PHP XPDF Documentation

Release 0.1

Romain Neutron - Alchemy

Apr 12, 2018

Contents

CHAPTER 1

Introduction

PHP-XPDF is an object oriented library to handle XPDF, an open source project self-described as "a viewer for Portable Document Format files. The Xpdf project also includes a PDF text extractor, PDF-to-PostScript converter, and various other utilities."

For the moment, there's only one handler for PDF2Text.

This library depends on Symfony Process Component and Monolog.

CHAPTER 2

Installation

PHP-XPDF relies on composer. If you do not yet use composer for your project, you can start with this `composer.json` at the root of your project:

```
{
    "require": {
        "php-xpdf/php-xpdf": "master"
    }
}
```

Install composer :

```
# Install composer
curl -s http://getcomposer.org/installer | php
# Upgrade your install
php composer.phar install
```

You now just have to autoload the library to use it :

```
<?php
require 'vendor/autoload.php';
```

This is a very short intro to composer. If you ever experience an issue or want to know more about composer, you will find help on its dedicated website http://getcomposer.org/.

CHAPTER 3

PDF2Text

3.1 Basic Usage

```php
<?php
use Monolog\Logger;
use Monolog\Handler\NullHandler;
use XPDF\PdfToText;

// Create a logger
$logger = new Logger('MyLogger');
$logger->pushHandler(new NullHandler());

// You have to pass a Monolog logger
// This logger provides some usefull infos about what's happening
$pdfToText = PdfToText::load($logger);

// open PDF
$pdfToText->open('PDF-book.pdf');

// PDF text is now in the $text variable
$text = $pdfToText->getText();
$pdfToText->close();
```

3.2 Using Custom Binary

The PDF2Text is automatically detected if its path install is in the PATH. if you want to use your own PdfTotext binary, use as follow :

```php
<?php
$pdfToText = new PdfToText('/path/to/your/binary', $logger);
```

or, on Windows platform :

```php
<?php
$pdfToText = new PdfToText('C:\XPDF\PDF2Text.exe', $logger);
```

3.3 Charset encoding

By default, output text is UTF-8 encoded. But if you want a custom output , use the `setOutputEncoding` method

```php
<?php
$pdfToText->setOutputEncoding('ISO-8859-5');
```

Note: The charset value should be an iconv compatible value.

3.4 Extract page range

You can restrict the text extraction on page range. For example to extract pages 3 to 6 ;

```php
<?php
$pdfToText->getText(3, 6);
```

3.5 Silex Service Provider

XPDF is bundled with a Silex Service Provider. Use it is very simple :

```php
<?php
use Silex\Application;
use XPDF\XPDFServiceProvider;

$app = new Application();
$app->register(new XPDFServiceProvider());

// You have access to PDF2Text
$app['xpdf.pdf2text']->open(...);
```

You can, of course, customize it :

```php
<?php
use Silex\Application;
use XPDF\XPDFServiceProvider;

$app = new Application();
$app->register(new XPDFServiceProvider(), array(
    'xpdf.pdf2text.binary' => '/your/custom/binary',
    'xpdf.logger'          => $my_logger,
));

// You have access to PDF2Text
$app['xpdf.pdf2text']->open(...);
```

CHAPTER 4

Handling Exceptions

XPDF throws 4 different types of exception :

- `\XPDF\Exception\BinaryNotFoundException` is thrown when no acceptable pdf2text binary is found.
- `\XPDF\Exception\InvalidFileArgumentException` is thrown when an invalid file is supplied for text extraction
- `\XPDF\Exception\LogicException` which extends SPL LogicException
- `\XPDF\Exception\RuntimeException` which extends SPL RuntimeException

All these Exception implements `\XPDF\Exception\Exception` so you can catch any of these exceptions by catching this exception interface.

CHAPTER 5

Report a bug

If you experience an issue, please report it in our issue tracker. Before reporting an issue, please be sure that it is not already reported by browsing open issues.

When reporting, please give us information to reproduce it by giving your platform (Linux / MacOS / Windows) and its version, the version of PHP you use (the output of `php --version`), and the version of xpdf you use (the output of `xpdf -v`).

CHAPTER 6

Ask for a feature

We would be glad you ask for a feature ! Feel free to add a feature request in the issues manager on GitHub !

CHAPTER 7

Contribute

You find a bug and resolved it ? You added a feature and want to share ? You found a typo in this doc and fixed it ? Feel free to send a Pull Request on GitHub, we will be glad to merge your code.

CHAPTER 8

Run tests

PHP-XPDF relies on PHPUnit for unit tests. To run tests on your system, ensure you have PHPUnit installed, and, at the root of PHP-XPDF, execute it :

```
phpunit
```

CHAPTER 9

About

PHP-XPDF has been written by Romain Neutron @ Alchemy for Phraseanet, our DAM software. Try it, it's awesome !

CHAPTER 10

License

PHP-XPDF is licensed under the MIT License