

Java



Programmatically



HOME > GUIDES > JAVA > REDACTION > PROGRAMMATICALLY

Programmatically redact PDFs in Java



COPY PAGE



You can create redactions programmatically via `addAnnotationJson`. The shape of the JSON passed to create the annotation can be found in the [Instant JSON](#) guide. In short, an array of rectangles is required to define the regions that should be covered by the redaction annotation.

You also have a few customization options for how a redaction should look, both while in its marked state, which is when the redaction has been created but not yet applied, and in its redacted state, which is when the redaction has been applied. It isn't possible to change the appearance once a redaction has been applied, since the redaction annotation will be removed from the document in the process of applying the redactions. Here's a list of available customization options for redactions:

- ❖ `overlayText` can be used to set the text that should be displayed at the specified region when a redaction has been applied.
- ❖ `repeatOverlayText` defines whether the overlay text should be drawn only once or repeated to fill the entire redaction area. This defaults to disabled, which means the overlay text is only drawn once. It has no effect if there's no overlay text specified.
- ❖ `color` can be used to change the color of the overlay text. It has no effect if there is no overlay text specified. This defaults to red, `#ff0000`.



ASK AI

- ✦ `fillColor` specifies the background color of the redaction area after it has been applied. The color is drawn on all the specified `rect` s. This defaults to black, `#000000`.
- ✦ `outlineColor` specifies the color used for the redaction's border in its marked state. This defaults to red, `#ff0000`.

```
JSONObject redactionAnnotation = new JSONObject();
redactionAnnotation.put("bbox", new float[]{10, 10, 200, 100});
redactionAnnotation.put("creatorName", "User");
redactionAnnotation.put("fillColor", "#000000");
redactionAnnotation.put("opacity", 1);
redactionAnnotation.put("pageIndex", 0);
redactionAnnotation.put("type", "pspdfkit/markup/redaction");
redactionAnnotation.put("v", 1);

document.getAnnotationProvider().addAnnotationJson(redactionAnnotation);
```

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).b
```

Was this helpful?

✓ YES

✗ NO



Copyright 2025 Nutrient. All rights reserved.