

暗号化API

RawIDを暗号化してSurveyIDに変換し、また復号化するためのシンプルなAPIサービスです。

機能

- 暗号化: RawIDを安全に暗号化してSurveyIDを生成
- 復号化: SurveyIDをRawIDに復号化
- ヘルスチェック: サービスの稼働状態を確認
- キー生成: 開発用の暗号化キーを生成

Renderへのデプロイ手順

1. GitHubリポジトリの準備

1. GitHubで新しいリポジトリを作成
2. 以下のファイルをリポジトリにアップロード：
 - `app.py`
 - `requirements.txt`
 - `render.yaml`
 - `runtime.txt`
 - `README.md`

2. 暗号化キーの生成

ローカルで以下のコマンドを実行して暗号化キーを生成：

```
bash

python3 -c "from cryptography.fernet import Fernet; print(Fernet.generate_key().decode())"
```

生成されたキーを安全に保存してください。

3. Renderでのデプロイ

1. [Render](#)にログイン
2. 「New +」 → 「Web Service」を選択
3. GitHubリポジトリを接続
4. 以下の設定を行う：

- **Name:** `encryption-api` (任意の名前)
- **Runtime:** Python
- **Build Command:** `pip install -r requirements.txt`
- **Start Command:** `gunicorn app:app`

5. 「Advanced」 セクションで環境変数を追加：

- **Key:** `FERNET_KEY`
- **Value:** 手順2で生成したキー

6. 「Create Web Service」をクリック

4. デプロイの確認

デプロイが完了したら、以下のURLでアクセス可能：

- `https://your-service-name.onrender.com/`

API エンドポイント

GET /

APIの基本情報を取得

GET /health

サービスの稼働状態を確認

POST /encrypt

RawIDを暗号化

リクエスト：

```
json

{
  "RawID": "12345"
}
```

レスポンス：

```
json

{
  "SurveyID": "gAAAAABh...",
  "status": "success"
}
```

POST /decrypt

SurveyIDを復号化

リクエスト:

```
json

{
  "SurveyID": "gAAAAABh..."
}
```

レスポンス:

```
json

{
  "RawID": "12345",
  "status": "success"
}
```

GET /generate-key

新しい暗号化キーを生成（開発用）

テスト方法

curlを使用した例

```
bash

# 暗号化
curl -X POST https://your-service-name.onrender.com/encrypt \
-H "Content-Type: application/json" \
-d '{"RawID": "test123"}'

# 復号化
curl -X POST https://your-service-name.onrender.com/decrypt \
-H "Content-Type: application/json" \
-d '{"SurveyID": "暗号化された文字列"}'
```

Pythonを使用した例

```
python
```

```
import requests

# APIのURL
base_url = "https://your-service-name.onrender.com"

# 暗号化
encrypt_response = requests.post(
    f"{base_url}/encrypt",
    json={"RawID": "12345"}
)
print(encrypt_response.json())

# 復号化
survey_id = encrypt_response.json()["SurveyID"]
decrypt_response = requests.post(
    f"{base_url}/decrypt",
    json={"SurveyID": survey_id}
)
print(decrypt_response.json())
```

セキュリティに関する注意事項

1. **FERNET_KEY**は絶対に公開しないでください
2. HTTPSを使用してAPIと通信してください
3. 本番環境では適切なアクセス制限を設定してください
4. 定期的にキーをローテーションすることを推奨します

トラブルシューティング

"Cipher not initialized" エラー

- Renderの環境変数に**FERNET_KEY**が正しく設定されているか確認

"Invalid or corrupted SurveyID" エラー

- 暗号化に使用したキーと復号化に使用しているキーが同じか確認
- SurveyIDが正しくコピーされているか確認

ライセンス

MIT License