



Processor



Variable Data



PROCESSOR > GUIDES > PDF GENERATION > VARIABLE DATA

Generate a PDF Containing Variable Data in Python



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](#).

When generating a PDF, it may not be possible to foresee all of its contents ahead of time. For that reason, we advise using a templating language such as [Mustache](#).

Templating languages allow for the injection of data at runtime. In regard to the PDF Generation feature, 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 take the letter example from the [\[creating a PDF from scratch\]\[creating-a-pdf\]](#) guide and implement date and name replacement with the use of Mustache in Python. (Note `{{name}}` and `{{date}}` in the following HTML):

```
1 <!DOCTYPE html>
2 <head>
3   ...
4 </head>
5 <html>
6   <body>
7     <div class="address">
8       John Smith<br/>
```



ASK AI

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

Next, replace your template data with something real. Define the data to replace the template arguments in code, though in practice, this data may come from an external source or database.

To perform the replacement, use the [Chevron](#) package, which is an implementation of the [Mustache](#) templating system.

The following will stream the `page.mustache` file (the HTML seen above) into Chevron to perform the replacement of `{{name}}` and `{{date}}`:

```
1  with open('page.mustache', 'r') as f:
2      final_html = chevron.render(f, {'name': 'John Smith Jr.', 'date': '29 Febr'
```

This example shows how you can use [Chevron](#) in Python, although it's worth noting that there are many other templating language alternatives that may better suit your environment.

Was this helpful?

✓ YES

✗ NO

Questions? [Contact us](#)