



Processor



Choose a Page



PROCESSOR > GUIDES

Error Status Codes



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](#).

Processor's `/build` API uses the following HTTP status codes to communicate success and failure of your requests. We use three main ranges to communicate the result of your requests:

- ✧ `200` — The API call was successful.
- ✧ `4XX` — Invalid data was supplied, or a precondition wasn't met.
- ✧ `5XX` — Internal error of Processor.

Here's a list of common status codes in the `4XX` range that you can expect, along with what they mean:

- ✧ `413` — Request exceeds the maximum input size, meaning either a single part, or the sum of all parts, is too large.
- ✧ `422` — The resulting PDF of your API call exceeds the maximum output file size.
- ✧ `401` — No API token is specified, or the API token you specified isn't valid.
- ✧ `400` — Any other user error with your requests. One possible response payloads is:

✧ `{ "details": "There was an error with one or more parts of the instructions." }`



ASK AI

Error Details

When PSPDFKit Processor encounters an error with the provided inputs, an error JSON object is returned. This object has the following syntax:

```
1  type FailingPath = {
2    path: string;
3    details: string;
4  };
5
6  type Error = {
7    details: string;
8    status: number;
9    requestId: string;
10   failingPaths: FailingPath[];
11  };
```

The most important properties are:

- ❖ `details` — A general description of the error.
- ❖ `status` — The HTTP status of the response. It's added in case the request status is modified by any proxy it might go through.
- ❖ `requestId` — An internal ID attached to each request. This parameter is helpful when contacting Support to help associate your support request with the internal logging.
- ❖ `failingPaths` — Points to the specific parts of the request that caused the failure. It helps pinpoint what to adjust to make your request work. In case of multiple errors in your request, it contains everything that could be found until processing had to be stopped.
 - ❖ `path` — The path that points to the exact key in the `instructions` object that caused the request to fail. This path will be in the JSONPath format.
 - ❖ `details` — A detailed error message for the failing path.

Examples

The following examples are meant to give an idea of how common errors returned by the API might look.

Wrong Page Index

Given the `instructions` for a PDF with only two pages:

```
1  {
2    "parts": [{ "file": "pdf-1", "pages": { "start": 2, "end": 2 } }]
3  }
```



You'd get the following error:

```
1  {
2    "details": "There was an error with one or more parts of the instructions.",
3    "status": 400,
4    "requestId": "random-id",
5    "failingPaths": [
6      {
7        "path": "$.instructions.parts[0].pages.end",
8        "details": "index 2 is out of range in document with 2 pages"
9      },
10     {
11       "path": "$.instructions.parts[0].pages.start",
12       "details": "index 2 is out of range in document with 2 pages"
13     }
14   ]
15 }
```



The `failingPaths` array tells you exactly which part of your request caused the failure. In this case, both the `start` page and `end` page are out of range.

Illegal File Format

Given the `instructions` for multiple files, including one text file:

```
1  {
2    "parts": [
3      { "file": "pdf-1" },
4      { "file": "pdf-2" },
5      { "file": "pdf-3" },
6      { "file": "text-4" }
7    ]
8  }
```



You'd get the following error:



```
1 {
2   "details": "There was an error with one or more parts of the instructions.",
3   "status": 400,
4   "requestId": "random-id",
5   "failingPaths": [
6     {
7       "path": "$.text-4",
8       "details": "is of the unsupported mimetype text/plain"
9     }
10  ]
11 }
```

The `failingPaths` array doesn't point to a location inside the `instructions` object. Instead, it points to `$.text-4`. This lets you know that the uploaded file with the name `text-4` is what caused the error to be returned.

Was this helpful?

✓ YES

✗ NO

Questions? [Contact us](#)

