pdf

The `pdf` endpoint is arguably the most useful, and most complex, endpoint the DynamicPDF API offers. Use the `pdf` endpoint to create simple one-page PDF documents (cover pages for example), create documents/reports using DLEX files, convert images to PDFs, and to merge PDFs into a combined PDF.

INFO

The `pdf` endpoint takes a JSON instructions document that provides instructions for creating, transforming, merging, and formatting inputs into a combined PDF. Refer to documentation on the instructions schema for information on how to use the `pdf` endpoint.

- Instructions Overview
- JSON Instructions Schema
- Instructions Schema Diagram
- Instructions Schema
- `pdf` Inputs Overview

Check out our blog for tips and tutorials!

You can also combine the results from these tasks to merge into a single PDF document. You perform one or more of the following tasks using the `pdf` endpoint.

CAUTION

The `pdf` endpoint is more complex than the other API endpoints. You must understand the `pdf` endpoint's schema for creating instructions if you use this endpoint via a REST call rather than one of our client libraries. pdf.instructions - JSON Schema.

TASK	DESCRIPTION
`page`	Create a simple one-page PDF document such as a cover-page.

TASK	DESCRIPTION
`dlex`	Create a report/document from a DLEX file combined with JSON data.
`html`	Converts an HTML document to a PDF.
`word`	Converts a Word document to a PDF.
`excel`	Converts an Excel document to a PDF.
`image`	Convert an image to a PDF.
`pdf`	Specify a pre-existing PDF to merge with other PDF documents.
One or more of the four listed tasks.	Combine any of the five previous tasks to create and return a combined PDF document.

Request/Response

REQUEST	REQUEST DATA	RESPONSE
POST	form fields	PDF document as byte-array

The `pdf` endpoint takes an HTTP POST form submission, the Instructions JSON document is sent as a file upload in the form's body.

```
POST https://api.dpdf.io/v1.0/pdf HTTP/1.1
Authorization: Bearer DP.xxx-api-key-xxx
```

Parameters

PARAMETERS	TYPE	DESCRIPTION	
Header			
`Authorization`	Bearer DP.V9xxxx	The API key used to authenticate endpoint.	*REQUIRED*
Form Field			
`Resource`	file	The resource path and filename on your local system.	
`Instructions`	file	Path to the local instructions JSON file.	*REQUIRED*

Example

```
curl https://api.dpdf.io/v1.0/pdf
-H"Authorization:Bearer DP.xxx-api-key-xxx"
-F "Instructions=@C:/instructions-simple-page.json"
-o simple-page.pdf
```

▶ Run in Postman

> **INFO**
>
> Refer to pdf in the Client Libraries Users Guide section for this example using the DynamicPDF API's client libraries.

Instruction Document

Understanding the `pdf` endpoint requires understanding the `pdf.instructions.schema.json` JSON schema, as the instructions document is where the `pdf` endpoint does its real work. Calling the `pdf` endpoint requires a JSON document that contains the instructions on the tasks you wish performing when calling the endpoint. You combine one or more of these tasks (or more accurately `inputs`) to create a final PDF.

Creating Instructions

> **INFO**
>
> Refer to pdf.instructions - JSON Schema endpoint for a detailed description of the `pdf.instructions` schema.

The first thing every instructions document must contain is at least one input type in the input array. The input is the source for the final PDF. Inputs can be images, DLEX files, and other PDFs. For example, in the following code snippet the instructions document contains two inputs which are merged to create a merged PDF.

```
{
  "author": "John Doe",
  "title": "Getting Started - Merge Two PDFs",
  "inputs": [
    {
      "type": "pdf",
      "resourceName": "getting-started/A.pdf"
    },
    {
      "type": "pdf",
      "resourceName": "getting-started/myimage.png"
    }
  ]
}
```

More on Resource

The resource field's value in `instructions.json` is either a file name or the path to the resource in your cloud storage. If the file is local to your system, specify the file name in the `instructions.json`. However, when using it locally, you must also specify the resource's location or content.

> **INFO**
>
> Remember, the local resource is actually binary that could come from multiple sources and not necessarily a physical file on your filesystem, as the pdf endpoint is a multipart-form POST submission. The only requirement is that the POST form submission submits the resource as bytes.

The following `instructions.json` specifies two resources; the first is a resource provided to the endpoint as a multipart-form submission. The other is a resource whose path is in cloud storage.

```
{
  "author": "John Doe",
  "inputs": [
    {
      "type": "pdf",
      "resourceName": "DocumentA.pdf"
    },
    {
      "type": "pdf",
      "resourceName": "example/DocumentB.pdf"
    }
  ]
}
```

Calling the preceding JSON using cURL specifies that `DocumentA.pdf` is on you local filesystem.

```
curl https://api.dpdf.io/v1.0/pdf -H "Authorization:Bearer DP.xxx-api-key-xxx"
-F "Instructions=@C:/temp/example/instructions.json"
-F "Resource=@c:/temp/example/DocumentA.pdf"
-o simple-out.pdf
```

Figure 1. An example request and response.

More on Inputs

Understanding the `inputs` array is fundamental to understanding the `pdf` endpoint and the processing that occurs to produce a finished PDF. **Two or more inputs indicate the two inputs will be merged into a new PDF document.** For example, consider a PDF input and an image that occur in that order.

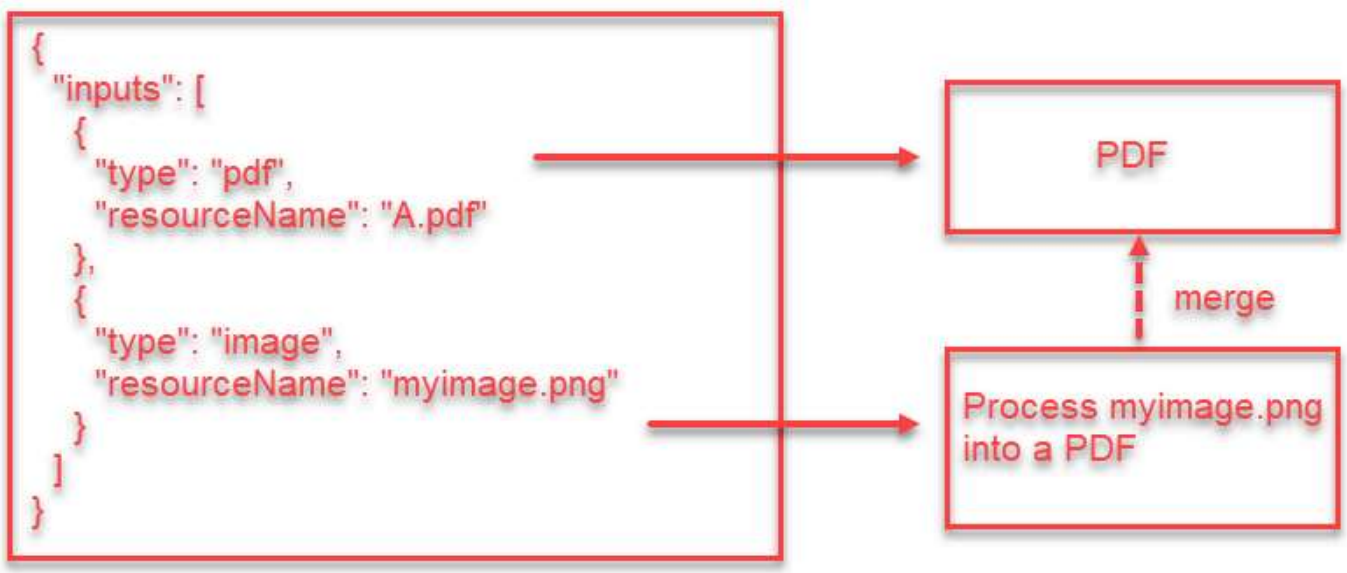

Figure 2. Merging two inputs to create a new PDF.

- Create a new PDF from `A.pdf`.
- Convert `myimage.png` into a PDF.
- Append the image PDF to the newly created already containing A.
- Return the resultant PDF.

> **INFO**
>
> The order that you specify the inputs in your instructions document is the order they are merged to create the final PDF.

Of course, you are not required to merge the outputs of one input type with another. You can also use the `pdf` endpoint as a standalone input. For instance, if you called the `pdf` endpoint and only included one `image` input type, then the `pdf` endpoint would convert the image to a PDF and then return that PDF.

> **TIP**

> Remember, two or more inputs in the same instructions document merges the output of those inputs into a combined PDF.

It is important you understand the `inputs` array and how they are combined to form a finished PDF. Undoubtedly you will use the `pdf` endpoint extensively, so it is important you understand how to use the `inputs` array into a finished PDF.

More on pdf Endpoint

This Users Guide topic only provided brief information on the `pdf` endpoint. But understanding the `pdf` endpoint requires understanding the endpoint's instructions JSON document. For more information on creating an instructions document, refer to one of the following topics. You can also chose to use one of the client libraries if you prefer to omit creating an instructions document. The client library abstracts the complexities of creating an instructions document and creates it for you behind the scenes.

- API Instructions Document JSON Schema
- JSON schema diagram
- Using a client library

Tags: pdf API Users Guide REST cURL JSON

*Last updated on **May 29, 2024***