***How to Connect to database using JDBC?***

The below are the Steps to Connect to database, before proceeding, you need to have MySQL Connector. You can download from here [Download MySQL Connector Jar](https://dev.mysql.com/downloads/connector/j/5.0.html) and add it the build path as we add selenium webdriver jar.

1. Load and Registering the Driver  
2. Establishing Connection.  
3. Creating Statement Object  
4. Execute the Statement  
5. Closing the connection.

*1. Load and Register the Driver:*

For registering the Driver we Load the Driver class using forName() method.

forName() is the static factory method which is present in predefined class called "Class". This method loads the class which is mentioned as parameter.

Class.forName("com.mysql.jdbc.Driver");// **class**.**forName** **load** **the** **Driver** **class**

Internally this Driver class will register the driver by using static method called registerDriver().

*2. Establishing Connection:*

For establishing connection with database we call static method called getConnection(...) present in DriverManager Class. This method contains three arguments of string type. i.e., url, username and password

DriverManager.getConnection("jdbc:mysql://localhost:3306/Employee","root","root");

URL contains "jdbc(main protocol):mysql(sub protocol for mySql)://localhost:3306(sub name for mysql (host:prot))/Employee(database)" and this method return type is Connection Object ie.,

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/Employee","root","root");

*3. Creating Statement Object:*

For creating statement object we need to call a method called createStatement() which is present in Connection Interface.

con.createStatement();

And this method returns Statement object and it is no argument method.

Statement st= con.createStatement();

*4.Executing Queries:*

For executing queries there are different methods present in Statement Interface for retrieving records and for updating records.

**Retrieving records:**  
for executing select queries(for fetching records) we call a method called executeQuery(String qry) by taking string as parameter.

st.executeQuery("**Select** \* **from** Employee");

This method returns ResultSet object.

Resultset rs= st.executeQuery("Select \* from Employee");// once executeQuery() executes the query **and** stores the records **in** to ResultSet object.

Now we need to get the records from ResultSet object. To access the resultset object it uses a method called next() which presents in ResultSet Interface.

By default Resultset reference 'rs' points to before first row. it moves rs to next row and returns true. When it returns true we retrieve the data in first row. next() returns false when rs points to after the last row. this next() will repeats the execution using while loop till it returns false.

To get the data from rows we use getXxx(..) taking string or integer as parameters. Here integer means column position and string means column name of the record. xxx indicates primitive datatypes or string object.

**while**(rs.next()) {

int EmpId= rs.getInt("EmpId");

**String** EmpName= rs.getString("EmpName");

**String** EmpAddress=rs.getString(3);

Double EmpSal= rs.getDouble(4);

**String** EmpDept=rs.getString("EmpDept");

System.out.println(EmpId+"\t"+EmpName+"\t"+EmpAddress+"\t"+EmpSal+"\t"+EmpDept);

}

**Actual Example :**

**import** org.testng.annotations.AfterTest;

**import** org.testng.annotations.BeforeTest;

**import** org.testng.annotations.Test;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.Statement;

**public** **class** SeleniumDatabaseTesting {

// Connection object

**static** Connection *con* = **null**;

// Statement object

**private** **static** Statement *stmt*;

// Constant for Database URL

**public** **static** String *DB\_URL* = "jdbc:mysql://localhost/Testdata";

//Database Username

**public** **static** String *DB\_USER* = "your\_user";

// Database Password

**public** **static** String *DB\_PASSWORD* = "your\_password";

@BeforeTest

**public** **void** setUp() **throws** Exception {

**try**{

// Database connection

String dbClass = "com.mysql.cj.jdbc.Driver";

Class.*forName*(dbClass).~~newInstance~~();

// Get connection to DB

Connection con = DriverManager.getConnection(*DB\_URL*, *DB\_USER*, *DB\_PASSWORD*);

// Statement object to send the SQL statement to the Database

*stmt* = con.createStatement();

}

**catch** (Exception e)

{

e.printStackTrace();

}

}

@Test

**public** **void** test() {

**try**{

String query = "select \* from testingdata";

// Get the contents of userinfo table from DB

ResultSet res = *stmt*.executeQuery(query);

// Print the result untill all the records are printed

// res.next() returns true if there is any next record else returns false

**while** (res.next())

{

System.***out***.print(res.getString(1));

System.***out***.print(" " + res.getString(2));

System.***out***.print(" " + res.getString(3));

System.***out***.println(" " + res.getString(4));

}

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

@AfterTest

**public** **void** tearDown() **throws** Exception {

// Close DB connection

**if** (*con* != **null**) {

*con*.close();

}

}

}