



Processor



Extraction



PROCESSOR > GUIDES > EXTRACTION

Extract text from PDFs and images easily



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](#).

This guide explains how to extract text from a PDF document.

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

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.

Sending the Request to Extract Data

To extract text from a document, post a multipart request to the `/build` API endpoint. In the instructions, specify the following output parameters:



ASK AI

- ❖ `type` specifies the output type. Set this to `json-content`.
- ❖ `plainText` is a Boolean value that determines whether to extract data as plain text.
- ❖ `structuredText` is a Boolean value that determines whether to extract data as structured text. Enabling this option gives you information about characters, lines, paragraphs, and words.
- ❖ `language` specifies the language used for recognizing text with optical character recognition (OCR). Sometimes, text is stored in a PDF or an image in a way that makes it so you cannot search or copy it. PSPDFKit's OCR engine allows you to recognize text and save it in a separate file where you can both search and copy and paste the text.

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": "json-content",  
11      "plainText": true,  
12      "structuredText": true,  
13      "language": "english"  
14    }  
15  }' \  
16   -o result.pdf
```

For more information on the `/build` instructions, refer to the [API Reference](#).

Example Data Extraction Response

```
1 {  
2   "pages": [  
3     {  
4       "pageIndex": 0,  
5       "plainText": "Lorem ipsum dolor sit amet, consectetur adipiscing elit."  
6       "structuredText": {  
7         "characters": [  
8           {  
9             "bbox": {  
10              "left": 0,
```

```
11         "top": 0,
12         "width": 100,
13         "height": 100
14     },
15     "value": "T"
16 }
17 ],
18 "lines": [
19     {
20         "bbox": {
21             "left": 0,
22             "top": 0,
23             "width": 100,
24             "height": 100
25         },
26         "firstWordIndex": 0,
27         "isRTL": false,
28         "isVertical": false,
29         "wordCount": 5
30     }
31 ],
32 "paragraphs": [
33     {
34         "bbox": {
35             "left": 0,
36             "top": 0,
37             "width": 100,
38             "height": 100
39         },
40         "firstLineIndex": 0,
41         "lineCount": 3
42     }
43 ],
44 "words": [
45     {
46         "bbox": {
47             "left": 0,
48             "top": 0,
49             "width": 100,
50             "height": 100
51         },
52         "characterCount": 4,
53         "firstCharacterIndex": 0,
54         "isFromDictionary": true,
55         "value": "word"
56     }
57 ]
58 }
59 }
60 ]
61 }
```

Was this helpful?

 YES

 NO

Questions? [Contact us](#)

