|  |  |
| --- | --- |
|  | Key-Value store using Servlet and JSP |
|  |  |
|  | Sakil Mallick (Roll – 001510501050)  Internet Technology Lab Assignment 2 Report  11/26/18 |

Report : Assignment 2 of Internet Tech Lab

# Problem statement :

Write a Servlet based query system, take query as input from the user in the text-field to extract the

data from Database and display it to the user in tabular format. All key-value pairs should be stored by

the server. Keys are strings, values could be any objects. If a value is not found, error.html page should

be displayed.

The query format could be any sequence of “get <key>” and/or “put <key> <value>”.

The server should be running on a TCP port. Display the local port, remote port and server port nos.

# Prerequisite (Software and Server Setup) :

1. First of all, we need to download and set up Eclipse. The version we used is Eclipse Photon.
2. Next, we need to set Tomcat Server. The version used here is 9.0.
3. Thereafter, we create a new project ( Dynamic Web Application ) in Eclipse.
4. For database purpose, we use Oracle Database 11g Express Edition. Username and password for the database must be set beforehand.
5. Then we need to create a table to store the key-value pairs.
6. We named the table mydb and it has two columns. First one is the key and second one is the value.
7. Last but not the least, we need to add ojdbc6.jar to the build path. In order to do that we copy the ojdbc6.jar file and paste it into the lib folder inside WEB-INF.
8. So, prerequisites are done. Now we can design the homepage (index.html) and write our servlets for get and put operation for key-value pairs. But at first we need to write code for database connection.

# Design and Working of the Application :

* At first, we create DBConnection.java ( a pure java class ), which contains the getConnection() method. This method creates connection our Oracle SQL server and returns the connection object.

**DBConnection.java :**

package com.sakilmallick;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class DBConnection {

public static Connection getConnection() {

Connection conn = null;

try {

DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());

conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "301931b0239");

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return conn;

}

}

* Now in order to create the database connection only once in the lifecycle of the web application we need to call this getConnection method only inside contextInitialized method of our DBConnectionServletContextListener.java.

**DBConnectionServletContextListener.java :**

package com.sakilmallick;

import java.sql.Connection;

import javax.servlet.ServletContext;

import javax.servlet.ServletContextEvent;

import javax.servlet.ServletContextListener;

import javax.servlet.annotation.WebListener;

/\*\*

\* Application Lifecycle Listener implementation class

\* DBConnectionServletContextListener

\*

\*/

@WebListener

public class DBConnectionServletContextListener implements ServletContextListener {

/\*\*

\* Default constructor.

\*/

public DBConnectionServletContextListener() {

// TODO Auto-generated constructor stub

}

/\*\*

\* @see ServletContextListener#contextDestroyed(ServletContextEvent)

\*/

public void contextDestroyed(ServletContextEvent sce) {

// TODO Auto-generated method stub

}

/\*\*

\* @see ServletContextListener#contextInitialized(ServletContextEvent)

\*/

public void contextInitialized(ServletContextEvent sce) {

// TODO Auto-generated method stub

ServletContext sc = sce.getServletContext();

Connection conn = DBConnection.getConnection();

sc.setAttribute("DBConnection", conn);

}

}

* Now we design the homepage (index.html). The homepage contains three forms, one for showing all entries in the database table, one for getting value for a certain key and another one for putting key-value pairs in the database table. The CSS file corresponding to the homepage is form1.css.

**index.html :**

<!DOCTYPE html>

<html>

<head>

<title>Servlet based GET & PUT</title>

<meta content=*"noindex, nofollow"* name=*"robots"*>

<!-- Include CSS File Here -->

<link href=*"form1.css"* rel=*"stylesheet"*>

</head>

<body>

<form method=*"get"* action=*"show"*>

<div>

<input type=*"submit"* value=*"SHOW ALL"*>

</div>

</form>

<form method=*"get"* action=*"get"*>

<ul class=*"errorMessages"*></ul>

<div>

<label>Enter key to get :</label>

<input type=*"text"* placeholder=*"Key"* name=*"key"* required>

<input type=*"submit"* value=*"GET"*>

</div>

</form>

<form method=*"post"* action=*"put"*>

<ul class=*"errorMessages"*></ul>

<div>

<label>Enter key :</label>

<input type=*"text"* placeholder=*"Key"* name=*"key"* required>

<label>Enter value :</label>

<input type=*"text"* placeholder=*"Value"* name=*"val"* required>

<input type=*"submit"* value=*"PUT"*>

</div>

</form>

<script>

**var** createAllErrors = **function**() {

**var** form = $( **this** ),

errorList = $( "ul.errorMessages", form );

**var** showAllErrorMessages = **function**() {

errorList.empty();

// Find all invalid fields within the form.

**var** invalidFields = form.find( ":invalid" ).each( **function**( index, node ) {

// Find the field's corresponding label

**var** label = $( "label[for=" + node.id + "] "),

// Opera incorrectly does not fill the validationMessage property.

message = node.validationMessage || 'Invalid value.';

errorList

.show()

.append( "<li><span>" + label.html() + "</span> " + message + "</li>" );

});

};

// Support Safari

form.on( "submit", **function**( event ) {

**if** ( **this**.checkValidity && !**this**.checkValidity() ) {

$( **this** ).find( ":invalid" ).first().focus();

event.preventDefault();

}

});

$( "input[type=submit], button:not([type=button])", form )

.on( "click", showAllErrorMessages);

$( "input", form ).on( "keypress", **function**( event ) {

**var** type = $( **this** ).attr( "type" );

**if** ( /date|email|month|number|search|tel|text|time|url|week/.test ( type )

&& event.keyCode == 13 ) {

showAllErrorMessages();

}

});

};

$( "form" ).each( createAllErrors );

</script>

</body>

</html>

**form1.css :**

@import *"http://fonts.googleapis.com/css?family=Raleway"*;

**body**{

font-family:*'Raleway',sans-serif*

}

**img***#logo*{

margin-left:*90px*

}

**div**{

width:*350px*;

height:*auto*;

margin:*50px* *auto* *0*;

padding:*50px*;

background-color:*#EEE*;

color:*#333*;

border:*2px* *solid* *#C2D6FF*;

border-radius:*40px* *0* *40px* *0*

}

**label**{

font-size:*15px*;

font-weight:*700*

}

**input**{

background-image:*url(../images/icon\_name.png)*;

background-repeat:*no-repeat*;

background-position:*6px*;

border:*1px* *solid* *#DADADA*;

margin-top:*10px*;

margin-bottom:*10px*;

padding-left:*35px*;

width:*310px*;

height:*30px*;

font-size:*14px*;

box-shadow:*0* *0* *10px*;

-webkit-box-shadow:*0* *0* *10px*;

/\* For I.E \*/

-moz-box-shadow:*0* *0* *10px*;

/\* For Mozilla Web Browser \*/

border-radius:*5px*;

-webkit-border-radius:*5px*;

/\* For I.E \*/

-moz-border-radius:*5px*

/\* For Mozilla Web Browser \*/

}

**input***#submit* {

background-color:*#cc7a66*;

border-radius:*5px*;

border:*none*;

padding:*10px* *25px*;

color:*#FFF*;

text-shadow:*1px* *1px* *1px* *#949494*;

margin-left:*120px*

}

**input***#submit:hover*{

background-color:*#FF9980*

}

* Pressing the show all button calls the ShowAllServlet which selects all the entries from the database table (mydb) and puts them in a resultset and passes it to the show.jsp file to render the view using HttpServletRequest object.

**ShowAllServlet.java :**

package com.sakilmallick;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class ShowAllServlet

\*/

@WebServlet("/show")

public class ShowAllServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public ShowAllServlet() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#service(HttpServletRequest request, HttpServletResponse

\* response)

\*/

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

String sql = "select \* from mydb";

Connection conn = (Connection) getServletContext().getAttribute("DBConnection");

PreparedStatement select = conn.prepareStatement(sql);

ResultSet result = select.executeQuery();

request.setAttribute("result", result);

RequestDispatcher rd = request.getRequestDispatcher("show.jsp");

rd.forward(request, response);

} catch (Exception e) {

e.printStackTrace();

}

}

}

**show.jsp :**

<%@ page language=*"java"* import=*"java.util.\*"* import=*"java.sql.\*"* import=*"javax.servlet.\*"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<style>

*#customers* {

font-family: *"Trebuchet MS", Arial, Helvetica, sans-serif*;

border-collapse: *collapse*;

width: *100%*;

}

*#customers* **td,** *#customers* **th** {

border: *1px solid #ddd*;

padding: *8px*;

}

*#customers* **tr***:nth-child(even)*{background-color: *#f2f2f2*;}

*#customers* **tr***:hover* {background-color: *#ddd*;}

*#customers* **th** {

padding-top: *12px*;

padding-bottom: *12px*;

text-align: *left*;

background-color: *#4CAF50*;

color: *white*;

}

</style>

</head>

<body>

<%

out.println("Local port : " + request.getLocalPort());

out.println("Remote port : " + request.getRemotePort());

out.println("Server port :" + request.getServerPort());

%>

<table id=*"customers"*>

<tr>

<th>Key</th>

<th>Value</th>

</tr>

<!-- Table contents below -->

<%

ResultSet result = (ResultSet)request.getAttribute("result");

String str="";

**while** (result.next()) {

str += "<tr><td>" + result.getString("key") + "</td><td>" + result.getString("val") + "</td></tr>";

// out.println(result.getString("key") + " : " + result.getString("val"));

}

out.println(str);

%>

</table>

</body>

</html>

* Next form in the homepage is for getting value for a certain key entered into the text field. Upon clicking the submit button, the GetServlet is called which searches in the table (mydb) for the entry corresponding to key entered by user. If there is any entry then it is shown in tabular form, otherwise the value field is shown as NULL. Get.jsp file is used create the view using the resultset obtained in the GetServlet shared through HttpServletRequest object.

**GetServlet.java :**

package com.sakilmallick;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class GetServlet

\*/

@WebServlet("/get")

public class GetServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public GetServlet() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#service(HttpServletRequest request, HttpServletResponse

\* response)

\*/

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

String key = request.getParameter("key");

String sql = "select key,val from mydb where key='" + key + "'";

PreparedStatement select;

Connection conn = (Connection) getServletContext().getAttribute("DBConnection");

select = conn.prepareStatement(sql);

ResultSet result = select.executeQuery();

request.setAttribute("result", result);

RequestDispatcher rd = request.getRequestDispatcher("get.jsp");

rd.forward(request, response);

// out.println("Local port : " + request.getLocalPort());

// out.println("Remote port : " + request.getRemotePort());

// out.println("Server port :" + request.getServerPort());

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

**get.jsp :**

<%@ page language=*"java"* import=*"java.util.\*"* import=*"java.sql.\*"* import=*"javax.servlet.\*"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<style>

*#customers* {

font-family: *"Trebuchet MS", Arial, Helvetica, sans-serif*;

border-collapse: *collapse*;

width: *100%*;

}

*#customers* **td,** *#customers* **th** {

border: *1px solid #ddd*;

padding: *8px*;

}

*#customers* **tr***:nth-child(even)*{background-color: *#f2f2f2*;}

*#customers* **tr***:hover* {background-color: *#ddd*;}

*#customers* **th** {

padding-top: *12px*;

padding-bottom: *12px*;

text-align: *left*;

background-color: *#4CAF50*;

color: *white*;

}

</style>

</head>

<body>

<%

out.println("Local port : " + request.getLocalPort());

out.println("Remote port : " + request.getRemotePort());

out.println("Server port :" + request.getServerPort());

%>

<table id=*"customers"*>

<tr>

<th>Key</th>

<th>Value</th>

</tr>

<!-- Table contents below -->

<%

ResultSet result = (ResultSet)request.getAttribute("result");

String str;

**if** (result.next()) {

str = "<tr><td>" + result.getString("key") + "</td><td>" + result.getString("val") + "</td></tr>";

out.println(str);

} **else** {

str = "<tr><td>" + result.getString("key") + "</td><td>" + "NULL" + "</td></tr>";

out.println(str);

}

%>

</table>

</body>

</html>

* The last form in the homepage is for inserting key-value pairs into the database table (mydb). User enters key and value, submits them. Upon submitting, the PutServlet is called, which searches for the key if it already exists in the table. If it does, then error page (error.jsp) is shown. Otherwise data is inserted into the table and confirmation page is shown to the user (put.jsp).

**PutServlet.java :**

package com.sakilmallick;

import java.io.IOException;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class PutServlet

\*/

@WebServlet("/put")

public class PutServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public PutServlet() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#service(HttpServletRequest request, HttpServletResponse

\* response)

\*/

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

try {

String key = request.getParameter("key");

String sql = "select key,val from mydb where key='" + key + "'";

PreparedStatement select;

Connection conn = (Connection) getServletContext().getAttribute("DBConnection");

select = conn.prepareStatement(sql);

ResultSet result = select.executeQuery();

if (result.next()) {

RequestDispatcher rd = request.getRequestDispatcher("error.jsp");

rd.forward(request, response);

} else {

String val = request.getParameter("val");

System.out.println("key : " + key);

System.out.println("val : " + val);

PreparedStatement ps = conn.prepareStatement("insert into mydb values(?,?)");

ps.setString(1, key);

ps.setString(2, val);

int ret = ps.executeUpdate();

if (ret > 0) {

RequestDispatcher rd = request.getRequestDispatcher("put.jsp");

rd.forward(request, response);

}

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

**error.jsp :**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

</head>

<body>

<h1>Key already exists!!</h1>

</body>

</html>

**put.jsp :**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"*>

<style>

*.alert* {

padding: *20px*;

background-color: *#008000*;

color: *white*;

}

*.closebtn* {

margin-left: *15px*;

color: *white*;

font-weight: *bold*;

float: *right*;

font-size: *22px*;

line-height: *20px*;

cursor: *pointer*;

transition: *0.3s*;

}

*.closebtn:hover* {

color: *black*;

}

</style>

</head>

<body>

<%

out.println("Local port : " + request.getLocalPort());

out.println("Remote port : " + request.getRemotePort());

out.println("Server port :" + request.getServerPort());

out.println("remote adress :" + request.getRemoteAddr());

out.println("local addr :" + request.getLocalAddr());

%>

<div class=*"alert"*>

<span class=*"closebtn"* onclick="this.parentElement.style.display='none';">&times;</span>

<strong>Data Inserted!</strong>

</div>

</body>

</html>

**P.S. – The URL mappings are done using @WebServlet() annotation. So web.xml only contains basic (default) entries.**

**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\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>

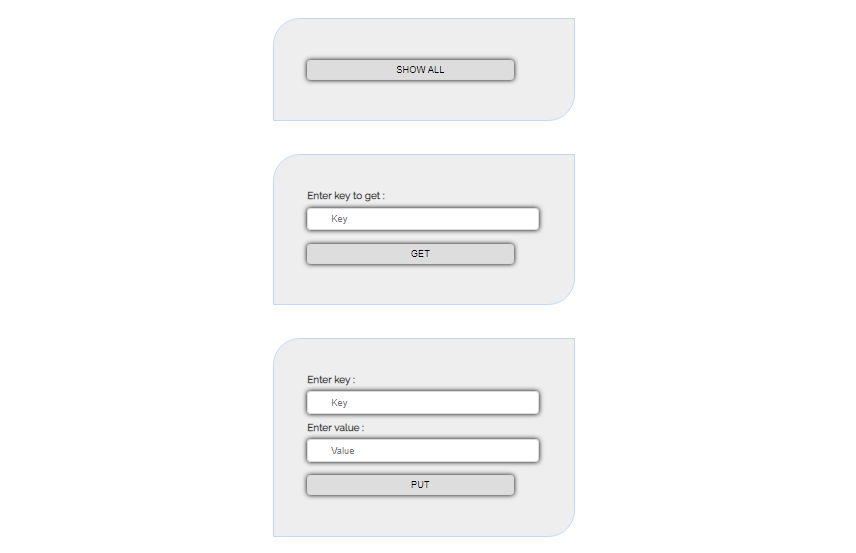
<listener>

<listener-class>com.sakilmallick.DBConnectionServletContextListener</listener-class>

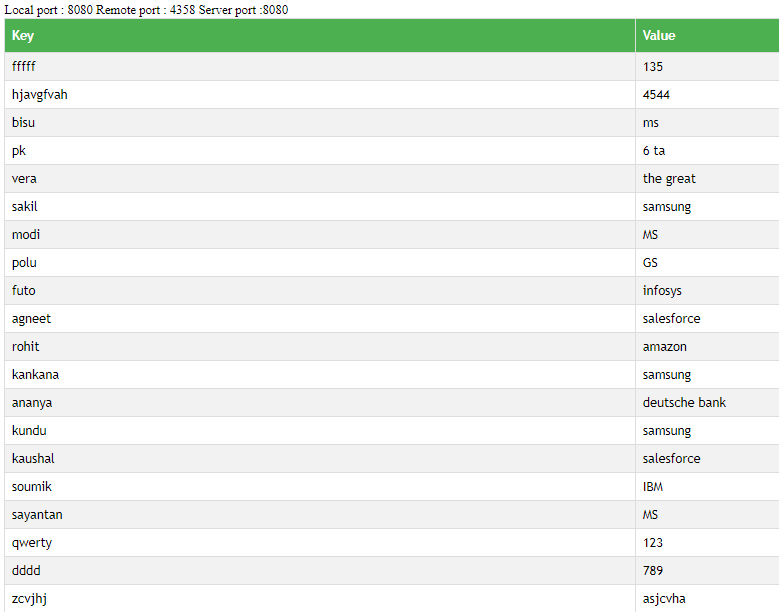
</listener>

</web-app>

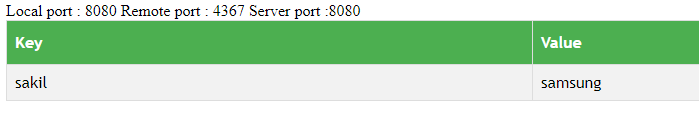
# Output :



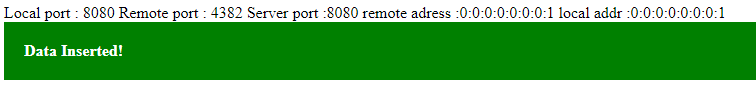
**Homepage ( index.html )**



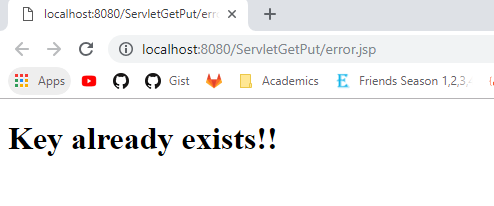
**Output of Show All form (ShowAllServlet.java -> show.jsp)**



**Output of Get form (GetServlet.java -> get.jsp)**



**Output of Put form (PutServlet.java -> put.jsp)**



**Output of Put form when key already exists in the table**

**(PutServlet.java -> error.jsp)**