Documents MCP - Documentation

Documents MCP

An MCP (Model Context Protocol) server that provides AI agents with tools to create and read PDF, DOCX, and PPTX documents.

Version 1.1.1 | MIT License

Features

- Create PDF - Generate PDF documents with text, headings, tables, images, and page numbers
- Create DOCX - Generate Word documents with headings, paragraphs, lists, tables, and images
- Create PPTX - Generate PowerPoint presentations with slides, text, shapes, tables, and charts
- Read PDF - Extract text content and metadata from PDF files with optional AI analysis
- Read DOCX - Extract text and HTML content from Word documents with optional AI analysis
- Read PPTX - Extract text content from PowerPoint presentations with optional AI analysis

Installation

Install via npm:

```
npm install documents-mcp
```

Or use directly with npx:

```
npx documents-mcp
```

Claude Desktop Configuration

Add the following to your Claude Desktop configuration file:

macOS/Linux: ~/.config/claude/claude_desktop_config.json

Windows: %APPDATA%\Claude\claude_desktop_config.json

Tool Reference

create-pdf

Create a PDF document with structured content including text, headings, tables, images, and page breaks.

Parameter	Type	Required	Description
title	string	Yes	Document title
author	string	No	Document author
content	array	Yes	Array of content items
outputPath	string	No	File path to save the PDF
pageSize	string	No	A4, Letter, or Legal

create-docx

Create a Word document with rich formatting including headings, paragraphs, lists, tables, and images.

Parameter	Type	Required	Description
title	string	Yes	Document title
author	string	No	Document author
content	array	Yes	Array of content items
outputPath	string	No	File path to save the DOCX

create-pptx

Create a PowerPoint presentation with slides containing text, shapes, tables, charts, and images.

Parameter	Type	Required	Description
title	string	Yes	Presentation title
author	string	No	Presentation author
slides	array	Yes	Array of slide objects
outputPath	string	No	File path to save the PPTX

read-pdf

Extract text content and metadata from a PDF file. Supports optional AI analysis.

Parameter	Type	Required	Description
filePath	string	No*	Path to the PDF file
base64Content	string	No*	Base64-encoded PDF content
prompt	string	No	Instruction for AI analysis

* Either filePath or base64Content is required

read-docx

Extract text content from a Word document. Supports text, HTML, or both output formats.

Parameter	Type	Required	Description
filePath	string	No*	Path to the DOCX file
base64Content	string	No*	Base64-encoded DOCX content
outputFormat	string	No	text, html, or both
prompt	string	No	Instruction for AI analysis

read-pptx

Extract text content from a PowerPoint presentation.

Parameter	Type	Required	Description
filePath	string	No*	Path to the PPTX file
base64Content	string	No*	Base64-encoded PPTX content
prompt	string	No	Instruction for AI analysis

Environment Variables

All environment variables are optional. Configure only the providers you need.

AI Provider API Keys

Variable	Description
OPENAI_API_KEY	OpenAI API key
ANTHROPIC_API_KEY	Anthropic API key
GOOGLE_API_KEY	Google AI API key
GEMINI_API_KEY	Gemini API key (alias for GOOG
PERPLEXITY_API_KEY	Perplexity API key
XAI_API_KEY	xAI (Grok) API key
GROQ_API_KEY	Groq API key

Local Model Endpoints

Variable	Default	Description
OLLAMA_BASE_URL	http://localhost:11434	Ollama server URL
LMSTUDIO_BASE_URL	http://localhost:1234	LM Studio server URL
VLLM_BASE_URL	http://localhost:8000	vLLM server URL

Output Configuration

Variable	Default	Description
OUTPUT_DIR	Current directory	Default directory for saved do

Client SDK

Use the client SDK to connect to the documents-mcp server programmatically.

Available Client Methods

Method	Description
connect()	Connect to the MCP server
disconnect()	Disconnect from the server
listTools()	List available tools
callTool(name, args)	Call any tool by name
createPdf(options)	Create a PDF document
createDocx(options)	Create a Word document
createPptx(options)	Create a PowerPoint presentati
readPdf(options)	Extract text from a PDF
readDocx(options)	Extract text from a Word docum
readPptx(options)	Extract text from a PowerPoint

HTTP Server

Start the HTTP server for web-based MCP clients:

```
npm run start:http
```

```
npx documents-mcp-http
```

The HTTP server exposes:

- GET /sse - SSE endpoint for MCP clients
- POST /messages?sessionId=<id> - Message endpoint for SSE sessions
- GET /health - Health check endpoint

Links

GitHub: https://github.com/HarjjotSinghh/documents-mcp

npm: https://www.npmjs.com/package/documents-mcp