Currency Converter Java Mini Project

Currency converter (or currency exchange) is a mini project coded in Java programming language. This simple application provides a web-based interface for exchanging/converting money from one currency (say $) to another currency (say €).

The complete source code of currency exchange application can be downloaded from the link below. As this is just a mini project, project report and documentation are not available. You can go through the description below for project abstract.

Currency Converter Project Abstract:

Different countries use different currency, and there is daily variation in these currencies relative to one another. Those who transfer money from one country to another (one currency to another) must be updated with the latest currency exchange rates in the market.

Currency converter mini project is built keeping this thing in mind. It is simply a calculator-like app developed using Ajax, Java servlets web features. In this application, there is regular update about currency of every country by which it displays present currency market value and conversion rate.

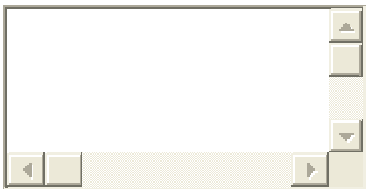
Such application can be used by any user, but it is mainly useful for business, shares, and finance related areas where money transfer and currency exchange takes place on a daily basis.

In this currency converter app, users are provided with an option to select the type of conversion, i.e. from “this” currency to “that” currency. This simple feature allows users to enter amount to be converted (say currency in Dollars), and display the converted amount (say currency in Euro).

Here’s a sample code of this mini project:

Currency Exchange (Convert.Java)

Java



```
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5  package com.exchange;
6
7  import java.io.*;
8  import java.net.*;
9  import java.util.*;
10 import javax.servlet.*;
11 import javax.servlet.http.*;
12 import java.io.InputStream;
13 import java.net.*;
14 import com.google.gson.*;
15
16
17 /**
18  *
19  * @author pakallis
20  */
21 class Recv
22 {
23                 private String lhs;
24                 private String rhs;
25                 private String error;
26                 private String icc;
27                 public Recv()
```

```
28              {
29
30              }
31              public String getLhs()
32              {
33                      return lhs;
34              }
35              public String getRhs()
36              {
37                      return rhs;
38              }
39  }
40  public class Convert extends HttpServlet {
41
42      /**
43       * Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods.
44       * @param request servlet request
45       * @param response servlet response
46       * @throws ServletException if a servlet-specific error occurs
47       * @throws IOException if an I/O error occurs
48       */
49      protected void processRequest(HttpServletRequest req, HttpServletResponse resp)
50              throws ServletException, IOException {
51          String query = "";
52          String amount = "";
53          String curTo = "";
54          String curFrom = "";
55          String submit = "";
56          String res = "";
57          HttpSession session;
58          resp.setContentType("text/html;charset=UTF-8");
```

```
59      PrintWriter out = resp.getWriter();
60
61
62      /*Read request parameters*/
63      amount = req.getParameter("amount");
64      curTo = req.getParameter("to");
65      curFrom = req.getParameter("from");
66
67
68
69      /*Open a connection to google and read the result*/
70      try {
71
72          query = "http://www.google.com/ig/calculator?hl=en&q=" + amount + curFrom + "=?" + curTo;
73          URL url = new URL(query);
74          InputStreamReader stream = new InputStreamReader(url.openStream());
75          BufferedReader in = new BufferedReader(stream);
76          String str = "";
77          String temp = "";
78
79          while ((temp = in.readLine()) != null) {
80              str = str + temp;
81          }
82
83
84          /*Parse the result which is in json format*/
85          Gson gson = new Gson();
86          Recv st = gson.fromJson(str, Recv.class);
87          String rhs = st.getRhs();
88          rhs = rhs.replaceAll("\u00a0", "");
89
```

```
90              /*we do the check in order to print the additional word(millions,billions etc)*/
91              StringTokenizer strto = new StringTokenizer(rhs);
92              String nextToken;
93              out.write(strto.nextToken());
94              nextToken = strto.nextToken();
95              if( nextToken.equals("million") || nextToken.equals("billion") || nextToken.equals("trillion"))
96              {
97                  out.println(" "+nextToken);
98              }
99
100
101         } catch (NumberFormatException e) {
102             out.println("The given amount is not a valid number");
103         }
104
105
106     }
107
108     // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
109     /**
110      * Handles the HTTP <code>GET</code> method.
111      * @param request servlet request
112      * @param response servlet response
113      * @throws ServletException if a servlet-specific error occurs
114      * @throws IOException if an I/O error occurs
115      */
116     @Override
117     protected void doGet(HttpServletRequest request, HttpServletResponse response)
118             throws ServletException, IOException {
119         processRequest(request, response);
120     }
```

```
121
122     /**
123      * Handles the HTTP <code>POST</code> method.
124      * @param request servlet request
125      * @param response servlet response
126      * @throws ServletException if a servlet-specific error occurs
127      * @throws IOException if an I/O error occurs
128      */
129     @Override
130     protected void doPost(HttpServletRequest request, HttpServletResponse response)
131             throws ServletException, IOException {
132         processRequest(request, response);
133     }
134
135     /**
136      * Returns a short description of the servlet.
137      * @return a String containing servlet description
138      */
139     @Override
140     public String getServletInfo() {
141         return "Short description";
142     }// </editor-fold>
143 }
```