PDFBox - PDF Text Extraction

Java PDF Library, pdftotext, PDF to text, java pdf text extraction

Table of contents

1. Extracting Text

See class:org.pdfbox.util.PDFTextStripper
See class:org.pdfbox.searchengine.lucene.LucenePDFDocument
See command line app:ExtractText

One of the main features of PDFBox is its ability to quickly and accurately extract text from a variety of PDF documents. This functionality is encapsulated in the org.pdfbox.util.PDFTextStripper and can be easily executed on the command line with org.pdfbox.ExtractText.

1.1. Lucene Integration

Lucene is an open source text search library from the Apache Jakarta Project. In order for Lucene to be able to index a PDF document it must first be converted to text. PDFBox provides a simple approach for adding PDF documents into a Lucene index.

```
Document luceneDocument = LucenePDFDocument.getDocument( ... );
```

Now that you hava a Lucene Document object, you can add it to the Lucene index just like you would if it had been created from a text or HTML file. The LucenePDFDocument automatically extracts a variety of metadata fields from the PDF to be added to the index, the javadoc shows details on those fields. This approach is very simple and should be sufficient for most users, if not then you can use some of the advanced text extraction techniques described in the next section.

1.2. Advanced Text Extraction

Some applications will have complex text extraction requiments and neither the command line application nor the LucenePDFDocument will be able to fulfill those requirements. It is possible for users to utilize or extend the PDFTextStripper class to meet some of these requirements.

1.2.1. Limiting The Extracted Text

There are several ways that we can limit the text that is extracted during the extraction process. The simplest is to specify the range of pages that you want to be extracted. For example, to only extract text from the second and third pages of the PDF document you could do this:

```
PDFTextStripper stripper = new PDFTextStripper();
```

```
            stripper.setStartPage( 2 );
            stripper.setEndPage( 3 );
            stripper.writeText( ... );
```

Note:
The startPage and endPage properties of PDFTextStripper are 1 based and inclusive.

If you wanted to start on page 2 and extract to the end of the document then you would just set the startPage property. By default all pages in the pdf document are extracted.

It is also possible to limit the extracted text to be between two bookmarks in the page. If you are not familiar with how to use bookmarks in PDFBox then you should review the Bookmarks page. Similar to the startPage/endPage properties, PDFTextStripper also has startBookmark/endBookmark properties. There are some caveats to be aware of when using this feature of the PDFTextStripper. Not all bookmarks point to a page in the current PDF document. The possible states of a bookmark are:

- null - The property was not set, this is the default.
- Points to page in the PDF - The property was set and points to a valid page in the PDF
- Bookmark does not point to anything - The property was set but the bookmark does not point to any page
- Bookmark points to external action - The property was set, but it points to a page in a different PDF or performs an action when activated

The table below will describe how PDFBox behaves in the various scenarios:

Start Bookmark	End Bookmark	Result
null	null	This is the default, the properties have no effect on the text extraction.
Points page in the PDF	null	Text extraction will begin on the page that this bookmark points to and go until the end of the document.
null	Points page in the PDF	Text extraction will begin on the first page and stop at the end of the page that this bookmark points to.
Bookmark does not point to anything	null	Because the PDFTextStripper cannot determine a start page based on the bookmark, it will start on the first page and go

		until the end of the document.
null	Bookmark does not point to anything	Because the PDFTextStripper cannot determine a end page based on the bookmark, it will start on the first page and go until the end of the document.
Bookmark does not point to anything	Bookmark does not point to anything	This is a special case! If the startBookmark and endBookmark are exactly the same then no text will be extracted. If they are different then it is not possible for the PDFTextStripper to determine that pages so it will include the entire document.
Bookmark points to external action	Bookmark points to external action	If either the startBookmark or the endBookmark refer to an external page or execute an action then an OutlineNotLocalException will be thrown to indicate to the user that the bookmark is not valid.

Note:

PDFTextStripper will check both the startPage/endPage and the startBookmark/endBookmark to determine if text should be extracted from the current page.