

MkDocs Mermaid to Image Plugin

MkDocs Mermaid to Image

None

None

目次

1. MkDocs Mermaid to Image Plugin	5
1.1 ✨ 特徴	5
1.2 開発ガイド	5
1.2.1 前提条件	5
1.2.2 セットアップ	5
1.2.3 ローカルでのビルドとインストール	5
1.2.4 よく使うコマンド	5
1.2.5 開発ワークフロー	6
2. アーキテクチャ設計	7
2.1 概要	7
2.2 プロジェクト構造	7
2.3 ファイル依存関係図	7
2.4 クラス図	8
2.5 プラグイン処理フロー	9
2.5.1 1. プラグイン初期化フロー (on_config)	9
2.5.2 2. ファイル処理フロー (on_files)	0
2.5.3 3. ページ処理フロー (on_page_markdown)	0
2.5.4 4. 画像生成フロー (MermaidImageGenerator.generate)	0
2.6 環境別処理戦略	0
2.6.1 モード判定	0
2.6.2 プラグイン有効化制御	0
2.6.3 ログレベル制御	0
2.7 プラグイン設定管理	0
2.7.1 設定スキーマ	0
2.7.2 設定検証	0
2.8 ファイル管理戦略	0
2.8.1 生成画像のFiles登録	0
2.8.2 画像の配置戦略	0
2.9 エラーハンドリング戦略	0
2.9.1 例外階層	0
2.9.2 エラー発生時の処理	0
2.9.3 ログ出力戦略	0

3. 使用方法	0
3.1 プラグイン設定	0
3.1.1 主要設定項目	0
3.1.2 高度な設定	0
3.2 PDF出力との組み合わせ	0
3.2.1 環境変数による条件付き画像化	0
3.2.2 使用方法	0
3.2.3 環境変数の判定仕様	0
3.2.4 実用的なワークフロー例	0
3.3 Mermaidダイアグラムの記述	0
3.3.1 基本的な記述方法	0
3.4 ビルドと実行	0
3.4.1 通常のビルド	0
3.4.2 ログレベル指定	0
3.5 生成される成果物	0
3.5.1 生成画像の確認	0
3.6 パフォーマンス最適化	0
3.6.1 キャッシュ活用	0
3.6.2 並列処理対応	0
4. Appendix	0
4.1 Claude Code向けPythonテンプレート	0
4.1.1 🚀 クイックスタート	0
4.2 Claude Code & Gemini Cli チートシート	0
4.2.1 📄 Claude Code チートシート	0
4.2.2 ✨ 主な特徴	0
4.2.3 📁 プロジェクト構造	0
4.2.4 📖 ドキュメント階層	0
4.2.5 ✅ 新規プロジェクト設定チェックリスト	0
4.2.6 🛠️ カスタマイズ	0
4.2.7 🔗 外部リソース	0
4.2.8 📄 ライセンス	0
4.3 mermaid-cli 完全攻略: PNG生成パラメータの網羅的解説と動的な画像サイズ自動制御戦略	0
4.3.1 序論: mermaid-cliの本質を理解する – 単なる画像コンバータではない	0
4.3.2 第1部: mermaid-cli パラメータ大全 – コマンドラインからの制御	0
4.3.3 第2部: 設定ファイルによる階層的カスタマイズ	0
4.3.4 第3部: 文字が潰れないための画像サイズ自動制御戦略	0
4.3.5 第4部: 実践的用例とトラブルシューティング	0

4.3.6 結論: mermaid-cliをマスターするための三つの鍵	0
4.4 MkDocsプラグインフックメソッド完全ガイド	0
4.4.1 フックメソッドの分類体系	0
4.4.2 完全なフックメソッド一覧	0
4.4.3 フックメソッドのライフサイクルと呼び出し順序	0
4.4.4 on_serveフックの詳細分析	0
4.4.5 プラグイン開発のベストプラクティス	0
4.4.6 実用的な実装パターン	0
4.4.7 結論	0
4.5 パッケージリリース戦略	0
4.5.1 基本概念	0
4.5.2 リリースフローの概念	0
4.5.3 setuptools_scmによるバージョン管理	0
4.5.4 バージョニング戦略	0
4.5.5 運用の要点	0
4.5.6 戦略の利点	0

1. MkDocs Mermaid to Image Plugin

MkDocs環境でMermaidダイアグラムを静的画像として事前レンダリングし、PDF出力に対応させるプラグインです。

- [Sample PDF](#)

1.1 ✨ 特徴

- MermaidダイアグラムをPNG/SVG画像として事前レンダリング
- PDF出力対応
- 標準テーマサポート
- キャッシュ機能による高速ビルド

1.2 開発ガイド

このセクションは `mkdocs-mermaid-to-image` プラグインの開発に参加するための総合的なガイドです。

1.2.1 前提条件

開発を始める前に、以下のツールがシステムにインストールされていることを確認してください。

- **Python**: 3.9 以上
- **Node.js**: 16.0.0 以上
- **npm**: 7.0.0 以上
- **uv**: 高速なPythonパッケージインストーラー
- **Make**: ビルド自動化ツール

1.2.2 セットアップ

開発環境のセットアップは、リポジトリのルートで以下のコマンドを実行するだけです。

```
make setup
```

このコマンドは、Pythonの依存関係とNode.jsの依存関係をインストールし、pre-commitフックを設定します。

1.2.3 ローカルでのビルドとインストール

開発中にプラグインの動作を確認するには、ローカルでビルドしてインストールする必要があります。ソースコードの変更を即座に反映させるために、編集可能モード (`-e`) でインストールすることを推奨します。

```
make install-dev
```

これにより、`mkdocs.yml` で `mermaid-to-image` プラグインを指定したMkDocsプロジェクトで、開発中のプラグインを直接テストできます。

1.2.4 よく使うコマンド

`Makefile` には、開発を効率化するためのコマンドが多数定義されています。

開発コマンド

- `make install-dev`: 開発用に編集可能モードでパッケージをインストールします。
- `make test`: すべてのテストを実行します。
- `make test-cov`: カバレッジレポート付きでテストを実行します。

品質チェック

- `make check`: 品質チェック（pre-commitフックと同等の内容）を実行します。
- `make check-security`: セキュリティチェック（bandit + pip-audit）を実行します。
- `make check-all`: 完全チェック（品質 + セキュリティ）を実行します。

MkDocsコマンド

- `uv run mkdocs serve`: 開発用のローカルサーバーを起動します。
- `uv run mkdocs build`: ドキュメントサイトをビルドします。
- `ENABLE_PDF_EXPORT=1 uv run mkdocs build`: PDF生成を有効にしてドキュメントサイトをビルドします。

Mermaid CLI

- `mmdc --version`: ローカルインストールされたMermaid CLIのバージョン確認。
- `npx mmdc --version`: npx経由でのMermaid CLIのバージョン確認（フォールバック）。

利用可能なすべてのコマンドについては、`make help` を実行して確認してください。

1.2.5 開発ワークフロー

日常的な開発

1. コード変更後:

```
make check # 品質チェック（自動修正含む）
```

2. テスト実行:

```
make test # 全テスト実行
```

コミット前

```
make check # pre-commitフックと同等のチェック
```

プルリクエスト前

```
make check-all # 品質 + セキュリティの完全チェック
```

PDF生成テスト

```
ENABLE_PDF_EXPORT=1 uv run mkdocs build
```

2. アーキテクチャ設計

2.1 概要

MkDocs Mermaid to Image Pluginは、MkDocsプロジェクト内のMermaid図をビルド時に静的画像（PNG/SVG）に変換するプラグインです。Mermaid CLIを利用してMarkdownファイル内のMermaidコードブロックを画像化し、Markdownの内容を画像参照タグに置き換えます。これにより、PDF出力やオフライン環境での閲覧に対応します。

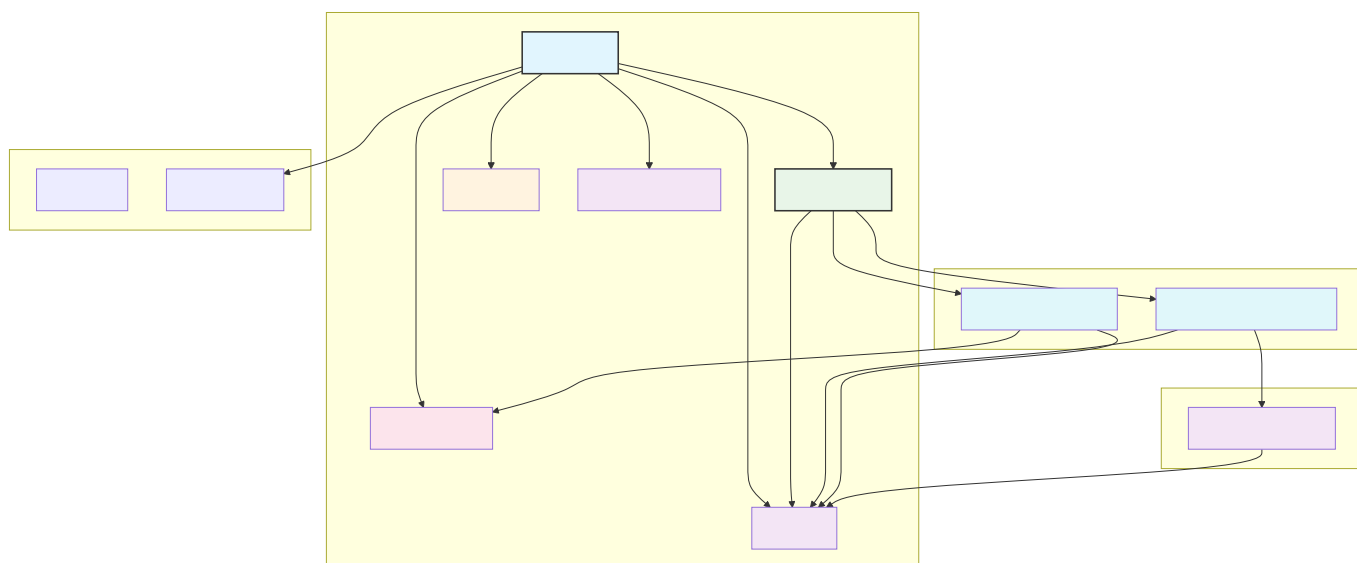
2.2 プロジェクト構造

```

mkdocs-mermaid-to-image/
├── src/
│   └── mkdocs_mermaid_to_image/
│       ├── __init__.py           # パッケージ初期化・バージョン情報
│       ├── plugin.py             # MkDocsプラグインメインクラス (MermaidToImagePlugin)
│       ├── processor.py          # ページ処理の統括 (MermaidProcessor)
│       ├── markdown_processor.py # Markdown解析 (MarkdownProcessor)
│       ├── image_generator.py    # 画像生成 (MermaidImageGenerator)
│       ├── mermaid_block.py     # Mermaidブロック表現 (MermaidBlock)
│       ├── config.py             # 設定スキーマ (MermaidPluginConfig, ConfigManager)
│       ├── types.py              # 型定義 (TypedDictなど)
│       ├── exceptions.py        # カスタム例外クラス
│       ├── logging_config.py    # ロギング設定・構造化フォーマッタ
│       └── utils.py              # ユティリティ関数

```

2.3 ファイル依存関係図



2.5 プラグイン処理フロー

2.5.1 1. プラグイン初期化フロー (on_config)

