| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PooledConnection.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/DataSource.html)   [**NEXT CLASS**](http://docs.google.com/javax/sql/RowSet.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sql/PooledConnection.html)    [**NO FRAMES**](http://docs.google.com/PooledConnection.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **javax.sql**

Interface PooledConnection

**All Known Subinterfaces:** [XAConnection](http://docs.google.com/javax/sql/XAConnection.html)

public interface **PooledConnection**

An object that provides hooks for connection pool management. A PooledConnection object represents a physical connection to a data source. The connection can be recycled rather than being closed when an application is finished with it, thus reducing the number of connections that need to be made.

An application programmer does not use the PooledConnection interface directly; rather, it is used by a middle tier infrastructure that manages the pooling of connections.

When an application calls the method DataSource.getConnection, it gets back a Connection object. If connection pooling is being done, that Connection object is actually a handle to a PooledConnection object, which is a physical connection.

The connection pool manager, typically the application server, maintains a pool of PooledConnection objects. If there is a PooledConnection object available in the pool, the connection pool manager returns a Connection object that is a handle to that physical connection. If no PooledConnection object is available, the connection pool manager calls the ConnectionPoolDataSource method getPoolConnection to create a new physical connection. The JDBC driver implementing ConnectionPoolDataSource creates a new PooledConnection object and returns a handle to it.

When an application closes a connection, it calls the Connection method close. When connection pooling is being done, the connection pool manager is notified because it has registered itself as a ConnectionEventListener object using the ConnectionPool method addConnectionEventListener. The connection pool manager deactivates the handle to the PooledConnection object and returns the PooledConnection object to the pool of connections so that it can be used again. Thus, when an application closes its connection, the underlying physical connection is recycled rather than being closed.

The physical connection is not closed until the connection pool manager calls the PooledConnection method close. This method is generally called to have an orderly shutdown of the server or if a fatal error has made the connection unusable.

A connection pool manager is often also a statement pool manager, maintining a pool of PreparedStatement objects. When an application closes a prepared statement, it calls the PreparedStatement method close. When Statement pooling is being done, the pool manager is notified because it has registered itself as a StatementEventListener object using the ConnectionPool method addStatementEventListener. Thus, when an application closes its PreparedStatement, the underlying prepared statement is recycled rather than being closed.

**Since:** 1.4

| **Method Summary** | |
| --- | --- |
| void | [**addConnectionEventListener**](http://docs.google.com/javax/sql/PooledConnection.html#addConnectionEventListener(javax.sql.ConnectionEventListener))([ConnectionEventListener](http://docs.google.com/javax/sql/ConnectionEventListener.html) listener)            Registers the given event listener so that it will be notified when an event occurs on this PooledConnection object. |
| void | [**addStatementEventListener**](http://docs.google.com/javax/sql/PooledConnection.html#addStatementEventListener(javax.sql.StatementEventListener))([StatementEventListener](http://docs.google.com/javax/sql/StatementEventListener.html) listener)            Registers a StatementEventListener with this PooledConnection object. |
| void | [**close**](http://docs.google.com/javax/sql/PooledConnection.html#close())()            Closes the physical connection that this PooledConnection object represents. |
| [Connection](http://docs.google.com/java/sql/Connection.html) | [**getConnection**](http://docs.google.com/javax/sql/PooledConnection.html#getConnection())()            Creates and returns a Connection object that is a handle for the physical connection that this PooledConnection object represents. |
| void | [**removeConnectionEventListener**](http://docs.google.com/javax/sql/PooledConnection.html#removeConnectionEventListener(javax.sql.ConnectionEventListener))([ConnectionEventListener](http://docs.google.com/javax/sql/ConnectionEventListener.html) listener)            Removes the given event listener from the list of components that will be notified when an event occurs on this PooledConnection object. |
| void | [**removeStatementEventListener**](http://docs.google.com/javax/sql/PooledConnection.html#removeStatementEventListener(javax.sql.StatementEventListener))([StatementEventListener](http://docs.google.com/javax/sql/StatementEventListener.html) listener)            Removes the specified StatementEventListener from the list of components that will be notified when the driver detects that a PreparedStatement has been closed or is invalid. |

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

### getConnection

[Connection](http://docs.google.com/java/sql/Connection.html) **getConnection**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Creates and returns a Connection object that is a handle for the physical connection that this PooledConnection object represents. The connection pool manager calls this method when an application has called the method DataSource.getConnection and there are no PooledConnection objects available. See the [interface description](http://docs.google.com/javax/sql/PooledConnection.html) for more information.

**Returns:**a Connection object that is a handle to this PooledConnection object **Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs SQLFeatureNotSupportedException - if the JDBC driver does not support this method**Since:** 1.4

### close

void **close**()  
 throws [SQLException](http://docs.google.com/java/sql/SQLException.html)

Closes the physical connection that this PooledConnection object represents. An application never calls this method directly; it is called by the connection pool module, or manager.

See the [interface description](http://docs.google.com/javax/sql/PooledConnection.html) for more information.

**Throws:** [SQLException](http://docs.google.com/java/sql/SQLException.html) - if a database access error occurs SQLFeatureNotSupportedException - if the JDBC driver does not support this method**Since:** 1.4

### addConnectionEventListener

void **addConnectionEventListener**([ConnectionEventListener](http://docs.google.com/javax/sql/ConnectionEventListener.html) listener)

Registers the given event listener so that it will be notified when an event occurs on this PooledConnection object.

**Parameters:**listener - a component, usually the connection pool manager, that has implemented the ConnectionEventListener interface and wants to be notified when the connection is closed or has an error**See Also:**[removeConnectionEventListener(javax.sql.ConnectionEventListener)](http://docs.google.com/javax/sql/PooledConnection.html#removeConnectionEventListener(javax.sql.ConnectionEventListener))

### removeConnectionEventListener

void **removeConnectionEventListener**([ConnectionEventListener](http://docs.google.com/javax/sql/ConnectionEventListener.html) listener)

Removes the given event listener from the list of components that will be notified when an event occurs on this PooledConnection object.

**Parameters:**listener - a component, usually the connection pool manager, that has implemented the ConnectionEventListener interface and been registered with this PooledConnection object as a listener**See Also:**[addConnectionEventListener(javax.sql.ConnectionEventListener)](http://docs.google.com/javax/sql/PooledConnection.html#addConnectionEventListener(javax.sql.ConnectionEventListener))

### addStatementEventListener

void **addStatementEventListener**([StatementEventListener](http://docs.google.com/javax/sql/StatementEventListener.html) listener)

Registers a StatementEventListener with this PooledConnection object. Components that wish to be notified when PreparedStatements created by the connection are closed or are detected to be invalid may use this method to register a StatementEventListener with this PooledConnection object.

**Parameters:**listener - an component which implements the StatementEventListener interface that is to be registered with this PooledConnection object

**Since:** 1.6

### removeStatementEventListener

void **removeStatementEventListener**([StatementEventListener](http://docs.google.com/javax/sql/StatementEventListener.html) listener)

Removes the specified StatementEventListener from the list of components that will be notified when the driver detects that a PreparedStatement has been closed or is invalid.

**Parameters:**listener - the component which implements the StatementEventListener interface that was previously registered with this PooledConnection object

**Since:** 1.6

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PooledConnection.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/DataSource.html)   [**NEXT CLASS**](http://docs.google.com/javax/sql/RowSet.html) | [**FRAMES**](http://docs.google.com/index.html?javax/sql/PooledConnection.html)    [**NO FRAMES**](http://docs.google.com/PooledConnection.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

[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).