



Wed 09 May 2018

## Convert HTML To PDF in PHP The Easy Way

In this tutorial you will learn how to easily convert web pages and raw HTML documents to PDF in your PHP applications. We will use the [Pdfcrowd API](#) for PDF generation. The API offers these benefits:

- The API is easy to use and fully supports HTML/CSS3/JavaScript.
- The integration takes only a few minutes.
- No third-party libraries are needed, just a single tiny PHP file.
- It does not consume CPU/memory on your computer, PDFs are created on the Pdfcrowd servers.

## Introduction

Let's start with an example:

```
<?php
require 'pdfcrowd.php';

$client = new \Pdfcrowd\HtmlToPdfClient("username", "apikey");
$pdf = $client->convertUrl('https://en.wikipedia.org/');
?>
```

This code converts *en.wikipedia.org* and stores the generated PDF to a string variable. You can save the result to a file or you can stream it to the browser, which we will discuss in detail shortly. You can click the thumbnail to open the generated PDF file:



Besides web pages, you can also convert a local HTML file or an HTML string:

```
<?php
$pdf = $client->convertFile('/path/to/your/file.html');
$pdf = $client->convertString('<b>bold</b> and <i>italic</i>');
?>
```

It is also possible to save the PDF directly to a file:

```
<?php
$client->convertUrlToFile('http://example.com/', 'example.pdf');
?>
```

## Basic Customization

Now that you know the basics, you may want to customize the generated PDF. Let's change the page format to Letter with half-inch margins:

```
<?php
$client->setPageSize("Letter");
$client->setPageMargins("0.5in", "0.5in", "0.5in", "0.5in");
?>
```

You can use metric units as well:

```
<?php
$client->setPageMargins("1cm", "1cm", "10mm", "10mm");
?>
```

You can also specify the appearance of the PDF when it is opened in a viewer:

```
<?php
$client->setInitialPdfZoomType(Pdfcrowd::FIT_PAGE);
$client->setPageLayout(Pdfcrowd::CONTINUOUS);
?>
```

The API provides many other options including password protection and fully customizable page headers and footers. Learn more about the available options in

the [HTML to PDF API - PHP SDK](#) documentation.

## Server Side PDF Generation

In this section we will show two common PDF generation scenarios.

### Generate PDF and send it to the browser

The following code converts *example.com* to PDF and sends it as a response:

```
<?php

require 'pdfcrowd.php';

try {
    $client = new \Pdfcrowd\HtmlToPdfClient("username", "apikey");
    $pdf = $client->convertUrl("https://example.com.com/");

    header("Content-Type: application/pdf");
    header("Cache-Control: no-cache");
    header("Accept-Ranges: none");
    header("Content-Disposition: inline; filename=\"example.pdf\"");

    echo $pdf;
}
catch(\Pdfcrowd\Error $why) {
    fwrite(STDERR, "Pdfcrowd Error: {$why}\n");
}

?>
```

To convert an HTML string you can use *convertString()* instead of *convertUrl()*:

```
<?php
$pdf = $client->convertString("<html><head>..</head><body>..</body></html>");

?>
```

Since Content-Disposition is set to *inline* the generated PDF is opened in the browser. If you change it to *attachment* the browser will pop up the file download dialog:

```
<?php
header("Content-Disposition: attachment; filename=\"mydomain.pdf\"");

?>
```

### Provide a PDF version of your web pages

This example shows how to enhance your PHP code so it can return a PDF version of your web pages. Let's look at the following helper function:

```
<?php
require 'pdfcrowd.php';

function generatePDF()
{
    if (!$_GET["pdf"])
        return False;

    try {
```

```

// build the url and remove the pdf field from the query string
$url = "http://" . $_SERVER["SERVER_NAME"] . $_SERVER["PHP_SELF"];
if (count($_GET) > 1) {
    unset($_GET["pdf"]);
    $url = $url . "?" . http_build_query($_GET, '', '&');
}

// call the API
$client = new \Pdfcrowd\HtmlToPdfClient("username", "apikey");
$pdf = $client->convertUrl($url);

// send the generated pdf to the browser
header("Content-Type: application/pdf");
header("Cache-Control: no-cache");
header("Accept-Ranges: none");
header("Content-Disposition: attachment; filename=\"created.pdf\"");

echo $pdf;
}
catch(\Pdfcrowd\Error $why) {
    fwrite(STDERR, "Pdfcrowd Error: {$why}\n");
}

return True;
}
?>

```

The *generatePDF()* function first checks if there is a *pdf* field in the query string. If yes, then the field is removed from the url. The function then passes the modified url to the API and finally sends the generated PDF to the browser.

You can use the function in your code like this:

```

<?php
if (generatePDF())
    return;

// your HTML rendering code
// ...
?>

```

*https://mydomain.com/page.php?pdf=1* now returns a PDF version of *https://mydomain.com/page.php*.

[Feed](#) | [Download as PDF](#)

[Contact](#) | [About](#) | [Customers](#) | [Blog](#) | [FAQ](#) | [Twitter](#) | [Press](#)

Copyright © 2009-2020 pdfcrowd.com [Legal](#) | [Privacy](#)