

## **HOW TO CREATE PDF USING JAVA WITH FORMATTING TEXT**

```
import java.io.FileOutputStream;

import java.util.Date;

import com.itextpdf.text.Anchor;

import com.itextpdf.text.BadElementException;

import com.itextpdf.text.BaseColor;

import com.itextpdf.text.Chapter;

import com.itextpdf.text.Document;

import com.itextpdf.text.DocumentException;

import com.itextpdf.text.Element;

import com.itextpdf.text.Font;

import com.itextpdf.text.List;

import com.itextpdf.text.ListItem;

import com.itextpdf.text.Paragraph;

import com.itextpdf.text.Phrase;

import com.itextpdf.text.Section;

import com.itextpdf.text.pdf.PdfPCell;

import com.itextpdf.text.pdf.PdfPTable;

import com.itextpdf.text.pdf.PdfWriter;
```

```
class FirstPdf {  
  
    private static String FILE = "C:\\Users\\PREMVIKASH\\Desktop\\FirstPdf1ww2.pdf";  
  
    private static Font catFont = new Font(Font.FontFamily.TIMES_ROMAN, 18,  
        Font.BOLD);  
  
    private static Font redFont = new Font(Font.FontFamily.TIMES_ROMAN, 12,  
        Font.NORMAL, BaseColor.RED);  
  
    private static Font subFont = new Font(Font.FontFamily.TIMES_ROMAN, 16,  
        Font.BOLD);  
  
    private static Font smallBold = new Font(Font.FontFamily.TIMES_ROMAN, 12,  
        Font.BOLD);  
  
  
    public static void main(String[] args)  
    {  
  
        try {  
  
            Document document = new Document();  
  
            PdfWriter pdfWriter = PdfWriter.getInstance(document, new FileOutputStream(FILE));  
  
            document.open();  
  
            addMetaData(document);  
  
            addTitlePage(document);  
  
            addContent(document);  
  
            document.close();  
  
        } catch (Exception e) {  
  
            e.printStackTrace();  
  
        }  
  
    }  
}
```

```
// iText allows to add metadata to the PDF which can be viewed in your Adobe
```

```
// Reader
```

```
// under File -> Properties
```

```
private static void addMetaData(Document document) {
```

```
    document.addTitle("My first PDF");
```

```
    document.addSubject("Using iText");
```

```
    document.addKeywords("Java, PDF, iText");
```

```
    document.addAuthor("Lars Vogel");
```

```
    document.addCreator("Lars Vogel");
```

```
}
```

```
private static void addTitlePage(Document document)
```

```
    throws DocumentException {
```

```
    Paragraph preface = new Paragraph();
```

```
    // We add one empty line
```

```
    addEmptyLine(preface, 1);
```

```
    // Lets write a big header
```

```
    preface.add(new Paragraph("// default to text display\n" +
```

```
"return new LabelView(elem);\n" +
```

```
"}\n" +
```

```
"}\n" +
```

```
" \n" +
```

```
"private class SectionView extends BoxView \n" +
```

```
"{\n" +
```

```
"private int pageNumber;\n" +  
"\n" +  
"\n" +  
"/**\n" +  
"* Creates a view from an element that spans the supplied axis\n" +  
"* @param element\n" +  
"* @param axis\n" +  
"* \n" +  
"public SectionView(Element element, int axis)\n" +  
"{\n" +  
"super(element, axis);\n" +  
"\n" +  
"// apply insets to width but not top/bottom as it distorts\n" +  
"// breaking calculations\n" +  
"setInsets((short) (0),\n" +  
"(short) (PAGE_INSET + PAGE_MARGIN.left),\n" +  
"(short) (0),\n" +  
"(short) (PAGE_INSET + PAGE_MARGIN.right));\n" +  
"}\n" +  
"\n" +  
"\n" +  
"protected void layout(int width, int height) {\n" +  
"width = PAGE_WIDTH - 2 * PAGE_INSET - PAGE_MARGIN.left - PAGE_MARGIN.right;\n" +  
"super.layout(width, height);\n" +  
"}\n" +
```

```
" \n" +
" \n" +
"public float getPreferredSpan(int axis) {\n" +
"float span = 0;\n" +
"if (axis == View.X_AXIS) {\n" +
"span = PAGE_WIDTH;\n" +
"} else {\n" +
"span = pageNumber * PAGE_HEIGHT;\n" +
"}\n" +
"return span;\n" +
"}", catFont));
```

```
addEmptyLine(preface, 1);
```

```
// Will create: Report generated by: _name, _date
```

```
preface.add(new Paragraph("Report generated by: " + System.getProperty("user.name") + ", " + new
Date(), //$NON-NLS-1$ //$NON-NLS-2$ //$NON-NLS-3$
```

```
smallBold));
```

```
addEmptyLine(preface, 3);
```

```
preface.add(new Paragraph("This document describes something which is very important ",
```

```
smallBold));
```

```
addEmptyLine(preface, 8);
```

```
preface.add(new Paragraph("This document is a preliminary version and not subject to your license
agreement or any other agreement with vogella.com ;-).",
```

```
redFont));
```

```
document.add(preface);  
  
// Start a new page  
  
document.newPage();  
  
}
```

```
private static void addContent(Document document) throws DocumentException
```

```
{  
  
    Anchor anchor = new Anchor("First Chapter", catFont);  
    anchor.setName("First Chapter");  
  
    // Second parameter is the number of the chapter  
    Chapter catPart = new Chapter(new Paragraph(anchor), 1);  
  
    Paragraph subPara = new Paragraph("Subcategory 1", subFont);  
    Section subCatPart = catPart.addSection(subPara);  
    subCatPart.add(new Paragraph("Hello"));  
  
    subPara = new Paragraph("Subcategory 2", subFont);  
    subCatPart = catPart.addSection(subPara);  
    subCatPart.add(new Paragraph("Paragraph 1"));  
    subCatPart.add(new Paragraph("Paragraph 2"));  
    subCatPart.add(new Paragraph("Paragraph 3"));  
  
    // add a list
```

```
createList(subCatPart);

Paragraph paragraph = new Paragraph();

addEmptyLine(paragraph, 5);

subCatPart.add(paragraph);

// add a table

createTable(subCatPart);

// now add all this to the document

document.add(catPart);

// Next section

anchor = new Anchor("Second Chapter", catFont);

anchor.setName("Second Chapter");

// Second parameter is the number of the chapter

catPart = new Chapter(new Paragraph(anchor), 1);

subPara = new Paragraph("Subcategory", subFont);

subCatPart = catPart.addSection(subPara);

subCatPart.add(new Paragraph("This is a very important message"));

// now add all this to the document

document.add(catPart);
```

```
}
```

```
private static void createTable(Section subCatPart)
    throws BadElementException {
    PdfPTable table = new PdfPTable(3);

    // t.setBorderColor(BaseColor.GRAY);
    // t.setPadding(4);
    // t.setSpacing(4);
    // t.setBorderWidth(1);

    PdfPCell c1 = new PdfPCell(new Phrase("Table Header 1"));
    c1.setHorizontalAlignment(Element.ALIGN_CENTER);
    table.addCell(c1);

    c1 = new PdfPCell(new Phrase("Table Header 2"));
    c1.setHorizontalAlignment(Element.ALIGN_CENTER);
    table.addCell(c1);

    c1 = new PdfPCell(new Phrase("Table Header 3"));
    c1.setHorizontalAlignment(Element.ALIGN_CENTER);
    table.addCell(c1);

    table.setHeaderRows(1);

    table.addCell("1.0");
```



```
table.addCell("1.1");
```

```
table.addCell("1.2");
```

```
table.addCell("2.1");
```

```
table.addCell("2.2");
```

```
table.addCell("2.3");
```

```
subCatPart.add(table);
```

```
}
```

```
private static void createList(Section subCatPart)
```

```
{
```

```
List list = new List(true, false, 10);
```

```
list.add(new ListItem("First point"));
```

```
list.add(new ListItem("Second point"));
```

```
list.add(new ListItem("Third point"));
```

```
subCatPart.add(list);
```

```
}
```

```
private static void addEmptyLine(Paragraph paragraph, int number)
```

```
{
```

```
for (int i = 0; i < number; i++)
```

```
{
```

```
paragraph.add(new Paragraph(" "));
```

```
}
```

}

}