TableReport.java

```
import com.cete.dynamicpdf.Grayscale;
import com.cete.dynamicpdf.Document;
import com.cete.dynamicpdf.Font;
import com.cete.dynamicpdf.Page;
import com.cete.dynamicpdf.PageSize;
import com.cete.dynamicpdf.TextAlign;
import com.cete.dynamicpdf.pageelements.*;
import com.cete.dynamicpdf.VAlign;
import java.sql.*;

import java.text.NumberFormat;
import java.util.Locale;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;

public class TableReport extends HttpServlet {

    ServletOutputStream sOut;

    Connection connection;

    public void init(ServletConfig servletConfig) throws ServletException {
        super.init(servletConfig);

    }

    public void doGet(HttpServletRequest req, HttpServletResponse res)
   throws IOException,ServletException
    {
        CeTeConnection ceTe = (CeTeConnection)getServletContext().getAttribute("cetecon");
        connection = ceTe.getConnection();
        sOut = res.getOutputStream();

        // Create a document and set it's properties
        Document objDocument = new Document();
        objDocument.setCreator("Table.java");
        objDocument.setAuthor("ceTe Software");
        objDocument.setTitle("Table Example");

        ResultSet data = getContactListData();
        Table2 table = new Table2(0, 0, 512, 676, Font.getHelvetica(), 12);
        table.getBorder().setWidth(1f);
```

TableReport.java

```
50          table.getCellDefault().getBorder().setWidth(1f);
51          table.setRepeatColumnHeaderCount(1);
52          table.setRepeatRowHeaderCount(1);
53
54          // Builds the report
55          buildTable(data, table);
56
57          addTableToPage(objDocument, table, "(1, 1)");
58          addTableToPage(objDocument, table.getOverflowColumns(), "(1, 2)");
59          Table2 objOverflowRowTable = table.getOverflowRows();
60          addTableToPage(objDocument, objOverflowRowTable, "(2, 1)");
61          addTableToPage(objDocument, objOverflowRowTable.getOverflowColumns(),
62                  "(2, 2)");
63
64          // Outputs the TableReport to the current web page
65          objDocument.drawToWeb(req, res, sOut, "TableReport.pdf");
66          ceTe.close();
67          sOut.close();
68      }
69
70      private void addTableToPage(Document document, Table2 table,
71              String pageLabel) {
72          Page page = new Page(PageSize.LETTER);
73          if (table != null) {
74              page.getElements().add(table);
75          }
76          page.getElements().add(new Label(pageLabel, 0, page.getDimensions()
77          .getBody().getHeight() - 12, page.getDimensions().getBody().getWidth(), 12,
78                  Font.getHelvetica(), 12, TextAlign.CENTER));
79          document.getPages().add(page);
80      }
81
82      private void buildTable(ResultSet data, Table2 table) {
83          createColumns(table);
84          createRowHeadings(table);
85          try {
86              while (data.next()) {
87                  createRow(table, data);
88              }
89          } catch (SQLException ex) {
90              System.err.println("cannot move ResultSet :"+ex.getMessage());
91          }
92      }
93
94      private void createRowHeadings(Table2 table)    {
95
96          Row2 row = table.getRows().add(40, Font.getTimesBold(), 12,
97                  Grayscale.getBlack(), Grayscale.getLightGrey());
98          row.getCellDefault().setAlign(TextAlign.CENTER);
```

TableReport.java

```
99          row.getCellDefault().setVAlign(VAlign.TOP);
100         row.getCells().add("ID");
101         row.getCells().add("Product Name");
102         row.getCells().add("Supplier ID");
103         row.getCells().add("Category ID");
104         row.getCells().add("Quantity Per Unit");
105         row.getCells().add("Unit Price");
106         row.getCells().add("Unit In Stock");
107         row.getCells().add("Units On Order");
108         row.getCells().add("Reorder Level");
109         row.getCells().add("Discontinued");
110     }
111
112     private void createRow(Table2 table, ResultSet data) {
113         Row2 row = table.getRows().add(20);
114         try {
115             row.getCells().add(String.valueOf(data.getInt(1)),
116                     Font.getHelvetica(), 12, Grayscale.getBlack(),
117                     Grayscale.getLightGrey(), 1);
118             String s1 = data.getString(2);
119             row.getCells().add(s1);
120             row.getCells().add(String.valueOf(data.getInt(3)));
121             row.getCells().add(String.valueOf(data.getInt(4)));
122             row.getCells().add(data.getString(5));
123             NumberFormat n = NumberFormat.getCurrencyInstance(Locale.US);
124             row.getCells().add(n.format(data.getBigDecimal(6)));
125             row.getCells().add(String.valueOf(data.getShort(7)));
126             row.getCells().add(String.valueOf(data.getShort(8)));
127             row.getCells().add(String.valueOf(data.getShort(9)));
128             row.getCells().add(String.valueOf(data.getBoolean(10)));
129         } catch (SQLException ex) {
130             System.err.println("cannot retrieve data from ResultSet :"+ex.
131                     getMessage());
132         }
133     }
134
135     private void createColumns(Table2 table) {
136         table.getColumns().add(25);
137         table.getColumns().add(150);
138         table.getColumns().add(90);
139         table.getColumns().add(90);
140         table.getColumns().add(120);
141         table.getColumns().add(60);
142         table.getColumns().add(90);
143         table.getColumns().add(90);
144         table.getColumns().add(90);
145         table.getColumns().add(90);
146     }
147
```

TableReport.java

```
148     private ResultSet getContactListData() {
149         // Creates a ResultSet for the report
150         ResultSet data = null;
151         try {
152             Statement st  = connection.createStatement();
153 
154             data  = st.executeQuery("SELECT * FROM "+
155                     "Products where (UnitPrice > 15)");
156         } catch (SQLException ex) {
157             ex.printStackTrace(System.err);
158         }
159         return data;
160     }
161 
162 
163 }
```