POC - > Creating PDF Programmatically Using Java

The aim is to creates pdf using java which consists of some text, images, tables, graphical data and charts.

- Libraries used are iText and JFreeChart.
- Implementing text, image, table.

Adding dependencies:

```
<!-- https://mvnrepository.com/artifact/com.itextpdf/itextpdf -->
<dependency>
    <groupId>com.itextpdf</groupId>
    <artifactId>itextpdf</artifactId>
    <version>5.5.13.3</version>
</dependency>

<!-- https://mvnrepository.com/artifact/jfree/jfreechart -->
<dependency>
    <groupId>jfree</groupId>
    <artifactId>jfreechart</artifactId>
    <version>1.0.13</version>
</dependency>
```

1. **Create a document.** The `com.itextpdf.text.Document` is the main class for PDF document generation.

2. **Create a PdfWriter.** The `com.itextpdf.text.pdf.PdfWriter` is responsible for writing the content of the document to the output stream. The first argument is a reference to the document. The second argument is the `FileOutputStream` in which the output will be written.

3. **Open document.** Once the document is opened, you can't write any Header- or Meta-information anymore. You have to open the document before you can begin to add content to the body of the document.

4. **Add content.** When the document is opened, you can start adding content.

5. **Close document.** Closes the document. Once all the content has been written in the body, you have to close the body. After that nothing can be written to the body anymore.

```
public class App
{
    public static void main( String[] args )    {



       //Enclose the code in try-catch block as it throws FileNotFoundException
       //and DocumentException
       try {
             Document document = new Document();

             PdfWriter.getInstance(document, new
             FileOutputStream("f:\\Sample.pdf"));

             document.open();

             //get instance of the image using get.Instance method in Image class
             Image image1 = Image.getInstance("cropped-Allps-logo-2-1-1.png");

             Image image2 = Image.getInstance("test.png");

             //pass the height and width of the image to scaleToFit
             image1.scaleToFit(150,150);

             //setting the alignment of the image
             image1.setAlignment(Element.ALIGN_CENTER);

             //passing the reference of image to add method to add in document
             document.add(image1);

             //to add a paragraph we need to pass the a paragraph object which
             //Takes string as argument.
             document.add(new Paragraph(" "));

             //To add fonts we have FontFactory class and which has a method
             //getFont(String fontname, float size, int style, BaseColor color)
             document.add(new Paragraph("Scale your business with ALLPS IT
             Teams",FontFactory.getFont(FontFactory.TIMES_BOLD,18,Font.BOLD,BaseC
             olor.RED)));

             document.add(new Paragraph("Hire high quality, cost-effective and
             dedicated remote IT teams through our extensive developer network.
             We are a Swiss company with Tech hub in India."));

             document.add(new Paragraph(" "));

             //To create a list we need to make a list object which takes Boolean
             for numbered as first argument and up to which number as second.
             List list = new List(true,10);

             //adding list item one by one
             list.add("IT Developers");

             list.add("Data Scientists");

             list.add("IT Architects");

             list.add("Project Managers");
```

```
            list.add("Automation Engineers");

            list.add("Niche skill Developers");

            //At last add the list to the document.
            document.add(list);

            document.add(new Paragraph(" "));

            //Create a table object and pass number of columns to constructor
            PdfPTable table = new PdfPTable(3);

            //create a pdf cell for the title of the table
            PdfPCell cell = new PdfPCell(new Paragraph("Milestones"));

            //defining the columns span for the cell
            cell.setColspan(4);

            //setting horizontal alignment of the cell
            cell.setHorizontalAlignment(Element.ALIGN_CENTER);

            //setting the background colour for the title cell
            cell.setBackgroundColor(BaseColor.GREEN);

            //add the cell to the table
            table.addCell(cell);

            table.addCell("Clients");

            table.addCell("Registered developers");

            table.addCell("Allps teams");

            table.addCell("16");

            table.addCell("40");

            table.addCell("7524");

            //adding table to the document
            document.add(table);

            document.add(new Paragraph(" "));

            image2.scaleToFit(400,400);

            image2.setAlignment(Element.ALIGN_CENTER);

            document.add(image2);

            //closing the document after writing all the content to the document
            document.close();

            System.out.println("created");

        }
        catch (Exception e) {

                JOptionPane.showMessageDialog(null, e);
```

```
       }

    }
}
```

- Implementing charts

```
import java.io.*;
import com.itextpdf.text.*;
import com.itextpdf.text.pdf.*;
import java.awt.Graphics2D;
import java.awt.geom.Rectangle2D;
// We will use DefaultPieDataset to define the data for the Pie Chart
import org.jfree.data.general.DefaultPieDataset;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.JFreeChart;
public class PDFPieChartExample {
     public static void main(String[] args){
        try {

        //We will define the data for the Pie Chart Using the Code below
        //Declare dataset object using the code below
        DefaultPieDataset myPiedataset = new DefaultPieDataset();
        //Define Values for the Pie Chart - Programming Languages
        //Percentage Difficulty
        myPiedataset.setValue("Java", 12.9);
        myPiedataset.setValue("C++", 37.9);
        myPiedataset.setValue("C", 86.5);
        myPiedataset.setValue("VB", 80.5);
        myPiedataset.setValue("Shell Script", 19.5);

        //With the dataset defined for Pie Chart, we can invoke a method in
        //ChartFactory object to create Pie Chart and Return a JFreeChart
        //object
        //This method returns a JFreeChart object back to us
        //We specify the chart title, dataset, legend, tooltip and URLs in
        //this method as input
        JFreeChart PDFPieChart=ChartFactory.createPieChart("Programming - Pie
        Chart Example",myPiedataset,true,true,false);

        //We have a Pie chart object, and now need to find a procedure to insert
        it into PDF using iText
        int width=640;
        //Width of our chart

        int height=480;
        //Height of our chart

        Document PieChart=new Document(new Rectangle(width,height));
        //Create a New Document Object for PDF
        //Create PDF Writer Object that will physically write the PDF file
        //to File Output Stream
        PdfWriter writer=PdfWriter.getInstance(PieChart,new
        FileOutputStream("Add_Pie_Chart_Using_JFreeChart.pdf"));

        //Ready with document objects, open the document object to push
        //contents
        PieChart.open();

        PieChart.addKeywords("iText,PieChart,JFreeChart,PDF,Example Tutorial");

        //Get Direct Content of the PDF document for writing
        PdfContentByte Add_Chart_Content= writer.getDirectContent();

        //Create a template using the PdfContent Byte object
        PdfTemplate
        template_Chart_Holder=Add_Chart_Content.createTemplate(width,height);

        //Create a 2D graphics object and Rectangle object as before to
        //write on the template
        Graphics2D
        Graphics_Chart=template_Chart_Holder.createGraphics(width,height,new
        DefaultFontMapper());
        Rectangle2D Chart_Region=new Rectangle2D.Double(0,0,540,380);
        //Invoke the draw method passing the Graphics and Rectangle 2D object to
        //draw the piechart
        PDFPieChart.draw(Graphics_Chart,Chart_Region);
        Graphics_Chart.dispose();

        //Add template to PdfContentByte and then to the PDF document
        Add_Chart_Content.addTemplate(template_Chart_Holder,0,0);

        //Close the Document, writer will create a beautiful Pie chart inside the
        //PDF document
        PieChart.close();
        }
        catch (Exception e)
        {
            System.out.println(e);
        }
    }
}
```

Submitted by: Mohit Verma

Designation: SDE Intern