



Processor



Redaction



PROCESSOR > GUIDES > REDACTION

Efficiently search and redact PDFs in Linux



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](#).

PSPDFKit Processor lets you create redactions using simple text search rules. Any piece of text matching a provided query will be covered by redaction annotations. To create redactions, use the `createRedactions` action with a `text` strategy.

To learn more about redaction strategies, go to the [API Reference](#).

You can configure some visual aspects of the redaction annotation, including its background color and overlay text, by passing an optional `content` object to the `createRedactions` action. To learn more about all configurable options, refer to the [API Reference](#).

The following example redacts any occurrence of the phrase ACME Bank:

```
1  {
2    "type": "createRedactions",
3    "strategy": "text",
4    "strategyOptions": {
5      "text": "ACME Bank"
6    }
7  }
```



ASK AI

Applying Redactions

After redaction annotations are created, they need to be applied to the document to effectively and permanently remove the covered content. You can achieve this by adding the `applyRedactions` action to the `/build` instructions.

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.

Creating and Applying Redactions in a File on Disk

Send a multipart request to the `/build` endpoint attached with the input file and the `instructions` JSON:

SHELL

HTTP

```
1 curl -X POST http://localhost:5000/api/build \  
2   -F file=@/path/to/example-document.pdf \  
3   -F instructions='{  
4     "parts": [  
5       {  
6         "file": "document",  
7         "actions": [  
8           {  
9             "type": "createRedactions",  
10            "strategy": "text",  
11            "strategyOptions": {  
12              "text": "ACME Bank"  
13            }  
14          },  
15          {  
16            "type": "applyRedactions"  
17          }  
18        ]  
19      }  
19 }
```



```
20 ]
21 }' \
22 -o result.pdf
```

This creates redaction annotations and applies them to the file, removing the content beneath them.

Creating and Applying Redactions in a File from a URL

Send a request to the `/build` endpoint and include a URL pointing to the file you want to redact:

SHELL

HTTP

```
1 curl -X POST http://localhost:5000/api/build \
2   -F instructions='{
3     "parts": [
4       {
5         "file": {
6           "url": "https://pspdfkit.com/downloads/examples/credit-card-application.pdf"
7         },
8         "actions": [
9           {
10            "type": "createRedactions",
11            "strategy": "text",
12            "strategyOptions": {
13              "text": "ACME Bank"
14            }
15          },
16          {
17            "type": "applyRedactions"
18          }
19        ]
20      }
21    ]
22  }' \
23   -o result.pdf
```

This creates redaction annotations and applies them to the file, removing the content beneath them.

Was this helpful?

✓ YES

✗ NO

Questions? [Contact us](#)

