



Web



PDF generation



WEB > GUIDES > PSPDFKIT SERVER > PDF GENERATION

Create PDFs from HTML with Mustache templating



PSPDFKit Server 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 will take you through the process of generating a PDF using data you supply.

JAVASCRIPT

PYTHON

JAVA

C# (.NET)

PHP

Using a templating language such as [Mustache](#) allows you to inject your data at runtime. In regard to PDF Generation, this means it gives you the possibility to customize HTML before sending it along for generation. Injecting data can be useful in situations such as name and date replacement, or when generating dynamic lists in invoices.

In this guide, you'll implement name and date replacement with the use of Mustache in JavaScript. Note

`{{name}}` and `{{date}}` in the following HTML:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <div class="address">
5       John Smith<br/>
6       123 Smith Street<br/>
```



ASK AI

```

7      90568 TA<br/>
8      <br/>
9      {{date}}
10     </div>
11     <div class="subject">Subject: PDF Generation FTW!</div>
12     <div>
13         <p>
14             PDF is great!
15         </p>
16     </div>
17     <div>{{name}}<br/></div>
18 </body>
19 </html>

```

First, the data for `{{name}}` and `{{date}}` needs to be defined. In practice, this data may come from an external source or database, but for this example, you'll define the data in a JSON file:

```

1  {
2    "name": "John Smith Jr.",
3    "date": "29 February, 2020"
4  }

```

Now, run the `mustache` command to produce a final HTML document with the template arguments replaced:

```

1  const mustache = require("mustache");
2  const fs = require("fs");
3
4  const page = fs.readFileSync("page.mustache").toString();
5  const data = JSON.parse(fs.readFileSync("data.json").toString());
6
7  const outputHTML = mustache.render(page, data);

```

Run the JavaScript with Node.js:

```
node mustache-example.js
```

Once you've created your HTML in JavaScript, send a request to the `/api/documents` endpoint, sending the PDF Generation schema with the HTML file you just created:

```

SHELL  HTTP
1  curl -X POST http://localhost:5000/api/documents \
2    -H "Authorization: Token token=<API token>" \
3    -F page.html=@/path/to/page.html \
4    -F generation='{
5
6

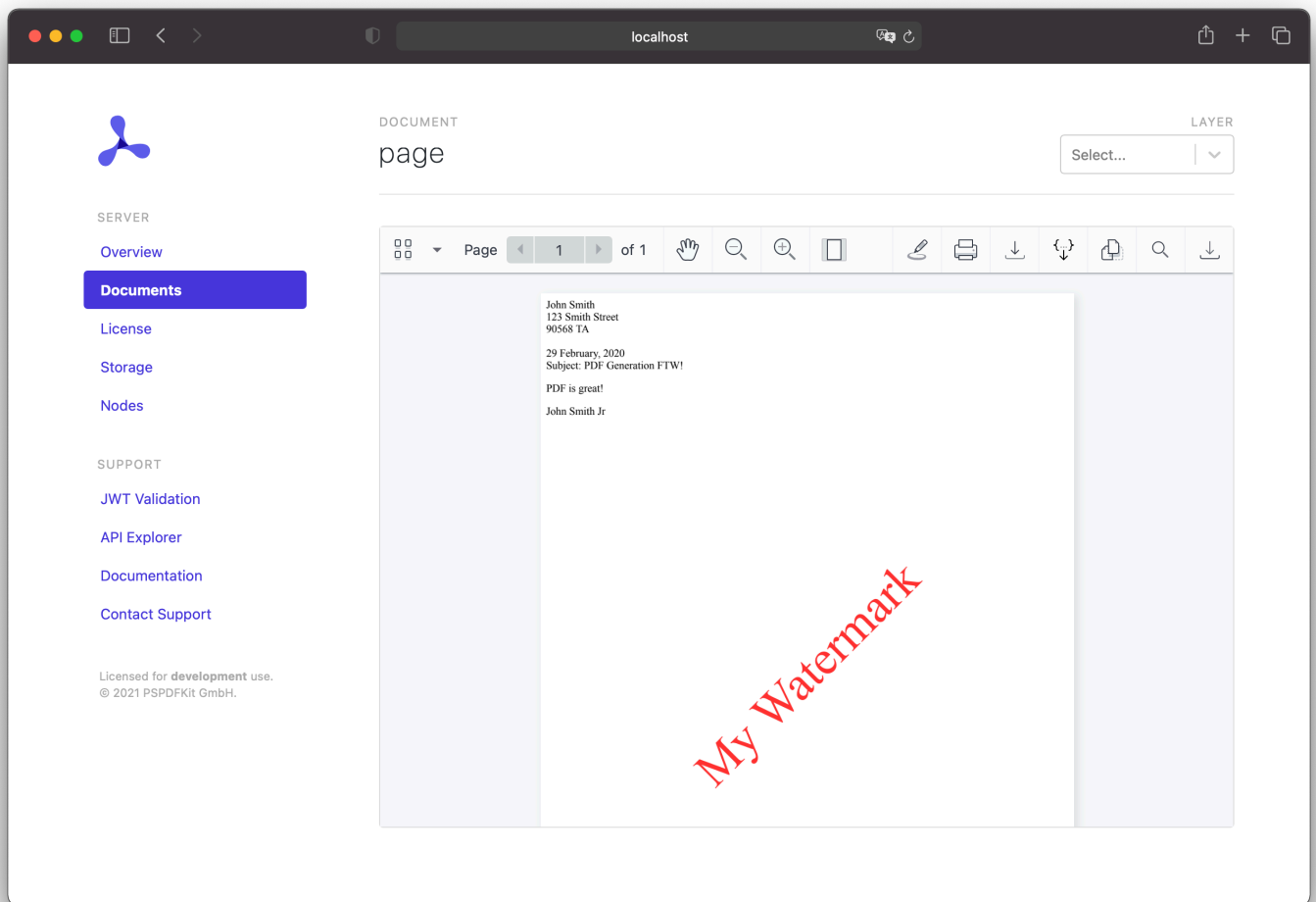
```

Adding watermarks

Once you have the basics of PDF generation down, you might want to watermark your PDF. You can do this by including some additional HTML:

```
1  <!DOCTYPE html>
2  <html>
3    <body>
4      <div style="position: fixed;
5        top: 50%;
6        left: 50%;
7        font-size: 72px;
8        color: red;
9        opacity: 80%;
10       transform: rotate(-45deg);
11       width: 500px;
12       height: 500px;
13       margin-top: -250px;
14       margin-left: -250px;
15       text-align: center;
16       vertical-align: middle;
17       line-height: 500px;">
18        My Watermark
19      </div>
20      <div class="address">
21        John Smith<br/>
22        123 Smith Street<br/>
23        90568 TA<br/>
24        <br/>
25        {{date}}
26      </div>
27      <div class="subject">Subject: PDF Generation FTW!</div>
28      <div>
29        <p>
30          PDF is great!
31        </p>
32      </div>
33      <div>{{name}}<br/></div>
34    </body>
35  </html>
```

This element will be rendered on all pages on top of all other content. The HTML above uses some text, but you could also use an image. The watermark can also be positioned any way you want.



Adding a cover page

There are two ways to add a cover page to a PDF you generated: You can generate the cover page as part of the HTML, or you can use a PDF you already have.

Using HTML

If you want to add a cover page to your generated PDF, you can do so by adding additional HTML:

```
1  <!DOCTYPE html>
2  <html>
3    <body>
4      <div style="display: block; width: 100%; height: 100%; page-break-after: a
5        <h1>My Cover Page</h1>
6      </div>
7      <div class="address">
8        John Smith<br/>
9        123 Smith Street<br/>
10       90568 TA<br/>
11       <br/>
12       {{date}}
13     </div>
14     <div class="subject">Subject: PDF Generation FTW!</div>
15     <div>
16       <p>
17         PDF is great!
```

```
18     </p>
19   </div>
20   <div>{{name}}<br/></div>
21 </body>
22 </html>
```

This will add another page before the rest of the content, and you can use the new page as your cover page.

Using a PDF

Instead of adding some HTML as a cover page, you can add a PDF. This can be done by applying an

`importDocument` operation on upload:

SHELL

HTTP

```
1 curl -X POST http://localhost:5000/api/documents \
2   -H "Authorization: Token token=<API token>" \
3   -F page.html=@/path/to/page.html \
4   -F cover.pdf=@/path/to/cover.pdf \
5   -F generation='{
6     "html": "page.html"
7   }' \
8   -F operations='{
9     "operations": [
10      {
```

```
11     "type": "importDocument",
12     "beforePageIndex": 0,
13     "document": "cover.pdf"
14   }
15 ]
16 }'
```

The provided PDF will be appended before the first page of the HTML, and it'll serve as your cover page.

Was this helpful?

✓ YES

✗ NO

Questions? [Contact us](#)

