

APACHE POI PPT - SLIDE LAYOUTS

http://www.tutorialspoint.com/apache_poi_ppt/apache_poi_ppt_slide_layouts.htm

Copyright © tutorialspoint.com

In the previous chapter, you have seen how to create empty slides and how to add slides to it. In this chapter, you will learn how to get the list of available slides, and how to create a slide with different layouts.

Available Slide layouts

PowerPoint presentations have slide layouts, and you can choose a desired layout to edit a slide. First of all, let us find out the list of all the slide layouts available.

- There are different slide masters and in each slide master, there are several slide layouts.
- You can get the list of the slide masters using the **getSlideMasters** method of the **XMLSlideShow** class.
- You can get the list of the slide layouts from each slide master using the **getSlideLayouts** method of the **XSLFSlideMaster** class.
- You can get the name of the slide layout from the layout object using the **getType** method of the **XSLFSlideLayout** class.

Note : All these classes belongs to *org.poi.xslf.usermodel* package.

Given below is the complete program to get the list of available slide layouts in the PPT:

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

import org.apache.poi.xslf.usermodel.XMLSlideShow;
import org.apache.poi.xslf.usermodel.XSLFSlideLayout;
import org.apache.poi.xslf.usermodel.XSLFSlideMaster;

public class SlideLayouts {

    public static void main(String args[]) throws IOException{

        //create an empty presentation
        XMLSlideShow ppt = new XMLSlideShow();
        System.out.println("Available slide layouts:")

        //getting the list of all slide masters
        for(XSLFSlideMaster master : ppt.getSlideMasters()){

            //getting the list of the layouts in each slide master
            for(XSLFSlideLayout layout : master.getSlideLayouts()){

                //getting the list of available slides
                System.out.println(layout.getType());
            }
        }
    }
}
```

Save the above Java code as **SlideLayouts.java** , and then compile and execute it from the command prompt as follows:

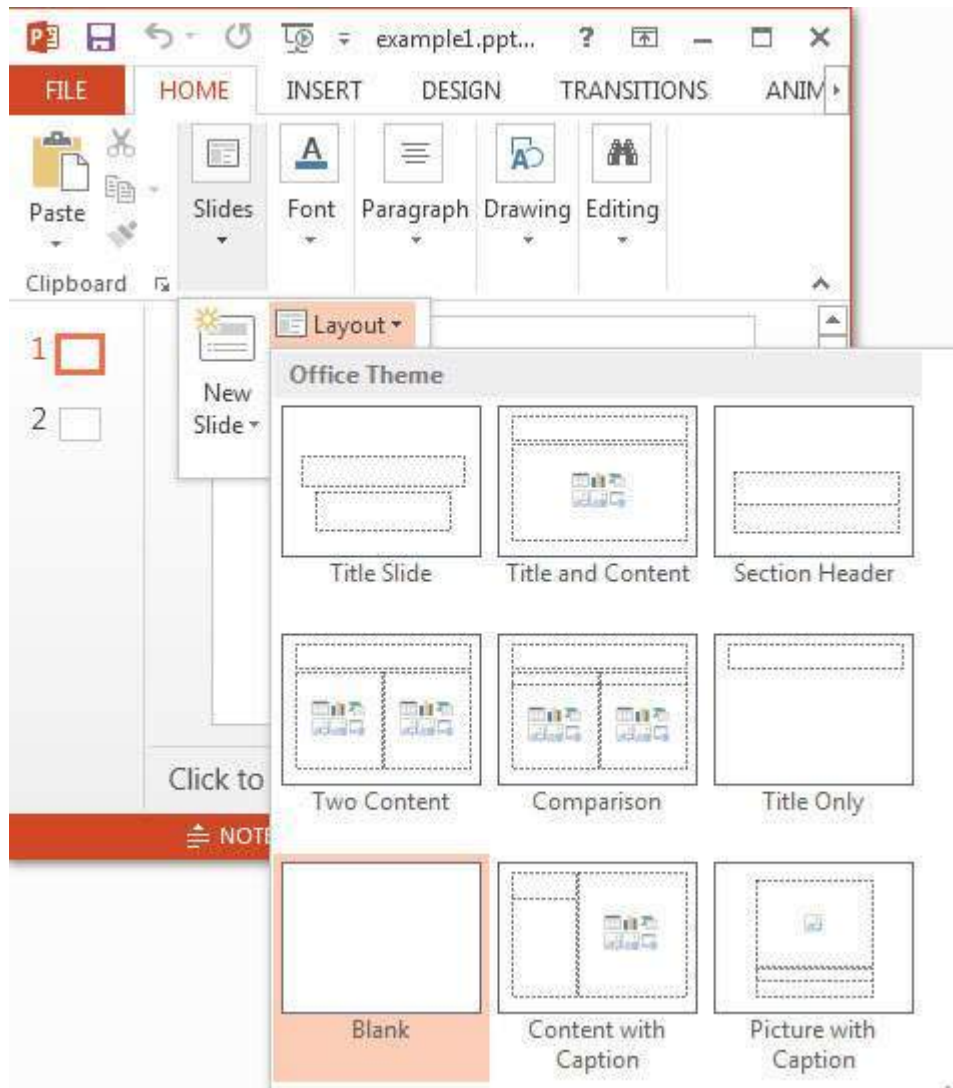
```
$javac SlideLayouts.java
$java SlideLayouts
```

It will compile and execute to generate the following output:

Available slide layouts:

TITLE
PIC_TX
VERT_TX
TWO_TX_TWO_OBJ
BLANK
VERT_TITLE_AND_TX
TITLE_AND_CONTENT
TITLE_ONLY
SECTION_HEADER
TWO_OBJ
OBJ_TX

Shown below are some of the sample slide layouts available with MS-Office 360, 2013 edition.



Title Layout

Let us create a slide in a PPT using Title layout. Follow the steps given below:

Step 1 : Create an empty presentation by instantiating the **XMLSlideShow** class as shown below:

```
XMLSlideShow ppt = new XMLSlideShow();
```

Step 2 : Get the list of slide masters using the **getSlideMasters** method. Thereafter, select the desired slide master using the index as shown below:

```
XSLFSlideMaster slideMaster = ppt.getSlideMasters()[0];
```

Here we are getting the default slide master which is in the 0th location of the slide masters array.

Step 3 : Get the desired layout using the **getLayout** method of the **XSLFSlideMaster** class. This method accepts a parameter where you have to pass one of the static variable of the **SlideLayoutclass**, which represents our desired layout. There are several variables in this class where each variable represents a slide layout.

The code snippet given below shows how to create a title layout:

```
XSLFSlideLayout titleLayout = slideMaster.getLayout(SlideLayout.TITLE);
```

Step 4 : Create a new slide by passing a slide layout object as parameter.

```
XSLFSlide slide = ppt.createSlide(titleLayout);
```

Step 5 : Select a placeholder using the **getPlaceholder** method of the **XSLFSlide** class. This method accepts an integer parameter. By passing 0 to it, you will get the **XSLFTextShape** object, using which you can access the title text area of the slide. Set the title using the **setText** method as shown below:

```
XSLFTextShape title1 = slide.getPlaceholder(0);  
//setting the title init  
title1.setText("Tutorials point");
```

Given below is the complete program to create a slide with Title layout in a presentation:

```
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
  
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.XSLFTextShape;  
  
public class TitleLayout {  
  
    public static void main(String args[]) throws IOException{  
  
        //creating presentation  
        XMLSlideShow ppt = new XMLSlideShow();  
  
        //getting the slide master object  
        XSLFSlideMaster slideMaster = ppt.getSlideMasters()[0];  
  
        //get the desired slide layout  
        XSLFSlideLayout titleLayout = slideMaster.getLayout(SlideLayout.TITLE);  
  
        //creating a slide with title layout  
        XSLFSlide slide1 = ppt.createSlide(titleLayout);  
  
        //selecting the place holder in it  
        XSLFTextShape title1 = slide1.getPlaceholder(0);  
  
        //setting the title init  
        title1.setText("Tutorials point");  
  
        //create a file object  
        File file=new File("C://POIPPT//Examples//Titlelayout.pptx");  
        FileOutputStream out = new FileOutputStream(file);  
  
        //save the changes in a PPT document  
        ppt.write(out);  
        System.out.println("slide cretated successfully");  
        out.close();  
    }  
}
```

```
}
```

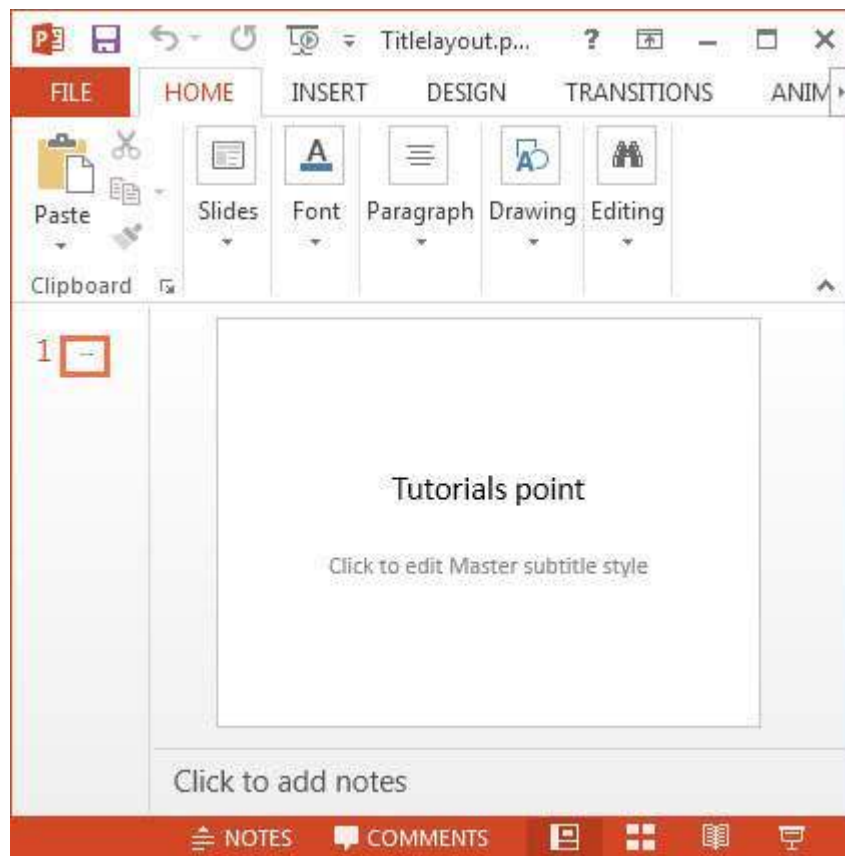
Save the above Java code as TitleLayout.java, and then compile and execute it from the command prompt as follows:

```
$javac TitleLayout.java  
$java TitleLayout
```

It will compile and execute to generate the following output.

```
slide created successfully
```

The PPT document with newly added Title layout slide appears as follows:



Title and content Layout

Let us create a slide in a PPT using Title and content layout. Follow the steps given below.

Step 1 : Create an empty presentation by instantiating the **XMLSlideShow** class as shown below:

```
XMLSlideShow ppt = new XMLSlideShow();
```

Step 2 : Get the list of slide masters using the **getSlideMasters** method. Select the desired slide master using the index as shown below:

```
XSLFSlideMaster slideMaster = ppt.getSlideMasters()[0];
```

Here we are getting the default slide master which is in the 0th location of the slide masters array.

Step 3 : Get the desired layout using the **getLayout** method of the **XSLFSlideMaster** class. This method accepts a parameter where you have to pass one of the static variable of the **SlideLayout** class which represents our desired layout. There are several variables in this class that represent slide layouts.

The following code snippet shows how to create title and content layout:

```
XSLFSlideLayout contentlayout = slideMaster.getLayout(SlideLayout.TITLE_AND_CONTENT);
```

Step 4 : Create a new slide by passing the slide layout object as parameter.

```
XSLFSlide slide = ppt.createSlide(SlideLayout.TITLE_AND_CONTENT);
```

Step 5 : Select a placeholder using the **getPlaceholder** method of the **XSLFSlide** class. This method accepts an integer parameter. By passing 1 to it, you will get the **XSLFTextShape** object, using which you can access the content area of the slide. Set the title using the **setText** method as shown below:

```
XSLFTextShape title1 = slide1.getPlaceholder(1);  
//setting the title init  
title1.setText("Introduction");
```

Step 6 : Clear the existing text in the slide using the **clearText** method of the **XSLFTextShape** class.

```
body.clearText();
```

Step 7 : Add new paragraph using the **addNewTextParagraph** method. Now add a new text run to the paragraph using the **addNewTextRun** method. Now to the text run, add text using the **setText** method as shown below:

```
body.addNewTextParagraph().addNewTextRun().setText("this is my first slide body");
```

Given below is the complete program to create a slide with Title layout in a presentation:

```
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
  
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.XSLFTextShape;  
  
public class TitleAndBodyLayout {  
  
    public static void main(String args[]) throws IOException{  
  
        //creating presentation  
        XMLSlideShow ppt = new XMLSlideShow();  
  
        //getting the slide master object  
        XSLFSlideMaster slideMaster = ppt.getSlideMasters()[0];  
  
        //select a layout from specified list  
        XSLFSlideLayout slidelayout = slideMaster.getLayout(SlideLayout.TITLE_AND_CONTENT);  
  
        //creating a slide with title and content layout  
        XSLFSlide slide = ppt.createSlide(slidelayout);  
        //selection of title place holder  
        XSLFTextShape title = slide.getPlaceholder(0);  
  
        //setting the title in it  
        title.setText("introduction");  
  
        //selection of body placeholder  
        XSLFTextShape body = slide.getPlaceholder(1);  
  
        //clear the existing text in the slide  
        body.clearText();  
    }  
}
```

```
//adding new paragraph
body.addNewTextParagraph().addNewTextRun().setText("this is my first slide body");

//create a file object
File file=new File("contentlayout.pptx");
FileOutputStream out = new FileOutputStream(file);

//save the changes in a file
ppt.write(out);
System.out.println("slide created successfully");
out.close();
}
}
```

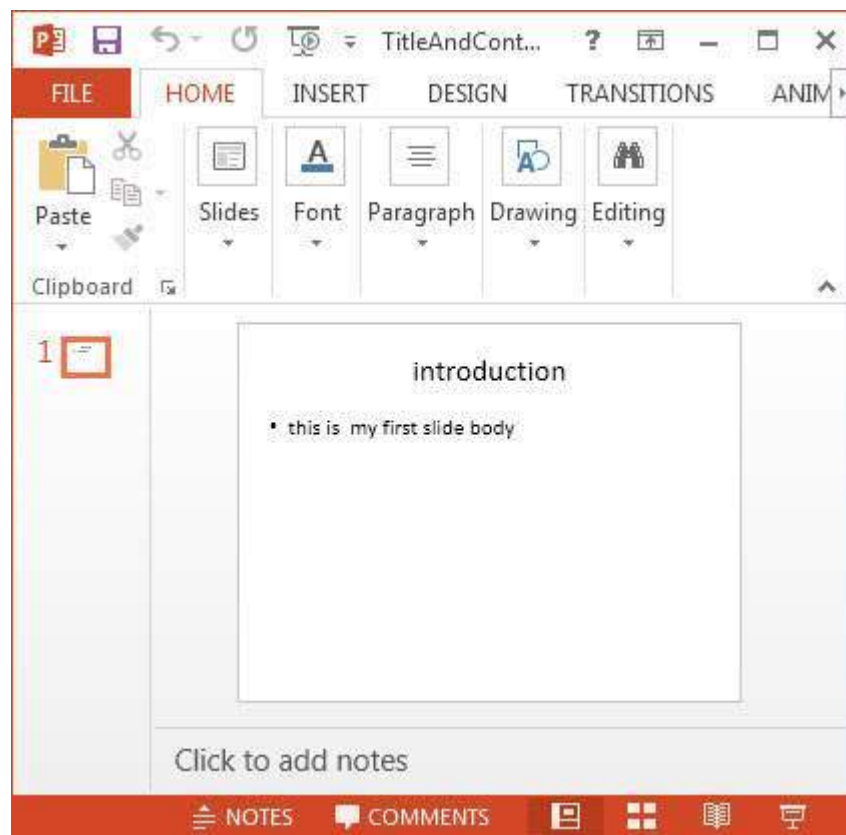
Save the above Java code as **TitleLayout.java**, and then compile and execute it from the command prompt as follows:

```
$javac TitleLayout.java
$java TitleLayout
```

It will compile and execute to generate the following output:

```
slide created successfully
```

The PPT document with newly added Title layout slide appears as follows:



In the same way, you can create slides with different layouts as well.

Loading [MathJax]/jax/output/HTML-CSS/jax.js