





# PDF support

Process PDFs with Claude. Extract text, analyze charts, and understand visual content from your documents.

 Copy page



You can now ask Claude about any text, pictures, charts, and tables in PDFs you provide. Some sample use cases:

- Analyzing financial reports and understanding charts/tables
- Extracting key information from legal documents
- Translation assistance for documents
- Converting document information into structured formats

## Before you begin

### Check PDF requirements

Claude works with any standard PDF. However, you should ensure your request size meets these requirements when using PDF support:

Requirement	Limit
Maximum request size	32MB
Maximum pages per request	100
Format	Standard PDF (no passwords/encryption)

Please note that both limits are on the entire request payload, including any other content sent alongside PDFs.

Since PDF support relies on Claude's vision capabilities, it is subject to the same limitations and considerations as other vision tasks.





PDF support is currently supported via direct API access and Google Vertex AI. All active models support PDF processing.

PDF support is now available on Amazon Bedrock with the following considerations:

## Amazon Bedrock PDF Support

When using PDF support through Amazon Bedrock's Converse API, there are two distinct document processing modes:

**Important:** To access Claude's full visual PDF understanding capabilities in the Converse API, you must enable citations. Without citations enabled, the API falls back to basic text extraction only. Learn more about [working with citations](#).

## Document Processing Modes


1. **Converse Document Chat** (Original mode - Text extraction only)
  - Provides basic text extraction from PDFs
  - Cannot analyze images, charts, or visual layouts within PDFs
  - Uses approximately 1,000 tokens for a 3-page PDF
  - Automatically used when citations are not enabled
2. **Claude PDF Chat** (New mode - Full visual understanding)
  - Provides complete visual analysis of PDFs
  - Can understand and analyze charts, graphs, images, and visual layouts
  - Processes each page as both text and image for comprehensive understanding
  - Uses approximately 7,000 tokens for a 3-page PDF
  - **Requires citations to be enabled** in the Converse API

## Key Limitations

- **Converse API:** Visual PDF analysis requires citations to be enabled. There is currently no option to use visual analysis without citations (unlike the InvokeModel API).
- **InvokeModel API:** Provides full control over PDF processing without forced citations.



If customers report that Claude isn't seeing images or charts in their PDFs when using the Converse API, they likely need to enable the citations flag. Without it, Converse falls back to basic text extraction only.

 This is a known constraint with the Converse API that we're working to address. For applications that require visual PDF analysis without citations, consider using the `InvokeModel` API instead.

 For non-PDF files like `.csv`, `.xlsx`, `.docx`, `.md`, or `.txt` files, see [Working with other file formats](#).

## Process PDFs with Claude

### Send your first PDF request

Let's start with a simple example using the Messages API. You can provide PDFs to Claude in three ways:

1. As a URL reference to a PDF hosted online
2. As a base64-encoded PDF in `document` content blocks
3. By a `file_id` from the [Files API](#)

### Option 1: URL-based PDF document

The simplest approach is to reference a PDF directly from a URL:

Shell ▾





```
-H "x-api-key: $ANTHROPIC_API_KEY" \  
-H "anthropic-version: 2023-06-01" \  
-d '{  
  "model": "claude-sonnet-4-5",  
  "max_tokens": 1024,  
  "messages": [{  
    "role": "user",  
    "content": [{  
      "type": "document",  
      "source": {  
        "type": "url",  
        "url": "https://assets.anthropic.com/m/1cd9d098ac3e6467/origi  
      }  
    },  
    {  
      "type": "text",  
      "text": "What are the key findings in this document?"  
    }  
  ]  
}]  
'
```

## Option 2: Base64-encoded PDF document

If you need to send PDFs from your local system or when a URL isn't available:

Shell ▾





```
# Method 2: Encode a local PDF file
# base64 document.pdf | tr -d '\n' > pdf_base64.txt

# Create a JSON request file using the pdf_base64.txt content
jq -n --rawfile PDF_BASE64 pdf_base64.txt '{
  "model": "claude-sonnet-4-5",
  "max_tokens": 1024,
  "messages": [{
    "role": "user",
    "content": [{
      "type": "document",
      "source": {
        "type": "base64",
        "media_type": "application/pdf",
        "data": $PDF_BASE64
      }
    }
  ],
  {
    "type": "text",
    "text": "What are the key findings in this document?"
  }
]}
}' > request.json

# Send the API request using the JSON file
curl https://api.anthropic.com/v1/messages \
  -H "content-type: application/json" \
  -H "x-api-key: $ANTHROPIC_API_KEY" \
  -H "anthropic-version: 2023-06-01" \
  -d @request.json
```

### Option 3: Files API

For PDFs you'll use repeatedly, or when you want to avoid encoding overhead, use the Files API:

Shell ▾





```
-H "x-api-key: $ANTHROPIC_API_KEY" \  
-H "anthropic-version: 2023-06-01" \  
-H "anthropic-beta: files-api-2025-04-14" \  
-F "file=@document.pdf"  
  
# Then use the returned file_id in your message  
curl https://api.anthropic.com/v1/messages \  
-H "content-type: application/json" \  
-H "x-api-key: $ANTHROPIC_API_KEY" \  
-H "anthropic-version: 2023-06-01" \  
-H "anthropic-beta: files-api-2025-04-14" \  
-d '{  
  "model": "claude-sonnet-4-5",  
  "max_tokens": 1024,  
  "messages": [{  
    "role": "user",  
    "content": [{  
      "type": "document",  
      "source": {  
        "type": "file",  
        "file_id": "file_abc123"  
      }  
    }],  
    {  
      "type": "text",  
      "text": "What are the key findings in this document?"  
    }  
  ]  
}'
```

## How PDF support works

When you send a PDF to Claude, the following steps occur:

### 1 The system extracts the contents of the document.

- The system converts each page of the document into an image.
- The text from each page is extracted and provided alongside each page's image.

### 2 Claude analyzes both the text and images to better understand the document.

- Documents are provided as a combination of text and images for analysis.

### 3 Claude responds, referencing the PDF's contents if relevant.

Claude can reference both textual and visual content when it responds. You can further improve performance by integrating PDF support with:

- **Prompt caching:** To improve performance for repeated analysis.
- **Batch processing:** For high-volume document processing.
- **Tool use:** To extract specific information from documents for use as tool inputs.

## Estimate your costs

The token count of a PDF file depends on the total text extracted from the document as well as the number of pages:

- **Text token costs:** Each page typically uses 1,500-3,000 tokens per page depending on content density. Standard API pricing applies with no additional PDF fees.
- **Image token costs:** Since each page is converted into an image, the same image-based cost calculations are applied.

You can use token counting to estimate costs for your specific PDFs.

## Optimize PDF processing

### Improve performance

Follow these best practices for optimal results:

- Place PDFs before text in your requests
- Use standard fonts
- Ensure text is clear and legible
- Rotate pages to proper upright orientation
- Use logical page numbers (from PDF viewer) in prompts
- Split large PDFs into chunks when needed
- Enable prompt caching for repeated analysis



For high-volume processing, consider these approaches:

## Use prompt caching

Cache PDFs to improve performance on repeated queries:

Shell ▾



```
# Create a JSON request file using the pdf_base64.txt content
jq -n --rawfile PDF_BASE64 pdf_base64.txt '{
  "model": "claude-sonnet-4-5",
  "max_tokens": 1024,
  "messages": [{
    "role": "user",
    "content": [{
      "type": "document",
      "source": {
        "type": "base64",
        "media_type": "application/pdf",
        "data": $PDF_BASE64
      },
      "cache_control": {
        "type": "ephemeral"
      }
    },
    {
      "type": "text",
      "text": "Which model has the highest human preference win rates ac
    }
  ]
}' > request.json

# Then make the API call using the JSON file
curl https://api.anthropic.com/v1/messages \
-H "content-type: application/json" \
-H "x-api-key: $ANTHROPIC_API_KEY" \
-H "anthropic-version: 2023-06-01" \
-d @request.json
```

## Process document batches

Use the Message Batches API for high-volume workflows:









```

"requests": [
  {
    "custom_id": "my-first-request",
    "params": {
      "model": "claude-sonnet-4-5",
      "max_tokens": 1024,
      "messages": [
        {
          "role": "user",
          "content": [
            {
              "type": "document",
              "source": {
                "type": "base64",
                "media_type": "application/pdf",
                "data": $PDF_BASE64
              }
            },
            {
              "type": "text",
              "text": "Which model has the highest human preference?"
            }
          ]
        }
      ]
    }
  },
  {
    "custom_id": "my-second-request",
    "params": {
      "model": "claude-sonnet-4-5",
      "max_tokens": 1024,
      "messages": [
        {
          "role": "user",
          "content": [
            {
              "type": "document",
              "source": {
                "type": "base64",
                "media_type": "application/pdf",
                "data": $PDF_BASE64
              }
            },
            {
              "type": "text",
              "text": "Extract 5 key insights from this document"
            }
          ]
        }
      ]
    }
  }
]

```



# Next steps

request.json

## Try PDF examples

# Then make the API call using the JSON file  
Explore practical examples of PDF processing in our cookbook recipe.  
<https://api.anthropic.com/docs/cookbook/recipes>

```
-H "content-type: application/json" \  
-H "x-api-key: $ANTHROPIC_API_KEY" \  
-H "anthropic-version: 2023-06-01" \  
</>d @request.json
```

## View API reference

See complete API documentation for PDF support.



- Solutions

AI agents

Code modernization

Coding

Customer support

Education

Financial services

Government

Life sciences

Partners

Amazon Bedrock

Google Cloud's Vertex AI
- Learn

Blog

Catalog

Courses

Use cases

Connectors

Customer stories

Engineering at Anthropic

Events

Powered by Claude

Service partners

Startups program

Company



---

Economic Futures

Research

News

Responsible Scaling Policy

Security and compliance

Transparency

Help and security

Availability

Status

Support

Discord

Terms and policies

Privacy policy

Responsible disclosure policy

Terms of service: Commercial

Terms of service: Consumer

Usage policy