Known Subinterfaces:** [FilteredRowSet](http://docs.google.com/javax/sql/rowset/FilteredRowSet.html), [JoinRowSet](http://docs.google.com/javax/sql/rowset/JoinRowSet.html)

public interface **WebRowSet**extends [CachedRowSet](http://docs.google.com/javax/sql/rowset/CachedRowSet.html)

The standard interface that all implementations of a WebRowSet must implement.

### 1.0 Overview

The WebRowSetImpl provides the standard reference implementation, which may be extended if required.

The standard WebRowSet XML Schema definition is available at the following URI:

<http://java.sun.com/xml/ns/jdbc/webrowset.xsd>  
 It describes the standard XML document format required when describing a RowSet object in XML and must be used be all standard implementations of the WebRowSet interface to ensure interoperability. In addition, the WebRowSet schema uses specific SQL/XML Schema annotations, thus ensuring greater cross platform inter-operability. This is an effort currently under way at the ISO organization. The SQL/XML definition is available at the following URI:

<http://standards.iso.org/iso/9075/2002/12/sqlxml>  
 The schema definition describes the internal data of a RowSet object in three distinct areas:

* **properties**

These properties describe the standard synchronization provider properties in addition to the more general RowSet properties.

* **metadata**

This describes the metadata associated with the tabular structure governed by a WebRowSet object. The metadata described is closely aligned with the metadata accessible in the underlying java.sql.ResultSet interface.

* **data**

This describes the original data (the state of data since the last population or last synchronization of the WebRowSet object) and the current data. By keeping track of the delta between the original data and the current data, a WebRowSet maintains the ability to synchronize changes in its data back to the originating data source.

### 2.0 WebRowSet States

The following sections demonstrates how a WebRowSet implementation should use the XML Schema to describe update, insert, and delete operations and to describe the state of a WebRowSet object in XML.

#### 2.1 State 1 - Outputting a WebRowSet Object to XML

In this example, a WebRowSet object is created and populated with a simple 2 column, 5 row table from a data source. Having the 5 rows in a WebRowSet object makes it possible to describe them in XML. The metadata describing the various standard JavaBeans properties as defined in the RowSet interface plus the standard properties defined in the CachedRowSetTM interface provide key details that describe WebRowSet properties. Outputting the WebRowSet object to XML using the standard writeXml methods describes the internal properties as follows:

<properties>  
 <command>select co1, col2 from test\_table</command>  
 <concurrency>1</concurrency>  
 <datasource/>  
 <escape-processing>true</escape-processing>  
 <fetch-direction>0</fetch-direction>  
 <fetch-size>0</fetch-size>  
 <isolation-level>1</isolation-level>  
 <key-columns/>  
 <map/>  
 <max-field-size>0</max-field-size>  
 <max-rows>0</max-rows>  
 <query-timeout>0</query-timeout>  
 <read-only>false</read-only>  
 <rowset-type>TRANSACTION\_READ\_UNCOMMITED</rowset-type>  
 <show-deleted>false</show-deleted>  
 <table-name/>  
 <url>jdbc:thin:oracle</url>  
 <sync-provider>  
 <sync-provider-name>.com.rowset.provider.RIOptimisticProvider</sync-provider-name>  
 <sync-provider-vendor>Sun Microsystems</sync-provider-vendor>  
 <sync-provider-version>1.0</sync-provider-name>  
 <sync-provider-grade>LOW</sync-provider-grade>  
 <data-source-lock>NONE</data-source-lock>  
 </sync-provider>  
 </properties>

The meta-data describing the make up of the WebRowSet is described in XML as detailed below. Note both columns are described between the column-definition tags.

<metadata>  
 <column-count>2</column-count>  
 <column-definition>  
 <column-index>1</column-index>  
 <auto-increment>false</auto-increment>  
 <case-sensitive>true</case-sensitive>  
 <currency>false</currency>  
 <nullable>1</nullable>  
 <signed>false</signed>  
 <searchable>true</searchable>  
 <column-display-size>10</column-display-size>   
 <column-label>COL1</column-label>  
 <column-name>COL1</column-name>  
 <schema-name/>  
 <column-precision>10</column-precision>  
 <column-scale>0</column-scale>  
 <table-name/>  
 <catalog-name/>  
 <column-type>1</column-type>  
 <column-type-name>CHAR</column-type-name>  
 </column-definition>  
 <column-definition>  
 <column-index>2</column-index>  
 <auto-increment>false</auto-increment>  
 <case-sensitive>false</case-sensitive>  
 <currency>false</currency>  
 <nullable>1</nullable>  
 <signed>true</signed>  
 <searchable>true</searchable>  
 <column-display-size>39</column-display-size>  
 <column-label>COL2</column-label>  
 <column-name>COL2</column-name>  
 <schema-name/>  
 <column-precision>38</column-precision>  
 <column-scale>0</column-scale>  
 <table-name/>  
 <catalog-name/>  
 <column-type>3</column-type>  
 <column-type-name>NUMBER</column-type-name>  
 </column-definition>  
 </metadata>

Having detailed how the properties and metadata are described, the following details how the contents of a WebRowSet object is described in XML. Note, that this describes a WebRowSet object that has not undergone any modifications since its instantiation. A currentRow tag is mapped to each row of the table structure that the WebRowSet object provides. A columnValue tag may contain either the stringData or binaryData tag, according to the SQL type that the XML value is mapping back to. The binaryData tag contains data in the Base64 encoding and is typically used for BLOB and CLOB type data.

<data>  
 <currentRow>  
 <columnValue>  
 firstrow  
 </columnValue>  
 <columnValue>  
 1  
 </columnValue>  
 </currentRow>  
 <currentRow>  
 <columnValue>  
 secondrow  
 </columnValue>  
 <columnValue>  
 2  
 </columnValue>  
 </currentRow>  
 <currentRow>  
 <columnValue>  
 thirdrow  
 </columnValue>  
 <columnValue>  
 3  
 </columnValue>  
 </currentRow>  
 <currentRow>  
 <columnValue>  
 fourthrow  
 </columnValue>  
 <columnValue>  
 4  
 </columnValue>  
 </currentRow>  
 </data>

#### 2.2 State 2 - Deleting a Row

Deleting a row in a WebRowSet object involves simply moving to the row to be deleted and then calling the method deleteRow, as in any other RowSet object. The following two lines of code, in which *wrs* is a WebRowSet object, delete the third row.

wrs.absolute(3);  
 wrs.deleteRow();

The XML description shows the third row is marked as a deleteRow, which eliminates the third row in the WebRowSet object.

<data>  
 <currentRow>  
 <columnValue>  
 firstrow  
 </columnValue>  
 <columnValue>  
 1  
 </columnValue>  
 </currentRow>  
 <currentRow>  
 <columnValue>  
 secondrow  
 </columnValue>  
 <columnValue>  
 2  
 </columnValue>  
 </currentRow>  
 <deleteRow>  
 <columnValue>  
 thirdrow  
 </columnValue>  
 <columnValue>  
 3  
 </columnValue>  
 </deleteRow>  
 <currentRow>  
 <columnValue>  
 fourthrow  
 </columnValue>  
 <columnValue>  
 4  
 </columnValue>  
 </currentRow>  
 </data>

#### 2.3 State 3 - Inserting a Row

A WebRowSet object can insert a new row by moving to the insert row, calling the appropriate updater methods for each column in the row, and then calling the method insertRow.

wrs.moveToInsertRow();  
 wrs.updateString(1, "fifththrow");  
 wrs.updateString(2, "5");  
 wrs.insertRow();

The following code fragment changes the second column value in the row just inserted. Note that this code applies when new rows are inserted right after the current row, which is why the method next moves the cursor to the correct row. Calling the method acceptChanges writes the change to the data source.

wrs.moveToCurrentRow();  
 wrs.next();  
 wrs.updateString(2, "V");  
 wrs.acceptChanges();  
 :

Describing this in XML demonstrates where the Java code inserts a new row and then performs an update on the newly inserted row on an individual field.

<data>  
 <currentRow>  
 <columnValue>  
 firstrow  
 </columnValue>  
 <columnValue>  
 1  
 </columnValue>  
 </currentRow>  
 <currentRow>  
 <columnValue>  
 secondrow  
 </columnValue>  
 <columnValue>  
 2  
 </columnValue>  
 </currentRow>  
 <currentRow>  
 <columnValue>  
 newthirdrow  
 </columnValue>  
 <columnValue>  
 III  
 </columnValue>  
 </currentRow>  
 <insertRow>  
 <columnValue>  
 fifthrow  
 </columnValue>  
 <columnValue>  
 5  
 </columnValue>  
 <updateValue>  
 V  
 </updateValue>  
 </insertRow>  
 <currentRow>  
 <columnValue>  
 fourthrow  
 </columnValue>  
 <columnValue>  
 4  
 </columnValue>  
 </currentRow>  
 </date>

#### 2.4 State 4 - Modifying a Row

Modifying a row produces specific XML that records both the new value and the value that was replaced. The value that was replaced becomes the original value, and the new value becomes the current value. The following code moves the cursor to a specific row, performs some modifications, and updates the row when complete.

wrs.absolute(5);  
 wrs.updateString(1, "new4thRow");  
 wrs.updateString(2, "IV");  
 wrs.updateRow();

In XML, this is described by the modifyRow tag. Both the original and new values are contained within the tag for original row tracking purposes.

<data>  
 <currentRow>  
 <columnValue>  
 firstrow  
 </columnValue>  
 <columnValue>  
 1  
 </columnValue>  
 </currentRow>  
 <currentRow>  
 <columnValue>  
 secondrow  
 </columnValue>  
 <columnValue>  
 2  
 </columnValue>  
 </currentRow>  
 <currentRow>  
 <columnValue>  
 newthirdrow  
 </columnValue>  
 <columnValue>  
 III  
 </columnValue>  
 </currentRow>  
 <currentRow>  
 <columnValue>  
 fifthrow  
 </columnValue>  
 <columnValue>  
 5  
 </columnValue>  
 </currentRow>  
 <modifyRow>  
 <columnValue>  
 fourthrow  
 </columnValue>  
 <updateValue>  
 new4thRow  
 </updateValue>  
 <columnValue>  
 4  
 </columnValue>  
 <updateValue>  
 IV  
 </updateValue>  
 </modifyRow>  
 </data>

**See Also:**[JdbcRowSet](http://docs.google.com/javax/sql/rowset/JdbcRowSet.html), [CachedRowSet](http://docs.google.com/javax/sql/rowset/CachedRowSet.html), [FilteredRowSet](http://docs.google.com/javax/sql/rowset/FilteredRowSet.html), [JoinRowSet](http://docs.google.com/javax/sql/rowset/JoinRowSet.html)

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**PUBLIC\_XML\_SCHEMA**](http://docs.google.com/javax/sql/rowset/WebRowSet.html#PUBLIC_XML_SCHEMA)            The public identifier for the XML Schema definition that defines the XML tags and their valid values for a WebRowSet implementation. |
| static [String](http://docs.google.com/java/lang/String.html) | [**SCHEMA\_SYSTEM\_ID**](http://docs.google.com/javax/sql/rowset/WebRowSet.html#SCHEMA_SYSTEM_ID)            The URL for the XML Schema definition file that defines the XML tags and their valid values for a WebRowSet implementation. |

| **Fields inherited from interface javax.sql.rowset.**[**CachedRowSet**](http://docs.google.com/javax/sql/rowset/CachedRowSet.html) |
| --- |
| [COMMIT\_ON\_ACCEPT\_CHANGES](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#COMMIT_ON_ACCEPT_CHANGES) |

| **Fields inherited from interface java.sql.**[**ResultSet**](http://docs.google.com/java/sql/ResultSet.html) |
| --- |
| [CLOSE\_CURSORS\_AT\_COMMIT](http://docs.google.com/java/sql/ResultSet.html#CLOSE_CURSORS_AT_COMMIT), [CONCUR\_READ\_ONLY](http://docs.google.com/java/sql/ResultSet.html#CONCUR_READ_ONLY), [CONCUR\_UPDATABLE](http://docs.google.com/java/sql/ResultSet.html#CONCUR_UPDATABLE), [FETCH\_FORWARD](http://docs.google.com/java/sql/ResultSet.html#FETCH_FORWARD), [FETCH\_REVERSE](http://docs.google.com/java/sql/ResultSet.html#FETCH_REVERSE), [FETCH\_UNKNOWN](http://docs.google.com/java/sql/ResultSet.html#FETCH_UNKNOWN), [HOLD\_CURSORS\_OVER\_COMMIT](http://docs.google.com/java/sql/ResultSet.html#HOLD_CURSORS_OVER_COMMIT), [TYPE\_FORWARD\_ONLY](http://docs.google.com/java/sql/ResultSet.html#TYPE_FORWARD_ONLY), [TYPE\_SCROLL\_INSENSITIVE](http://docs.google.com/java/sql/ResultSet.html#TYPE_SCROLL_INSENSITIVE), [TYPE\_SCROLL\_SENSITIVE](http://docs.google.com/java/sql/ResultSet.html#TYPE_SCROLL_SENSITIVE) |

| **Method Summary** | |
| --- | --- |
| void | [**readXml**](http://docs.google.com/javax/sql/rowset/WebRowSet.html#readXml(java.io.InputStream))([InputStream](http://docs.google.com/java/io/InputStream.html) iStream)            Reads a stream based XML input to populate this WebRowSet object. |
| void | [**readXml**](http://docs.google.com/javax/sql/rowset/WebRowSet.html#readXml(java.io.Reader))([Reader](http://docs.google.com/java/io/Reader.html) reader)            Reads a WebRowSet object in its XML format from the given Reader object. |
| void | [**writeXml**](http://docs.google.com/javax/sql/rowset/WebRowSet.html#writeXml(java.io.OutputStream))([OutputStream](http://docs.google.com/java/io/OutputStream.html) oStream)            Writes the data, properties, and metadata for this WebRowSet object to the given OutputStream object in XML format. |
| void | [**writeXml**](http://docs.google.com/javax/sql/rowset/WebRowSet.html#writeXml(java.sql.ResultSet,%20java.io.OutputStream))([ResultSet](http://docs.google.com/java/sql/ResultSet.html) rs, [OutputStream](http://docs.google.com/java/io/OutputStream.html) oStream)            Populates this WebRowSet object with the contents of the given ResultSet object and writes its data, properties, and metadata to the given OutputStream object in XML format. |
| void | [**writeXml**](http://docs.google.com/javax/sql/rowset/WebRowSet.html#writeXml(java.sql.ResultSet,%20java.io.Writer))([ResultSet](http://docs.google.com/java/sql/ResultSet.html) rs, [Writer](http://docs.google.com/java/io/Writer.html) writer)            Populates this WebRowSet object with the contents of the given ResultSet object and writes its data, properties, and metadata to the given Writer object in XML format. |
| void | [**writeXml**](http://docs.google.com/javax/sql/rowset/WebRowSet.html#writeXml(java.io.Writer))([Writer](http://docs.google.com/java/io/Writer.html) writer)            Writes the data, properties, and metadata for this WebRowSet object to the given Writer object in XML format. |

| **Methods inherited from interface javax.sql.rowset.**[**CachedRowSet**](http://docs.google.com/javax/sql/rowset/CachedRowSet.html) |
| --- |
| [acceptChanges](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#acceptChanges()), [acceptChanges](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#acceptChanges(java.sql.Connection)), [columnUpdated](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#columnUpdated(int)), [columnUpdated](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#columnUpdated(java.lang.String)), [commit](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#commit()), [createCopy](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#createCopy()), [createCopyNoConstraints](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#createCopyNoConstraints()), [createCopySchema](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#createCopySchema()), [createShared](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#createShared()), [execute](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#execute(java.sql.Connection)), [getKeyColumns](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#getKeyColumns()), [getOriginal](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#getOriginal()), [getOriginalRow](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#getOriginalRow()), [getPageSize](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#getPageSize()), [getRowSetWarnings](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#getRowSetWarnings()), [getShowDeleted](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#getShowDeleted()), [getSyncProvider](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#getSyncProvider()), [getTableName](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#getTableName()), [nextPage](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#nextPage()), [populate](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#populate(java.sql.ResultSet)), [populate](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#populate(java.sql.ResultSet,%20int)), [previousPage](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#previousPage()), [release](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#release()), [restoreOriginal](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#restoreOriginal()), [rollback](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#rollback()), [rollback](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#rollback(java.sql.Savepoint)), [rowSetPopulated](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#rowSetPopulated(javax.sql.RowSetEvent,%20int)), [setKeyColumns](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#setKeyColumns(int%5B%5D)), [setMetaData](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#setMetaData(javax.sql.RowSetMetaData)), [setOriginalRow](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#setOriginalRow()), [setPageSize](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#setPageSize(int)), [setShowDeleted](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#setShowDeleted(boolean)), [setSyncProvider](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#setSyncProvider(java.lang.String)), [setTableName](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#setTableName(java.lang.String)), [size](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#size()), [toCollection](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#toCollection()), [toCollection](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#toCollection(int)), [toCollection](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#toCollection(java.lang.String)), [undoDelete](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#undoDelete()), [undoInsert](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#undoInsert()), [undoUpdate](http://docs.google.com/javax/sql/rowset/CachedRowSet.html#undoUpdate()) |

| **Methods inherited from interface javax.sql.**[**RowSet**](http://docs.google.com/javax/sql/RowSet.html) |
| --- |
| [addRowSetListener](http://docs.google.com/javax/sql/RowSet.html#addRowSetListener(javax.sql.RowSetListener)), [clearParameters](http://docs.google.com/javax/sql/RowSet.html#clearParameters()), [execute](http://docs.google.com/javax/sql/RowSet.html#execute()), [getCommand](http://docs.google.com/javax/sql/RowSet.html#getCommand()), [getDataSourceName](http://docs.google.com/javax/sql/RowSet.html#getDataSourceName()), [getEscapeProcessing](http://docs.google.com/javax/sql/RowSet.html#getEscapeProcessing()), [getMaxFieldSize](http://docs.google.com/javax/sql/RowSet.html#getMaxFieldSize()), [getMaxRows](http://docs.google.com/javax/sql/RowSet.html#getMaxRows()), [getPassword](http://docs.google.com/javax/sql/RowSet.html#getPassword()), [getQueryTimeout](http://docs.google.com/javax/sql/RowSet.html#getQueryTimeout()), [getTransactionIsolation](http://docs.google.com/javax/sql/RowSet.html#getTransactionIsolation()), [getTypeMap](http://docs.google.com/javax/sql/RowSet.html#getTypeMap()), [getUrl](http://docs.google.com/javax/sql/RowSet.html#getUrl()), [getUsername](http://docs.google.com/javax/sql/RowSet.html#getUsername()), [isReadOnly](http://docs.google.com/javax/sql/RowSet.html#isReadOnly()), [removeRowSetListener](http://docs.google.com/javax/sql/RowSet.html#removeRowSetListener(javax.sql.RowSetListener)), [setArray](http://docs.google.com/javax/sql/RowSet.html#setArray(int,%20java.sql.Array)), [setAsciiStream](http://docs.google.com/javax/sql/RowSet.html#setAsciiStream(int,%20java.io.InputStream)), [setAsciiStream](http://docs.google.com/javax/sql/RowSet.html#setAsciiStream(int,%20java.io.InputStream,%20int)), [setAsciiStream](http://docs.google.com/javax/sql/RowSet.html#setAsciiStream(java.lang.String,%20java.io.InputStream)), [setAsciiStream](http://docs.google.com/javax/sql/RowSet.html#setAsciiStream(java.lang.String,%20java.io.InputStream,%20int)), [setBigDecimal](http://docs.google.com/javax/sql/RowSet.html#setBigDecimal(int,%20java.math.BigDecimal)), [setBigDecimal](http://docs.google.com/javax/sql/RowSet.html#setBigDecimal(java.lang.String,%20java.math.BigDecimal)), [setBinaryStream](http://docs.google.com/javax/sql/RowSet.html#setBinaryStream(int,%20java.io.InputStream)), [setBinaryStream](http://docs.google.com/javax/sql/RowSet.html#setBinaryStream(int,%20java.io.InputStream,%20int)), [setBinaryStream](http://docs.google.com/javax/sql/RowSet.html#setBinaryStream(java.lang.String,%20java.io.InputStream)), [setBinaryStream](http://docs.google.com/javax/sql/RowSet.html#setBinaryStream(java.lang.String,%20java.io.InputStream,%20int)), [setBlob](http://docs.google.com/javax/sql/RowSet.html#setBlob(int,%20java.sql.Blob)), [setBlob](http://docs.google.com/javax/sql/RowSet.html#setBlob(int,%20java.io.InputStream)), [setBlob](http://docs.google.com/javax/sql/RowSet.html#setBlob(int,%20java.io.InputStream,%20long)), [setBlob](http://docs.google.com/javax/sql/RowSet.html#setBlob(java.lang.String,%20java.sql.Blob)), [setBlob](http://docs.google.com/javax/sql/RowSet.html#setBlob(java.lang.String,%20java.io.InputStream)), [setBlob](http://docs.google.com/javax/sql/RowSet.html#setBlob(java.lang.String,%20java.io.InputStream,%20long)), [setBoolean](http://docs.google.com/javax/sql/RowSet.html#setBoolean(int,%20boolean)), [setBoolean](http://docs.google.com/javax/sql/RowSet.html#setBoolean(java.lang.String,%20boolean)), [setByte](http://docs.google.com/javax/sql/RowSet.html#setByte(int,%20byte)), [setByte](http://docs.google.com/javax/sql/RowSet.html#setByte(java.lang.String,%20byte)), [setBytes](http://docs.google.com/javax/sql/RowSet.html#setBytes(int,%20byte%5B%5D)), [setBytes](http://docs.google.com/javax/sql/RowSet.html#setBytes(java.lang.String,%20byte%5B%5D)), [setCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setCharacterStream(int,%20java.io.Reader)), [setCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setCharacterStream(int,%20java.io.Reader,%20int)), [setCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setCharacterStream(java.lang.String,%20java.io.Reader)), [setCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setCharacterStream(java.lang.String,%20java.io.Reader,%20int)), [setClob](http://docs.google.com/javax/sql/RowSet.html#setClob(int,%20java.sql.Clob)), [setClob](http://docs.google.com/javax/sql/RowSet.html#setClob(int,%20java.io.Reader)), [setClob](http://docs.google.com/javax/sql/RowSet.html#setClob(int,%20java.io.Reader,%20long)), [setClob](http://docs.google.com/javax/sql/RowSet.html#setClob(java.lang.String,%20java.sql.Clob)), [setClob](http://docs.google.com/javax/sql/RowSet.html#setClob(java.lang.String,%20java.io.Reader)), [setClob](http://docs.google.com/javax/sql/RowSet.html#setClob(java.lang.String,%20java.io.Reader,%20long)), [setCommand](http://docs.google.com/javax/sql/RowSet.html#setCommand(java.lang.String)), [setConcurrency](http://docs.google.com/javax/sql/RowSet.html#setConcurrency(int)), [setDataSourceName](http://docs.google.com/javax/sql/RowSet.html#setDataSourceName(java.lang.String)), [setDate](http://docs.google.com/javax/sql/RowSet.html#setDate(int,%20java.sql.Date)), [setDate](http://docs.google.com/javax/sql/RowSet.html#setDate(int,%20java.sql.Date,%20java.util.Calendar)), [setDate](http://docs.google.com/javax/sql/RowSet.html#setDate(java.lang.String,%20java.sql.Date)), [setDate](http://docs.google.com/javax/sql/RowSet.html#setDate(java.lang.String,%20java.sql.Date,%20java.util.Calendar)), [setDouble](http://docs.google.com/javax/sql/RowSet.html#setDouble(int,%20double)), [setDouble](http://docs.google.com/javax/sql/RowSet.html#setDouble(java.lang.String,%20double)), [setEscapeProcessing](http://docs.google.com/javax/sql/RowSet.html#setEscapeProcessing(boolean)), [setFloat](http://docs.google.com/javax/sql/RowSet.html#setFloat(int,%20float)), [setFloat](http://docs.google.com/javax/sql/RowSet.html#setFloat(java.lang.String,%20float)), [setInt](http://docs.google.com/javax/sql/RowSet.html#setInt(int,%20int)), [setInt](http://docs.google.com/javax/sql/RowSet.html#setInt(java.lang.String,%20int)), [setLong](http://docs.google.com/javax/sql/RowSet.html#setLong(int,%20long)), [setLong](http://docs.google.com/javax/sql/RowSet.html#setLong(java.lang.String,%20long)), [setMaxFieldSize](http://docs.google.com/javax/sql/RowSet.html#setMaxFieldSize(int)), [setMaxRows](http://docs.google.com/javax/sql/RowSet.html#setMaxRows(int)), [setNCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setNCharacterStream(int,%20java.io.Reader)), [setNCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setNCharacterStream(int,%20java.io.Reader,%20long)), [setNCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setNCharacterStream(java.lang.String,%20java.io.Reader)), [setNCharacterStream](http://docs.google.com/javax/sql/RowSet.html#setNCharacterStream(java.lang.String,%20java.io.Reader,%20long)), [setNClob](http://docs.google.com/javax/sql/RowSet.html#setNClob(int,%20java.sql.NClob)), [setNClob](http://docs.google.com/javax/sql/RowSet.html#setNClob(int,%20java.io.Reader)), [setNClob](http://docs.google.com/javax/sql/RowSet.html#setNClob(int,%20java.io.Reader,%20long)), [setNClob](http://docs.google.com/javax/sql/RowSet.html#setNClob(java.lang.String,%20java.sql.NClob)), [setNClob](http://docs.google.com/javax/sql/RowSet.html#setNClob(java.lang.String,%20java.io.Reader)), [setNClob](http://docs.google.com/javax/sql/RowSet.html#setNClob(java.lang.String,%20java.io.Reader,%20long)), [setNString](http://docs.google.com/javax/sql/RowSet.html#setNString(int,%20java.lang.String)), [setNString](http://docs.google.com/javax/sql/RowSet.html#setNString(java.lang.String,%20java.lang.String)), [setNull](http://docs.google.com/javax/sql/RowSet.html#setNull(int,%20int)), [setNull](http://docs.google.com/javax/sql/RowSet.html#setNull(int,%20int,%20java.lang.String)), [setNull](http://docs.google.com/javax/sql/RowSet.html#setNull(java.lang.String,%20int)), [setNull](http://docs.google.com/javax/sql/RowSet.html#setNull(java.lang.String,%20int,%20java.lang.String)), [setObject](http://docs.google.com/javax/sql/RowSet.html#setObject(int,%20java.lang.Object)), [setObject](http://docs.google.com/javax/sql/RowSet.html#setObject(int,%20java.lang.Object,%20int)), [setObject](http://docs.google.com/javax/sql/RowSet.html#setObject(int,%20java.lang.Object,%20int,%20int)), [setObject](http://docs.google.com/javax/sql/RowSet.html#setObject(java.lang.String,%20java.lang.Object)), [setObject](http://docs.google.com/javax/sql/RowSet.html#setObject(java.lang.String,%20java.lang.Object,%20int)), [setObject](http://docs.google.com/javax/sql/RowSet.html#setObject(java.lang.String,%20java.lang.Object,%20int,%20int)), [setPassword](http://docs.google.com/javax/sql/RowSet.html#setPassword(java.lang.String)), [setQueryTimeout](http://docs.google.com/javax/sql/RowSet.html#setQueryTimeout(int)), [setReadOnly](http://docs.google.com/javax/sql/RowSet.html#setReadOnly(boolean)), [setRef](http://docs.google.com/javax/sql/RowSet.html#setRef(int,%20java.sql.Ref)), [setRowId](http://docs.google.com/javax/sql/RowSet.html#setRowId(int,%20java.sql.RowId)), [setRowId](http://docs.google.com/javax/sql/RowSet.html#setRowId(java.lang.String,%20java.sql.RowId)), [setShort](http://docs.google.com/javax/sql/RowSet.html#setShort(int,%20short)), [setShort](http://docs.google.com/javax/sql/RowSet.html#setShort(java.lang.String,%20short)), [setSQLXML](http://docs.google.com/javax/sql/RowSet.html#setSQLXML(int,%20java.sql.SQLXML)), [setSQLXML](http://docs.google.com/javax/sql/RowSet.html#setSQLXML(java.lang.String,%20java.sql.SQLXML)), [setString](http://docs.google.com/javax/sql/RowSet.html#setString(int,%20java.lang.String)), [setString](http://docs.google.com/javax/sql/RowSet.html#setString(java.lang.String,%20java.lang.String)), [setTime](http://docs.google.com/javax/sql/RowSet.html#setTime(int,%20java.sql.Time)), [setTime](http://docs.google.com/javax/sql/RowSet.html#setTime(int,%20java.sql.Time,%20java.util.Calendar)), [setTime](http://docs.google.com/javax/sql/RowSet.html#setTime(java.lang.String,%20java.sql.Time)), [setTime](http://docs.google.com/javax/sql/RowSet.html#setTime(java.lang.String,%20java.sql.Time,%20java.util.Calendar)), [setTimestamp](http://docs.google.com/javax/sql/RowSet.html#setTimestamp(int,%20java.sql.Timestamp)), [setTimestamp](http://docs.google.com/javax/sql/RowSet.html#setTimestamp(int,%20java.sql.Timestamp,%20java.util.Calendar)), [setTimestamp](http://docs.google.com/javax/sql/RowSet.html#setTimestamp(java.lang.String,%20java.sql.Timestamp)), [setTimestamp](http://docs.google.com/javax/sql/RowSet.html#setTimestamp(java.lang.String,%20java.sql.Timestamp,%20java.util.Calendar)), [setTransactionIsolation](http://docs.google.com/javax/sql/RowSet.html#setTransactionIsolation(int)), [setType](http://docs.google.com/javax/sql/RowSet.html#setType(int)), [setTypeMap](http://docs.google.com/javax/sql/RowSet.html#setTypeMap(java.util.Map)), [setURL](http://docs.google.com/javax/sql/RowSet.html#setURL(int,%20java.net.URL)), [setUrl](http://docs.google.com/javax/sql/RowSet.html#setUrl(java.lang.String)), [setUsername](http://docs.google.com/javax/sql/RowSet.html#setUsername(java.lang.String)) |

| **Methods inherited from interface java.sql.**[**ResultSet**](http://docs.google.com/java/sql/ResultSet.html) |
| --- |
| [absolute](http://docs.google.com/java/sql/ResultSet.html#absolute(int)), [afterLast](http://docs.google.com/java/sql/ResultSet.html#afterLast()), [beforeFirst](http://docs.google.com/java/sql/ResultSet.html#beforeFirst()), [cancelRowUpdates](http://docs.google.com/java/sql/ResultSet.html#cancelRowUpdates()), [clearWarnings](http://docs.google.com/java/sql/ResultSet.html#clearWarnings()), [close](http://docs.google.com/java/sql/ResultSet.html#close()), [deleteRow](http://docs.google.com/java/sql/ResultSet.html#deleteRow()), [findColumn](http://docs.google.com/java/sql/ResultSet.html#findColumn(java.lang.String)), [first](http://docs.google.com/java/sql/ResultSet.html#first()), [getArray](http://docs.google.com/java/sql/ResultSet.html#getArray(int)), [getArray](http://docs.google.com/java/sql/ResultSet.html#getArray(java.lang.String)), [getAsciiStream](http://docs.google.com/java/sql/ResultSet.html#getAsciiStream(int)), [getAsciiStream](http://docs.google.com/java/sql/ResultSet.html#getAsciiStream(java.lang.String)), [getBigDecimal](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(int)), [getBigDecimal](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(int,%20int)), [getBigDecimal](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(java.lang.String)), [getBigDecimal](http://docs.google.com/java/sql/ResultSet.html#getBigDecimal(java.lang.String,%20int)), [getBinaryStream](http://docs.google.com/java/sql/ResultSet.html#getBinaryStream(int)), [getBinaryStream](http://docs.google.com/java/sql/ResultSet.html#getBinaryStream(java.lang.String)), [getBlob](http://docs.google.com/java/sql/ResultSet.html#getBlob(int)), [getBlob](http://docs.google.com/java/sql/ResultSet.html#getBlob(java.lang.String)), [getBoolean](http://docs.google.com/java/sql/ResultSet.html#getBoolean(int)), [getBoolean](http://docs.google.com/java/sql/ResultSet.html#getBoolean(java.lang.String)), [getByte](http://docs.google.com/java/sql/ResultSet.html#getByte(int)), [getByte](http://docs.google.com/java/sql/ResultSet.html#getByte(java.lang.String)), [getBytes](http://docs.google.com/java/sql/ResultSet.html#getBytes(int)), [getBytes](http://docs.google.com/java/sql/ResultSet.html#getBytes(java.lang.String)), [getCharacterStream](http://docs.google.com/java/sql/ResultSet.html#getCharacterStream(int)), [getCharacterStream](http://docs.google.com/java/sql/ResultSet.html#getCharacterStream(java.lang.String)), [getClob](http://docs.google.com/java/sql/ResultSet.html#getClob(int)), [getClob](http://docs.google.com/java/sql/ResultSet.html#getClob(java.lang.String)), [getConcurrency](http://docs.google.com/java/sql/ResultSet.html#getConcurrency()), [getCursorName](http://docs.google.com/java/sql/ResultSet.html#getCursorName()), [getDate](http://docs.google.com/java/sql/ResultSet.html#getDate(int)), [getDate](http://docs.google.com/java/sql/ResultSet.html#getDate(int,%20java.util.Calendar)), [getDate](http://docs.google.com/java/sql/ResultSet.html#getDate(java.lang.String)), [getDate](http://docs.google.com/java/sql/ResultSet.html#getDate(java.lang.String,%20java.util.Calendar)), [getDouble](http://docs.google.com/java/sql/ResultSet.html#getDouble(int)), [getDouble](http://docs.google.com/java/sql/ResultSet.html#getDouble(java.lang.String)), [getFetchDirection](http://docs.google.com/java/sql/ResultSet.html#getFetchDirection()), [getFetchSize](http://docs.google.com/java/sql/ResultSet.html#getFetchSize()), [getFloat](http://docs.google.com/java/sql/ResultSet.html#getFloat(int)), [getFloat](http://docs.google.com/java/sql/ResultSet.html#getFloat(java.lang.String)), [getHoldability](http://docs.google.com/java/sql/ResultSet.html#getHoldability()), [getInt](http://docs.google.com/java/sql/ResultSet.html#getInt(int)), [getInt](http://docs.google.com/java/sql/ResultSet.html#getInt(java.lang.String)), [getLong](http://docs.google.com/java/sql/ResultSet.html#getLong(int)), [getLong](http://docs.google.com/java/sql/ResultSet.html#getLong(java.lang.String)), [getMetaData](http://docs.google.com/java/sql/ResultSet.html#getMetaData()), [getNCharacterStream](http://docs.google.com/java/sql/ResultSet.html#getNCharacterStream(int)), [getNCharacterStream](http://docs.google.com/java/sql/ResultSet.html#getNCharacterStream(java.lang.String)), [getNClob](http://docs.google.com/java/sql/ResultSet.html#getNClob(int)), [getNClob](http://docs.google.com/java/sql/ResultSet.html#getNClob(java.lang.String)), [getNString](http://docs.google.com/java/sql/ResultSet.html#getNString(int)), [getNString](http://docs.google.com/java/sql/ResultSet.html#getNString(java.lang.String)), [getObject](http://docs.google.com/java/sql/ResultSet.html#getObject(int)), [getObject](http://docs.google.com/java/sql/ResultSet.html#getObject(int,%20java.util.Map)), [getObject](http://docs.google.com/java/sql/ResultSet.html#getObject(java.lang.String)), [getObject](http://docs.google.com/java/sql/ResultSet.html#getObject(java.lang.String,%20java.util.Map)), [getRef](http://docs.google.com/java/sql/ResultSet.html#getRef(int)), [getRef](http://docs.google.com/java/sql/ResultSet.html#getRef(java.lang.String)), [getRow](http://docs.google.com/java/sql/ResultSet.html#getRow()), [getRowId](http://docs.google.com/java/sql/ResultSet.html#getRowId(int)), [getRowId](http://docs.google.com/java/sql/ResultSet.html#getRowId(java.lang.String)), [getShort](http://docs.google.com/java/sql/ResultSet.html#getShort(int)), [getShort](http://docs.google.com/java/sql/ResultSet.html#getShort(java.lang.String)), [getSQLXML](http://docs.google.com/java/sql/ResultSet.html#getSQLXML(int)), [getSQLXML](http://docs.google.com/java/sql/ResultSet.html#getSQLXML(java.lang.String)), [getStatement](http://docs.google.com/java/sql/ResultSet.html#getStatement()), [getString](http://docs.google.com/java/sql/ResultSet.html#getString(int)), [getString](http://docs.google.com/java/sql/ResultSet.html#getString(java.lang.String)), [getTime](http://docs.google.com/java/sql/ResultSet.html#getTime(int)), [getTime](http://docs.google.com/java/sql/ResultSet.html#getTime(int,%20java.util.Calendar)), [getTime](http://docs.google.com/java/sql/ResultSet.html#getTime(java.lang.String)), [getTime](http://docs.google.com/java/sql/ResultSet.html#getTime(java.lang.String,%20java.util.Calendar)), [getTimestamp](http://docs.google.com/java/sql/ResultSet.html#getTimestamp(int)), [getTimestamp](http://docs.google.com/java/sql/ResultSet.html#getTimestamp(int,%20java.util.Calendar)), [getTimestamp](http://docs.google.com/java/sql/ResultSet.html#getTimestamp(java.lang.String)), [getTimestamp](http://docs.google.com/java/sql/ResultSet.html#getTimestamp(java.lang.String,%20java.util.Calendar)), [getType](http://docs.google.com/java/sql/ResultSet.html#getType()), [getUnicodeStream](http://docs.google.com/java/sql/ResultSet.html#getUnicodeStream(int)), [getUnicodeStream](http://docs.google.com/java/sql/ResultSet.html#getUnicodeStream(java.lang.String)), [getURL](http://docs.google.com/java/sql/ResultSet.html#getURL(int)), [getURL](http://docs.google.com/java/sql/ResultSet.html#getURL(java.lang.String)), [getWarnings](http://docs.google.com/java/sql/ResultSet.html#getWarnings()), [insertRow](http://docs.google.com/java/sql/ResultSet.html#insertRow()), [isAfterLast](http://docs.google.com/java/sql/ResultSet.html#isAfterLast()), [isBeforeFirst](http://docs.google.com/java/sql/ResultSet.html#isBeforeFirst()), [isClosed](http://docs.google.com/java/sql/ResultSet.html#isClosed()), [isFirst](http://docs.google.com/java/sql/ResultSet.html#isFirst()), [isLast](http://docs.google.com/java/sql/ResultSet.html#isLast()), [last](http://docs.google.com/java/sql/ResultSet.html#last()), [moveToCurrentRow](http://docs.google.com/java/sql/ResultSet.html#moveToCurrentRow()), [moveToInsertRow](http://docs.google.com/java/sql/ResultSet.html#moveToInsertRow()), [next](http://docs.google.com/java/sql/ResultSet.html#next()), [previous](http://docs.google.com/java/sql/ResultSet.html#previous()), [refreshRow](http://docs.google.com/java/sql/ResultSet.html#refreshRow()), [relative](http://docs.google.com/java/sql/ResultSet.html#relative(int)), [rowDeleted](http://docs.google.com/java/sql/ResultSet.html#rowDeleted()), [rowInserted](http://docs.google.com/java/sql/ResultSet.html#rowInserted()), [rowUpdated](http://docs.google.com/java/sql/ResultSet.html#rowUpdated()), [setFetchDirection](http://docs.google.com/java/sql/ResultSet.html#setFetchDirection(int)), [setFetchSize](http://docs.google.com/java/sql/ResultSet.html#setFetchSize(int)), [updateArray](http://docs.google.com/java/sql/ResultSet.html#updateArray(int,%20java.sql.Array)), [updateArray](http://docs.google.com/java/sql/ResultSet.html#updateArray(java.lang.String,%20java.sql.Array)), [updateAsciiStream](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(int,%20java.io.InputStream)), [updateAsciiStream](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(int,%20java.io.InputStream,%20int)), [updateAsciiStream](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(int,%20java.io.InputStream,%20long)), [updateAsciiStream](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(java.lang.String,%20java.io.InputStream)), [updateAsciiStream](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(java.lang.String,%20java.io.InputStream,%20int)), [updateAsciiStream](http://docs.google.com/java/sql/ResultSet.html#updateAsciiStream(java.lang.String,%20java.io.InputStream,%20long)), [updateBigDecimal](http://docs.google.com/java/sql/ResultSet.html#updateBigDecimal(int,%20java.math.BigDecimal)), [updateBigDecimal](http://docs.google.com/java/sql/ResultSet.html#updateBigDecimal(java.lang.String,%20java.math.BigDecimal)), [updateBinaryStream](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(int,%20java.io.InputStream)), [updateBinaryStream](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(int,%20java.io.InputStream,%20int)), [updateBinaryStream](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(int,%20java.io.InputStream,%20long)), [updateBinaryStream](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(java.lang.String,%20java.io.InputStream)), [updateBinaryStream](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(java.lang.String,%20java.io.InputStream,%20int)), [updateBinaryStream](http://docs.google.com/java/sql/ResultSet.html#updateBinaryStream(java.lang.String,%20java.io.InputStream,%20long)), [updateBlob](http://docs.google.com/java/sql/ResultSet.html#updateBlob(int,%20java.sql.Blob)), [updateBlob](http://docs.google.com/java/sql/ResultSet.html#updateBlob(int,%20java.io.InputStream)), [updateBlob](http://docs.google.com/java/sql/ResultSet.html#updateBlob(int,%20java.io.InputStream,%20long)), [updateBlob](http://docs.google.com/java/sql/ResultSet.html#updateBlob(java.lang.String,%20java.sql.Blob)), [updateBlob](http://docs.google.com/java/sql/ResultSet.html#updateBlob(java.lang.String,%20java.io.InputStream)), [updateBlob](http://docs.google.com/java/sql/ResultSet.html#updateBlob(java.lang.String,%20java.io.InputStream,%20long)), [updateBoolean](http://docs.google.com/java/sql/ResultSet.html#updateBoolean(int,%20boolean)), [updateBoolean](http://docs.google.com/java/sql/ResultSet.html#updateBoolean(java.lang.String,%20boolean)), [updateByte](http://docs.google.com/java/sql/ResultSet.html#updateByte(int,%20byte)), [updateByte](http://docs.google.com/java/sql/ResultSet.html#updateByte(java.lang.String,%20byte)), [updateBytes](http://docs.google.com/java/sql/ResultSet.html#updateBytes(int,%20byte%5B%5D)), [updateBytes](http://docs.google.com/java/sql/ResultSet.html#updateBytes(java.lang.String,%20byte%5B%5D)), [updateCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(int,%20java.io.Reader)), [updateCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(int,%20java.io.Reader,%20int)), [updateCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(int,%20java.io.Reader,%20long)), [updateCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(java.lang.String,%20java.io.Reader)), [updateCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(java.lang.String,%20java.io.Reader,%20int)), [updateCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateCharacterStream(java.lang.String,%20java.io.Reader,%20long)), [updateClob](http://docs.google.com/java/sql/ResultSet.html#updateClob(int,%20java.sql.Clob)), [updateClob](http://docs.google.com/java/sql/ResultSet.html#updateClob(int,%20java.io.Reader)), [updateClob](http://docs.google.com/java/sql/ResultSet.html#updateClob(int,%20java.io.Reader,%20long)), [updateClob](http://docs.google.com/java/sql/ResultSet.html#updateClob(java.lang.String,%20java.sql.Clob)), [updateClob](http://docs.google.com/java/sql/ResultSet.html#updateClob(java.lang.String,%20java.io.Reader)), [updateClob](http://docs.google.com/java/sql/ResultSet.html#updateClob(java.lang.String,%20java.io.Reader,%20long)), [updateDate](http://docs.google.com/java/sql/ResultSet.html#updateDate(int,%20java.sql.Date)), [updateDate](http://docs.google.com/java/sql/ResultSet.html#updateDate(java.lang.String,%20java.sql.Date)), [updateDouble](http://docs.google.com/java/sql/ResultSet.html#updateDouble(int,%20double)), [updateDouble](http://docs.google.com/java/sql/ResultSet.html#updateDouble(java.lang.String,%20double)), [updateFloat](http://docs.google.com/java/sql/ResultSet.html#updateFloat(int,%20float)), [updateFloat](http://docs.google.com/java/sql/ResultSet.html#updateFloat(java.lang.String,%20float)), [updateInt](http://docs.google.com/java/sql/ResultSet.html#updateInt(int,%20int)), [updateInt](http://docs.google.com/java/sql/ResultSet.html#updateInt(java.lang.String,%20int)), [updateLong](http://docs.google.com/java/sql/ResultSet.html#updateLong(int,%20long)), [updateLong](http://docs.google.com/java/sql/ResultSet.html#updateLong(java.lang.String,%20long)), [updateNCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateNCharacterStream(int,%20java.io.Reader)), [updateNCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateNCharacterStream(int,%20java.io.Reader,%20long)), [updateNCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateNCharacterStream(java.lang.String,%20java.io.Reader)), [updateNCharacterStream](http://docs.google.com/java/sql/ResultSet.html#updateNCharacterStream(java.lang.String,%20java.io.Reader,%20long)), [updateNClob](http://docs.google.com/java/sql/ResultSet.html#updateNClob(int,%20java.sql.NClob)), [updateNClob](http://docs.google.com/java/sql/ResultSet.html#updateNClob(int,%20java.io.Reader)), [updateNClob](http://docs.google.com/java/sql/ResultSet.html#updateNClob(int,%20java.io.Reader,%20long)), [updateNClob](http://docs.google.com/java/sql/ResultSet.html#updateNClob(java.lang.String,%20java.sql.NClob)), [updateNClob](http://docs.google.com/java/sql/ResultSet.html#updateNClob(java.lang.String,%20java.io.Reader)), [updateNClob](http://docs.google.com/java/sql/ResultSet.html#updateNClob(java.lang.String,%20java.io.Reader,%20long)), [updateNString](http://docs.google.com/java/sql/ResultSet.html#updateNString(int,%20java.lang.String)), [updateNString](http://docs.google.com/java/sql/ResultSet.html#updateNString(java.lang.String,%20java.lang.String)), [updateNull](http://docs.google.com/java/sql/ResultSet.html#updateNull(int)), [updateNull](http://docs.google.com/java/sql/ResultSet.html#updateNull(java.lang.String)), [updateObject](http://docs.google.com/java/sql/ResultSet.html#updateObject(int,%20java.lang.Object)), [updateObject](http://docs.google.com/java/sql/ResultSet.html#updateObject(int,%20java.lang.Object,%20int)), [updateObject](http://docs.google.com/java/sql/ResultSet.html#updateObject(java.lang.String,%20java.lang.Object)), [updateObject](http://docs.google.com/java/sql/ResultSet.html#updateObject(java.lang.String,%20java.lang.Object,%20int)), [updateRef](http://docs.google.com/java/sql/ResultSet.html#updateRef(int,%20java.sql.Ref)), [updateRef](http://docs.google.com/java/sql/ResultSet.html#updateRef(java.lang.String,%20java.sql.Ref)), [updateRow](http://docs.google.com/java/sql/ResultSet.html#updateRow()), [updateRowId](http://docs.google.com/java/sql/ResultSet.html#updateRowId(int,%20java.sql.RowId)), [updateRowId](http://docs.google.com/java/sql/ResultSet.html#updateRowId(java.lang.String,%20java.sql.RowId)), [updateShort](http://docs.google.com/java/sql/ResultSet.html#updateShort(int,%20short)), [updateShort](http://docs.google.com/java/sql/ResultSet.html#updateShort(java.lang.String,%20short)), [updateSQLXML](http://docs.google.com/java/sql/ResultSet.html#updateSQLXML(int,%20java.sql.SQLXML)), [updateSQLXML](http://docs.google.com/java/sql/ResultSet.html#updateSQLXML(java.lang.String,%20java.sql.SQLXML)), [updateString](http://docs.google.com/java/sql/ResultSet.html#updateString(int,%20java.lang.String)), [updateString](http://docs.google.com/java/sql/ResultSet.html#updateString(java.lang.String,%20java.lang.String)), [updateTime](http://docs.google.com/java/sql/ResultSet.html#updateTime(int,%20java.sql.Time)), [updateTime](http://docs.google.com/java/sql/ResultSet.html#updateTime(java.lang.String,%20java.sql.Time)), [updateTimestamp](http://docs.google.com/java/sql/ResultSet.html#updateTimestamp(int,%20java.sql.Timestamp)), [updateTimestamp](http://docs.google.com/java/sql/ResultSet.html#updateTimestamp(java.lang.String,%20java.sql.Timestamp)), [wasNull](http://docs.google.com/java/sql/ResultSet.html#wasNull()) |

| **Methods inherited from interface java.sql.**[**Wrapper**](http://docs.google.com/java/sql/Wrapper.html) |
| --- |
| [isWrapperFor](http://docs.google.com/java/sql/Wrapper.html#isWrapperFor(java.lang.Class)), [unwrap](http://docs.google.com/java/sql/Wrapper.html#unwrap(java.lang.Class)) |

| **Methods inherited from interface javax.sql.rowset.**[**Joinable**](http://docs.google.com/javax/sql/rowset/Joinable.html) |
| --- |
| [getMatchColumnIndexes](http://docs.google.com/javax/sql/rowset/Joinable.html#getMatchColumnIndexes()), [getMatchColumnNames](http://docs.google.com/javax/sql/rowset/Joinable.html#getMatchColumnNames()), [setMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int)), [setMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(int%5B%5D)), [setMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(java.lang.String)), [setMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#setMatchColumn(java.lang.String%5B%5D)), [unsetMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(int)), [unsetMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(int%5B%5D)), [unsetMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(java.lang.String)), [unsetMatchColumn](http://docs.google.com/javax/sql/rowset/Joinable.html#unsetMatchColumn(java.lang.String%5B%5D)) |

| **Field Detail** |
| --- |

### PUBLIC\_XML\_SCHEMA

static final [String](http://docs.google.com/java/lang/String.html) **PUBLIC\_XML\_SCHEMA**

The public identifier for the XML Schema definition that defines the XML tags and their valid values for a WebRowSet implementation.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.sql.rowset.WebRowSet.PUBLIC_XML_SCHEMA)

### SCHEMA\_SYSTEM\_ID

static final [String](http://docs.google.com/java/lang/String.html) **SCHEMA\_SYSTEM\_ID**

The URL for the XML Schema definition file that defines the XML tags and their valid values for a WebRowSet implementation.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.sql.rowset.WebRowSet.SCHEMA_SYSTEM_ID)

| **Method Detail** |
| --- |

### readXml

void **readXml**([Reader](http://docs.google.com/java/io/Reader.html) reader)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Reads a WebRowSet object in its XML format from the given Reader object.

**Parameters:**reader - the java.io.Reader stream from which this WebRowSet object will be populated **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs

### readXml

void **readXml**([InputStream](http://docs.google.com/java/io/InputStream.html) iStream)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Reads a stream based XML input to populate this WebRowSet object.

**Parameters:**iStream - the java.io.InputStream from which this WebRowSet object will be populated **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a data source access error occurs [IOException](http://docs.google.com/java/io/IOException.html) - if an IO exception occurs

### writeXml

void **writeXml**([ResultSet](http://docs.google.com/java/sql/ResultSet.html) rs,  
 [Writer](http://docs.google.com/java/io/Writer.html) writer)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Populates this WebRowSet object with the contents of the given ResultSet object and writes its data, properties, and metadata to the given Writer object in XML format.

NOTE: The WebRowSet cursor may be moved to write out the contents to the XML data source. If implemented in this way, the cursor **must** be returned to its position just prior to the writeXml() call.

**Parameters:**rs - the ResultSet object with which to populate this WebRowSet objectwriter - the java.io.Writer object to write to. **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if an error occurs writing out the rowset contents in XML format

### writeXml

void **writeXml**([ResultSet](http://docs.google.com/java/sql/ResultSet.html) rs,  
 [OutputStream](http://docs.google.com/java/io/OutputStream.html) oStream)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Populates this WebRowSet object with the contents of the given ResultSet object and writes its data, properties, and metadata to the given OutputStream object in XML format.

NOTE: The WebRowSet cursor may be moved to write out the contents to the XML data source. If implemented in this way, the cursor **must** be returned to its position just prior to the writeXml() call.

**Parameters:**rs - the ResultSet object with which to populate this WebRowSet objectoStream - the java.io.OutputStream to write to **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a data source access error occurs [IOException](http://docs.google.com/java/io/IOException.html) - if a IO exception occurs

### writeXml

void **writeXml**([Writer](http://docs.google.com/java/io/Writer.html) writer)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Writes the data, properties, and metadata for this WebRowSet object to the given Writer object in XML format.

**Parameters:**writer - the java.io.Writer stream to write to **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if an error occurs writing out the rowset contents to XML

### writeXml

void **writeXml**([OutputStream](http://docs.google.com/java/io/OutputStream.html) oStream)  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Writes the data, properties, and metadata for this WebRowSet object to the given OutputStream object in XML format.

**Parameters:**oStream - the java.io.OutputStream stream to write to **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a data source access error occurs [IOException](http://docs.google.com/java/io/IOException.html) - if a IO exception occurs

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/WebRowSet.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/sql/rowset/RowSetWarning.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?javax/sql/rowset/WebRowSet.html)    [**NO FRAMES**](http://docs.google.com/WebRowSet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#26in1rg) | CONSTR | [METHOD](#1ksv4uv) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).