



Processor



Rendering



PROCESSOR > GUIDES

Upgrade to Document Engine for PDF rendering



PSPDFKit Processor has been deprecated and replaced by [Document Engine](#). To migrate to Document Engine and unlock advanced document processing capabilities, refer to our migration guide. Learn more about these enhancements on our [blog](#).

To render pages in a document, post a multipart request to the `/build` API endpoint, including the desired page index or indexes in the `instructions.output.pages` field. To learn more about the schema for `/build instructions`, go to the [API Reference](#).

In the `/build` request, set the `instructions.output.type` parameter to `image`. By default, the `output.type` for `/build instructions` is set to `pdf`. The page index is provided via the `instructions.output.pages` field, and the rendered dimensions are provided via a `width`, `height`, or `dpi` option.

To learn more about available options for rendering, go to the [API Reference](#).

Only one option — `width`, `height`, or `dpi` — can be chosen. Other dimensions are calculated before rendering, so as to preserve the page aspect ratio of the rendered image.

The format of the rendered image can be controlled via the `format` option. Supported image formats are PNG, JPEG, WEBP, and TIFF.

Before you get started, make sure [Processor is up and running](#).



ASK AI

You can download and use either of the following sample documents for the examples in this guide:

❖ [Example eight-page PDF](#)

❖ [Example four-page PDF](#)

You'll be sending [multipart POST requests](#) with [instructions](#) to Processor's `/build` endpoint. To learn more about multipart requests, refer to our blog post on the topic, [A Brief Tour of Multipart Requests](#).

Check out the [API Reference](#) to learn more about the `/build` endpoint and all the actions you can perform on PDFs with PSPDFKit Processor.

Rendering a Page of a File on Disk

You can render a single page or multiple pages of a document.

Rendering a Single Page

The following example renders the second page (with index `1`) of a document. The response to this request is a PNG image file:

SHELL

HTTP

```
1 curl -X POST http://localhost:5000/api/build \  
2   -F document=@/path/to/example-document.pdf \  
3   -F instructions='{  
4     "parts": [  
5       {  
6         "file": "document"  
7       }  
8     ],  
9     "output": {  
10      "type": "image",  
11      "format": "png",  
12      "pages": {  
13        "start": 1,  
14        "end": 1  
15      }  
16    }  
17  }' \  
18   -o output-image.png
```



Rendering Multiple Pages

The following example renders the first, second, and third page (indexes `0`, `1` and `2`) of a document. The response to this request is a ZIP file containing all the images:

SHELL

HTTP

```
1 curl -X POST http://localhost:5000/api/build \  
2   -F document=@/path/to/example-document.pdf \  
3   -F instructions='{  
4     "parts": [  
5       {  
6         "file": "document"  
7       }  
8     ],  
9     "output": {  
10      "type": "image",  
11      "format": "png",  
12      "pages": {  
13        "start": 0,  
14        "end": 2  
15      }  
16    }  
17  }' \  
18   -o result.zip
```



Rendering a Page of a File from a URL

You can specify the document to render by providing a URL to the document instead of a path to the file:

SHELL

HTTP

```
1 curl -X POST http://localhost:5000/api/build \  
2   -F instructions='{  
3     "parts": [  
4       {  
5         "file": {  
6           "url": "https://pspdfkit.com/downloads/examples/paper.pdf"  
7         }  
8       }  
9     ],  
10    "output": {  
11      "type": "image",  
12      "format": "png",
```



```
13     "pages": {
14         "start": 1,
15         "end": 1
16     }
17 }
18 }' \
19 -o output-image.png
```

Was this helpful?

✓ YES

✗ NO

Questions? [Contact us](#)

