

サンプルドキュメント

Table of Contents

1. はじめに	1
1.1. 目的	1
2. システム概要	1
2.1. アーキテクチャ図	1
2.2. プロセスフロー	2
3. 技術詳細	3
3.1. 使用技術	3
3.2. 日本語対応	3
4. サンプルコード	4
4.1. Python例	4
4.2. Shell例	4
5. まとめ	4

1. はじめに

このドキュメントは、AsciiDocからPDFへの変換システムのサンプルです。日本語フォントとKrokiを使用した図表生成機能をテストします。

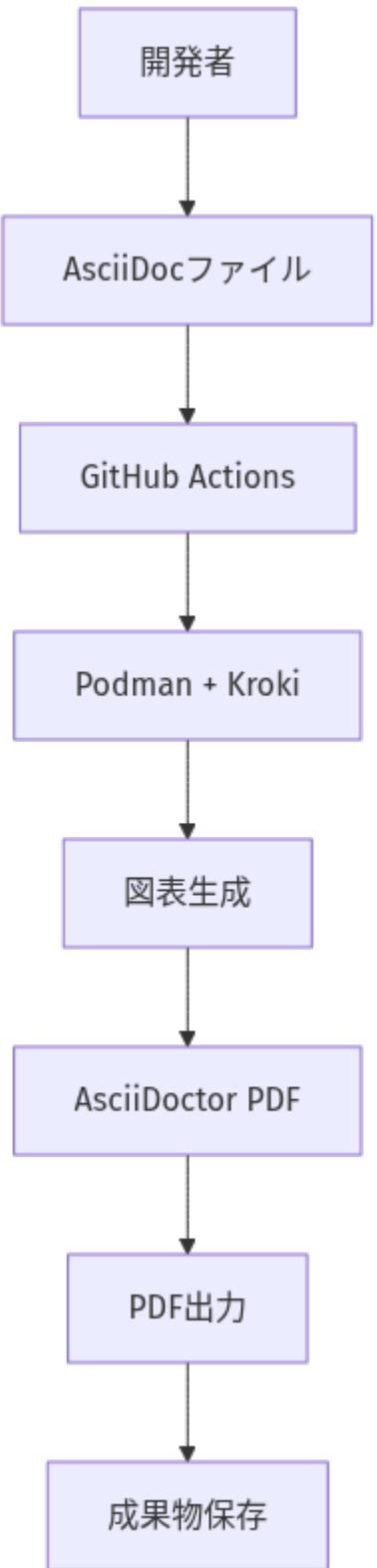
1.1. 目的

- AsciiDocの基本的な機能の確認
- 日本語フォントの表示テスト
- Mermaid図表の生成確認
- PlantUML図表の生成確認

2. システム概要

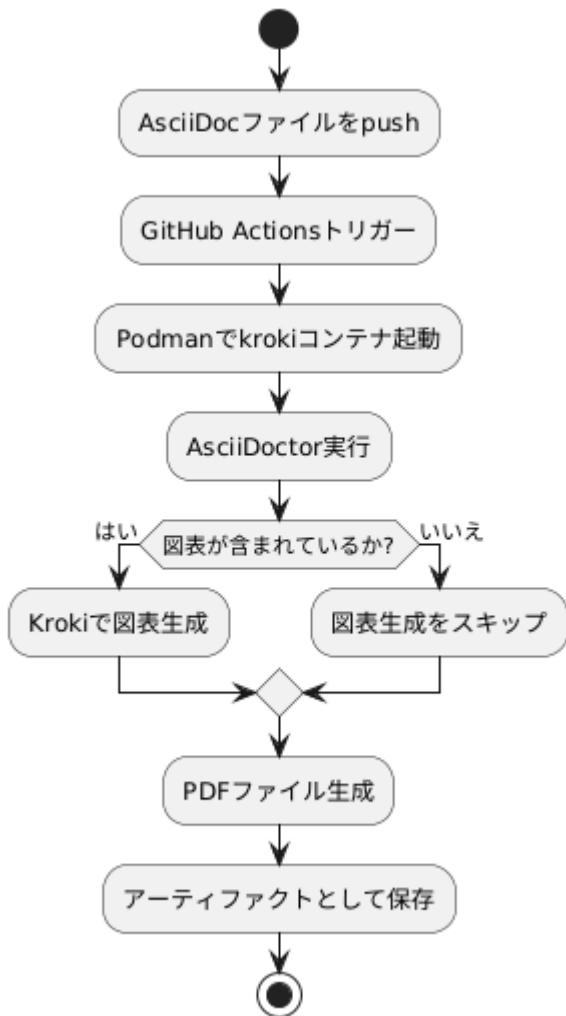
2.1. アーキテクチャ図

以下のMermaid図は、システムの全体的なアーキテクチャを示しています。



2.2. プロセスフロー

次のPlantUML図は、変換プロセスの詳細なフローを表示します。



3. 技術詳細

3.1. 使用技術

分類	技術	説明
文書作成	AsciiDoc	軽量マークアップ言語
図表生成	Kroki	各種図表フォーマットのレンダリング
コンテナ	Podman	Dockerの代替コンテナランタイム
CI/CD	GitHub Actions	自動化ワークフロー
PDF生成	AsciiDoctor PDF	AsciiDocからPDFへの変換

3.2. 日本語対応

このシステムでは以下の日本語対応を行っています：

1. フォント設定

- Noto Sans CJK (ゴシック体)
- Noto Serif CJK (明朝体)

2. 文字エンコーディング

- UTF-8での処理

3. レイアウト

- A4サイズ対応
- 適切な行間設定

4. サンプルコード

4.1. Python例

```
def hello_world():
    """日本語を含むサンプル関数"""
    print("こんにちは、世界！")
    return "Hello, World!"

if __name__ == "__main__":
    message = hello_world()
    print(f"メッセージ: {message}")
```

4.2. Shell例

```
#!/bin/bash
# PDFコンバーターの実行例

echo "変換を開始します..."
asciidoctor-pdf \
--theme asciidoctor-pdf-theme.yml \
--out-dir output \
docs/sample.adoc
echo "変換が完了しました。"
```

hogeホゲ

hogeホゲ

5. まとめ

このシステムにより、以下の機能が実現されています：

- 日本語を含むAsciiDoc文書のPDF変換
- MermaidやPlantUMLを使用した図表の自動生成
- GitHub Actionsによる自動化されたビルドプロセス
- Podmanを使用したコンテナベースの環境構築

今後の拡張として、以下のような機能追加も検討できます：

- 複数の文書ファイルの一括変換
- カスタムテーマの追加
- 他の図表フォーマットのサポート
- メール通知機能

WARNING

このシステムを使用する際は、Krokiサーバーが正常に起動していることを確認してください。

NOTE

PDF生成時に日本語フォントが正しく適用されない場合は、フォントファイルのパスを確認してください。