HtmlAreaTagsAndStyles.java

```
1
2  import com.cete.dynamicpdf.Document;
3  import com.cete.dynamicpdf.Page;
4  import com.cete.dynamicpdf.pageelements.html.HtmlArea;
5  import java.io.File;
6  import java.io.IOException;
7  import java.net.URI;
8  import java.net.URISyntaxException;
9  import javax.servlet.ServletConfig;
10 import javax.servlet.ServletException;
11 import javax.servlet.ServletOutputStream;
12 import javax.servlet.http.HttpServlet;
13 import javax.servlet.http.HttpServletRequest;
14 import javax.servlet.http.HttpServletResponse;
15
16 /*
17  * To change this license header, choose License Headers in Project Properties.
18  * To change this template file, choose Tools | Templates
19  * and open the template in the editor.
20  */
21 /**
22  *
23  * @author MohammedSiddiqUlla
24  */
25 public class HtmlAreaTagsAndStyles extends HttpServlet {
26
27     ServletOutputStream sOut;
28
29     public void init(ServletConfig servletConfig) throws ServletException {
30         super.init(servletConfig);
31
32     }
33
34     public void doGet(HttpServletRequest req, HttpServletResponse res)
35             throws IOException, ServletException {
36
37         sOut = res.getOutputStream();
38         // Create a document and set it's properties
39         Document document = new Document();
40         document.setCreator("HtmlAreaTagsAndStyles.aspx");
41         document.setAuthor("ceTe Software");
42         document.setTitle("HtmlArea Tags And Styles");
43
44         //Create a uri and get the input Html file.
45         URI uri = null;
46
47         File file = new File(getServletContext().getRealPath("html" + "/HtmlAreaTagsAndStyles.html"));
48         uri = file.toURI();
49
```

```
50          // Create a HtmlArea and set it's Dimensions
51          HtmlArea htmlArea = new HtmlArea(uri, 0, 0, 500, 680);
52          Page page;
53          int x = 1;
54          do {
55              // Create a page to add to the document
56              page = new Page();
57              if (x == 3) {
58                  // set the page dimensions for 3rd page
59                  page.getDimensions().setWidth(400);
60                  page.getDimensions().setHeight(550);
61              }
62
63              // Add htmlArea to page
64              page.getElements().add(htmlArea);
65
66              // Set the html area object equal to the rest of the html area that did not fit
67              // if all the html area did fit, GetOverflowHtmlArea will return null
68              htmlArea = htmlArea.getOverflowHtmlArea(0, 0, 680);
69
70              // Add page to document
71              document.getPages().add(page);
72              x++;
73
74          } while (htmlArea != null);
75
76          // Outputs the document to the current web page
77          document.drawToWeb(req, res, sOut, "HtmlAreaTagsAndStyles.pdf");
78
79          sOut.close();
80      }
81  }
```