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.math.BigDecimal;
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);
```

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

```
        row.getCellDefault().setAlign(TextAlign.CENTER);
        row.getCellDefault().setVAlign(VAlign.TOP);
        row.getCells().add("ID");
        row.getCells().add("Product Name");
        row.getCells().add("Supplier ID");
        row.getCells().add("Category ID");
        row.getCells().add("Quantity Per Unit");
        row.getCells().add("Unit Price");
        row.getCells().add("Unit In Stock");
        row.getCells().add("Units On Order");
        row.getCells().add("Reorder Level");
        row.getCells().add("Discontinued");
    }

    private void createRow(Table2 table, ResultSet data) {
        Row2 row = table.getRows().add(20);
        try {
            row.getCells().add(String.valueOf(data.getInt(1)),
                    Font.getHelvetica(), 12, Grayscale.getBlack(),
                    Grayscale.getLightGrey(), 1);
            String s1 = data.getString(2);
            row.getCells().add(s1);
            row.getCells().add(String.valueOf(data.getInt(3)));
            row.getCells().add(String.valueOf(data.getInt(4)));
            row.getCells().add(data.getString(5));
            BigDecimal rate = data.getBigDecimal(6);
            rate.setScale(2, BigDecimal.ROUND_HALF_EVEN);
            NumberFormat n = NumberFormat.getCurrencyInstance(Locale.US);
            row.getCells().add(n.format(rate));
            row.getCells().add(String.valueOf(data.getShort(7)));
            row.getCells().add(String.valueOf(data.getShort(8)));
            row.getCells().add(String.valueOf(data.getShort(9)));
            row.getCells().add(String.valueOf(data.getBoolean(10)));
        } catch (SQLException ex) {
            System.err.println("cannot retrieve data from ResultSet :"+ex.
                    getMessage());
        }
    }

    private void createColumns(Table2 table) {
        table.getColumns().add(25);
        table.getColumns().add(150);
        table.getColumns().add(90);
        table.getColumns().add(90);
        table.getColumns().add(120);
        table.getColumns().add(60);
        table.getColumns().add(90);
        table.getColumns().add(90);
        table.getColumns().add(90);
```

```
148         table.getColumns().add(90);
149     }
150 
151     private ResultSet getContactListData() {
152         // Creates a ResultSet for the report
153         ResultSet data = null;
154         try {
155             Statement st  = connection.createStatement();
156 
157             data  = st.executeQuery("SELECT * FROM "+
158                     "Products where (UnitPrice > 15)");
159         } catch (SQLException ex) {
160             ex.printStackTrace(System.err);
161         }
162         return data;
163     }
164 
165 
166 }
```