



Processor



Extraction



PROCESSOR &gt; GUIDES &gt; EXTRACTION

# Extract data from PDFs using Document Engine



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 data from PDFs using Processor.

You can extract the following pieces of information from a PDF document:

- ✧ Text
- ✧ Tables
- ✧ Key-value pairs. For more information, refer to the guide on [how key-value pair extraction works](#).

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. For more about multipart requests, refer to our blog post on the topic, [A Brief Tour of Multipart Requests](#).



ASK AI

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 data on all pages of a document, post a multipart request to the `/build` API endpoint. In the instructions, specify the following output parameters:

- ✧ `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.
- ✧ `keyValuePairs` is a Boolean value that determines whether to extract key-value pairs.
- ✧ `tables` is a Boolean value that determines whether to extract table data.
- ✧ `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      "keyValuePairs": true,  
14      "tables": true,  
15      "language": "english"  
16    }  
17  }' \  
18   -o result.pdf
```



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

# Interpreting the Data Extraction Response

The API response provides information about the data you included in the API request, such as:

- ✧ Plain text
- ✧ Structured text with information about characters, lines, paragraphs, and words
- ✧ Extracted key-value pairs
- ✧ Tables

## 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     "keyValuePairs": [
60         {
61             "confidence": 95.4,
62             "key": {
63                 "bbox": {
64                     "left": 0,
65                     "top": 0,
66                     "width": 100,
67                     "height": 100
68                 },
69                 "content": "#"
70             },
71             "value": {
72                 "bbox": {
73                     "left": 0,
74                     "top": 0,
75                     "width": 100,
76                     "height": 100
77                 },
78                 "content": "€",
79                 "dataType": "Currency"
80             }
81         }
82     ],
83     "tables": [
84         {
85             "confidence": 95.4,
86             "bbox": {
87                 "left": 0,
88                 "top": 0,

```



```

89         "width": 100,
90         "height": 100
91     },
92     "cells": [
93         {
94             "bbox": {
95                 "left": 0,
96                 "top": 0,
97                 "width": 100,
98                 "height": 100
99             },
100             "rowIndex": 0,
101             "columnIndex": 0,
102             "isHeader": true,
103             "text": "Invoice number"
104         }
105     ],
106     "columns": [
107         {
108             "bbox": {
109                 "left": 0,
110                 "top": 0,
111                 "width": 100,
112                 "height": 100
113             }
114         }
115     ],
116     "lines": [
117         {
118             "bbox": {
119                 "left": 0,
120                 "top": 0,
121                 "width": 100,
122                 "height": 100
123             },
124             "isVertical": false,
125             "thickness": 0
126         }
127     ],
128     "rows": [
129         {
130             "bbox": {
131                 "left": 0,
132                 "top": 0,
133                 "width": 100,
134                 "height": 100
135             }
136         }
137     ]
138 }
139 ]
140 }
141 ]
142 }

```



---

Was this helpful?

 YES

 NO

---

Questions? [Contact us](#)

