

Class MySQLDB

java.lang.Object  
MySQLDB

```
public class MySQLDB
extends java.lang.Object
```

Class for interacting with the database

Field Summary

Fields	
Modifier and Type	Field and Description
private java.sql.Connection	<code>conn</code>
private static java.lang.String	<code>password</code>
private static java.lang.String	<code>uri</code>
private static java.lang.String	<code>user</code>

Constructor Summary

Constructors	
Constructor and Description	
<code>MySQLDB()</code>	Instantiates the database and connects to it

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method and Description	
boolean	<code>close()</code> closes database	
boolean	<code>connect()</code> closes opens, then connects to the database	
boolean	<code>endTrans()</code> Ends the transaction by committing changes and turning auto commit back on	
java.sql.ResultSet	<code>getData(java.sql.PreparedStatement stmt)</code> Get data from database	
java.sql.ResultSet	<code>getData(java.lang.String sqlString)</code> get data from database using a regular query.	
java.sql.PreparedStatement	<code>PrepStmt(java.lang.String sqlQuery)</code> Prepares statements	
void	<code>rollbackTrans()</code> Rolls back the transaction if a failure occurs	
int	<code>setData(java.sql.PreparedStatement stmt)</code> Used to change data in the database using a prepared statement	
int	<code>setData(java.lang.String sqlString)</code> Used to update data in database, does not use prepared statements	
boolean	<code>startTrans()</code> Starts a transaction by turning off auto commit	

Methods inherited from class java.lang.Object

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

Field Detail

uri

private static final java.lang.String uri

See Also:  
Constant Field Values

user

private static final java.lang.String user

See Also:  
Constant Field Values

password

private static final java.lang.String password

See Also:  
Constant Field Values

conn

private java.sql.Connection conn

Constructor Detail

MySQLDB

public MySQLDB()

Instantiates the database and connects to it

Method Detail

connect

public boolean connect()

closes opens, then connects to the database

Returns:  
boolean

close

public boolean close()

closes database

Returns:  
boolean

getData

public java.sql.ResultSet getData(java.sql.PreparedStatement stmt)

Get data from database

Parameters:  
stmt - the prepared statement to execute

Returns:  
the resultset from the database

getData

```
public java.sql.ResultSet getData(java.lang.String sqlString)
```

get data from database using a regular query. DOES NOT CHECK FOR SQL INJECTION ASSUMES query is clean (precondition)

**Parameters:**

sqlString - the create, update, or delete sql string

**Returns:**

ResultSet of query

### setData

```
public int setData(java.sql.PreparedStatement stmt)
    throws DLEException
```

Used to change data in the database using a prepared statement

**Parameters:**

stmt - the prepared statment to execute

**Returns:**

the amount of records changed

**Throws:**

DLEException

### setData

```
public int setData(java.lang.String sqlString)
    throws DLEException
```

Used to update data in database, does not use prepared statements

**Parameters:**

sqlString - the update sql string

**Throws:**

DLEException

### startTrans

```
public boolean startTrans()
```

Starts a transaction by turning off auto commit

**Returns:**

boolean

### endTrans

```
public boolean endTrans()
```

Ends the transaction by committing changes and turning auto commit back on

**Returns:**

boolean

### rollbackTrans

```
public void rollbackTrans()
    throws java.sql.SQLException
```

Rolls back the transaction if a failure occurs

**Throws:**

java.sql.SQLException

### PrepStmt

```
public java.sql.PreparedStatement PrepStmt(java.lang.String sqlQuery)
    throws DLEException
```

Prepares statements

**Parameters:**

sqlQuery - String

**Returns:**

PreparedStatement

**Throws:**

SQLException