LEARN APACHE POI-PPT

java PPT library

tutorialspoint

SIMPLYEASYLEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia

https://twitter.com/tutorialspoint

About the Tutorial

This tutorial provides a basic understanding of Apache POI library and its features. Here we will learn how to read, write, and manage MS-PowerPoint documents using Java programs.

Audience

This tutorial is designed for all the readers working on Java and especially those who want to create, read, write, and modify PPT files using Java.

Prerequisites

A general awareness of Java programing with JDK1.5 or later versions and IO concepts in Java are the only prerequisites to understand this tutorial.

Copyright & Disclaimer

The API of Apache POI contains a number of methods and classes. In this tutorial, we have used only some of those for demonstration purpose. We encourage the readers to refer the complete API document for a comprehensive understanding.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial 1

Audience 1

Prerequisites 1

Copyright & Disclaimer 1

Table of Contents 2

1. APACHE POI – OVERVIEW 5

What is Apache POI? 5

Components of Apache POI 5

2. FLAVORS OF JAVA PPT API 7

Aspose Slides for Java 7

Apache POI 7

3. APACHE POI – INSTALLATION 9

System Requirements 9

Step 1: Verify your Java Installation 9

Step 2: Set your Java Environment 10

Step 3: Install Apache POI Library 10

4. CLASSES AND METHODS 13

Presentation 13

Slide 14

Slide Master 15

Slide Layout 15

Text Paragraph 16

Text Run 17

Text shape 18

HyperLink 19

5. PRESENTATION 20

Creating Empty Presentation 20

Editing an Existing Presentation 22

6. SLIDE LAYOUTS 25

Available Slide layouts 25

Title Layout 27

Title and content Layout 31

7. MANAGEMENT OF SLIDES 36

Changing a Slide 36

Reordering Slides 39

Deleting Slides 43

8. IMAGES 47

Adding Image 47

Reading Image 50

9. CREATING HYPERLINKS 53

Creating Hyperlinks 53

10. READING SHAPES 58

Reading Shapes from a Presentation 58

11. FORMATTING TEXT 61

Formatting Text in a Presentation 61

12. MERGING 68

Merging Multiple Presentations..........**68**

13. PPT TO IMAGE 73

Converting Presentation to Image..........**73**

1. APACHE POI – OVERVIEW

Many a time, a software application is required to generate reports in Microsoft Office file format. Sometimes, an application is even expected to receive MS- Office files as input data.

Any Java programmer who wants to produce MS Office files as output must use a predefined and read-only API to do so.

What is Apache POI?

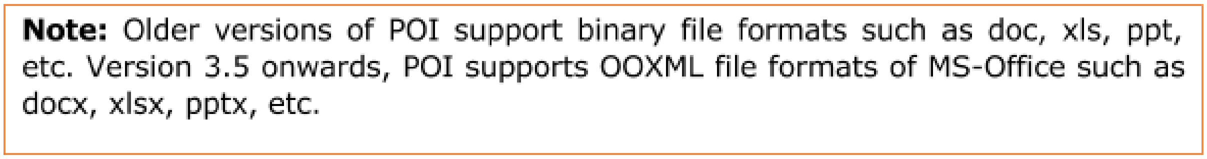
Apache POI is a popular API that allows programmers to create, modify, and display MS-Office files using Java programs. It is an open source library developed and distributed by Apache Software Foundation. It contains classes and methods to decode the user input data, or a file into MS Office documents.

Components of Apache POI

Apache POI contains classes and methods to work on all OLE2 Compound documents of MS-Office. The list of components of this API is given below:

- **POIFS** (Poor Obfuscation Implementation File System): This component is the basic factor of all other POI elements. It is used to read different files explicitly.
- **HSSF** (Horrible SpreadSheet Format): It is used to read and write .xls format of MS-Excel files.
- **XSSF** (XML SpreadSheet Format): It is used for .xlsx file format of MS-Excel.
- **HPSF** (Horrible Property Set Format): It is used to extract property sets of the MS-Office files.
- **HWPF** (Horrible Word Processor Format): It is used to read and write .**doc** extension files of MS-Word.
- **XWPF** (XML Word Processor Format): It is used to read and write .docx extension files of MS-Word.
- **HSLF** (Horrible Slide Layout Format): It is used to read, create, and edit PowerPoint presentations.
- **HDGF** (Horrible DiaGram Format): It contains classes and methods for MS-Visio binary files.
- **HPBF** (Horrible PuBlisher Format): It is used to read and write MS-Publisher files.

This tutorial guides you through the process of working on Microsoft PowerPoint presentation using Java. Therefore the discussion is confined to **XSLF component**.



Note: Older versions of POI support binary file formats such as doc, xls, ppt, etc. Version 3.5 onwards, POI supports OOXML file formats of MS-Office such as docx, xlsx, pptx, etc.

2. FLAVORS OF JAVA PPT API

This chapter takes you through some of the flavors of Java PowerPoint API and their features. There are many vendors who provide Java PPT related APIs; some of them are considered in this chapter.

Aspose Slides for Java

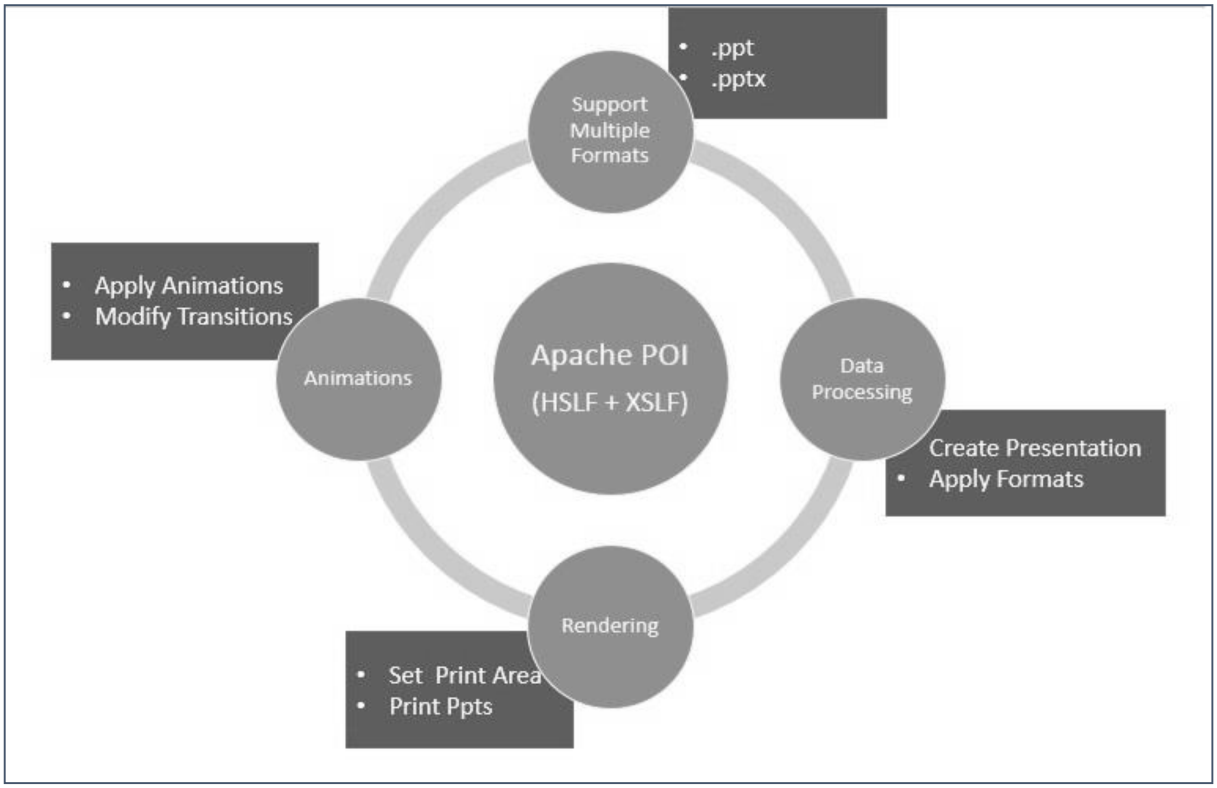
Aspose slides for Java is a purely licensed Java PPT API, developed and distributed by the vendor **Aspose**. The latest version of this API is 8.1.2, released in July 2014. It is a rich and heavy API (combination of plain Java classes and AWT classes) for designing the PPT component that can read, write, and manage slides.

The common uses of this API are as follows:

- Build dynamic presentations
- Render and print high-fidelity presentations
- Generate, edit, convert, and print presentations

Apache POI

Apache POI is a 100% open source library provided by Apache Software Foundation. Most of the small and medium scale application developers depend heavily on Apache POI (HSLF + XSLF). It supports all the basic features of PPT libraries; however, rendering and text extraction are its main features. Given below is the architecture of Apache POI for PPT.



Architecture – Apache POI for PPT

3. APACHE POI – INSTALLATION

This chapter takes you through the process of setting up Apache POI on Windows and Linux based systems. Apache POI can easily be installed and integrated with your current Java environment, following a few simple steps without any complex setup procedures. User administration is required for installation.

System Requirements

JDK	Java SE 2 JDK 1.5 or above
Memory	1 GB RAM (recommended)
Disk Space	No minimum requirement
Operating System Version	Windows XP or above, Linux

Let us now proceed with the steps to install Apache POI.

Step 1: Verify your Java Installation

First of all, you need to have Java Software Development Kit (SDK) installed on your system. To verify this, execute any of the following two commands depending on the platform you are working on.

If the Java installation has been done properly, then it will display the current version and specification of your Java installation. A sample output is given in the following table.

Platform	Command	Sample Output
Windows	Open command console and type: \>java –version	Java version "1.7.0_60" Java (TM) SE Run Time Environment (build 1.7.0_60-b19) Java Hotspot (TM) 64-bit Server VM (build 24.60-b09,mixed mode)
Linux	Open command terminal and type: $java –version	java version "1.7.0_25" Open JDK Runtime Environment (rhel-2.3.10.4.el6_4-x86_64) Open JDK 64-Bit Server VM (build 23.7-b01, mixed mode)

We assume that the readers of this tutorial have Java SDK version 1.7.0_60 installed on their system.

In case you do not have Java SDK, download its current version from http://www.oracle.com/technetwork/java/javase/downloads/index.html and install it.

Step 2: Set your Java Environment

Set the environment variable JAVA_HOME to point to the base directory location where Java is installed on your machine. For example,

Platform	Description
Windows	Set JAVA_HOME to C:\ProgramFiles\java\jdk1.7.0_60
Linux	Export JAVA_HOME=/usr/local/java-current

Append the full path of Java compiler location to the System Path.

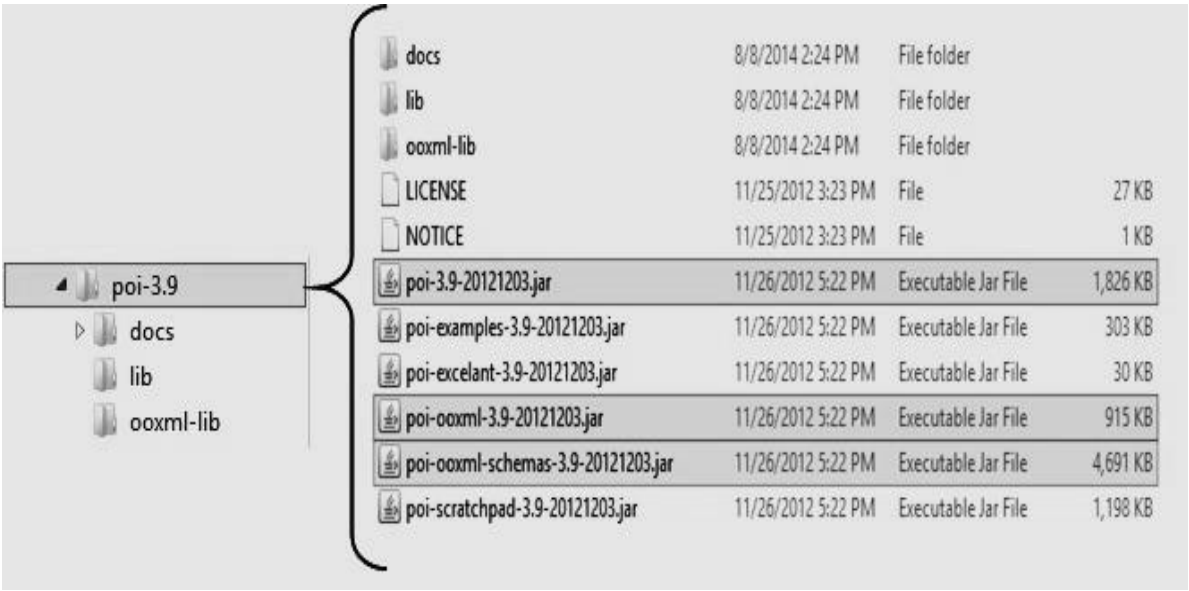
Platform	Description
Windows	Append the String "C:\Program Files\Java\jdk1.7.0_60\bin" to the end of the system variable PATH.
Linux	Export PATH=$PATH:$JAVA_HOME/bin/

Execute the command **java -version** from the command prompt as explained above.

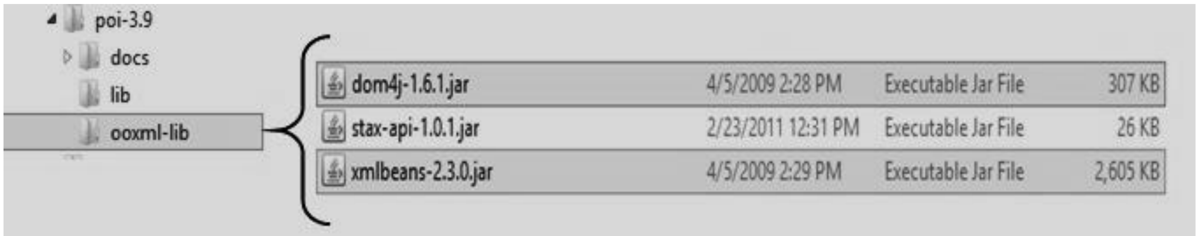
Step 3: Install Apache POI Library

Download the latest version of Apache POI from http://poi.apache.org/download.html and unzip its contents to a folder from where the required libraries can be linked to your Java program. Let us assume the files are collected in a folder on C drive.

The following images show the directories and the file structures inside the downloaded folder:



docs	8/8/2014 2:24 PM	File folder	
lib	8/8/2014 2:24 PM	File folder	
ooxml-lib	8/8/2014 2:24 PM	File folder	
LICENSE	11/25/2012 3:23 PM	File	27 KB
NOTICE	11/25/2012 3:23 PM	File	1 KB
poi-3.9-20121203.jar	11/26/2012 5:22 PM	Executable Jar File	1,826 KB
poi-examples-3.9-20121203.jar	11/26/2012 5:22 PM	Executable Jar File	303 KB
poi-excelant-3.9-20121203.jar	11/26/2012 5:22 PM	Executable Jar File	30 KB
poi-ooxml-3.9-20121203.jar	11/26/2012 5:22 PM	Executable Jar File	915 KB
poi-ooxml-schemas-3.9-20121203.jar	11/26/2012 5:22 PM	Executable Jar File	4,691 KB
poi-scratchpad-3.9-20121203.jar	11/26/2012 5:22 PM	Executable Jar File	1,198 KB



dom4j-1.6.1.jar	4/5/2009 2:28 PM	Executable Jar File	307 KB
stax-api-1.0.1.jar	2/23/2011 12:31 PM	Executable Jar File	26 KB
xmlbeans-2.3.0.jar	4/5/2009 2:29 PM	Executable Jar File	2,605 KB

Add the complete path of the five **jars** as highlighted in the above image to the CLASSPATH.

Platform	Description
Windows	Append the following strings to the end of the user variable CLASSPATH: “C:\poi-3.9\poi-3.9-20121203.jar;” “C:\poi-3.9\poi-ooxml-3.9-20121203.jar;” “C:\poi-3.9\poi-ooxml-schemas-3.9-20121203.jar;” “C:\poi-3.9\ooxml-lib\dom4j-1.6.1.jar;” “C:\poi-3.9\ooxml-lib\xmlbeans-2.3.0.jar;.;”
Linux	Export CLASSPATH=$CLASSPATH: /usr/share/poi-3.9/poi-3.9-20121203.tar: /usr/share/poi-3.9/poi-ooxml-schemas-3.9-20121203.tar: /usr/share/poi-3.9/poi-ooxml-3.9-20121203.tar: /usr/share/poi-3.9/ooxml-lib/dom4j-1.6.1.tar: /usr/share/poi-3.9/ooxml-lib/xmlbeans-2.3.0.tar

4. CLASSES AND METHODS

In this chapter, we will learn about a few classes and methods under Apache POI API that are crucial to work on PPT files using Java programs.

Presentation

To create and manage a presentation, you have a class called XMLSlideShow in the package *org.apache.poi.xslf.usermodel.* Given below are some important methods and a constructor of this class.

Class: XMLSlideShow

Package: org.apache.poi.xslf.usermodel

S. No.	Constructor and Description
1	XMLSlideShow(java.io.InputStream inputStream) You can instantiate this class by passing an inputstream class object to it.

S. No.	Methods and Description
1	int addPicture (byte[] pictureData, int format) Using this method, you can add a picture to a presentation.
2	XSLFSlide createSlide() Creates a blank slide in a presentation.
3	XSLFSlide createSlide(XSLFSlideLayout layout) Creates a slide with a given slide layout.
4	java.util.List<XSLFPictureData> getAllPictures() Returns an array of all the pictures in a presentation.
5	java.awt.Dimension getPageSize() Using this method, you can get to know the current page size.

6	XSLFSlideMaster[] getSlideMasters() Returns the array of all the slides in a presentation.
7	XSLFSlide[] getSlides() Returns all the slides in a presentation.
8	XSLFSlide removeSlide(int index) Using this method, you can remove a slide from a presentation.
9	void setPageSize(java.awt.Dimension pgSize) Using this method, you can reset the page size.
10	void setSlideOrder(XSLFSlide slide, int newIndex) Using this method, you can reorder the slides.

Slide

To create and manage a slide in a presentation, the methods of the **XSLFSlide** class are used. Some important methods of this class are mentioned below.

Class: XSLFSlide

Package: org.apache.poi.xslf.usermodel

S. No.	Methods and Description
1	XSLFBackground getBackground() Returns the **XSLFBackground** object which can be used to retrieve details like color and anchor of the background of the slide. You can also draw shapes in the slide using this object.
2	XSLFSlideLayout getSlideLayout() Provides access to the **XSLFSlideLayout** object of the current slide.
3	XSLFSlideMaster getSlideMaster() Provides access to the slide master of the current slide.