



Processor



Choose a Page



PROCESSOR &gt; GUIDES

# PDF Rendering API for Linux



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](#).



The `POST /render` API has been deprecated, and it may be removed in a future version of PSPDFKit. To render images from a PDF, please use the [build API](#) with `image` as the output format.

## Rendering a Document

To render a document, submit a `multipart/form-data` request to the `POST /render` API endpoint. The response depends on whether the source document contains a single page or multiple pages:

- ⚙ Responds with `application/zip`, which contains the rendered pages when rendering multiple pages.
- ⚙ Responds with the image data directly when rendering a single page.

## Request Headers

Available headers for `POST /render` are outlined below.



ASK AI

- Optional: `Authorization` — The [JSON Web Token \(JWT\)](#).
- Optional: `pspdfkit-pdf-password` — The password required for the PDF document to be processed.
- Optional: `X-Request-Id` — If this is set, the log statements associated with the HTTP request are marked with a `request_id` label. Logs correlated with the same request have the same request ID. This helps you determine which request triggered a specific response and what errors or warnings were emitted during the request processing. The request ID needs to be between 20 and 200 characters long.

## Request Parameters

Available parameters for `POST /render` are outlined below.

- `"file"`, `"url"`, Or `"generation"` :
  - `"file"` — The document be processed.
  - `"url"` — The URL of the document to be processed.
  - `"generation"` — A JSON object describing how the document should be generated. See the [PDF Generation schema](#) guide for more information.
- Optional: `"operations"` — The JSON object describing the operations to be performed on the supplied document. For all available operations, see the [available operations](#) guide.
- Optional: Attachment data for the operations — For example, the XFDF to be imported when using the `applyXfdf` document operation.

## Request

SHELL

HTTP

```
1 curl -X POST http://localhost:5000/render \  
2   -F file=@Example.pdf \  
3   -F options='{  
4     "options": {  
5       "pageIndex": 0,  
6       "width": 1000,  
7       "renderAnnotations": true,  
8       "format": "png"  
9     }  
10  
11
```



## Response

```
1 HTTP/1.1 200 OK
2 Content-Type: image/png
3
4 <image data>
```



## Options

The following options can be used in the `POST /render` API:

```
1 type Options = {
2   width?: number,
3   height?: number,
4   pageIndex?: number | Array<number> | 'all',
5   renderAnnotations?: boolean,
6   format?: 'png' | 'jpeg' | 'webp',
7 };
```



### width/height

The `width` and `height` options control the required dimensions of the rendered image.

Either the `width` or `height` parameter is required.

If both `width` and `height` parameters are missing or were provided at the same time, an HTTP response with the status `400` will be returned:

```
1 HTTP/1.1 400 Bad Request
2 Content-Type: application/json
```



### pageIndex

The optional `pageIndex` option sets the pages that will be rendered. The allowed values are:

- ❖ `number` — Renders a single page with this page index.
- ❖ `Array<number>` — Renders a list of pages with these page indexes.
- ❖ `"all"` — Renders all pages in a document.

This defaults to rendering all pages if the `pageIndex` option is missing.

Page indexes are zero-based non-negative integers. If an invalid `pageIndex` is used, an HTTP response with the status `400` will be returned:

```
1 HTTP/1.1 400 Bad Request
2 Content-Type: application/json
3
4 {
5     "description": "`pageIndex` must be a non-negative integer, list of non-ne
6     "reason": "invalid_options"
7 }
```

If using a `pageIndex` for a page that isn't present in a document, an HTTP response with the status `400` will be returned:

```
1 HTTP/1.1 400 Bad Request
2 Content-Type: application/json
3
4 {
5     "description": "Page index is out of bounds.",
6     "reason": "no_such_page"
7 }
```

## renderAnnotations

The optional `renderAnnotations` option controls whether or not annotations should be rendered. It defaults to `true`.

## format

The optional `format` option controls the format of the rendered image. The supported values of this option are `"png"`, `"jpeg"`, and `"webp"`.

If not specified, the default rendering format is resolved from the value of the `Accept` header. If multiple supported image MIME types are present, the rendering format is resolved in this priority order: `image/webp` , `image/jpeg` , and `image/png` . This defaults to `image/png` if the `Accept` header is missing or doesn't contain any supported image MIME type.

---

Was this helpful?



Questions? [Contact us](#)

