



Processor



PDF Generation



PROCESSOR > GUIDES > PDF GENERATION

Understanding PDF generation schema in 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](#).

To achieve a flexible layout and custom page configuration, PSPDFKit exposes multiple configuration options in a JSON object sent to the `/build` API endpoint.

To find out more about the `/build` API endpoint and how the PDF Generation JSON object should be sent, go to the [API Reference](#).

PDF Generation schema declaration

The following block outlines all the available options in the PDF Generation schema, which consists of input files, assets, and layout options:

```
1 type Orientation = "landscape" | "portrait";
2 type PageSize =
3   | "A0"
4   | "A1"
5   | "A2"
6   | "A3"
7   | "A4"
8   | "A5"
```



ASK AI

```

9   | "A6"
10  | "A7"
11  | "A8"
12  | "Letter"
13  | "Legal";
14
15  type PdfGenerationSchema = {
16    html: string, // The HTML file passed in the multipart request.
17    assets?: Array<string>, // All assets imported in the HTML. Reference the na
18    layout?: {
19      orientation?: Orientation,
20      size?: {
21        width: number,
22        height: number
23      } | PageSize, // {width, height} in mm or page size preset.
24      margin?: {
25        // Margin sizes in mm.
26        left: number,
27        top: number,
28        right: number,
29        bottom: number
30      }
31    }
32  };

```

`html` is mandatory, and if you provide this field only, it yields the minimum configurable operation. All the other fields will default to the following:

```

1  {
2    "assets": [],
3    "layout": {
4      "orientation": "portrait",
5      "size": "A4",
6      "margin": {
7        "left": 0,
8        "top": 0,
9        "right": 0,
10       "bottom": 0
11     }
12   }
13 }

```

Referencing assets

When designing an HTML page, it's common to split the design into multiple files, such as an HTML file, a CSS file, and image files. The PDF Generation command expects a flat directory structure, so any

referenced assets have to reside next to the HTML file and not in subdirectories.

The following shows how you would send a CSS file that's referenced in the HTML file:

```
1  <!DOCTYPE html>
2  <head>
3    <link rel="stylesheet" href="style.css" />
4  </head>
5  <html>
6    <body>
7      <h1>PDF Generation Header</h1>
8      
9    </body>
10 </html>
```

```
1  h1 {
2    font-size: xx-large;
3  }
```

CURL

HTTP

```
1  curl -X POST http://localhost:5000/build \
2    -F page.html=@/path/to/page.html \
3    -F style.css=@/path/to/style.css \
4    -F my-image.jpg=@/path/to/my-image.jpg \
5    -F instructions='{
6      "parts": [
7        {
8          "html": "page.html",
9          "assets": [
10             "style.css",
11             "my-image.jpg"
12           ]
13        }
14      ]
15    }' \
16    --output result.pdf
```

Note that JavaScript assets currently aren't supported in PDF Generation.

Assets passed in the multipart request must match the name used to reference the file in HTML. For example, if you have an image block, ``, the data representing the image in

the multipart request should have the name `my-image.jpg` .

Page layout

The `layout` object, which is part of the [PDF Generation schema](#), allows for customization of the PDF page layout and dimensions. All figures in this object are in reference to millimeters, and all pages will take on this configuration.

Was this helpful?

☒ YES

☐ NO

Questions? [Contact us](#)

