

# Introduction to JDBC

Difficulty Level : Hard Last Updated : 26 Jul, 2021

## Introduction

JDBC or Java Database Connectivity is a specification from Sun microsystems that provides a standard abstraction(that is API or Protocol) for java applications to communicate with various databases. It provides the language with java database connectivity standard. It is used to write programs required to access databases. JDBC along with the database driver is capable of accessing databases and spreadsheets. The enterprise data stored in a relational database(RDB) can be accessed with the help of JDBC APIs.

## Definition of JDBC(Java Database Connectivity)

JDBC is an API(Application programming interface) which is used in java programming to interact with databases.

*The classes and interfaces of JDBC allows application to send request made by users to the specified database.*

## Purpose of JDBC

Enterprise applications that are created using the JAVA EE technology need to interact with databases to store application-specific information. So, interacting with a database requires efficient database connectivity which can be achieved by using the ODBC(Open database connectivity) driver. This driver is used with JDBC to interact or communicate with various kinds of databases such as Oracle, MS Access, Mysql and SQL server database.

## Components of JDBC

There are generally four main components of JDBC through which it can interact with a database. They are as mentioned below:

1. **JDBC API:** It provides various methods and interfaces for easy communication with the database. It provides two packages as follows which contains the java SE and java EE platforms to exhibit WORA(write once run everywhere) capabilities.

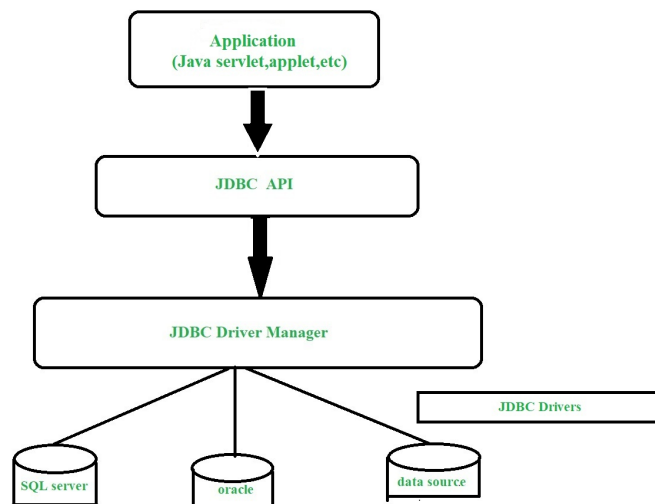
1. `java.sql.*;`
2. `javax.sql.*;`

It also provides a standard to connect a database to a client application.

2. **JDBC Driver manager:** It loads database-specific driver in an application to establish a connection with a database. It is used to make a database-specific call to the database to process the user request.
3. **JDBC Test suite:** It is used to test the operation(such as insertion, deletion, updation) being performed by JDBC Drivers.
4. **JDBC-ODBC Bridge Drivers:** It connects database drivers to the database. This bridge translates JDBC method call to the ODBC function call. It makes the use of `sun.jdbc.odbc`

package that includes native library to access ODBC characteristics.

### Architecture of JDBC



### **Description:**

1. **Application:** It is a java applet or a servlet which communicates with a data source.
2. **The JDBC API:** The JDBC API allows Java programs to execute SQL statements and retrieve results. Some of the important classes and interfaces defined in JDBC API are as follows:

DriverManager

Driver

Connection

Statement

PreparedStatement

CallableStatement

ResultSet

SQL data

3. **DriverManager:** It plays an important role in the JDBC architecture. It uses some database-specific drivers to effectively connect enterprise applications to databases.
4. **JDBC drivers:** To communicate with a data source through JDBC, you need a JDBC driver that intelligently communicates with the respective data source.

### Types of JDBC Architecture(2-tier and 3-tier)

The JDBC architecture consists of two-tier and three-tier processing models to access a database. They are as described below:

1. **Two-tier model:** A java application communicates directly to the data source. The JDBC driver enables the communication between the application and the data source. When a user sends a query to the data source, the answers for those queries are sent back to the user in the form of results.  
The data source can be located on a different machine on a network to which a user is connected. This is known as a **client/server configuration**, where the user's machine acts as a client and the machine having the data source running acts as the server.
2. **Three-tier model:** In this, the user's queries are sent to middle-tier services, from which the commands are again sent to the data source. The results are sent back to the middle tier, and from there to the user.

This type of model is found very useful by management information system directors.

### Working of JDBC

Java application that needs to communicate with the database has to be programmed using JDBC API. JDBC Driver supporting data sources such as Oracle and SQL server has to be added in java application for JDBC support which can be done dynamically at run time. This JDBC driver intelligently communicates the respective data source.

### **Creating a simple JDBC application**

```
package com.vinayak.jdbc;
import java.sql.*;
public class JDBCdemo {
    public static void main(
        String args[]) throws SQLException,
        ClassNotFoundException
    {
        String driverClassName
            = "sun.jdbc.odbc.JdbcOdbcDriver";
        String url = "jdbc:odbc:XE";
        String username = "scott";
        String password = "tiger";
        String query
```

```
        = "insert into students values(109, 'bhatt')";

    // Load driver class
    Class.forName(driverClassName);

    // Obtain a connection
    Connection con = DriverManager.getConnection(
        url, username, password);

    // Obtain a statement
    Statement st = con.createStatement();

    // Execute the query
    int count = st.executeUpdate(query);
    System.out.println(
        "number of rows affected by this query= " + count);

    // Closing the connection as per the
    // requirement with connection is completed
    con.close();
}
} // class
```

The above example demonstrates the basic steps to access a database using JDBC. The application uses the JDBC-ODBC bridge driver to connect to the database. You must import **java.sql** package to provide basic SQL functionality and use the classes of the package.