

SlideShow Maker with Java

Although the author and publisher have made every effort to ensure that the information in this writing was correct at press time, the author and publisher do not assume and hereby disclaim any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from negligence, accident, or any other cause.

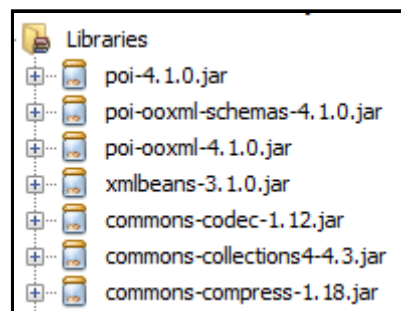
paid link



97 Day Money-Back Guarantee
Web & WordPress Hosting
\$2.59 /mo
DreamHost
Get Started

Though not as popular as video, slideshow can be a great marketing tool. There are sites that allow users to publish slide presentations and share the slides with others. This tutorial is written for anyone who want to create their own slideshow building tool. It outputs Microsoft PowerPoint Presentation (.pptx) file.

To start with, you need to download *Apache POI*, a Java API for manipulating Microsoft Documents. Download the library here: <https://poi.apache.org/>. Unzip it and add the required jar files to your project.



Here is the program:

```
import java.awt.Color;
import java.awt.Rectangle;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
```

```

import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import javax.swing.*;
import org.apache.commons.compress.utils.IOUtils;
import org.apache.poi.sl.usermodel.PictureData;
import org.apache.poi.xslf.usermodel.SlideLayout;
import org.apache.poi.xslf.usermodel.XMLSlideShow;
import org.apache.poi.xslf.usermodel.XSLFSlide;
import org.apache.poi.xslf.usermodel.XSLFSlideLayout;
import org.apache.poi.xslf.usermodel.XSLFSlideMaster;
import org.apache.poi.xslf.usermodel.XSLFTextBox;
import org.apache.poi.xslf.usermodel.XSLFTextParagraph;
import org.apache.poi.xslf.usermodel.XSLFTextRun;
import org.apache.poi.xslf.usermodel.XSLFPictureData;
import org.apache.poi.xslf.usermodel.XSLFPictureShape;

public class SlideShowMaker{
    public SlideShowMaker() throws FileNotFoundException, IOException{
        ArrayList<String> arText = new ArrayList<>();
        arText.add('Relaxing Places');
        arText.add('Crowd in Beach');
        arText.add('Long Bridge');
        arText.add('Coffee Cafe');
        arText.add('Big Ranch');

        ArrayList<String> arPic = new ArrayList<>();
        arPic.add('images/cinema.jpg');
        arPic.add('images/beach.jpg');
        arPic.add('images/bridge.jpg');
        arPic.add('images/cafe.jpg');
        arPic.add('images/ranch.jpg');

        XMLSlideShow ppt = new XMLSlideShow();
        XSLFSlideMaster slideMaster = ppt.getSlideMasters().get(0);
        XSLFSlideLayout slideLayout = slideMaster.getLayout(SlideLayout.BLANK);
        XSLFSlide slide[] = new XSLFSlide[arText.size()];

        int slideWidth = ppt.getPageSize().width;
        int slideHeight = ppt.getPageSize().height;

        for(int i=0;i<slide.length;i++){
            slide[i] = ppt.createSlide(slideLayout);
            //slide[i].getBackground().setFillColor(Color.BLUE);

            int textWidth = slideWidth*80/100;
            int textHeight = slideHeight*20/100;
            int textX = (slideWidth - textWidth) / 2;
            int textY = 10;

            XSLFTextBox text = slide[i].createTextBox();
            text.setAnchor(new Rectangle(textX,textY,textWidth,textHeight));
            XSLFTextParagraph paragraph = text.addNewTextParagraph();
            XSLFTextRun run = paragraph.addNewTextRun();
            run.setText(arText.get(i));
            run.setFontSize(72.0);

            int picWidth = slideWidth*80/100;
            int picHeight = slideHeight*60/100;
            int picX = (slideWidth - picWidth) / 2;
            int picY = textY + textHeight + 10;

            byte[] picByte = IOUtils.toByteArray(new FileInputStream(arPic.get(i)));
            XSLFPictureData picData = ppt.addPicture(picByte, PictureData.PictureType.JPEG);
            XSLFPictureShape picShape = slide[i].createPicture(picData);
            picShape.setAnchor(new Rectangle(picX,picY,picWidth,picHeight));
        }

        File file=new File('slideshow.pptx');
        FileOutputStream out=new FileOutputStream(file);
        ppt.write(out);
    }
}

```

```

        out.close();
    }

    public static void main(String[] args) throws IOException{
        new SlideShowMaker();
    }
}

```

Explanation:

Two arrays are used to store slide content. The `ArrayList arText` contains text that will be inserted into the slides. The other `ArrayList, arPic`, stores the paths of images that will accompany the text.

Basically, `XMLSlideShow` creates an object that will be used to write a slideshow. Slide layout is determined using `slide master`. As you can see, all of the slides use `BLANK` layout.

```

XMLSlideShow ppt = new XMLSlideShow();
XSLFSlideMaster slideMaster = ppt.getSlideMasters().get(0);
XSLFSlideLayout slideLayout = slideMaster.getLayout(SlideLayout.BLANK);

```

The code below creates an array of slides. The number of slides depend on the number of items in the `ArrayList arText`.

```

XSLFSlide slide[] = new XSLFSlide[arText.size()];

```

The width and height of the slide are determined in order to calculate text (and image) size and placement.

```

int slideWidth = ppt.getPageSize().width;
int slideHeight = ppt.getPageSize().height;

```

A blank slide is created.

```

slide[i] = ppt.createSlide(slideLayout);

```

A text box is created. The text from the `ArrayList arText` is written in the text box.

```

XSLFTextBox text = slide[i].createTextBox();

```

The code below deals with the size and placement of the text box.

```

text.setAnchor(new Rectangle(textX,textY,textWidth,textHeight));

```

Text size, color, and other attributes can be customized with the help of the class `XSLFTextRun`.

```

XSLFTextParagraph paragraph = text.addNewTextParagraph();
XSLFTextRun run = paragraph.addNewTextRun();
run.setText(arText.get(i));
run.setFontSize(72.0);

```

Before an image is added to the slide, image data is first converted into a byte array.

```

byte[] picByte = IOUtils.toByteArray(new FileInputStream(arPic.get(i)));

```

The image is then added to the slide.

```

XSLFPictureData picData = ppt.addPicture(picByte, PictureData.PictureType.JPEG);
XSLFPictureShape picShape = slide[i].createPicture(picData);

```

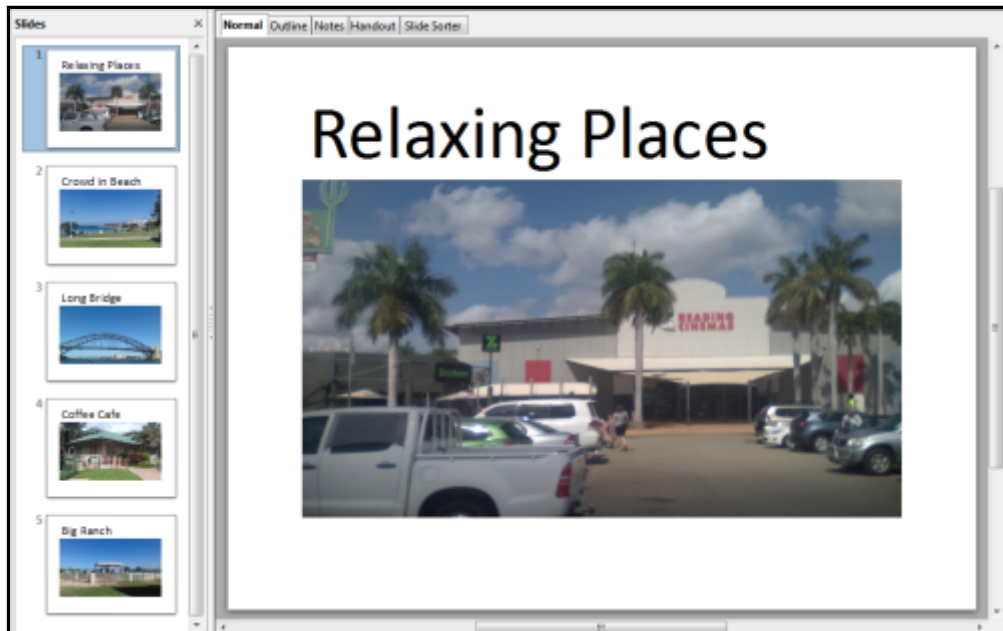
With the help of the method `setAnchor`, the image can be resized and drawn in a pre specified location.

```

picShape.setAnchor(new Rectangle(picX,picY,picWidth,picHeight));

```

Here is the output:



You can improve the program, for instance, by making it as a template based system.

advertisement

Fully Managed VPS Hosting

Big or small, website or application - there is a VPS configuration for you.

[Click here](#)

www.liberpaper.com