



PDFTOPIC By Ilyes-El-Majouti

 version **v0.1.3**

donate [PayPal](#)

minified size **37.9 kB**

downloads **1k/month**

license **Apache-2.0**

stars **9**

last commit **february**

The **pdftopic** package offers an advanced solution for converting PDF files into various image formats such as PNG, JPG, etc. With a particular focus on rendering quality, **pdftopic** distinguishes itself by producing output of exceptional quality, surpassing the performance of other packages available on the market. Designed to meet the critical needs of demanding users, **pdftopic** effectively and reliably solves the rendering quality problems that many other packages present, offering an unrivalled and reliable PDF conversion solution for developers and IT professionals.

IMPORTANT NOTE: We strongly encourage you to support this library by making a donation via [PayPal](#). Your generosity is highly appreciated, and we also welcome contributors with gratitude!

Prerequisites

- node >= 12.x
- imagemagick
- ghostscript (**windows only**)

Don't have imagemagick yet ?

Follow [this](#) guide to install the required dependencies.

Installation

```
npm install --save pdftopic
```

Usage

Converting specific page of PDF from buffer, then saving as image file

```
import pdftopic from "pdftopic";
import fs from "fs";

(async () => {
  const curriculum_vitae =
    fs.readFileSync('./pdf_files/curriculum_vitae.pdf');

  const converted_result = await pdftopic.pdfbuffer(pdf, 0);
  fs.writeFileSync(`./png_files/curriculum_vitae.png`,
    converted_result[0]);
})();
```

Converting specific pages of PDF from buffer, then saving as image file

```
import pdftopic from "pdftopic";
import fs from "fs";

(async () => {
  const curriculum_vitae =
    fs.readFileSync('./pdf_files/curriculum_vitae.pdf');

  const converted_result = await pdftopic.pdfbuffer(pdf, [ 1, 3 ]);
  converted_result.forEach((file, index) => {
    fs.writeFileSync(`./curriculum_vitae-${index}.png`, file);
  });
})();
```

Converting all pages of PDF from buffer, then saving as image file

```
import pdftopic from "pdftopic";
import fs from "fs";

(async () => {
  const curriculum_vitae =
    fs.readFileSync('./pdf_files/curriculum_vitae.pdf');

  const converted_result = await pdftopic.pdfbuffer(pdf, "all");
```

```
converted_result.forEach((file, index) => {  
    fs.writeFileSync(`./png_files/curriculum_vitae-${index}.png`,  
file);  
});  
})();
```

Converting all pages of PDF from buffer, then concat images and save image file

```
import pdftopic from "pdftopic";  
import fs from "fs";  
  
(async () => {  
    const curriculum_vitae =  
fs.readFileSync('./pdf_files/curriculum_vitae.pdf');  
  
    const converted_result = await pdftopic.pdfbuffer(pdf, "all");  
    const concat_converted_result = await  
pdftopic.bufferstoappend(allfiles);  
  
    fs.writeFileSync(`./png_files/curriculum_vitae-big.png`,  
concat_converted_result);  
})();
```

Provide practical demonstrations.

Additional usage examples are available for reference [here](#).

PDFTOPIC API

- [pdfbuffer\(buffer, page, progress\)](#)
- [bufferstoappend\(buffers, progress\)](#)
- [pdfcount\(buffer\)](#)
- [getDimensions\(buffers\)](#)

pdfbuffer(buffer, page, progress)

Initiate the conversion of a PDF to a buffer by supplying the function with a PDF and a specific page number or 'all' for the entire document. The function validates the input and processes the conversion accordingly, with an optional progress bar to track the process.

Functions

Convert a specific page of the PDF to Image/Buffer by supplying a file path.

```
pdfbuffer(buffer, page, progress)
```

- buffer - buffer pdf file
 - page - page number to convert to an image
 - progress - progress converting.
-

Converts PDF to Image/Buffer by supplying a file path

```
pdftobuffer(buffer, pages, progress)
```

- buffer - buffer pdf file
 - pages - page numbers to convert to image
 - set **pages** to **"all"** to convert all pages
 - **pages** also accepts an array indicating the page number e.g. **[1,2,3]**
 - also accepts number e.g. **1**
 - progress - progress converting
 - set **false** to disable progressbar (in default)
 - set **true** to enable progressbar
-

bufferstoappend(buffers, progress)

Kick-start the concatenation of multiple buffers by feeding the function with an array of buffers.

Functions

Concatenate multiple buffers into a single buffer by providing an array of buffers to the function. The function processes each buffer, appends them together, and returns the combined buffer.

```
bufferstoappend(buffers, progress)
```

- buffers - array of buffers images
-

pdftocount(buffer)

Start the process of counting the number of pages in a PDF by providing a PDF buffer to the function.

Functions

Determine the total number of pages in a PDF document by supplying the PDF to the function. The function loads the PDF and returns the page count.

```
pdftocount(buffer)
```

- buffer - buffer pdf file

getDimmentions(buffers)

To use the `getDimmentions` function, you need to provide an array of buffers as input. Each element of this array must represent an image in buffer form. Make sure that each buffer in the array corresponds to a valid image. Once you've supplied this array of buffers, you can use it as an argument to call the `getDimmentions` function.

Functions

The `getDimmentions` function is an asynchronous function that takes an array of buffers as an argument. The function returns an object containing the following information:

- `outputImgWidth`: the maximum width of all the images in the array.
- `outputImgHeight`: the sum total of the heights of all the images in the array.
- `dimmentions`: an array containing the dimensions (width and height) of each image in {width, height} format.
- `channels`: the number of elements in the dimensions array, corresponding to the number of images processed.

```
getDimmentions(buffers)
```

- buffers - array of buffers images

New updates

[UPDATE] Simple file conversion

```
ilyeselmajouti@MacBook-Pro-de-Ilyes pdftopic % npm run test

> pdftopic test
> jest

PASS ./index.test.js
  ✓ converting pdftopic@0.1.3 (single file) (470 ms)
  ✓ converting pdftopic@1.0 (single file) (296 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        1.006 s
Ran all test suites.
ilyeselmajouti@MacBook-Pro-de-Ilyes pdftopic %
```

As part of this update, a comparative evaluation of conversion speed was carried out on a single file between version `pdftopic@0.1.3` and the latest version `pdftopic@1.0.0`. This analysis revealed a significant improvement in conversion performance in the latest version, demonstrating a clear acceleration in the conversion process.

[UPDATE] Multiple file conversion

```
ilyeselmajouti@MacBook-Pro-de-Ilyes pdftopic % npm run test

> pdftopic test
> jest

PASS ./index.test.js
  ✓ converting pdftopic@0.1.3 (multiple file) (2752 ms)
  ✓ converting pdftopic@1.0 (multiple file) (855 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        3.741 s, estimated 4 s
Ran all test suites.
ilyeselmajouti@MacBook-Pro-de-Ilyes pdftopic %
```

In line with this update, a comparative analysis of conversion speed was carried out between the previous version `pdftopic@0.1.3` and the current version `pdftopic@1.0.0`, based on the processing of several files. This evaluation demonstrated a significant improvement in conversion performance in the more recent version, highlighting a clear acceleration in the conversion process.

[ADDITION] Concatenate multiple images into a single visual entity

```
ilyeselmajouti@MBP-de-Ilyes pdftopic % npm run test

> pdftopic@1.0.0 test
> jest

PASS ./index.test.js
  ✓ converting pdftopic@1.0.0 (all images in one image) (113 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.267 s, estimated 1 s
Ran all test suites.
ilyeselmajouti@MBP-de-Ilyes pdftopic %
```

In line with this update, the current version of `pdftopic@1.0.0` has been enhanced by the integration of an image merge feature, enabling multiple images to be combined into a single visual entity. Although this feature is still in the beta phase, we are currently working on an improved beta version, offering even faster merging speeds than those currently available.