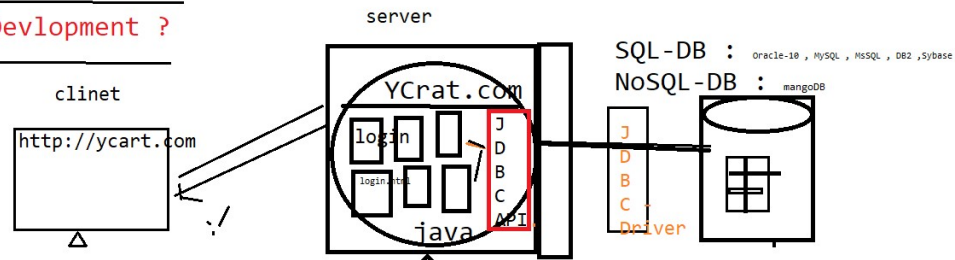


JDBC [Java Data base Connectivity]

What is JDBC ?

It's a J2AEE API , which is used to connect to any data base from the java code

Why JDBC Development ?



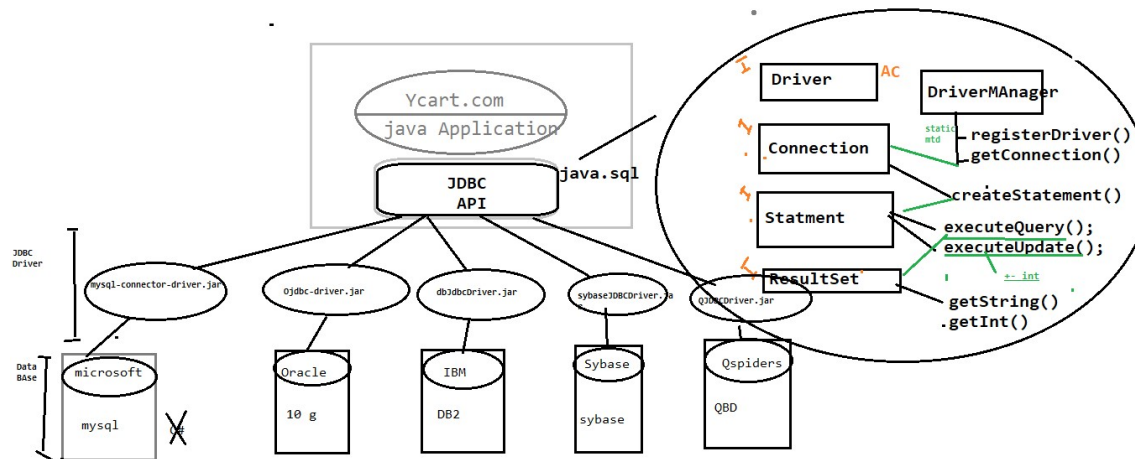
Every Application required physical database to store the data in the form table

Why JDBC in Development ?

JDBC helps java program to store / retrieve data from the physical database

In Real time, every application required data base to store/retrieve the user information in data base

What is JDBC Drivers & JDBC API ?



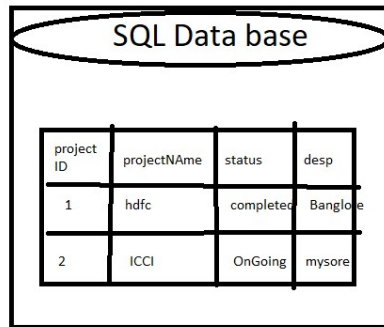
JDBC API :its an API's available in java.sql package , which is collection interfaces with abstract method & class , & it's database independent

JDBC Driver : its implementation class of JDBC API provided by data base vender . it's a db dependent

There are types of database

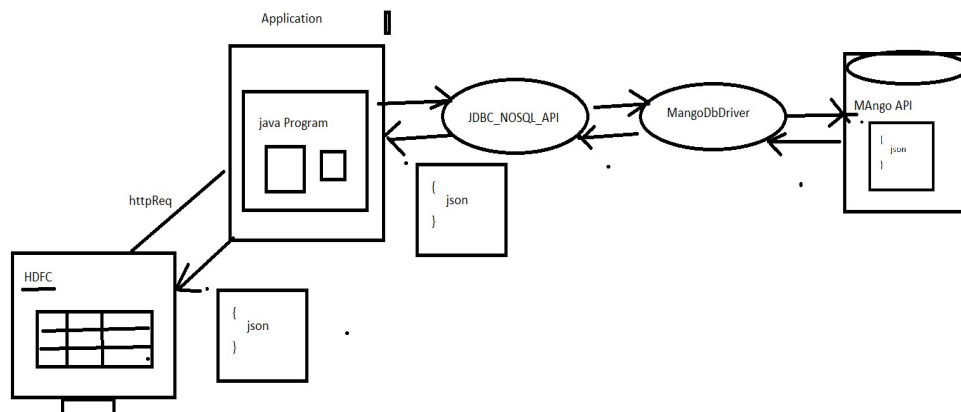
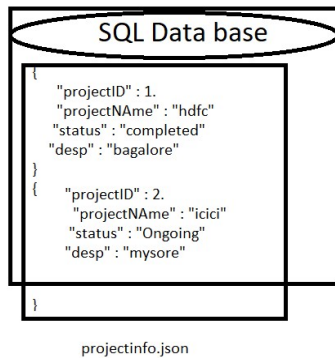
1. SQL Database

- ⇒ Data will be stored in form of table
EG :mysql , MsSql , Oracle 11 g , DB2 ,



2. No SQL Database

- ⇒ Data will be stored in the from JSON format
EG : Mango db , Casandra , NoSQL

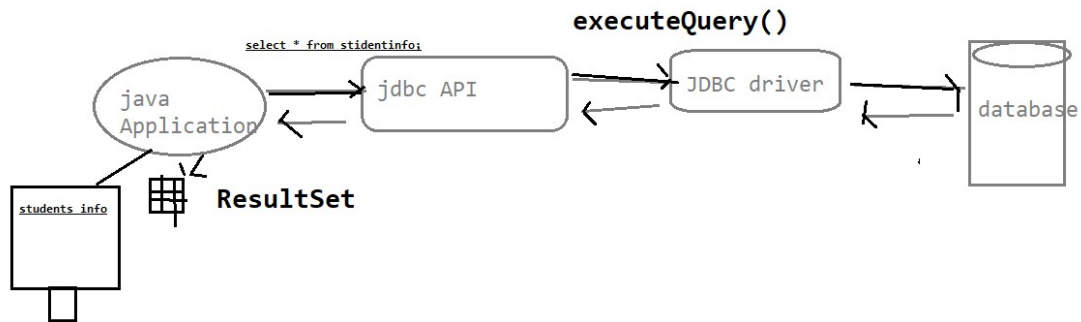


Dependency code to get MYSQL drivers

```
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.27</version>
</dependency>
```

There are two types Query ?

Select Query



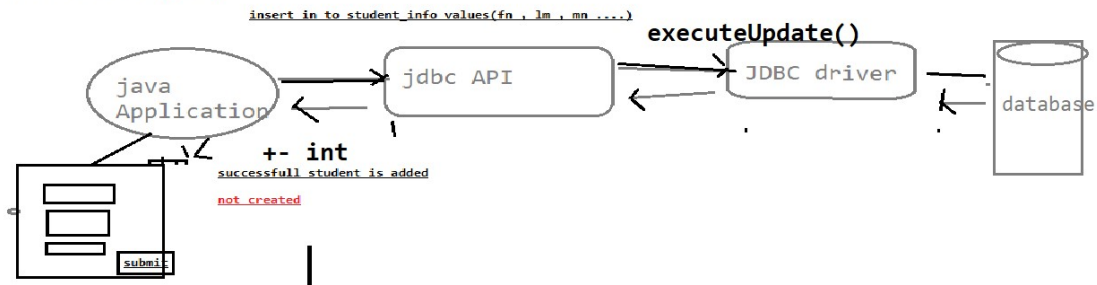
```

1 package practice;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.ResultSet;
5 import java.sql.Statement;
6 import com.mysql.jdbc.Driver;
7 public class SelectQuery_SampleCode {
8     public static void main(String[] args) throws Throwable {
9         Connection conn = null;
10        try {
11            Driver driverRef = new Driver();
12            /* step 1 : load /register *mysql *the databade */
13            DriverManager.registerDriver(driverRef);
14
15            /* step 2 : connect to db */
16            conn= DriverManager.getConnection("jdbc:mysql://localhost:3306/projects", "root", "root");
17            System.out.println("connection is done");
18            /* step 3 : create query statement */
19            Statement stat= conn.createStatement();
20            String query = "select * from projec";
21
22            /* step 4 : execute the Query */
23            ResultSet resultset = stat.executeQuery(query);
24            while (resultset.next()) {
25                System.out.println(resultset.getString(1) + "\t" + resultset.getString(2) + "\t"
26                    + resultset.getString(3) + "\t" + resultset.getString(4));
27            }
28        } catch (Exception e) {
29        } finally {
30            /*step 5: close the connection */
31            conn.close();
32            System.out.println("=====close db connection=====");
33        }
34    }
35 }
36

```

JDBC [Java Data base Connectivity]

Non Select Query



```
SDET_29 - Comcast_selenium_Framework/src/test/java/practice/NonSelectQuery_SampleCode2.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
SampleTest.java ReadDataFromProp... commonData.prope... CreateOrgTest.java Comcast_selenium... testdata.xlsx *SelectQuery_Sam... *NonSelectQuery_S... *ProjectUnitTestj...
11 public class NonSelectQuery_SampleCode2 {
12     public static void main(String[] args) throws Throwable {
13         Connection conn = null;
14         int result = 0;
15         try {
16             Driver driverRef = new Driver();
17             /* step 1 : load /register *mysql *the database */
18             DriverManager.registerDriver(driverRef);
19             /* step 2 : connect to db */
20             conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/projects", "root", "root");
21             System.out.println("connection is done");
22             /* step 3 : create query statement */
23             Statement stat = conn.createStatement();
24             String query = "insert into project values('TY_PROJ_3002','deepak','12/01/2022','BankOFBoroda_2','On Going',10)";
25             /* step 4 : execute the Query */
26             result = stat.executeUpdate(query);
27         } catch (Exception e) {
28         } finally {
29             // TODO: handle exception
30             if(result==1) {
31                 System.out.println("Project inserted succesfully");
32             } else {
33                 System.err.println("Project is not inserted--!");
34             }
35             /*step 5: close the connection */
36             conn.close();
37             System.out.println("=====close db connection=====");
38         }
39     }
40 }
```

Why JDBC in Automation ?

In Automation also required JDBC Connection , Most of the test case Preconditions , required few set of data in Database &also validation GUI WRT Database

Case 1:

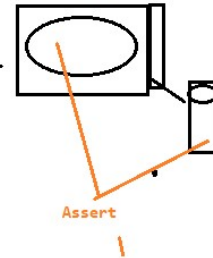
testCase : sendAmount

precondition : FAN should have minimum 10 K balance

insert into account values (Acc1, active , 10000)

1. login to app
2. navigate to transaction page
3. enter FAN . TAN , Amount(6k) & click on send button
4. verify the sucessfull msg
verify the balance in GUI
verify the balance in Database
select * from account where accountNum = Acc1;

testEnv



Case 2

precondition : insert product in to database , which is req for testcase

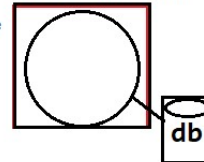
@test

1. login to Amazon
2. enter *product* in serch feild
3. click on srach button
4. verify the product list & verify

testData

productName	cat	brand	qty
iphone12	apple	pple	10
iphone-7	mobile	apple	20
samsung	mobile	sao	55
moto			

Amazon.com



Case :3

Test Case : crete Order

2 min

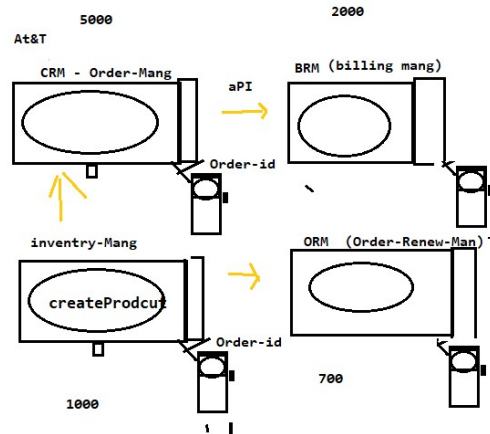
1. login to CRM
2. Go to Odrer page
3. create an Order
4. verify an Odrer

fk-010

Test Case : BillOnOrder

precondtion is : create an order in database via Query

1. login to BRM
2. Go to Billing Page
3. do payemnet
4. verify

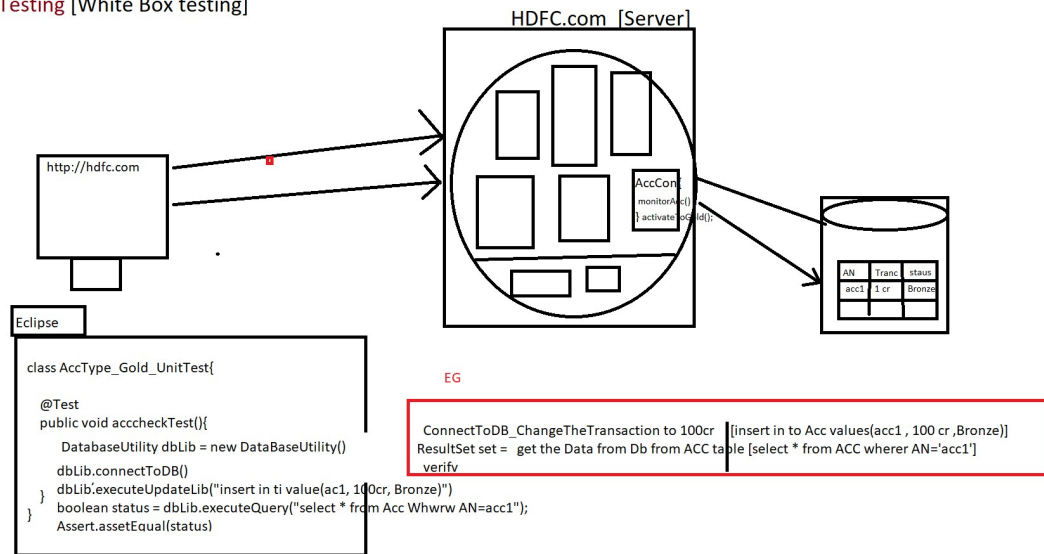


JDBC [Java Data base Connectivity]

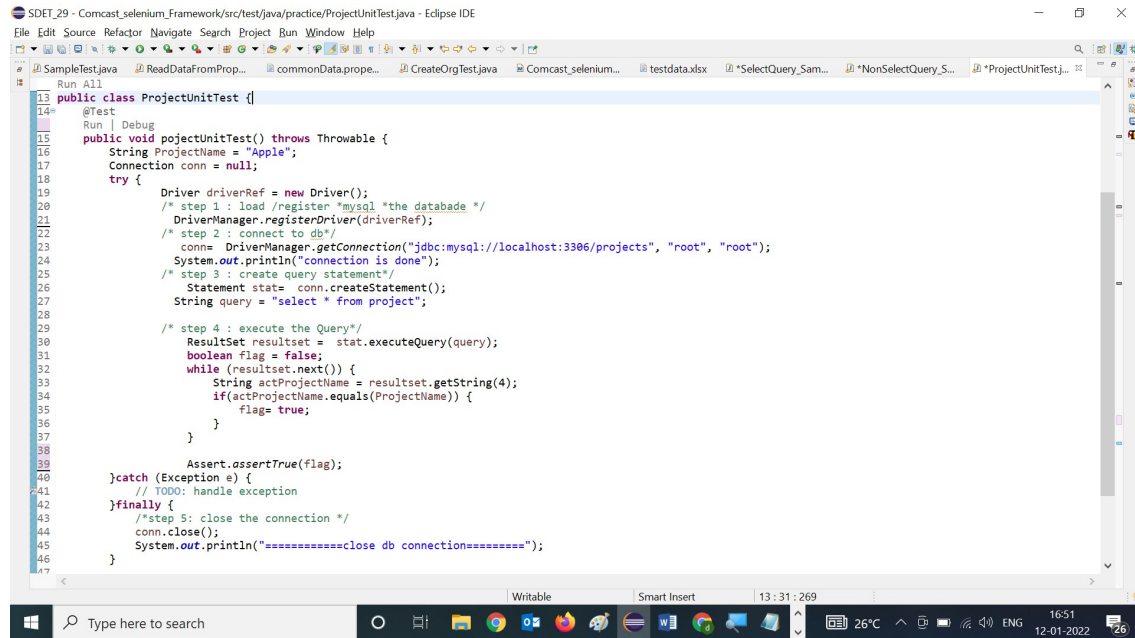
JDBC usage in UNIT Testing[White box Testing]

In case unit testing , [jdbc]database code is required to connect to database & get the data ten verify the data WRT to business logic

Unit Testing [White Box testing]



JDBC [Java Data base Connectivity]



```
13 public class ProjectUnitTest {
14     @Test
15     public void projectUnitTest() throws Throwable {
16         String projectName = "Apple";
17         Connection conn = null;
18         try {
19             Driver driverRef = new Driver();
20             /* step 1 : load /register "mysql" the database */
21             DriverManager.registerDriver(driverRef);
22             /* step 2 : connect to db */
23             conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/projects", "root", "root");
24             System.out.println("connection is done");
25             /* step 3 : create query statement */
26             Statement stat = conn.createStatement();
27             String query = "select * from project";
28
29             /* step 4 : execute the Query */
30             ResultSet resultSet = stat.executeQuery(query);
31             boolean flag = false;
32             while (resultSet.next()) {
33                 String actProjectName = resultSet.getString(4);
34                 if (actProjectName.equals(projectName)) {
35                     flag = true;
36                 }
37             }
38             Assert.assertTrue(flag);
39         } catch (Exception e) {
40             // TODO: handle exception
41         } finally {
42             /* step 5: close the connection */
43             conn.close();
44             System.out.println("=====close db connection=====");
45         }
46     }
47 }
```