PDFShift API Documentation (v3.0

Introduction

Welcome to the PDFShift API. This documentation contains all the details needed to use the API and how to convert documents from HTML to PDF.

You will find code samples in Shell, Javascript, Python and PHP.

Authentication

Authenticate your account by including your secret key in API requests.
You can manage your API keys in the Dashboard.

Authentication to the API is performed via HTTP Basic Auth.

In order to connect, provide the basic auth values with "api" as the username, and your API Key as the password.

> Note that the username value is ignored on PDFShift.
> We suggest to set to "api" to have some values, but anything will work as long as it's compatible with the standard.

```
Authorization: Basic api:your_api_key
```

Sandbox

We highly recommend you to set the `sandbox` parameter to `true` when starting with PDFShift, and in development mode.

When enabled, the `sandbox` parameter adds a watermark to your generated PDF, but *does not count* the conversions made via PDFShift.

That way, you can set up your code and run multiple tests without worrying about your remaining credits.

Once all is set, you can run a last test locally with the `sandbox` removed or set to `false` to ensure the PDF renders exactly like you want.

Note In case you forgot to set the `sandbox` parameter to `true` while working on you
Simply reach out to us and we'll remove some of your credits usage for free during t

Rate Limit

Rate limiting is only forced for **unauthenticated accounts** with a limit of 2 requests per minutes.

As soon as you are authenticated, the restriction is lifted and you can convert as many documents as you want.

When reaching the rate limit, you will get an HTTP status code of `429`.

Each request will contain three headers to let you know your usage:

KEY	EXPLANATION
X-RateLimit-Remaining	Indicates the number of requests before hitting the rate limit.
X-RateLimit-Reset	Indicates the number of requests you can make per minutes (always 45).
X-RateLimit-Reset	Indicates when the rate limit will reset.

```
# HTTP response from PDFShift's API will contain these three headers:

X-RateLimit-Remaining: 30
X-RateLimit-Limit: 45
X-RateLimit-Reset: 1466368960
```

Getting an API Key

In order to generate PDF without the PDFShift's watermark, you will need an API Key.
You can request one by creating an account on PDFShift's website.

Once you have submitted the form, we will send you an email containing your API Key.
You can also create other API Key if you want, to split your keys accross your project.
This can be done via your Dashboard.

You can test and play with our API without an API Key. Simply don't provide a "Basic A
request will be done, with the difference that a watermark will be applied to the docu
Note that applying a watermark makes the generation slower, so you will have better result when using your API Key.

Errors

Most of the errors you will encounter when using our API can be found here. In case something is missing, feel free to let us know.

The PDFShift API uses the following error codes:

ERROR CODE	MEANING
400	Bad Request -- Your request is invalid. Often, it means a parameter was wrongly set.
401	Unauthorized -- No API key were found.
403	Forbidden -- The provided API key is invalid.
404	Not Found -- The page you tried to reach was not found.
405	Method Not Allowed -- The endpoint you tried to reach is not available with this HTTP method.
408	A timeout error occured when trying to convert a document.
429	Too Many Requests -- You sent too many request. Please see the Rate limiting section for more details.
500	Internal Server Error -- We had a problem with our server.

Support

If you need any help converting your HTML documents, feel free to reach out to our support team.

We are available via support@pdfshift.io and we will do our best to help you convert your documents the way you want to.

Convert

Converts an html document to pdf

Body parameter

source `required` string
Original document to convert to PDF. PDFShift will automatically detect if it's an URL and load it, or an HTML document and charge it.
You can also send an array of documents to convert if parallel conversions is enabled on your account. In that case, you will also need to provide the `webhook` parameters as this operation is asynchronous.

sandbox boolean
Will generates documents that doesn't count in the credits. The generated document will come with a watermark.

encode boolean
Will return the generated PDF in Base64 encoded format, instead of raw.

filename string
Name of the destination file. Only an alphanumerical value with \"-\" or \"_\", of at least 7 chars accepted. If given, the response **will not** be the PDF, but a JSON response containing an `url` parameter to an Amazon S3 bucket, to download the file. The file will be kept for 2 days, then automatically deleted. See Saving the document to Amazon S3 for an example.

webhook string
An URL where we will send a POST request containing a JSON body similar to when you use the filename parameter. The JSON response will contain a URL key that points to your file, stored on Amazon S3.

s3_destination string

Path to your S3 bucket, in order to save the converted PDF directly into your AWS S3 account. See Saving to your Amazon S3 for more details. Use a full path value like `s3://doc-example-bucket/pdfshift/upload/86aa3ede7d05.pdf`.

`timeout` number
If provided, will kill the page loading at a specified time without stopping with a Timeout error. Value in seconds.

`wait_for` string
Name of a function available globally. When present, PDFShift will wait for this function to return a truthy value (true, 1, a string, etc) or up to 30 seconds, then proceed to the conversion.

`landscape` boolean
Will set the view in landscape mode instead of portrait.

`css` string
Will append this CSS styles to the document before saving it. Can be an URL or a String of CSS rules.

`javascript` string
Will execute the given Javascript before saving the document. Can be an URL or a String of JS code.

`disable_images` boolean
Images will not be included in the final document.

`disable_javascript` boolean
Will not execute the javascript at all in the document.

`disable_links` boolean
The link in the document will not point anywhere.

`disable_backgrounds` boolean
The final document will not have the background images.

`remove_blank` boolean
Remove the last page if it is considered empty.

`delay` number
In milliseconds. Will wait for this duration before capturing the document. Up to 10 seconds max.

`use_print` boolean
Use the print stylesheet instead of the general one.

`format` string
Format of the document. You can either use the standard values (Letter, Legal, Tabloid, Ledger, A0, A1, A2, A3, A4, A5) or a custom `{width}x{height}` value. For {width} and {height}, you can indicate the following units: in, cm, mm.

`pages` string
Pages to print. Can be one number (`3`), a range (`1-5`), a list (`4,5,6`) or a combination of both (`1-3,6,7`). If the number is higher than the real number of pages, that number will be ignored.

zoom number

A value between 0 and 2. Allows you to increase the zoom in the document for speci
default zoom, lower is smaller, higher is bigger.

margin object

Empty spaces between the outer and the beginning of the content. See the Margin s
details.

- **top** string
  Space between the top and the content.
- **right** string
  Space between the right and the content.
- **bottom** string
  Space between the bottom and the content.
- **left** string
  Space between the left and the content.

auth object

Object containing `username` and `password` for accessing password-protected content.

- **username** string
  Username value from the Basic Authentication scheme.
- **password** string
  Password value from the Basic Authentication scheme.

cookies array

List of cookies you want to send along with the requests when loading the source. They must be provided as an array of objects with the following properties:

- **name** required string
  Name of the cookie.
- **value** required string
  Value for the specified cookie.
- **secure** boolean
  If set to true, This cookie will only be available for secure (https) connections.
- **http_only** boolean
  If set to true, this cookie will only be available to HTTP request only (no javascript).

http_headers object

List of HTTP headers that you can pass to the request.

header object
Defines a custom header. See the Header/Footer section for more details.

source required string
Element to add in the header part of the document. You can use variables, indicated at the end of the document. PDFShift will automatically detect if it's an URL and load it, or an HTML data and charge it.
Accepted variables are:

VARIABLE	DESCRIPTION
{{date}}	Formatted print date
{{title}}	Title of the HTML document
{{url}}	Page URL
{{page}}	Current page
{{total}}	Total number of pages

height string
A spacing between the header and the content.

footer object
Same as header (See the Header/Footer section for more details).

source required string
Element to add in the footer part of the document. You can use variables, indicated at the end of the document. PDFShift will automatically detect if it's an URL and load it, or an HTML data and charge it.
Accepted variables are:

VARIABLE	DESCRIPTION

{{date}}

Formatted print date

{{title}}

Title of the HTML document

{{url}}

Page URL

{{page}}

Current page

{{total}}

Total number of pages

`height` string

Space between the body and the footer

`protection` object

Will add restrictions on the PDF document. See the Protection section for more details.

`author` string

Document's author name.

`user_password` `required` string

A user who has the password will be able to view the document and perform operations allowed by the permission options.

`owner_password` `required` string

A user who has the password will have unlimited access to the PDF, including changing the passwords and permission options.

`no_print` boolean

When set to true, printing will be disabled.

`no_copy` boolean

When set to true, the possibility to copy any text will be disabled.

`no_modify` boolean

When set to true, the possibility to modify the document will be disabled.

`watermark` object

Add a watermark to the generated document. The watermark will always be placed at the center of the document. See the Watermark section for more details.

image string
Image file to add on top of the generated PDF. PDFShift will automatically detect
data, and act accordingly. When sending as data, the value must be base64 enc

text object
You can add a text as a watermark on your document easily.

value required string
Text to add on top of the generated PDF.

size number
Size of the font, in points.

font string
Font Family native to PDF. Allowed fonts are Helvetica, Times and Courier.

color string
Color of the font, in hexadecimal.

opacity number
Opacity of the font, between 0 (invisible) and 100.

bold boolean
Set the font in Bold.

italic boolean
Set the font in italic.

rotate number
Rotation of the watermark, in degrees

POST /convert/pdf

RESPONSE

%PDF-1.4....

HTTP STATUS CODE	DESCRIPTION
200 - OK	

Returns a pdf file.

400 - Bad Request
Bad request, returned when the request was not properly written.

401 - Unauthorized
Invalid api key given.

403 - Forbidden
No remaining credits left.

429 - Too Many Requests
You have been rate-limited.

Details

Margin

You can define margin for the document (space between the limits of the document and the beginning of the content).
You can either pass an object as defined below, or use a CSS like string, like the following:

VALUE	DESCRIPTION
10px	Will set a margin of 10px for all four borders.
10px 0	Will set a margin of 10px for top and bottom, and a margin of 0 for left and right.
10px 0 20px	Will set a margin of 10px for top, 0 for left and right and 20px for the bottom.
10px 20px 30px 40px	Will set a margin of 10px for top, 20px for right, 30px for bottom and 40px for left.

Otherwise, you can use an object to directly target a specific margin, using the following:

PARAMETER	TYPE	DEFAULT	DESCRIPTION
top	Integer or String	null	Space between the top and the content.
right	Integer or String	null	Space between the right and the content.
bottom	Integer or String	null	Space between the bottom and the content.
left	Integer or String	null	Space between the left and the content.

Header/Footer

You can configure the aspect of your header and footer document using the following values.

> The footer and header are **independant** from the rest of the document.
>
> As such, the CSS style defined in your body won't apply on your header/footer.
> To style your header/footer, you need to set a specific style either using <style> tag first, or adding style="" on your DOM elements.

PARAMETER
TYPE
DEFAULT
DESCRIPTION
source
String or URL
null
Element to add in the header/footer part of the document. You can use variables, indicated at the end of the document. PDFShift will automatically detect if it's an URL and load it, or an HTML data and charge it.
height
Integer or String
null
A spacing between the header or footer and the content. For header, it's the space between the header and the beginning of the document. For the footer, it's the space between the end of the document and the bottom of the page.
start_at
Integer
1
Start to display the header/footer at that given page. **Important**: If you send header AND footer, and set a start_at higher than 1, it must be the same for header and footer. For instance, header.start_at = 1 and footer.start_at = 5 is possible. But header.start_at = 2 and footer.start_at = 3 is NOT possible.

Header/Footer variables

VARIABLE
DESCRIPTION
{{date}}
Formatted print date
{{title}}
Title of the HTML document
{{url}}

Page URL

{{page}}
Current page

{{total}}
Total number of pages

Protection

You can restrict access to your generated document using the following rules. The encryption is made in 128bits.

> Some PDF Reader don't make the distinction between **user** and **owner** in a PDF Document. This means that when the user password has been entered, some PDF reader ignore the restrictions (no print, no copy, etc).
> So, setting a blank password for the user is similar to no security.

PARAMETER
TYPE
DEFAULT
DESCRIPTION

author
String
null
Document's author name

user_password
String
null
A user who has the password will be able to view the document and perform operations allowed by the permission options

owner_password
String
null

A user who has the password will have unlimited access to the PDF, including changing the passwords and permission options.

no_print
Boolean
false
When set to true, printing will be disabled.

no_copy
Boolean
false
When set to true, the possibility to copy any text will be disabled.

no_modify
Boolean
false
When set to true, the possibility to modify the document will be disabled.

Watermark

You can add a watermark to your documents via three alternatives:

- Via a PDF
- Via Text
- Via Image

Each alternatives has a set of options, which is detailed here:

Via PDF

PARAMETER
TYPE
DEFAULT
DESCRIPTION

source
URL or Base64 encoded PDF content
required

You can provide the source either as an URL, or a **base 64 encoded** PDF content. Raw PDF content will be refused. We recommend you to send a one page PDF content because only the [illegible] each of your generated content.

Via Text

PARAMETER
TYPE
DEFAULT
DESCRIPTION
text
String
required
The text to display as watermark.
font_size
Integer
16
The font size used.
font_family
String
Helvetica
The font family used.
font_color
String
000000
The color of the text.
font_opacity
Integer
100
The opacity for the text.
font_bold
Boolean
false

Set to true if you want the text bold.
font_italic
Boolean
false
Set to true if you want the text in italic.
offset_x
String
center
The X position. Can be either a String (left, center, right) or a number with unit (defaults to pixels. Allowed units are 'px', 'in', 'cm', 'mm', 'pt').
offset_y
String
middle
The Y position. Can be either a string (top, middle, bottom) or a number with unit (defaults to pixels. Allowed units are 'px', 'in', 'cm', 'mm', 'pt').
rotate
Integer
-45
The degree for the rotated element.

Via Image

PARAMETER
TYPE
DEFAULT
DESCRIPTION
image
URL or Base64 encoded image content
required
The image to display as watermark.
offset_x
String
center
The X position. Can be either a String (left, center, right) or a number with unit (defaults to pixels.

Allowed units are 'px', 'in', 'cm', 'mm', 'pt').

offset_y
String
middle
The Y position. Can be either a string (top, middle, bottom) or a number with unit (defaults to pixels. Allowed units are 'px', 'in', 'cm', 'mm', 'pt').

rotate
Integer
-45
The degree for the rotated element.

Webhooks

> When the conversion fail, we also do a `post` request to your endpoint, but with an `error` key instead.
> We recommend you to first check if the body contains the `error` before processing the document, and act accordingly.

If the `webhook` parameter is defined, the call to PDFShift's API will return a Queued response along with a 202 status code, like the following:

`{"queued": true}`

Once the processing has been done, we will send a `post` request to your webhook endpoint, containing a JSON payload with the URL to your converted document, stored at Amazon S3 (for two days).

Here's a sample of the payload we will send you:


```
{
  "duration": 3121.766417985782,
  "filesize": 259972,
  "response": {
    "duration": 2562,
    "status-code": 200
  },
  "success": true,
  "url": "https://s3.amazonaws.com/pdfshift/d/2/2019-05/99c456250a01448686d81752a3fb5beb/154660
}
```

In case there is any error while processing your document, we will instead send you a payload containing an `error` key, like the following:


```
{
  "error": {
    "code": 400,
    "error": "The requested page took too long to load.",
    "identifier": "A74",
    "success": false
  }
}
```

Saving to your Amazon S3

With PDFShift, you can save the converted PDF direclty into your S3 bucket at Amazon, allowing to post process it or deliver it to your clients using your own system.

In order to be able to save on your Bucket, you'll need to update the policy of your bucket destination.

We recommend you to use the following policy, but if you're an expert at AWS S3, feel free to customize it:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "Only allow writes to this bucket with bucket owner full control",
    "Effect": "Allow",
    "Principal": {
      "AWS": ["arn:aws:iam::804461045055:user/pdfshift"]
    },
    "Action": ["s3:PutObject"],
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
    "Condition": {
      "StringEquals": {
        "s3:x-amz-acl": "bucket-owner-full-control"
      }
    }
  }]
}
```

(The file is also available at Pastebin)

> Don't forget to change the name **DOC-EXAMPLE-BUCKET** to your appropriate bucket's name!

Once this policy in place, you'll be able to use the `s3_destination` parameter and use a value like `s3://doc-example-bucket/pdfshift/upload/86aa3ede7d05.pdf`

Misc

When converting a document, if successful, the HTTP response from PDFShift's API will contain the following header:

HEADER	DESCRIPTION
X-Response-StatusCode	The status code from your URL source, when an URL is provided. This can be useful to ensure the URL worked correctly.

Examples

Converting an URL

> The above command returns a PDF in binary format.

This endpoint the given URL to PDF.

Javascript | Php | Python | Ruby | Java | Csharp | Go

```
// Use the code available at
// https://gist.github.com/cnicodeme/28ade69b269ca0a4af0a7c29c479b747

pdfshift.convert('your_api_key', { source: 'https://www.example.com' }).then(function (response) {
    fs.writeFile('example.com.pdf', response.data, "binary", function () {})
}).catch(function({message, code, response, errors = null}) {})
```

Saving the document to Amazon S3

> The above command returns JSON structured like this:


```
{
  "success": true,
  "url": "https://s3.amazonaws.com/pdfshift/d/2/2019-02/47fc3918791a4818a6bf655
  "filesize": 13370,
  "duration": 5
}
```


By passing the "filename" parameter, the endpoint won't return the binary PDF, but an URL from Amazon S3 where the document will be stored for 2 days before being automatically deleted.

This can be useful if you don't want to download a large PDF to your server to then serve it to your users, but instead redirect them directly to that document.

Javascript | Php | Python | Ruby | Java | Csharp | Go

```
// Use the code available at
// https://gist.github.com/cnicodeme/28ade69b269ca0a4af0a7c29c479b747

pdfshift.convert('your_api_key', { source: 'https://www.example.com', filename: 'result.pdf' }).then(functio
    // The URL is on
    console.log(response.data.url);
}).catch(function({message, code, response, errors = null}) {})
```

Accessing secured pages

> The above command returns a PDF in binary format.

If your documents are located inside a protected area requiring a Basic Auth access, you can use the `auth` parameter from PDFShift's API to connect to your website.
Here's an example on how to do so.

Javascript | Php | Python | Ruby | Java | Csharp | Go

```
// Use the code available at
// https://gist.github.com/cnicodeme/28ade69b269ca0a4af0a7c29c479b747

pdfshift.convert('your_api_key', { source: 'https://httpbin.org/basic-auth/user/passwd', auth: { user: 'user',
    fs.writeFileSync('basic-auth.pdf', response.data, "binary", function () {})
})
```

Using Cookies

> The above command returns a PDF in binary format.

On the contrary, if your endpoint requires a more advanced authentication format, like a PHP session. You can add cookies to the parameter to simulate an active session.

This can be easily done with the `cookies` parameter from our API.

Javascript | Php | Python | Ruby | Java | Csharp | Go

```
// Use the code available at
// https://gist.github.com/cnicodeme/28ade69b269ca0a4af0a7c29c479b747

pdfshift.convert('your_api_key', { source: 'https://httpbin.org/cookies', cookies: [{name: 'session', value: '4
    fs.writeFileSync('cookies.pdf', response.data, "binary", function () {})
})
```

Adding a custom footer

> The above command returns a PDF in binary format.

One frequent action when converting a web document to PDF is to add header and footer.
This is useful to add page number for instance, or the name of your company at the top of each pages.

This is easily done in PDFShift with the `header` / `footer` parameter.

Javascript | Php | Python | Ruby | Java | Csharp | Go

```
// Use the code available at
// https://gist.github.com/cnicodeme/28ade69b269ca0a4af0a7c29c479b747

pdfshift.convert('your_api_key', { source: 'https://www.example.com', footer: { source: '<div>Page {{page
    fs.writeFileSync('footer.pdf', response.data, "binary", function () {})
})
```

Sending an invoice by email

> The above command returns a PDF in binary format.

Here's a complete example of how PDFShift can be integrated in one of your project.

A frequent use case is to use PDFShift to convert a locally generated invoice made in HTML (displayed in the back-office of your customer), converted in PDF and then sent by email.

Javascript | Php | Python | Ruby | Java | Csharp | Go

```
// Use the code available at
// https://gist.github.com/cnicodeme/28ade69b269ca0a4af0a7c29c479b747

const express = require('express');
const fs = require('fs');
const nodemail = require('nodemailer');

const app = express();

app.get('/send/',  (req, res, next) => {
    let invoice = fs.readFileSync('invoice.html', 'utf8');

    pdfshift.convert('your_api_key', { source: invoice }).then(function (response) {
        let transporter = nodemailer.createTransport({
            host: "smtp.gmail.com",
            port: 587,
            secure: true,
            auth: {
                user: account.user,
                pass: account.pass
            }
        });

        let mailOptions = {
            from: '"Billing at Your-Site" <billing@your-site.tld>',
            to: "customer@gmail.com"
            subject: "Thank you for your purchase",
            text: fs.readFileSync('templates/emails/invoice.txt', 'utf8'),
            html: fs.readFileSync('templates/emails/invoice.html', 'utf8'),
            attachments: [
                {
                    filename: 'invoice.pdf',
                    contentType: 'application/pdf',
                    content: response.data
                }
            ]
        };

        // send mail with defined transport object
```

```
    await transporter.sendMail(mailOptions)

    // Then, we redirect
    res.redirect(301, '/thank-you');
  })
})
```


Credits

Credits usage

Returns the current credits usage.

```
GET /credits/usage
```

RESPONSE

```
{
  "credits": {
    "base": 50000,
    "remaining": 49881,
    "total": 50000,
    "used": 119
  },
  "success": true
}
```

HTTP STATUS CODE	DESCRIPTION
200 - OK	Returns a valid json containing the credits details.