






Java Import or export edits [HOME](#)  [GUIDES](#)  [JAVA](#)  [EDITOR](#)  [IMPORT OR EXPORT EDITS](#)

Import or export PDF edits in Java



COPY PAGE



Document Editor operations can be written in JSON and imported using the `DocumentEditor.addOperations` method. The JSON format is described in detail in the Web API documentation. Nutrient Java SDK uses the same format.

Operations can be chained up in code. For example, the following removes a page, rotates another page, and adds page labels:

```
// Create the operations using `JSONObject`s.
JSONObject movePagesJson = new JSONObject();
movePagesJson.put("type", "removePages");
movePagesJson.put("pageIndexes", new int[] {3, 4});
JSONObject rotatePagesJson = new JSONObject();
rotatePagesJson.put("type", "rotatePages");
rotatePagesJson.put("pageIndexes", new int[] {0, 1});
JSONObject addLabelsJson = new JSONObject();
addLabelsJson.put("type", "setPageLabel");
addLabelsJson.put("pageIndexes", new int[] {0});
addLabelsJson.put("pageLabel", "Great page!");

// Pack into a JSON array.
JSONArray operationsArray = new JSONArray();
operationsArray.put(movePagesJson);
```



ASK AI

```
operationsArray.put(rotatePagesJson);
operationsArray.put(addLabelsJson);
```

The same thing can be achieved with a JSON string, either in code or loaded from a file. You can also add a blue page to spice things up a bit:

```
final String jsonString = "[ "
    + "{ 'type': 'removePages', 'pageIndexes': [1] }, "
    + "{ 'type': 'rotatePages', 'pageIndexes': [0], 'rotateBy': 90 }, "
    + "{ 'type': 'setPageLabel', 'pageIndexes': [0], 'pageLabel': 'My custom label' }, "
    + "{ 'type': 'addPage', 'afterPageIndex': 0, 'backgroundColor': '#0000FF' } ]";
final JSONArray operationsArray = new JSONArray(jsonString);
```

These are applied using `addOperations`:

```
final DocumentEditor documentEditor = pdfDocument.createDocumentEditor();
documentEditor.addOperations(operationsArray);
// Don't forget to save the operations.
final File file = new File("documentEditorOutput.pdf");
documentEditor.saveDocument(new FileDataProvider(file));
```

Was this helpful?

✓ YES

✗ NO



Copyright 2025 Nutrient. All rights reserved.