

## View.java

```
1 package com.tavor.project;
2
3 import java.awt.Dimension;
4 import java.awt.EventQueue;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7 import java.awt.event.MouseEvent;
8 import javax.swing.*;
9 import org.apache.log4j.Logger;
10 import org.apache.log4j.PropertyConfigurator;
11
12 @SuppressWarnings({ "serial", "rawtypes" })
13 public class View extends JFrame {
14 //class fields
15     private JButton jButtonConvert;
16     private JButton jButtonClear;
17     private JComboBox jComboBoxFrom;
18     private JComboBox jComboBoxTo;
19     private JLabel jLabelEnterAmount;
20     private JLabel jLabelFrom;
21     private JLabel jLabelTo;
22     private JLabel jLabelDate;
23     private JLabel jLabelResult;
24     private JScrollPane jScrollPane1;
25     private static JTable jTableRates;
26     private JTextField jTextFieldInput;
27     private JTextField jTextFieldDate;
28     private JTextField jTextFieldResult;
29     private static Model model=new Model(); //logic module
30     public static Logger myLogger = Logger.getLogger(View.class); //log file
31
32     //constructor
33     public View() {
34         initComponents();
35     }
36
37     @SuppressWarnings("unchecked")
38     private void initComponents() { //creating gui components
39         jTextFieldInput = new JTextField();
40         jLabelEnterAmount = new JLabel();
41         jComboBoxFrom = new JComboBox();
42         jLabelFrom = new JLabel();
43         jComboBoxTo = new JComboBox();
44         jLabelTo = new JLabel();
45         jScrollPane1 = new JScrollPane();
46         jTableRates = new JTable();
47         jButtonConvert = new JButton();
```

## View.java

```
48     jTextFieldDate = new JTextField();
49     jTextFieldResult = new JTextField();
50     jLabelDate = new JLabel();
51     jLabelResult = new JLabel();
52     jButtonClear=new JButton();
53
54     setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
55     setSize(new Dimension(500,500));
56     setLocationRelativeTo(null);
57     setResizable(false);
58     setTitle("Currency Exchange Rate");
59     jTextFieldInput.setText("Enter here");
60     requestFocusInWindow(true);
61     setVisible(true);
62     jLabelEnterAmount.setText("Enter Amount");
63     requestFocusInWindow() ;
64
65     jComboBoxFrom.setModel(new DefaultComboBoxModel(new String[]
66 {
67         //set From button options
68         "ILS-Israel", "AUD-Australia", "CAD-Canada",
69         "CHF-Switzerland",
70         "DKK-Denmark", "EGP-Egypt", "EUR-EMU", "GBP-Great Britain",
71         "JOD-Jordan", "JPY-Japan", "LBP-Lebanon", "NOK-Norway",
72         "SEK-Sweden", "USD-USA", "ZAR-South Africa" }));
73
74     jLabelFrom.setText("From");
75
76     jComboBoxTo.setModel(new DefaultComboBoxModel(new String[]
77 {
78         //set To button options
79         "ILS-Israel", "AUD-Australia", "CAD-Canada",
80         "CHF-Switzerland",
81         "DKK-Denmark", "EGP-Egypt", "EUR-EMU", "GBP-Great Britain",
82         "JOD-Jordan", "JPY-Japan", "LBP-Lebanon", "NOK-Norway",
83         "SEK-Sweden", "USD-USA", "ZAR-South Africa" }));
84
85     jLabelTo.setText("To");
86     jLabelDate.setText("Date");
87     jLabelResult.setText("Result");
88
89     // table- Currencies Rates
90     jTableRates.setModel(new javax.swing.table.DefaultTableModel(
91         new Object[][] { { null, null }, { null, null },
92             { null, null }, { null, null }, { null, null },
93             { null, null }, { null, null }, { null, null },
94             { null, null }, { null, null }, { null, null } },
95         new String[] { "Currency", "In Shekel" }));
```

## View.java

```
91     jTableRates.setShowVerticalLines(true);
92     jScrollPane1.setViewportViewView(jTableRates);
93     jButtonConvert.setBackground(new java.awt.Color(255, 102, 0));
94     jButtonConvert.setText("Convert");
95     jButtonClear.setBackground(java.awt.Color.CYAN);
96     jButtonClear.setText("Clear");
97
98 //add all the component into the container
99 GroupLayout layout = new GroupLayout(getContentPane());
100 getContentPane().setLayout(layout);
101 layout.setHorizontalGroup(
102     layout.createParallelGroup(GroupLayout.Alignment.LEADING)
103         .addGroup(layout.createSequentialGroup()
104             .addGap(24, 24, 24)
105             .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING)
106                 .addComponent(jTextFieldInput, GroupLayout.PREFERRED_SIZE, 121,
107                     GroupLayout.PREFERRED_SIZE)
108                 .addComponent(jLabelEnterAmount))
109             .addGap(76, 76, 76)
110             .addGroup(layout.createParallelGroup(GroupLayout.Alignment.TRAILING)
111                 .addGroup(layout.createSequentialGroup()
112                     .addComponent(jLabelTo)
113                     .addGap(306, Short.MAX_VALUE))
114                 .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING,
115                     false)
116                     .addComponent(jComboBoxTo, 0, 176, Short.MAX_VALUE)
117                     .addComponent(jLabelFrom, GroupLayout.Alignment.LEADING,
118                         GroupLayout.PREFERRED_SIZE,
119                         Short.MAX_VALUE))
120                 .addGroup(layout.createSequentialGroup()
121                     .addComponent(jScrollPane1, GroupLayout.PREFERRED_SIZE, 179,
122                         GroupLayout.PREFERRED_SIZE)
123                     .addGap(67, 67, 67)
124                     .addGroup(layout.createParallelGroup(GroupLayout.Alignment.LEADING, false)
125                         .addComponent(jLabelResult)
126                         .addComponent(jButtonConvert, GroupLayout.PREFERRED_SIZE, 105,
127                             GroupLayout.PREFERRED_SIZE)
128                         .addComponent(jButtonClear, GroupLayout.PREFERRED_SIZE, 105,
129                             GroupLayout.PREFERRED_SIZE)
130                         .addComponent(jTextFieldResult, GroupLayout.DEFAULT_SIZE, 194,
131                             Short.MAX_VALUE)
132                         .addComponent(jLabelDate, GroupLayout.PREFERRED_SIZE, 70,
```

## View.java

```
        GroupLayout.PREFERRED_SIZE)
131    .addComponent(jTextFieldDate)))
132    .addGroup(layout.createSequentialGroup())
133    .addGap(177, 177, 177)))
134    .addContainerGap(GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
135);
136 layout.setVerticalGroup(
137     layout.createParallelGroup(GroupLayout.Alignment.LEADING)
138     .addGroup(layout.createSequentialGroup())
139     .addContainerGap()
140     .addGap(8, 8, 8)
141     .addGroup(layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
142     .addComponent(jLabelEnterAmount)
143     .addComponent(jLabelFrom))
144     .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
145     .addGroup(layout.createParallelGroup(GroupLayout.Alignment.BASELINE)
146     .addComponent(jTextFieldInput, GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
147     .addComponent(jComboBoxFrom, GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
148     .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)
149     .addComponent(jLabelTo)
150     .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
151     .addComponent(jComboBoxTo, GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
152     .addGroup(layout.createParallelGroup(GroupLayout.Alignment.TRAILING)
153     .addGroup(layout.createSequentialGroup())
154     .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED, 18,
        Short.MAX_VALUE)
155     .addComponent(jScrollPane1, GroupLayout.PREFERRED_SIZE, 253,
        GroupLayout.PREFERRED_SIZE))
156     .addGroup(layout.createSequentialGroup())
157     .addGap(26, 26, 26)
158     .addComponent(jButtonConvert, GroupLayout.PREFERRED_SIZE, 67,
        GroupLayout.PREFERRED_SIZE)
159     .addComponent(jButtonClear, GroupLayout.PREFERRED_SIZE, 67,
        GroupLayout.PREFERRED_SIZE)
160     .addGap(18, 18, 18)
161     .addComponent(jLabelResult)
162     .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
163     .addComponent(jTextFieldResult, GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
164     .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)
165     .addComponent(jLabelDate)
166     .addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)
167     .addComponent(jTextFieldDate, GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
```

## View.java

```
168     .addGap(0, 0, Short.MAX_VALUE)))
169     .addGap(21, 21, 21))
170     );
171 pack();
172
173 // Listeners
174
175     jTextFieldInput.addMouseListener(new java.awt.event.MouseListener() {
176         //jTextFieldInput listener
177         @Override
178         public void mouseReleased(MouseEvent e) {
179             // TODO Auto-generated method stub
180         }
181
182         @Override
183         public void mousePressed(MouseEvent e) {
184             // TODO Auto-generated method stub
185         }
186
187         @Override
188         public void mouseExited(MouseEvent e) {
189             // TODO Auto-generated method stub
190         }
191
192         @Override
193         public void mouseEntered(MouseEvent e) {
194             // TODO Auto-generated method stub
195         }
196
197         @Override
198         public void mouseClicked(MouseEvent e) { //delete message on
199             click jTextFieldInput
200                 if(jTextFieldInput.getText().equals("Enter here"))
201                     {jTextFieldInput.setText("");}
202             }
203         });
204
205     jButtonClear.addActionListener(new ActionListener() { //jButtonClear
206         listener
207         public void actionPerformed(ActionEvent e) { //clear fields
208             jComboBoxFrom.setSelectedItem("ILS-Israel");
209             jComboBoxTo.setSelectedItem("ILS-Israel");
210             jTextFieldInput.setText("Enter here");
211             jTextFieldDate.setText("");
```

## View.java

```
212         jTextFieldResult.setText("");
213         myLogger.info("the user pressed on clear");
214     }
215 });
216
217     jButtonConvert.addActionListener(new ActionListener() {
218         //jButtonConvert listener
219
220         public void actionPerformed(ActionEvent arg0) {
221
222             String amount = jTextFieldInput.getText(); //get amount input
223             String from = jComboBoxFrom.getSelectedItem().toString();
224             String to = jComboBoxTo.getSelectedItem().toString();
225
226             from = from.substring(0, 3); //get country code
227             to = to.substring(0, 3); //get country code
228             try { //Convert the data using model
229                 double a = 0;
230                 a = Double.parseDouble(amount);
231                 jTextFieldResult.setText(Double.toString(model.Convert(fro
232 m, a, to)));
233                 jTextFieldDate.setText(model.getDate());
234                 myLogger.info("The user convert " + amount + " " + from+ "
235 to " + to + ".");
236             } catch (NumberFormatException e) { //exception cause by
237                 illegal input
238                 myLogger.error("illegal input");
239                 JOptionPane.showMessageDialog(new JFrame(),
240                     "Please insert numbers only!");
241             } catch (NullPointerException e) { //exception
242                 NullPointerException
243                 myLogger.error("Error NullPointerException");
244             }
245         }
246     });
247 }
248 //end of initComponents
249
250 //update the table, using model module
251 public static void fillTableData() {
252     String data[]=model.getData();
253     for (int i = 0,j=0; i < data.length; i++,++j) {
254         jTableRates.setValueAt(data[i], j, 0);
255         jTableRates.setValueAt(data[++i], j, 1);
256     }
257 }
```

## View.java

```
254     myLogger.info("table update!");
255
256 }
257
258
259 //run application
260 private static void Start() {
261     PropertyConfigurator.configure(".\\log4j.properties"); //set log path
properties
262     myLogger.info("application runing");
263     try { //set gui properties
264
265         for (UIManager.LookAndFeelInfo info : UIManager
266             .getInstalledLookAndFeels()) {
267             if ("Nimbus".equals(info.getName())) {
268                 UIManager.setLookAndFeel(info.getClassName());
269                 break;
270             }
271         }
272
273     } catch (ClassNotFoundException ex) {
274         myLogger.error("gui fail",ex);
275     } catch (InstantiationException ex) {
276         myLogger.error("gui fail",ex);
277
278     } catch (IllegalAccessException ex) {
279         myLogger.error("gui fail",ex);
280
281     } catch (javax.swing.UnsupportedLookAndFeelException ex) {
282         myLogger.error("gui fail",ex);
283
284     }
285
286
287 //thread thats run the application
288 EventQueue.invokeLater(new Runnable() {
289     public void run() {
290         new View(); //run gui
291         fillTableData(); //update table 1st time
292
293     }
294 });
295
296 //separated thread thats update the table every x milliseconds
297 Thread t = new Thread(new Runnable() {
298     public void run() {
299         while (true) {
```

## View.java

```
300         try {
301             Thread.sleep(180000);
302             fillTableData();
303         } catch (InterruptedException e) {
304             myLogger.error("Error table",e);
305
306         }
307     }
308 }
309 });
310 t.start();
311 }
312
313 public static void main(String args[]) {
314     Start();
315 }
316 }
317
```