

```
1
2 import com.cete.dynamicpdf.*;
3 import com.cete.dynamicpdf.pageelements.Label;
4 import com.cete.dynamicpdf.pageelements.Row2;
5 import com.cete.dynamicpdf.pageelements.Table2;
6 import java.text.SimpleDateFormat;
7 import java.util.Calendar;
8 import java.util.Date;
9 import java.util.GregorianCalendar;
10 import java.io.IOException;
11 import javax.servlet.ServletException;
12 import javax.servlet.ServletOutputStream;
13 import javax.servlet.http.HttpServlet;
14 import javax.servlet.http.HttpServletRequest;
15 import javax.servlet.http.HttpServletResponse;
16
17
18 public class TableCalendar extends HttpServlet {
19     ServletOutputStream sOut;
20     public void doGet(HttpServletRequest req, HttpServletResponse res)
21     throws IOException, ServletException {
22         sOut = res.getOutputStream();
23         Document document = new Document(); // Create a New PDF Document.
24         document.setCreator("TableCalendar.java");
25         document.setAuthor("Your Name");
26         document.setTitle("Table Calendar Example");
27         Calendar cal = new GregorianCalendar();
28         document.getPages().add(createCalendarPage(cal));
29         cal.add(Calendar.MONTH, 1);
30         document.getPages().add(createCalendarPage(cal));
31         cal.add(Calendar.MONTH, 1);
32         document.getPages().add(createCalendarPage(cal));
33         document.drawToWeb(req, res, sOut, "TableCalendar.pdf");
34         sOut.close();
35     }
36
37     private Page createCalendarPage(Calendar cal) {
38         // Create a New Page.
39         Page page = new Page(PageSize.LETTER, PageOrientation.LANDSCAPE);
40
41         // Uncomment the line below to add a layout grid to the page.
42         //objPage.getElements().add( new LayoutGrid() );
43
44         // Create a table and set it's properties
45         Table2 table = new Table2(0, 40, page.getDimensions().getBody().getWidth(),
46             page.getDimensions().getBody().getHeight() - 40,
47             Font.getHelvetica(), 12);
48         table.getBorder().setWidth(2);
49         table.getCellDefault().setTextColor(RgbColor.getNavy());
```

```
50 table.getCellDefault().getBorder().setColor(IColor.getCornflowerBlue());
51 table.getBorder().setColor(IColor.getCornflowerBlue());
52 addColumns(table, page.getDimensions().getBody().getWidth());
53
54 createDaysOfWeekHeader(table);
55
56 cal.set(cal.DAY_OF_MONTH, 1);
57 int startDay = cal.get(Calendar.DAY_OF_WEEK) - 1;
58 int daysInMonth = cal.getActualMaximum(cal.DAY_OF_MONTH);
59
60 int requiredRows = (int)Math.ceil((daysInMonth + startDay) / 7.0f);
61
62 float rowHeight = (page.getDimensions().getBody().getHeight() - 60) / requiredRows;
63
64 Row2 row = null;
65 int position = 0;
66
67 // Add a blank entry for Sunday
68 if (startDay != 0) {
69     row = table.getRows().add(rowHeight, Font.getHelvetica(), 14);
70     row.getCells().add(" ", Font.getHelvetica(), 15, IColor.getRed(),
71         IColor.getAliceBlue(), 1);
72     position++;
73 }
74
75 // Add any additional blank entries days of previous month
76 while (position < startDay) {
77     row.getCells().add(" ");
78     position++;
79 }
80
81 // Add the days of the month to the myTable
82 for (int day=1; day <= daysInMonth; day++) {
83     if (position++ % 7 == 0) {
84         row = table.getRows().add(rowHeight, Font.getHelvetica(), 14);
85         row.getCells().add(Integer.toString(day), Font.getHelvetica(),
86             14, IColor.getRed(), IColor.getAliceBlue(), 1);
87     } else {
88         row.getCells().add(Integer.toString(day));
89     }
90 }
91
92 // Add any additional blank entries to the end of the table
93 while (position++ % 7 != 0) {
94     row.getCells().add(" ");
95 }
96
97 // Add the table to the Page
98 page.getElements().add(table);
99 Date d1 = cal.getTime();
```

TableCalendar.java

```
99     SimpleDateFormat sd = new SimpleDateFormat("MMMM yyyy");
100     // Create and add a Month label to the calendar
101     Label label = new Label(sd.format(d1), 0, 0,
102         page.getDimensions().getBody().getWidth(), 50,
103         Font.getHelvetica(), 23, TextAlign.CENTER,
104         RgbColor.getDeepPink());
105     page.getElements().add(label);
106
107
108     return page;
109 }
110
111 private void addColumns(Table2 table, float width) {
112     for (int i = 0; i < 7; i++) {
113         table.getColumns().add(width / 7);
114     }
115 }
116
117 private void createDaysOfWeekHeader(Table2 table) {
118     Row2 row = table.getRows().add(20, Font.getHelvetica(), 12);
119     row.getCells().add("Sunday", Font.getHelveticaBold(),
120         12, RgbColor.getRed(), RgbColor.getAliceBlue(), 1);
121     row.getCells().add("Monday");
122     row.getCells().add("Tuesday");
123     row.getCells().add("Wednesday");
124     row.getCells().add("Thursday");
125     row.getCells().add("Friday");
126     row.getCells().add("Saturday");
127 }
128 }
```