

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

```

50      //Create a uri and get the input Html file.
51
52      URI uri = null;
53
54      File file = new File(getServletContext().getRealPath("html" + "/HtmlTags.html"));
55      uri = file.toURI();
56
57      // Create a HtmlArea and set it's Dimensions
58      HtmlArea htmlArea = new HtmlArea(uri, 0, 0, 500, 680);
59      Page page;
60      do {
61          // Create a page to add to the document
62          page = new Page();
63
64          // Add htmlArea to page
65          page.getElements().add(htmlArea);
66
67          // Add page to document
68          document.getPages().add(page);
69
70          // Set the html area object equal to the rest of the html area that did not fit
71          // if all the html area did fit, GetOverflowHtmlArea will return null
72          htmlArea = htmlArea.getOverflowHtmlArea(0, 0, 680);
73
74      } while (htmlArea != null);
75
76      // Add the template to the document
77      document.setTemplate(header);
78
79      // Outputs the document to the current web page
80      document.drawToWeb(req, res, sOut, "HtmlTags.pdf");
81
82      sOut.close();
83  }
84 }

```