

# 劇評ツール API使用ガイド

---

このドキュメントでは、劇評ツールでのAI API（AnthropicとOpenAI）の使用方法について詳しく説明します。

## 目次

---

- [1. APIの概要](#)
- [2. APIキーの設定方法](#)
- [3. Anthropic APIの使用方法](#)
- [4. OpenAI APIの使用方法](#)
- [5. APIプロバイダーの切り替え](#)
- [6. APIパラメータの調整](#)
- [7. エラー処理とリトライ](#)
- [8. APIコスト管理](#)

## APIの概要

---

劇評ツールでは、以下の処理でAI APIを使用しています：

- 意見抽出:** コメントから意見を抽出する
- クラスターラベリング:** クラスターにラベルを付ける
- クラスター要約:** クラスターの要約を生成する
- 全体概要生成:** 全体の概要を生成する

これらの処理は、AnthropicまたはOpenAIのAPIを使用して実行されます。

# APIキーの設定方法

## APIキーの取得方法

### Anthropic APIキー

1. [Anthropicのウェブサイト](#)にアクセスし、アカウントを作成します。
2. APIキーを発行します。

### OpenAI APIキー

1. [OpenAIのウェブサイト](#)にアクセスし、アカウントを作成します。
2. APIキーを発行します。

## APIキーの設定方法

APIキーは、以下の2つの方法で設定できます：

### 1. 環境変数を使用する方法（推奨）

環境変数を使用すると、APIキーをコードやconfigファイルに直接記述する必要がなくなり、セキュリティが向上します。

#### Windowsの場合：

コマンドプロンプトで：

```
1 | set ANTHROPIC_API_KEY=your_api_key_here
```

PowerShellで：

```
1 | $env:ANTHROPIC_API_KEY="your_api_key_here"
```

#### macOS/Linuxの場合：

```
1 | export ANTHROPIC_API_KEY=your_api_key_here
```

同様に、OpenAI APIキーを設定する場合は、`OPENAI_API_KEY` 環境変数を使用します。

## 2. config.jsonファイルに直接記述する方法

config.jsonファイルの`ai` セクションに、APIキーを直接記述することもできます：

```
1  "ai": {  
2    "provider": "anthropic",  
3    "model": "claude-3-7-sonnet-20250219",  
4    "temperature": 0.2,  
5    "sample_size": 5,  
6    "api_key": "your_api_key_here",  
7    "openai_api_key": "your_openai_api_key_here",  
8    "anthropic_api_key": "your_anthropic_api_key_here"  
9  }
```

注意：

- `api_key` は一般的なAPIキーとして使用されます。
- `provider` が `anthropic` の場合、`anthropic_api_key` または `api_key` が使用されます。
- `provider` が `openai` の場合、`openai_api_key` または `api_key` が使用されます。
- 環境変数が設定されている場合は、config.jsonの設定よりも優先されます。

## Anthropic APIの使用方法

### 設定

config.jsonファイルで、以下のように設定します：

```
1 "ai": {  
2   "provider": "anthropic",  
3   "model": "claude-3-7-sonnet-20250219",  
4   "temperature": 0.2,  
5   "sample_size": 5,  
6   "anthropic_api_key": "your_anthropic_api_key_here"  
7 }
```

## サポートされているモデル

Anthropicでは、以下のモデルがサポートされています：

- claude-3-7-sonnet-20250219
- claude-3-opus-20240229
- claude-3-sonnet-20240229
- claude-3-haiku-20240307
- claude-2.1
- claude-2.0
- claude-instant-1.2

最新のモデル情報については、[Anthropicの公式ドキュメント](#)を参照してください。

## APIパラメータ

- model: 使用するモデル
- temperature: 生成の多様性 (0.0～1.0)
- max\_tokens: 生成するトークンの最大数

# OpenAI APIの使用方法

---

## 設定

config.jsonファイルで、以下のように設定します：

```
1  "ai": {  
2    "provider": "openai",  
3    "model": "gpt-4",  
4    "temperature": 0.2,  
5    "sample_size": 5,  
6    "openai_api_key": "your_openai_api_key_here"  
7  }
```

## サポートされているモデル

OpenAIでは、以下のモデルがサポートされています：

- gpt-4
- gpt-4-turbo
- gpt-3.5-turbo

最新のモデル情報については、[OpenAIの公式ドキュメント](#)を参照してください。

## APIパラメータ

- model: 使用するモデル
- temperature: 生成の多様性 (0.0～1.0)
- max\_tokens: 生成するトークンの最大数

## APIプロバイダーの切り替え

---

APIプロバイダーを切り替えるには、以下の方法があります：

## 1. config.jsonファイルを編集する

config.jsonファイルの`ai.provider`を変更します：

```
1 "ai": {  
2   "provider": "anthropic", // または "openai"  
3   ...  
4 }
```

## 2. コマンドラインオプションを使用する

コマンドラインオプション`--api`を使用して、APIプロバイダーを指定します：

```
1 python process_all.py --api anthropic
```

または

```
1 python process_all.py --api openai
```

## APIパラメータの調整

### 温度（temperature）

温度パラメータは、生成の多様性を制御します。値が低いほど決定的な応答になり、値が高いほど多様な応答になります。

config.jsonファイルで設定：

```
1 "ai": {  
2   "temperature": 0.2, // 0.0～1.0の値  
3   ...  
4 }
```

## サンプルサイズ

サンプルサイズは、一度に処理するコメントの数を指定します。大きな値を設定すると処理が速くなりますが、APIの制限に達する可能性があります。

config.jsonファイルで設定：

```
1 "ai": {  
2   "sample_size": 5, // 一度に処理するコメントの数  
3   ...  
4 }
```

## 最大トークン数

生成するテキストの最大トークン数を指定します。

config.jsonファイルで設定：

```
1 "ai": {  
2   "max_tokens": 4000, // 生成するトークンの最大数  
3   ...  
4 }
```

## エラー処理とリトライ

APIリクエストが失敗した場合、自動的にリトライが行われます。リトライの回数と遅延時間は、以下のパラメータで制御できます：

```
1 {  
2   "retry_attempts": 3, // リトライの最大回数  
3   "retry_delay": 5    // リトライ間の遅延時間（秒）  
4 }
```

これらのパラメータは、config.jsonのトップレベルに追加できます。

## APIコスト管理

AIモデルの使用にはコストがかかります。コストを管理するためのヒント：

1. **小さなデータセットでテスト:** 本番環境で大量のデータを処理する前に、小さなデータセットでテストしてください。
2. **低コストのモデルを使用:** 高性能なモデル（例：GPT-4、Claude 3 Opus）は高コストです。必要に応じて、低コストのモデル（例：GPT-3.5-Turbo、Claude 3 Haiku）を使用してください。
3. **キャッシュを活用:** 同じクエリに対する応答をキャッシュすることで、API コールを減らすことができます。
4. **バッチ処理を最適化:** `sample_size` パラメータを調整して、バッチ処理を最適化してください。

## コスト見積もり

以下は、一般的なデータセットサイズに基づくコスト見積もりの例です：

- 100件のコメント、Claude 3 Sonnet使用: 約\$0.5～\$2.0
- 1,000件のコメント、Claude 3 Sonnet使用: 約\$5.0～\$20.0

注意: これらは概算であり、実際のコストは使用するモデル、生成するテキストの量、APIの価格設定によって異なります。