(/) PDF Parser (/)

PDF Parser (/)

PHP library to parse PDF files and extract elements like text. (/)

build failing (https://travis-ci.org/smalot/pdfparser) downloads 3.42 M (https://packagist.org/packages/smalot/pdfparser)
stable v0.17.1 (https://github.com/smalot/pdfparser)

Download from GitHub (https://github.com/smalot/pdfparser)

Use it from Packagist (https://packagist.org/packages/smalot/pdfparser)

Home (/) / Documentation

Documentation

PdfParser, a standalone PHP library, provides various tools to extract data from a PDF file.
Currently, secured documents are not supported.

This Library is still under active development. As a result, users must expect BC breaks when using the master version.

This project is supported by Actualys (http://www.actualys.com/).

Prerequisites

This library requires **PHP 5.3**.
PDFParser is built on top of TCPDF parser.
This library will be automatically downloaded through Composer command line.

Installation

Using Composer

Add PDFParser to your composer.json file :

```
1. {
2.     "require": {
3.         "smalot/pdfparser": "*"
4.     }
5. }
```

Now ask for composer to download the bundle by running the command:

```
1. $ composer update smalot/pdfparser
```

As standalone library

First of all, download the library from Github by choosing a specific release (https://github.com/smalot/pdfparser/releases) or directly the master (https://github.com/smalot/pdfparser/archive/master.zip).

Once done, unzip it and run the following command line using composer (http://getcomposer.org/download/).

```
$ composer update
```

This command will download any dependencies (*Atoum library*) and create the 'autoload.php' file.

Now create a new file with this content, in the same folder :

```
<?php

// Include 'Composer' autoloader.
include 'vendor/autoload.php';

// Your code
// ...

?>
```

Unit tests with Atoum

Run Atoum unit tests (with code coverage - if xdebug installed) :

```
$ vendor/bin/atoum -d vendor/smalot/pdfparser/src/Smalot/PdfParser/Tests/
```

Once this command is ended, the folder "coverage/" will contain html pages with a code coverage summary.

Use

This sample will parse all the pdf file and extract text from each page.

```
<?php

// Include Composer autoloader if not already done.
include 'vendor/autoload.php';

// Parse pdf file and build necessary objects.
$parser = new \Smalot\PdfParser\Parser();
$pdf    = $parser->parseFile('document.pdf');

$text = $pdf->getText();
echo $text;

?>
```

You can too extract text from each page handly or for a specific page.

```php
<?php

// Include Composer autoloader if not already done.
include 'vendor/autoload.php';

// Parse pdf file and build necessary objects.
$parser = new \Smalot\PdfParser\Parser();
$pdf    = $parser->parseFile('document.pdf');

// Retrieve all pages from the pdf file.
$pages  = $pdf->getPages();

// Loop over each page to extract text.
foreach ($pages as $page) {
    echo $page->getText();
}

?>
```

Here a sample code to extract metadata from document (Author, Creator, CreationDate, ...).

```php
<?php

// Include Composer autoloader if not already done.
include 'vendor/autoload.php';

// Parse pdf file and build necessary objects.
$parser = new \Smalot\PdfParser\Parser();
$pdf    = $parser->parseFile('document.pdf');

// Retrieve all details from the pdf file.
$details  = $pdf->getDetails();

// Loop over each property to extract values (string or array).
foreach ($details as $property => $value) {
    if (is_array($value)) {
        $value = implode(', ', $value);
    }
    echo $property . ' => ' . $value . "\n";
}

?>
```