

PDF Toolkit APIs Doc

PDF Toolkit APIs – All-in-One PDF Processing Platform

A powerful, developer-friendly API suite to process, secure, and manipulate PDF files programmatically. Built for SaaS products, automation workflows, compliance systems, and internal tools — without UI, SDKs, or complexity.

Base URLs (<BASE_DOMAIN>)

Complete toolkit API: pdf-handling-apis.p.rapidapi.com

Encryption API: pdf-encryption-api.p.rapidapi.com

Decryption API: pdf-decryption-api.p.rapidapi.com

Watermark API: pdf-watermark-api.p.rapidapi.com

Redaction API: pdf-redaction-hide-sensitive-info-api.p.rapidapi.com

Compression API: pdf-compression-api.p.rapidapi.com

Why Choose PDF Toolkit API?

- ⚡ Fast and reliable PDF processing
 - 🔒 Secure file handling (temporary storage only)
 - 🧩 Modular APIs — use only what you need
 - 🌐 Language-agnostic (works with any stack)
 - 📦 Scales from startups to enterprise
 - 💰 Simple subscription-based pricing
 - 🛠️ Available PDF Tools
-

PDF Encryption API (Base64)

Encrypt a PDF file by passing the base64 string of it using a password.

cURL Example

(Replace <> with their corresponding values)

```
curl -OJ --location
'https://pdf-handling-apis.p.rapidapi.com/api/encrypt-pdf-base64' \
--header 'Content-Type: application/json' \
--data '{
  "file_base64" : <file Base64 string>;,
  "password" : <pdf-password>;
}'
```

Success Response

Returns the encrypted PDF file directly.

Content-Type: application/pdf Content-Disposition: attachment;
filename="sample_encrypted.pdf"

Error Response

```
{ "status": "error", "status_code": 500, "message": "Internal Server Error",  
  "data": null }
```

PDF Encryption API

Encrypt a PDF file using a password.

cURL Example

(Replace <> with their corresponding values)

```
curl -OJ --location 'https://<BASE_DOMAIN>/api/encrypt-pdf-' \\  
--header 'X-RapidAPI-Host: <BASE_DOMAIN>' \\  
--header 'X-RapidAPI-Key: <API-key>' \\  
--form 'file=@<Absolute-Filepath>' \\  
--form 'password=<pdf-password>'
```

Success Response

Returns the encrypted PDF file directly.

Content-Type: application/pdf Content-Disposition: attachment;
filename="sample_encrypted.pdf"

Error Response

```
{ "status": "error", "status_code": 500, "message": "Internal Server Error",  
  "data": null }
```

PDF Decryption API

Remove password protection from an encrypted PDF.

cURL Example

(Replace <> with their corresponding values)

```
curl -OJ --location 'https://<BASE_DOMAIN>/api/decrypt-pdf' \
--header 'X-RapidAPI-Host: <BASE_DOMAIN>' \
--header 'X-RapidAPI-Key: <API-key>' \
--form 'file=@<Absolute-Filepath>' \
--form 'password=<pdf-password>' \
```

Error Response

```
{ "status": "error", "status_code": 500, "message": "Internal Server Error",
"data": null }
```

PDF Watermark API

Send the pdf file to be watermarked along with a file which contains a single page of the watermark as form-data through the API and the API returns the watermarked version of the file. The backend deletes the received files as soon as it returns the watermarked pdf file, hence keeping the user's data secure. Moreover, the pdf upload size is limited to 10MB.

cURL Example

(Replace <> with their corresponding values)

```
curl -OJ --location
'https://pdf-handling-apis.p.rapidapi.com/api/watermark-pdf' \
--header 'X-RapidAPI-Host: pdf-handling-apis.p.rapidapi.com' \
--header 'X-RapidAPI-Key: <API-key>' \
--form 'file=@<Absolute-Filepath>' \
--form 'watermark_file=@<Absolute-Filepath-having-Watermark>' \
```

Error Response

```
{ "status": "error", "status_code": 500, "message": "Internal Server Error",
"data": null }
```

PDF Redaction API

Send the pdf file to be redacted (hide sensitive data) along with an array of words that need to be redacted (like emails, passwords etc). The backend deletes the received files as soon as it returns the redacted pdf file and doesn't store data in any form, hence keeping the user's data secure. Moreover, the pdf upload size is limited to 10MB.

cURL Example

(Replace <> with their corresponding values)

```
curl -OJ --location 'https://pdf-handling-apis.p.rapidapi.com/api/redact-pdf'
--header 'X-RapidAPI-Host: pdf-handling-apis.p.rapidapi.com' \
--header 'X-RapidAPI-Key: <API-key>' \
--form 'file=@<Absolute-Filepath>' \
--form 'words_to_redact=["<abc@gmail.com>", "<1234567890>"]' \
```

Error Response

```
{ "status": "error", "status_code": 500, "message": "Internal Server Error",
"data": null }
```

PDF Compression API (Low)

Send the pdf file to be compressed and it returns you the low level compressed file. The backend deletes the received files as soon as it returns the compressed pdf file and doesn't store data in any form, hence keeping the user's data secure. Following is the sample Curl request and possible response structure. Moreover, the pdf upload size is limited to 10MB. Please note that this level is more effective with textual PDFs and not PDFs with images.

cURL Example

(Replace <> with their corresponding values)

```
curl -OJ --location
'https://pdf-handling-apis.p.rapidapi.com/api/compress-pdf/low' \
--header 'X-RapidAPI-Host: pdf-handling-apis.p.rapidapi.com' \
--header 'X-RapidAPI-Key: <API-key>' \
--form 'file=@<Absolute-Filepath>' \
```

Error Response

```
{ "status": "error", "status_code": 500, "message": "Internal Server Error",  
"data": null }
```

PDF Compression API (Medium)

Send the pdf file to be compressed and it returns you the medium level compressed file. The backend deletes the received files as soon as it returns the compressed pdf file and doesn't store data in any form, hence keeping the user's data secure. Following is the sample Curl request and possible response structure. Moreover, the pdf upload size is limited to 10MB. Please note that this level is more effective with PDFs with images.

cURL Example

(Replace <> with their corresponding values)

```
curl -OJ --location  
'https://pdf-handling-apis.p.rapidapi.com/api/compress-pdf/mid' \  
--header 'X-RapidAPI-Host: pdf-handling-apis.p.rapidapi.com' \  
--header 'X-RapidAPI-Key: <API-key>' \  
--form 'file=@<Absolute-Filepath>'" \
```

Error Response

```
{ "status": "error", "status_code": 500, "message": "Internal Server Error",  
"data": null }
```

PDF Compression API (High)

Send the pdf file to be compressed and it returns you the high level compressed file. The backend deletes the received files as soon as it returns the compressed pdf file and doesn't store data in any form, hence keeping the user's data secure. Following is the sample Curl request and possible response structure. Moreover, the pdf upload size is limited to 10MB. Please note that this level is more effective with PDFs with images.

cURL Example

(Replace <> with their corresponding values)

```
curl -OJ --location  
'https://pdf-handling-apis.p.rapidapi.com/api/compress-pdf/high' \  
--header 'X-RapidAPI-Host: pdf-handling-apis.p.rapidapi.com' \  

```

```
--header 'X-RapidAPI-Key: &lt;API-key&gt;' \
--form 'file=@&lt;Absolute-Filepath&gt;''
```

Error Response

```
{ "status": "error", "status_code": 500, "message": "Internal Server Error",
"data": null }
```

Security & Data Handling

- Files are processed in-memory or temporary storage
- Automatically deleted after processing
- No files or passwords are logged or shared

Testing & Integration

- Works seamlessly with Postman
- Copy-paste cURL on your terminal after updating the parameters
- APIs can simply be integrated in your projects.

Example Use Cases

- SaaS document processing
- Legal & compliance platforms
- Fintech document security
- HR & payroll systems
- Internal workflow automation

API Design Philosophy

- One tool per endpoint
- Predictable request/response formats
- File-in → file-out wherever possible
- JSON errors for easy debugging

Support & Feedback

For issues or feature requests: nidhi.business.infotech@gmail.com

NOTE: To access specific APIs, please reach out using the above-mentioned email ID.