



Java



Regex patterns

[HOME](#)  [GUIDES](#)  [JAVA](#)  [REDACTION](#)  [REGEX PATTERNS](#)

# Using regex patterns to redact PDFs in Java



COPY PAGE



To customize search data even further, Nutrient supports searching via regular expressions. Regular expressions can be used either when you're trying to match a word/phrase, or for more complicated expressions, such as when pattern matching complicated identification numbers. If you ever find that Nutrient is missing a preset, you can implement it as an expression:

```
RedactionProcessor.create()  
  .addRedactionTemplates(new RedactionRegEx.Builder("A").build())  
  .redact(document);
```

In the above example, the `RedactionProcessor` API will redact any instances of the uppercase character A.

It's possible to take this further and define a complicated pattern for removal — for example, a four-digit code that's followed by three letters:



ASK AI

```
RedactionProcessor.create()
```

```
.addRedactionTemplates(new RedactionRegex.Builder("\\d{4}[A-Za-z]{3}").build()  
.redact(document);
```

Note that to provide any regular expression escape sequence, it's necessary that you double escape the character in order for the literal string to be valid, just like with '\d', which is equivalent to '\\d' in regular expressions.

## Applying redactions

All that's required to apply the redaction annotations is to pass an option to the [ `save` ] API. From there, Nutrient will do the processing and redact all the relevant information:

```
document.save(new DocumentSaveOptions.Builder().applyRedactionAnnotations(true).build());
```

---

### Was this helpful?

✓ YES

✗ NO



Copyright 2025 Nutrient. All rights reserved.