



Processor



PDF Generation



PROCESSOR &gt; GUIDES &gt; PDF GENERATION

# Edit a Generated PDF



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](#).

After generating a PDF, you may still want to perform extra operations to finalize your document. Examples of these operations include adding a cover page, adding watermarks, setting page labels, and more. Any `action` provided by Processor can be used with PDF Generation. To learn more about the actions you can perform with Processor, see the [API Reference](#).

This guide shows how PDF Generation and additional actions are combined in one command to produce the final 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.



ASK AI

# Generating a PDF from HTML

The example consists of a simple styled form, which can be seen in the following HTML:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <style>
5       body {
6         font-size: 16px;
7         line-height: 1.6em;
8         font-family: "Helvetica", Arial, sans-serif;
9         color: #3d434e;
10      }
11
12      h1 {
13        font-size: 42px;
14        font-weight: normal;
15        color: #142132;
16        margin-block-start: 0.3em;
17      }
18
19      h2 {
20        font-size: 32px;
21        font-weight: normal;
22        color: #142132;
23        margin-block-start: 0.3em;
24      }
25
26      h3 {
27        font-size: 12px;
28        font-weight: normal;
29        text-transform: uppercase;
30        letter-spacing: 0.1em;
31        color: #4739e5;
32      }
33
34      .container {
35        max-width: 500px;
36        margin: 100px auto auto;
37        page-break-after: always;
38      }
39
40      .grid {
41        display: flex;
42        flex-direction: row;
43        justify-content: space-between;
44      }
45
46      .column {
47        display: flex;
48        flex-direction: column;
```



```

49     }
50
51     .form-list {
52         list-style: none;
53         padding: 0;
54     }
55
56     .list-item {
57         margin-bottom: 1.5rem;
58     }
59
60     .form-border {
61         padding: 15px 10px;
62         border-width: 2px;
63     }
64
65     .form-text {
66         font-size: 18px;
67         line-height: 1.5;
68     }
69
70     .full-width {
71         width: 100%;
72     }
73
74     .expanded-checkbox {
75         width: 30px;
76         height: 30px;
77     }
78
79     .checkbox-label {
80         display: flex;
81         align-items: center;
82         padding-left: 10px;
83     }
84
85     .checkbox-list {
86         display: flex;
87     }
88 </style>
89 <title>Data Science Online Course</title>
90 </head>
91
92 <body>
93     <div class="container">
94         <h3>Data Science</h3>
95         <h1>Course Signup</h1>
96         <p>
97             Learn how to make informed decisions, create beautiful visualizations,
98             and even try to predict future events through Machine Learning.
99         </p>
100
101         <form>
102             <ul class="form-list">
103                 <li class="list-item">

```



```

104         <label for="knowledge-level"></label>
105         <select
106             id="knowledge-level"
107             class="full-width form-text form-border"
108         >
109             <option selected>-- Knowledge level --</option>
110             <option value="beginner">Beginner</option>
111             <option value="intermediate">Intermediate</option>
112             <option value="professional">Professional</option>
113         </select>
114     </li>
115     <li class="list-item">
116         <div class="grid">
117             <div class="column">
118                 <label for="first-name">First Name</label>
119                 <input
120                     type="text"
121                     id="first-name"
122                     class="form-text form-border"
123                 />
124             </div>
125             <div class="column">
126                 <label for="last-name">Last Name</label>
127                 <input
128                     type="text"
129                     id="last-name"
130                     class="form-text form-border"
131                 />
132             </div>
133         </div>
134     </li>
135     <li class="list-item">
136         <div class="column">
137             <label for="email">Email</label>
138             <input type="text" id="email" class="form-text form-border" />
139         </div>
140     </li>
141     <li class="list-item">
142         <div class="column">
143             <label for="phone-number">Phone Number</label>
144             <input
145                 type="text"
146                 id="phone-number"
147                 class="form-text form-border"
148             />
149         </div>
150     </li>
151     <li class="checkbox-list">
152         <input type="checkbox" id="terms" class="expanded-checkbox" />
153         <label class="checkbox-label" for="terms"
154             >I have read and agreed with the terms and conditions.</label>
155     >
156 </li>
157 </ul>
158 </form>

```



```
159     </div>
160     <div class="container">
161         <h2>Terms and Conditions</h2>
162         <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. ...</p>
163     </div>
164 </body>
165 </html>
```

This form includes some text boxes, a select box, and a checkbox. These have all been styled to ensure they display nicely on an A4 page.

In addition to the form, there's a Terms and Conditions section, which is referenced on the form. To ensure this section is generated on a separate page, set the `page-break-after` property to `always` in the `container` class. This forces a page break when generated.

To generate a PDF with the form and the terms and conditions, send a multipart message using the `curl` example below. This code has margins defined for the page layout of the generated PDF:

SHELL

HTTP

```
1  curl -X POST http://localhost:5000/api/build \
2    -F page.html=@/path/to/page.html \
3    -F instructions='{
4      "parts": [
5        {
6          "html": "page.html",
7          "layout": {
8            "size": "a4",
9            "margin": {
10             "left": 10,
11             "right": 10,
12             "top": 40,
13             "bottom": 10
14           }
15         }
16       ]
17     }' \
18    -o result.pdf
```



## Adding a Cover Page

To add a cover page to the generated PDF, send a request like in the example below to the `/build` endpoint. To learn more about combining various documents and pages, read our [merging documents](#) guide.

The name of the cover page in the multipart request needs to be the same name referenced by `file` in the `FilePart`. If you don't have a suitable PDF document, use the sample [cover page PDF](#):

SHELL

HTTP

```
1 curl -X POST http://localhost:5000/api/build \  
2   -F page.html=@/path/to/page.html \  
3   -F cover-page=@/path/to/cover-page.pdf \  
4   -F instructions='{  
5     "parts": [  
6       {  
7         "file": "cover-page"  
8       },  
9       {  
10        "html": "page.html",  
11        "layout": {  
12          "size": "a4",  
13          "margin": {  
14            "left": 10,  
15            "right": 10,  
16            "top": 40,  
17            "bottom": 10  
18          }  
19        }  
20      }  
21    ]  
22  }' \  
23   -o result.pdf
```



## Adding a Watermark

To add a `watermark` to the document, use the example below.

To learn more about how to add watermarks to documents, refer to our [watermark](#) guide.

SHELL

HTTP

```
1 curl -X POST http://localhost:5000/api/build \  
2   -F page.html=@/path/to/page.html \  
3   -F cover-page=@/path/to/cover-page.pdf \  
4   -F instructions='{  
5     "parts": [  
6       {  
7         "file": "cover-page",  
8         "watermark": "My Watermark"  
9       },  
10      {  
11        "html": "page.html",  
12        "layout": {  
13          "size": "a4",  
14          "margin": {  
15            "left": 10,  
16            "right": 10,  
17            "top": 40,  
18            "bottom": 10  
19          }  
20        }  
21      }  
22    ]  
23  }' \  
24   -o result.pdf
```



```
4 -F instructions='{
5   "parts": [
6     {
7       "file": "cover-page"
8     },
9     {
10      "html": "page.html",
11      "layout": {
12        "size": "a4",
13        "margin": {
14          "left": 10,
15          "right": 10,
16          "top": 40,
17          "bottom": 10
18        }
19      }
20    }
21  ],
22  "actions": [
23    {
24      "type": "watermark",
25      "text": "Property of PSPDFKit",
26      "width": 250,
27      "height": 200,
28      "left": 0,
29      "bottom": "50%",
30      "opacity": 0.5,
31      "rotation": 10,
32      "fontSize": 50,
33      "fontColor": "#FF0000",
34      "fontStyle": [
35        "italic",
36        "bold"
37      ],
38      "fontFamily": "Helvetica"
39    }
40  ]
41 }' \
42 -o result.pdf
```

To learn more, go to the [API Reference](#) and the [overview](#) guide.

---

Was this helpful?

✓ YES

✗ NO

Questions? [Contact us](#)

