

9 Dec 2022:

JDBC Steps:

1. Load the driver / register the driver with the Driver Manager.

```
Class.forName('com.jdbc.....Driver');
```

2. Get the connection by providing the url, username, password

```
Connection connection= DriverManager.getConnection(url, username, password)
```

3. Create a pipeline to send sql queries to database from java application

```
Statement statement= connection.createStatement(); // creates a pipeline
```

4. use statement object to add queries

example: insert an employee into db.

```
statement.execute(query)
```

5. close the connection

Problem Statement: Insert data into mysql db

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;

public class InsertData {

    public static void main(String[] args) throws
ClassNotFoundException, SQLException {
```

```

        Class.forName("com.mysql.cj.jdbc.Driver");

        Connection
connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/Empl
oyeesDB", "root", "ipl2021@dubai");

        Statement statement = connection.createStatement();

        // insert data into the database.

        String query = "insert into customer values(14,
9014,29000,'anil')";

        statement.execute(query);

    }

```

Problem statement: retrieve the data from the db - table

```

public class InsertData {

    public static void main(String[] args) throws
ClassNotFoundException, SQLException {

        Class.forName("com.mysql.cj.jdbc.Driver");

        Connection
connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/Empl
oyeesDB", "root", "ipl2021@dubai");

        Statement statement = connection.createStatement();

```

```

        // get all the customers data from db

String query= "select * from customer";

ResultSet resultset=statement.executeQuery(query);

System.out.println("Resultset"+ resultset);

        while(resultset.next()) {
            System.out.println(resultset.getInt("id")+ "\t"+
resultset.getInt("accountNumber")+ "\t"+ resultset.getInt("balance")+ "\t"+
resultset.getString("name"));
        }

```

Problem Statement:

Get column names dynamically.

```

package com.firstapp;

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

public class InsertData {

    public static void main(String[] args) throws
ClassNotFoundException, SQLException {

```

```
Class.forName("com.mysql.cj.jdbc.Driver");
```

```
Connection
```

```
connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/EmployeesDB", "root", "ipl2021@dubai");
```

```
Statement statement = connection.createStatement();
```

```
// Get column names dynamically.
```

```
String query= "select * from customer";
```

```
ResultSet resultset=statement.executeQuery(query);
```

```
ResultSetMetaData rsmd=resultset.getMetaData();
```

```
int count = rsmd.getColumnCount();
```

```
int i=1;
```

```
while( i <=count) {
```

```
    System.out.println(rsmd.getColumnLabel(i));
```

```
}
```

```
}  
  
}
```

13 Dec 2022: CRUD application using MYSQL and JDBC:

```
package com.firstapp;
```

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.util.Scanner;
```

```
public class CRUD {
```

```
    Statement statement= null;
```

```
    Scanner sc= new Scanner(System.in);
```

```
    public static void main(String[] args) throws SQLException {
```

```
        CRUD c = new CRUD();  
        c.getDBConnection();  
        System.out.println(c.statement);  
        // c.add();  
        //c.getAllCustomers();
```

```

        c.deleteCustomer();
    }

    public void getDBConnection() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection
connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/Empl
yeesDB", "root", "ipl2021@dubai");
            statement = connection.createStatement();

        }
        catch(Exception e) {
            System.out.println(e);
        }
    }
}

```

```

public void addCustomer() throws SQLException {

```

```

    System.out.println("Enter id");
    long id = sc.nextLong();

```

```

    System.out.println("Enter accno");
    long accNo = sc.nextLong();

```

```

    System.out.println("Enter balance");
    long accBalance = sc.nextLong();

```

```

    System.out.println("Enter name");
    String name = sc.next();

```

```

    System.out.println(accNo + "\t" + accBalance + "\t" + name);

```

```

    // insert into customer values(10, 9011,9000,'kiran');

```

```
String query = "insert into customer values("+id+ ","+accNo+ "," +  
accBalance + ","+ name+"");
```

```
System.out.println("query" + query);
```

```
statement.execute(query);
```

```
boolean result = getCustomerById(id);
```

```
if(result == true) {
```

```
    System.out.println("customer added successfully!");
```

```
}
```

```
else {
```

```
    System.out.println("Failure in addition of the customer");
```

```
}
```

```
}
```

```
public boolean getCustomerById(long id) throws SQLException {
```

```
    String query = "select * from customer where id="+id;
```

```
    ResultSet rs= statement.executeQuery(query);
```

```
    return rs.next();
```

```
}
```

```
public void deleteCustomer() throws SQLException {
```

```
    System.out.println("Enter id");
```

```
    long id = sc.nextLong();
```

```
String query = "delete from customer where id="+id;
```

```

        System.out.println("query" + query);

        statement.execute(query);

    }

    public void updateCustomer() {

        // please write the code in busy schedule
    }

    public void getAllCustomers() throws SQLException {

String query = "select * from customer";

        ResultSet rs= statement.executeQuery(query);

        System.out.println("All Customers Data");
        System.out.println("-----");

        while(rs.next()) {

            System.out.println(rs.getLong(1) + "\t"+
rs.getLong(2)+"\t"+rs.getLong(3)+"\t"+ rs.getString(4));

        }

    }

}

/*
 *
 * Implement the interactive system for the above example

```



```

*
*      1 add 2 remove 3 get all customers 4 update 5 exit
*
*      1
*
*      customer added successfully
*
*
*      1 add 2 remove 3 get all customers 4 update 5 exit
*
*      2
*
*
*
*
*/

```

14 Dec 2022:

Connection to MongoDB Database:

```
package com.firstapp;
```

```
import com.mongodb.MongoClient;
```

```
import com.mongodb.client.FindIterable;
```

```
import com.mongodb.client.MongoCollection;
```

```
import com.mongodb.client.MongoDatabase;
```

```
public class MongoDBConnection {
```

```
    public static void main(String[] args) {
```

```
        MongoClient mongo = new MongoClient( "localhost" , 27017 );
```

```
        System.out.println(mongo);
```

```
MongoDatabase md = mongo.getDatabase("7am");

System.out.println("md" + md);

MongoCollection collection = md.getCollection("products");

System.out.println("collection"+ collection);

FindIterable fi = collection.find();

System.out.println(fi.first());

    }

}
```

JDBC Java Doc:

<https://docs.oracle.com/javase/8/docs/api/java/sql/package-summary.html>

Connection - Interface

Ref:

<https://docs.oracle.com/javase/8/docs/api/java/sql/Connection.html>

Statement:

The statement interface object is used to send static Queries to the database.

Ref:

<https://docs.oracle.com/javase/8/docs/api/java/sql/Statement.html>

Prepared Statement:

Ref:

<https://docs.oracle.com/javase/8/docs/api/java/sql/PreparedStatement.html>

PreparedStatement Demo:

```
package com.firstapp;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;

public class PreparedStatementDemo {

    public static void main(String[] args) throws ClassNotFoundException, SQLException {

        Class.forName("com.mysql.cj.jdbc.Driver");

        Connection
connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/EmployeesDB", "root",
"ipl2021@dubai");

        System.out.println("connection" + connection);

        PreparedStatement pstmt = connection.prepareStatement("insert into customer
values(?,?,?,?)");
        pstmt.setInt(1, 17);
```

```

        pstmt.setInt(2, 8788);
        pstmt.setInt(3, 9000);
        pstmt.setString(4, "raj");
        int i = pstmt.executeUpdate();
        System.out.println(i);
    }
}

```

15 Dec 2022:

Batch Processing:

```

package com.firstapp;

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

public class BatchProcessingDemo {

    public static void main(String[] args) throws ClassNotFoundException,
    SQLException {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection connection=
        DriverManager.getConnection("jdbc:mysql://localhost:3306/EmployeesDB",
        "root", "ipl2021@dubai");
        connection.setAutoCommit(false);
        Statement statement = connection.createStatement();
        statement.addBatch("insert into customer values(204,
        2015,9000,'user 5')");
        statement.addBatch("insert into customer values(205,
        2016,19000,'user 6')");
    }
}

```

```

        statement.addBatch("insert into customer values(206,
2017,29000,'user 7')");
        statement.addBatch("insert into customer values(207,
2018,39000,'user 8')");
        statement.executeBatch();
        connection.commit();
        System.out.println("finished");
    }
}

```

Transaction management:

1. **class** FetchRecords{
2. **public static void** main(String args[])**throws** Exception{
3. Class.forName("oracle.jdbc.driver.OracleDriver");
4. Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system",
"oracle");
5. con.setAutoCommit(**false**);
- 6.
7. Statement stmt=con.createStatement();
8. stmt.executeUpdate("insert into user420 values(190,'abhi',40000)");
9. stmt.executeUpdate("insert into user420 values(191,'umesh',50000)");
- 10.
11. con.commit();
12. con.close();
- 13.}}

Meta data:

ResultSetMetadata:

```
package com.firstapp;
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;
```

```
public class ResultSetMetaDataDemo {
```

```
    public static void main(String[] args) throws ClassNotFoundException,
    SQLException {
```

```
        Class.forName("com.mysql.cj.jdbc.Driver");
```

```
        Connection
```

```
connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/Emplo
yeesDB", "root", "ipl2021@dubai");
```

```
        Statement statement = connection.createStatement();
```

```
        ResultSet rs = statement.executeQuery("select * from customer");
```

```
        ResultSetMetaData rsmd = rs.getMetaData();
```

```
//
```

```
//      System.out.println(rsmd.getColumnCount());
```

```
//      System.out.println(rsmd.getColumnLabel(4));
```

```
//      System.out.println(rsmd.getColumnLabel(2));
```

```
        int i=1;
```

```
        while(i <=rsmd.getColumnCount()) {
```

```

        System.out.println(rsmd.getColumnLabel(i));
        i++;
    }

}

}

}

```

Database Metadata:

```

package com.firstapp;

import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseMetaDataDemo {

    public static void main(String[] args) throws ClassNotFoundException,
SQLException {

        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection
connection=DriverManager.getConnection("jdbc:mysql://localhost:3306/Empl
oyeesDB", "root", "ipl2021@dubai");
        DatabaseMetaData dmd = connection.getMetaData();
        System.out.println(dmd.getDriverName());
        System.out.println(dmd.getDriverVersion());
        System.out.println(dmd.getDatabaseProductName());
    }
}

```

```
        System.out.println(dmd.getMaxConnections());  
  
    }  
  
}
```

16 Dec 2022:

Modern Approach:

Front End App (Server) : UI (Dynamic) + Show data - Libraries / Frameworks

Backend (Server) : Data Processing - RestFul WebServices

Traditional Approach for Application Development:

There used to only one server side application (Servlets) that handles data processing and also UI.

Servlets:

Servlet is a web component that enhances the capability of a web server (apache tomcat).

Using Servlets, we can create web applications.

javax.servlet:

Servlet is an interface that provides few life cycle methods to provide dynamic content to the end users.

Servlet Interface

implemented by

GenericServlet

extended by

HttpServlet

There are 3 ways to create a Servlet.

1. By implementing Servlet Interface.

```
package com.digitalync.learning;
```

```
import java.io.IOException;
```

```
import javax.servlet.Servlet;
```

```
import javax.servlet.ServletConfig;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.ServletRequest;
```

```
import javax.servlet.ServletResponse;
```

```
public class FirstServlet implements Servlet{
```

```
    @Override
```

```
    public void destroy() {
```

```
        // TODO Auto-generated method stub
```

```

        System.out.println("destroy");
    }

    @Override
    public ServletConfig getServletConfig() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public String getServletInfo() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public void init(ServletConfig arg0) throws ServletException {
        // TODO Auto-generated method stub

        System.out.println("init method");
    }

    @Override
    public void service(ServletRequest arg0, ServletResponse arg1) throws
ServletException, IOException {
        // TODO Auto-generated method stub

        System.out.println("Service method");
    }
}

```

2. By extending GenericServlet class :

```
public class FirstServlet extends GenericServlet{

    @Override
    public void service(ServletRequest arg0, ServletResponse arg1) throws
ServletException, IOException {
        // TODO Auto-generated method stub

    }

}
```

3. By extending HttpServlet class

```
public class FirstServlet extends HttpServlet{

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {
        // TODO Auto-generated method stub
        super.doGet(req, resp);
    }

}
```

First App:

web.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID"
version="3.0">
```

```
<servlet>
<servlet-name>FirstServlet</servlet-name>
<servlet-class>com.digitalync.learning.FirstServlet</servlet-class>
</servlet>
```

```
<servlet-mapping>
<servlet-name>FirstServlet</servlet-name>
<url-pattern>/home</url-pattern>
```

```
</servlet-mapping>
```

```
</web-app>
```

FirstServlet.java:

```
package com.digitalync.learning;

import java.io.IOException;

import javax.servlet.GenericServlet;
import javax.servlet.Servlet;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
public class FirstServlet implements Servlet{

    @Override
    public void destroy() {
        // TODO Auto-generated method stub

        System.out.println("destroy");
    }

    @Override
    public ServletConfig getServletConfig() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public String getServletInfo() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public void init(ServletConfig arg0) throws ServletException {
        // TODO Auto-generated method stub

        System.out.println("init method");
    }

    @Override
    public void service(ServletRequest arg0, ServletResponse arg1) throws
ServletException, IOException {
        // TODO Auto-generated method stub
    }
}
```

```
        System.out.println("Service method");
    }
}
```

19Dec2022:
ServletRequest (HttpServletRequest) &
ServletResponse(HttpServletResponse)

Web.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID"
version="3.0">

    <servlet>
        <servlet-name>s1</servlet-name>
        <servlet-class>com.learn.servlet.ServletX</servlet-class>
    </servlet>

    <servlet-mapping>

        <servlet-name>s1</servlet-name>
        <url-pattern>/otp</url-pattern>

    </servlet-mapping>
```

</web-app>

ServletX.java:

```
package com.learn.servlet;
```

```
import java.io.IOException;
```

```
import java.io.PrintWriter;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
public class ServletX extends HttpServlet {  
    private static final long serialVersionUID = 1L;
```

```
    @Override
```

```
    protected void doGet(HttpServletRequest req, HttpServletResponse  
resp) throws ServletException, IOException {
```

```
        PrintWriter out = resp.getWriter();
```

```
        out.print("<html><body><form action='./otp' method='POST'>  
<label> Enter OTP: </label> <input type='number' name='otp'>  
<button>Submit</button> </form> </body></html> ");
```

```
    }
```

```
    @Override
```

```
    protected void doPost(HttpServletRequest req,  
HttpServletResponse resp) throws ServletException, IOException {
```

```
        PrintWriter out = resp.getWriter();
```

```

    int otp= Integer.parseInt(req.getParameter("otp"));

    if(otp == 9011) {

        out.print("OTP is successful. please check your email after
some time");
    }

    else {

        out.print("OTP Failure");
    }

}

}

```

20 Dec 2022:

Servlet Config:

User.java:

```

public class User extends HttpServlet {
    Connection connection= null;
    private static final long serialVersionUID = 1L;

    @Override
    public void init(ServletConfig config) throws ServletException {
        System.out.println("driver:"+config.getInitParameter("driver"));
        System.out.println("init method called");
        String driverName = config.getInitParameter("driver");
    }
}

```



```

        String url = config.getInitParameter("url");
        String username = config.getInitParameter("username");
        String password = config.getInitParameter("password");
        try {
            Class.forName(driverName);

connection=DriverManager.getConnection(url,username,password);
            System.out.println("connection"+connection);
        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

```

Web.xml:

```

<web-app
    xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" version="3.1">
    <servlet>
        <servlet-name>s1</servlet-name>
        <servlet-class>com.ecommerce.User</servlet-class>
        <init-param>
            <param-name>driver</param-name>
            <param-value>com.mysql.cj.jdbc.Driver</param-value>
        </init-param>

        <init-param>
            <param-name>url</param-name>

```

```

<param-value>jdbc:mysql://localhost:3306/EmployeesDB</param-value>
  </init-param>
  <init-param>
    <param-name>username</param-name>
    <param-value>root</param-value>
  </init-param>
  <init-param>
    <param-name>password</param-name>
    <param-value>ipl2021@dubai</param-value>
  </init-param>
</servlet>
<servlet-mapping>
  <servlet-name>s1</servlet-name>
  <url-pattern>/user</url-pattern>
</servlet-mapping>
</web-app>

```

ServletContext:

Web.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" version="3.1">
  <context-param>
    <param-name>code</param-name>
    <param-value>mdbvsbdvbwdbjb</param-value>
  </context-param>

  <context-param>
    <param-name>access_key</param-name>

```

```
<param-value>AHGFH123452MVCVJBS</param-value>
</context-param>

<servlet>
  <servlet-name>s1</servlet-name>
  <servlet-class>com.servletcontext.demo.Servlet1</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>s1</servlet-name>
  <url-pattern>/servlet1</url-pattern>
</servlet-mapping>

<servlet>
  <servlet-name>s2</servlet-name>
  <servlet-class>com.servletcontext.demo.Servlet2</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>s2</servlet-name>
  <url-pattern>/servlet2</url-pattern>
</servlet-mapping>

<servlet>
  <servlet-name>s3</servlet-name>
  <servlet-class>com.servletcontext.demo.Servlet3</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>s3</servlet-name>
  <url-pattern>/servlet3</url-pattern>
</servlet-mapping>
</web-app>
```

Servlet1.java:

```
package com.servletcontext.demo;
```

```

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class Servlet1 extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {

        PrintWriter out= resp.getWriter();

        System.out.println("I am in Servlet 1");

        ServletContext context = getServletContext();
        out.print("In Servlet one");
        out.println(context.getInitParameter("code"));
        out.println(context.getInitParameter("access_key"));

    }

}

```

Servlet2.java:

```

package com.servletcontext.demo;

import java.io.IOException;

```

```

import java.io.PrintWriter;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class Servlet2 extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req,
        HttpServletResponse resp) throws ServletException, IOException {

        PrintWriter out= resp.getWriter();

        System.out.println("I am in Servlet 2");

        ServletContext context = getServletContext();
        out.print("In Servlet two");
        out.println(context.getInitParameter("code"));
        out.println(context.getInitParameter("access_key"));

    }

}

```

Servlet-JDBC:

User.java:

```

package com.ecommerce;

import java.io.IOException;
import java.io.PrintWriter;

```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class User extends HttpServlet {
    Connection connection= null;
    private static final long serialVersionUID = 1L;

    @Override
    public void init(ServletConfig config) throws ServletException {
        System.out.println("driver:"+config.getInitParameter("driver"));
        System.out.println("init method called");
        String driverName = config.getInitParameter("driver");
        String url = config.getInitParameter("url");
        String username = config.getInitParameter("username");
        String password = config.getInitParameter("password");
        try {
            Class.forName(driverName);

            connection=DriverManager.getConnection(url,username,password);
            System.out.println("connection"+connection);
        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

```

```

@Override
protected void doGet(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {

    PrintWriter out = resp.getWriter();

    out.print("<html><body><form action='./user' method='POST'>
<label> Enter id: </label> <input type='number' name='id'> <br><br> <label>
Enter Acc no: </label> <input type='number' name='accountNumber'>
<br><br><label> Enter balance: </label> <input type='number'
name='balance'> <br><br> <label> Enter Name: </label> <input type='text'
name='name'> <br><br> <button>Submit</button> </form> </body></html>
");

}

```

```

@Override
protected void doPost(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {
    PrintWriter out = resp.getWriter();
    int id = Integer.parseInt(req.getParameter("id"));
    int accountNumber =
Integer.parseInt(req.getParameter("accountNumber"));
    int balance = Integer.parseInt(req.getParameter("balance"));
    String name = req.getParameter("name");

    try {
        PreparedStatement pstmt =
connection.prepareStatement("insert into customer values(?,?,?,?)");
        pstmt.setInt(1, id);
        pstmt.setInt(2, accountNumber);
        pstmt.setInt(3, balance);
        pstmt.setString(4, name);
        int i = pstmt.executeUpdate();
        if(i ==1) {

```

```

        out.println("<h1>Success </h1>");
    }
    else {
        out.println("<h1>Failure </h1>");
    }
}
catch(Exception e) {
    System.out.println(e);
    out.println("<h1>Failure </h1>" + e.getMessage());
}
}

}

```

Web.xml:

```

<web-app
    xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" version="3.1">
    <servlet>
        <servlet-name>s1</servlet-name>
        <servlet-class>com.ecommerce.User</servlet-class>
        <init-param>

```



```

        <param-name>driver</param-name>
        <param-value>com.mysql.cj.jdbc.Driver</param-value>
    </init-param>

    <init-param>
        <param-name>url</param-name>

        <param-value>jdbc:mysql://localhost:3306/EmployeesDB</param-value>
    </init-param>
    <init-param>
        <param-name>username</param-name>
        <param-value>root</param-value>
    </init-param>
    <init-param>
        <param-name>password</param-name>
        <param-value>ipl2021@dubai</param-value>
    </init-param>
</servlet>
<servlet-mapping>
    <servlet-name>s1</servlet-name>
    <url-pattern>/user</url-pattern>
</servlet-mapping>
</web-app>

```

Session Management:

Cookies:

FirstServlet.java:

```
package com.cookie.demo;
```

```
import java.io.IOException;
```

```
import java.io.PrintWriter;
```

```

import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class FirstServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {

        Cookie cookie = new Cookie("token", "sjhdvjsvcj");
        resp.addCookie(cookie);

        PrintWriter out=resp.getWriter();

        out.print("<html><a href='./f2'>SecondServlet</a> </html>");

    }

}

```

SecondServlet.java:

```

package com.cookie.demo;

import java.io.IOException;

import javax.servlet.ServletException;

```

```

import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class SecondServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {

        Cookie cookies[]=req.getCookies();

        System.out.println(cookies[0].getName() + "\t"+
cookies[0].getValue());
    }
}

```

Web.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID"
version="3.0">

```

```

    <servlet>
    <servlet-name>f1</servlet-name>
    <servlet-class>com.cookie.demo.FirstServlet</servlet-class>
    </servlet>
    <servlet-mapping>
    <servlet-name>f1</servlet-name>
    <url-pattern>/f1</url-pattern>

```

```
</servlet-mapping>

<servlet>
<servlet-name>f2</servlet-name>
<servlet-class>com.cookie.demo.SecondServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>f2</servlet-name>
  <url-pattern>/f2</url-pattern>
</servlet-mapping>

</web-app>
```

HttpSession:

FirstServlet.java:

```
package com.httpsession.demo;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class FirstServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
```

@Override

```

        protected void doGet(HttpServletRequest req, HttpServletResponse
resp) throws ServletException, IOException {

    //            Cookie cookie = new Cookie("token", "sjhdvjsvcj");
    //            resp.addCookie(cookie);

            HttpSession session = req.getSession();
            session.setAttribute("token", new Integer(10000));
            PrintWriter out=resp.getWriter();

            out.print("<html><a href='./f2'>SecondServlet</a> </html>");

        }

    }

```

SecondServlet.java:

```

package com.httpsession.demo;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class SecondServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

```

@Override

```
protected void doGet(HttpServletRequest req, HttpServletResponse  
resp) throws ServletException, IOException {
```

```
    HttpSession session = req.getSession(false);
```

```
    Integer i= (Integer)session.getAttribute("token");
```

```
    System.out.println("Token" + i);
```

```
    }  
}
```

Web.xml:

```
<?xml version="1.0" encoding="UTF-8"?>  
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns="http://java.sun.com/xml/ns/javaee"  
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID"  
version="3.0">
```

```
    <servlet>  
    <servlet-name>f1</servlet-name>  
    <servlet-class>com.httpsession.demo.FirstServlet</servlet-class>  
    </servlet>  
    <servlet-mapping>  
        <servlet-name>f1</servlet-name>  
        <url-pattern>/f1</url-pattern>  
    </servlet-mapping>
```

```
    <servlet>  
    <servlet-name>f2</servlet-name>  
    <servlet-class>com.httpsession.demo.SecondServlet</servlet-class>
```

```
</servlet>
<servlet-mapping>
  <servlet-name>f2</servlet-name>
  <url-pattern>/f2</url-pattern>
</servlet-mapping>

</web-app>
```

Annotations:

```
package com.firstapp;
```

```
class Version{

    public String getVersion(){
        return "java 8";
    }

}
```

```
class UpdatedVersion extends Version{

    @Override
    public String getVersion() {
        return "java 19";
    }

}
```

```
public class Client {

    public static void main(String[] args) {
        UpdatedVersion u = new UpdatedVersion();
        Version v = new UpdatedVersion();
        System.out.println(v.getVersion());
    }
}
```

```
}  
  
}
```

Servlet Annotations:

```
package com.annotations.demo;
```

```
import java.io.IOException;
```

```
import java.io.PrintWriter;
```

```
import javax.servlet.ServletConfig;
```

```
import javax.servlet.ServletException;
```

```
import javax.servlet.annotation.WebInitParam;
```

```
import javax.servlet.annotation.WebServlet;
```

```
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;
```

```
/**
```

```
 * Servlet implementation class FirstServlet
```

```
 */
```

```
@WebServlet(value = "/f1",initParams = {  
    @WebInitParam(name = "drivername", value=  
"org.oracle.OracleDriver")  
})
```

```
public class FirstServlet extends HttpServlet {  
    private static final long serialVersionUID = 1L;
```

```
    @Override
```

```
        protected void doGet(HttpServletRequest req,  
HttpServletResponse resp) throws ServletException, IOException {  
            PrintWriter out = resp.getWriter();
```



```

        out.print("<html> <body>I am from First Servlet</body></html>");

        ServletConfig config= getServletConfig();

        System.out.println(config.getInitParameter("drivename"));

    }

}

```

Jar file:

http://www.java2s.com/Code/Jar/j/Downloadjavaservlet30jar.htm#google_vignette

27 Dec 2022:

Pom.xml:

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.firstapp.hibernate</groupId>
  <artifactId>hibernatedemo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>hibernatedemo</name>
  <url>http://maven.apache.org</url>

  <properties>

```

```
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

<dependencies>

<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.3.1.Final</version>
</dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.29</version>
</dependency>
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>3.8.1</version>
  <scope>test</scope>
</dependency>
</dependencies>
</project>
```

Product.java:

```
package com.firstapp.hibernate.hibernatedemo;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.Table;
```

```
@Entity
```

```
@Table
```

```
public class Product {

    @Id
    @Column
    private int id;
    @Column
    private String name;
    @Column
    private String seller;
    @Column
    private int price;
    @Column
    private String imageUrl;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getSeller() {
        return seller;
    }
    public void setSeller(String seller) {
        this.seller = seller;
    }
    public int getPrice() {
        return price;
    }
    public void setPrice(int price) {
        this.price = price;
    }
}
```

```

    }
    public String getImageUrl() {
        return imageUrl;
    }
    public void setImageUrl(String imageUrl) {
        this.imageUrl = imageUrl;
    }
}

```

App.java:

```

package com.firstapp.hibernate.hibernatedemo;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.boot.Metadata;
import org.hibernate.boot.MetadataSources;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

/**
 * Hello world!
 */
public class App
{
    public static void main( String[] args )
    {
        StandardServiceRegistry ssr = new
StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();

```

```

        Metadata meta = new
MetadataSources(ssr).getMetadataBuilder().build();

SessionFactory factory = meta.getSessionFactoryBuilder().build();
Session session = factory.openSession();
Transaction transaction= session.beginTransaction();
Product product = new Product();
product.setId(101);
product.setImageUrl("some url 2");
product.setName("product 2");
product.setPrice(1900);
product.setSeller("some seller 2");
session.save(product);
transaction.commit();
    }
}

```

CRUD app:

```

package com.firstapp.hibernate.hibernatedemo;

import java.util.Iterator;
import java.util.List;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.boot.Metadata;
import org.hibernate.boot.MetadataSources;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

/**
 * Hello world!

```

```

*
*/
public class App
{
    public static void main( String[] args )
    {
        StandardServiceRegistry ssr = new
StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();
        Metadata meta = new
MetadataSources(ssr).getMetadataBuilder().build();

        SessionFactory factory = meta.getSessionFactoryBuilder().build();
        Session session = factory.openSession();
        Transaction transaction= session.beginTransaction();
        /*
        // Saving a product
        Product product = new Product();
        product.setId(102);
        product.setImageUrl("some url 3");
        product.setName("product 3");
        product.setPrice(2900);
        product.setSeller("some seller 3");
        session.save(product);

        // updating a product

        Product product = new Product();
        product.setId(103);
        product.setImageUrl("some url 3");
        product.setName("product 3");
        product.setPrice(4000);
        product.setSeller("some seller 3");
        session.update(product);

        // iterating a list of products

```

```

List<Product> products = session.createQuery("from Product").list();
Iterator it = products.iterator();

while(it.hasNext()) {

    Product prod = (Product)it.next();

    System.out.println(prod.getId() + "\t" + prod.getImageUrl() + "\t" +
prod.getName() + "\t" + prod.getPrice());

}

// deleting a product

Product product = session.get(Product.class, 102);
session.delete(product);

*/
try {
    Product product = session.get(Product.class, 102);
    session.delete(product);
}
catch(Exception e) {
    System.out.println(e.getMessage());
}
transaction.commit();
}
}

```

First Level Cache:

hibernate.cfg.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC

```

```

"-//Hibernate/Hibernate Configuration DTD 5.3//EN"
"http://www.hibernate.org/dtd/hibernate-configuration-5.3.dtd">

<hibernate-configuration>

    <session-factory>
        <property
name="dialect">org.hibernate.dialect.MySQL5InnoDBDialect</property>
        <property
name="connection.url">jdbc:mysql://localhost:3306/ecommerce_8am</propert
y>
        <property name="connection.username">root</property>
        <property name="connection.password">ipl2021@dubai</property>
        <property
name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property name="hbm2ddl.auto">update</property>
        <property name="show_sql">true</property>

        <mapping
class="com.firstapp.hibernate.hibernatedemo.Product"></mapping>

        <!-- <property name="hbm2ddl.auto">update</property>
        <property name="show_sql">true</property>
        <mapping class="com.hibernate.demo.Employee"/> -->

    </session-factory>

</hibernate-configuration>

```

Product.java:

```

package com.firstapp.hibernate.hibernatedemo;

import javax.persistence.Column;
import javax.persistence.Entity;

```



```
import javax.persistence.Id;
import javax.persistence.Table;
```

```
@Entity
```

```
@Table
```

```
public class Product {
```

```
    @Id
```

```
    @Column
```

```
    private int id;
```

```
    @Column
```

```
    private String name;
```

```
    @Column
```

```
    private String seller;
```

```
    @Column
```

```
    private int price;
```

```
    @Column
```

```
    private String imageUrl;
```

```
    public int getId() {
```

```
        return id;
```

```
    }
```

```
    public void setId(int id) {
```

```
        this.id = id;
```

```
    }
```

```
    public String getName() {
```

```
        return name;
```

```
    }
```

```
    public void setName(String name) {
```

```
        this.name = name;
```

```
    }
```

```
    public String getSeller() {
```

```
        return seller;
```

```
    }
```

```
    public void setSeller(String seller) {
```

```
        this.seller = seller;
```

```
    }
```

```

        public int getPrice() {
            return price;
        }
        public void setPrice(int price) {
            this.price = price;
        }
        public String getImageUrl() {
            return imageUrl;
        }
        public void setImageUrl(String imageUrl) {
            this.imageUrl = imageUrl;
        }
    }
}

```

App.java:

```

package com.firstapp.hibernate.hibernatedemo;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

import org.hibernate.boot.Metadata;
import org.hibernate.boot.MetadataSources;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

/**
 * Hello world!
 */
public class App
{
    public static void main( String[] args )
    {
        StandardServiceRegistry ssr = new
StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();

```

```

        Metadata meta = new MetadataSources(ssr).getMetadataBuilder().build();

        SessionFactory factory = meta.getSessionFactoryBuilder().build();
        Session session = factory.openSession(); // user 1
        Product product= session.get(Product.class, 101);
        System.out.println(product.getName());
        Product product2= session.get(Product.class, 101);
        System.out.println(product2.getName());
        // Product product3= session.get(Product.class, 103);
        // System.out.println(product3.getName());
        //
        // Product product4= session.get(Product.class, 103);
        // System.out.println(product4.getName());
        session.close();

        Session session2 = factory.openSession(); // user2
        Product p1= session2.get(Product.class, 101);
        System.out.println(p1.getName());

    }
}

```

Pom.xml:

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.firstapp.hibernate</groupId>
  <artifactId>hibernatedemo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>hibernatedemo</name>
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

```

```
</properties>

<dependencies>

<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.3.1.Final</version>
</dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.29</version>
</dependency>
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>3.8.1</version>
  <scope>test</scope>
</dependency>

</dependencies>
</project>
```

Second Level Cache:

Hibernate.cfg.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
  "-//Hibernate/Hibernate Configuration DTD 5.3//EN"
  "http://www.hibernate.org/dtd/hibernate-configuration-5.3.dtd">

  <hibernate-configuration>
```

```

    <session-factory>
      <property
name="dialect">org.hibernate.dialect.MySQL5InnoDBDialect</property>
      <property
name="connection.url">jdbc:mysql://localhost:3306/ecommerce_8am</propert
y>
      <property name="connection.username">root</property>
      <property name="connection.password">ipl2021@dubai</property>
      <property
name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
      <property name="hbm2ddl.auto">update</property>
      <property name="show_sql">true</property>
      <property name="cache.use_second_level_cache">true</property>
    <property
name="cache.region.factory_class">org.hibernate.cache.ehcache.EhCacheR
egionFactory</property>
      <mapping
class="com.firstapp.hibernate.hibernatedemo.Product"></mapping>

    <!-- <property name="hbm2ddl.auto">update</property>
      <property name="show_sql">true</property>
      <mapping class="com.hibernate.demo.Employee"/> -->

    </session-factory>

  </hibernate-configuration>

```

Product.java:

```

package com.firstapp.hibernate.hibernatedemo;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

```

@Entity

@Table

public class Product {

 @Id

 @Column

 private int id;

 @Column

 private String name;

 @Column

 private String seller;

 @Column

 private int price;

 @Column

 private String imageUrl;

 public int getId() {

 return id;

 }

 public void setId(int id) {

 this.id = id;

 }

 public String getName() {

 return name;

 }

 public void setName(String name) {

 this.name = name;

 }

 public String getSeller() {

 return seller;

 }

 public void setSeller(String seller) {

 this.seller = seller;

 }

 public int getPrice() {

 return price;

 }

 public void setPrice(int price) {

 this.price = price;

 }

 public String getImageUrl() {

 return imageUrl;

 }

 public void setImageUrl(String imageUrl) {

 this.imageUrl = imageUrl;

```
}  
  
}
```

App.java:

```
package com.firstapp.hibernate.hibernatedemo;  
  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
  
import org.hibernate.boot.Metadata;  
import org.hibernate.boot.MetadataSources;  
import org.hibernate.boot.registry.StandardServiceRegistry;  
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;  
  
/**  
 * Hello world!  
 */  
public class App  
{  
    public static void main( String[] args )  
    {  
        StandardServiceRegistry ssr = new  
StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();  
        Metadata meta = new MetadataSources(ssr).getMetadataBuilder().build();  
  
        SessionFactory factory = meta.getSessionFactoryBuilder().build();  
        Session session = factory.openSession(); // user 1  
        Product product= session.get(Product.class, 101);  
        System.out.println(product.getName());  
        Product product2= session.get(Product.class, 101);  
        System.out.println(product2.getName());  
        // Product product3= session.get(Product.class, 103);  
        // System.out.println(product3.getName());  
        //  
        // Product product4= session.get(Product.class, 103);  
        // System.out.println(product4.getName());  
        session.close();  
  
        Session session2 = factory.openSession(); // user2  
        Product p1= session2.get(Product.class, 101);
```

```
System.out.println(p1.getName());
```

```
}
```

```
}
```