



[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS NEXT CLASS

[FRAMES](#) [NO FRAMES](#) [All Classes](#)SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#)DETAIL: FIELD | CONSTR | [METHOD](#)

de.dlr.xisum.mysql

Interface DataBaseConnector

All Known Implementing Classes:[DataBaseConnectorImpl](#)public interface **DataBaseConnector**

Interface for the Implmentation of the DataBaseConnector.

Version:

\$Revision: 1.1 \$

Author:

Thijs Metsch

Method Summary

void	executeDeleteQuery (java.lang.String query)
void	executeInsertQuery (java.lang.String query)
java.sql.ResultSet	executeSelectQuery (java.lang.String query)
void	executeUpdateQuery (java.lang.String query)

Method Detail

executeSelectQuerypublic java.sql.ResultSet **executeSelectQuery**(java.lang.String query)

throws java.sql.SQLException

Throws:

java.sql.SQLException

executeUpdateQuery

```
public void executeUpdateQuery(java.lang.String query)
    throws java.sql.SQLException
```

Throws:

java.sql.SQLException

executeDeleteQuery

```
public void executeDeleteQuery(java.lang.String query)
    throws java.sql.SQLException
```

Throws:

java.sql.SQLException

executeInsertQuery

```
public void executeInsertQuery(java.lang.String query)
    throws java.sql.SQLException
```

Throws:

java.sql.SQLException

[Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

de.dlr.xisum.mysql

Class DataBaseConnectorImpl

java.lang.Object

└ de.dlr.xisum.mysql.DataBaseConnectorImpl

All Implemented Interfaces:

[DataBaseConnector](#)

public class **DataBaseConnectorImpl**

extends java.lang.Object

implements [DataBaseConnector](#)

Implementation of a JDBC connector.

Version:

\$Revision: 1.2 \$

Author:

Thijs Metsch

Constructor Summary

[DataBaseConnectorImpl](#)(java.lang.String dataBaseUrl)

Connect to the Database

Method Summary

void	executeDeleteQuery (java.lang.String query) Deletes an Record form the Database.
void	executeInsertQuery (java.lang.String query) Inserts an Record into the Database .
java.sql.ResultSet	executeSelectQuery (java.lang.String query) Executes an Query which Selects several items from

	the Database.
void	executeUpdateQuery (java.lang.String query) Executes an Query which updates an Record from the Database.

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

DataBaseConnectorImpl

```
public DataBaseConnectorImpl( java.lang.String dataBaseUrl )
```

Connect to the Database

Parameters:

dataBaseUrl - Name of the Database

Method Detail

executeSelectQuery

```
public java.sql.ResultSet executeSelectQuery( java.lang.String query)  
throws java.sql.SQLException
```

Executes an Query which Selects several items from the Database.

Specified by:

[executeSelectQuery](#) in interface [DataBaseConnector](#)

Parameters:

query - The SELECT Query.

Returns:

Throws:

java.sql.SQLException

executeUpdateQuery

```
public void executeUpdateQuery(java.lang.String query)
    throws java.sql.SQLException
```

Executes an Query which updates an Record from the Database.

Specified by:

[executeUpdateQuery](#) in interface [DataBaseConnector](#)

Parameters:

query - The UPDATE Query.

Throws:

java.sql.SQLException

executeDeleteQuery

```
public void executeDeleteQuery(java.lang.String query)
    throws java.sql.SQLException
```

Deletes an Record form the Database.

Specified by:

[executeDeleteQuery](#) in interface [DataBaseConnector](#)

Parameters:

query - DELETE Query.

Throws:

java.sql.SQLException

executeInsertQuery

```
public void executeInsertQuery(java.lang.String query)
    throws java.sql.SQLException
```

Inserts an Record into the Database .

Specified by:

[executeInsertQuery](#) in interface [DataBaseConnector](#)

Parameters:

query - INSERT Query.

Throws:

java.sql.SQLException

[Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

PREV CLASS NEXT CLASS

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

```

/*
 * DataBaseConnector.java
 *
 * Interface for the Implmentation of the DataBaseConnector.
 *
 * Created: 12.11.2003 Thijs Metsch <thijs.metsch@dlr.de>
 *
 * Version: $Id: $
 *
 * Copyright (C) 1998-2003 DLR/SISTEC, Germany
 *
 * All rights reserved
 *
 * http://www.dlr.de/tent
 */
package de.dlr.xisum.mysql;

import java.sql.ResultSet;
import java.sql.SQLException;

/**
 * Interface for the Implmentation of the DataBaseConnector.
 *
 * @version $Revision: 1.1 $
 * @author Thijs Metsch
 */
public interface DataBaseConnector {

    public ResultSet executeSelectQuery(String query) throws SQLException;
    public void executeUpdateQuery(String query) throws SQLException;
    public void executeDeleteQuery(String query) throws SQLException;
    public void executeInsertQuery(String query) throws SQLException;

}

```



```

/*
 * DataBaseConnectorImpl.java
 *
 * Implementation of a JDBC connector.
 *
 * Created: 12.10.2003 Thijs Metsch <thijs.metsch@dlr.de>
 *
 * Version: $Id: MySQL.java,v 1.1 2003/10/28 15:20:09 befreax Exp $
 *
 * Copyright (C) 1998-2003 DLR/SISTEC, Germany
 *
 * All rights reserved
 *
 * http://www.dlr.de/tent
 */
package de.dlr.xisum.mysql;

import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

/**
 * Implementation of a JDBC connector.
 *
 * @version $Revision: 1.2 $
 * @author Thijs Metsch
 */
public class DataBaseConnectorImpl implements DataBaseConnector {
    Statement statement;

    /**
     * Connect to the Database
     *
     * @param dataBaseUrl Name of the Database
     */
    public DataBaseConnectorImpl(String dataBaseUrl) {
        try {
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            java.sql.Connection conn = DriverManager.getConnection(dataBaseUrl);
            statement = conn.createStatement();
        }
        catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    /**
     * Executes an Query which Selects several items from the Database.
     *
     * @param query The <code>SELECT</code> Query.
     * @return
     * @throws SQLException
     */
    public ResultSet executeSelectQuery(String query) throws SQLException {
        return statement.executeQuery(query);
    }

    /**
     * Executes an Query which updates an Record from the Database.
     *
     * @param query The <code>UPDATE</code> Query.
     */

```

```
    * @throws SQLException
    */
public void executeUpdateQuery(String query) throws SQLException {
    statement.executeUpdate(query);
}

/**
 * Deletes an Record form the Database.
 *
 * @param query <code>DELETE</code> Query.
 * @throws SQLException
 */
public void executeDeleteQuery(String query) throws SQLException {
    statement.execute(query);
}

/**
 * Inserts an Record into the Database .
 *
 * @param query <code>INSERT</code> Query.
 * @throws SQLException
 */
public void executeInsertQuery(String query) throws SQLException {
    statement.execute(query);
}
}
```