

Youtube Diff Checker

- 新規作成
 - 2022年10月7日 (parcels-dollar.0p@icloud.com)

システム概要

- Youtubeに投稿されている動画をまとめたプレイリスト機能における削除・非公開となった動画が何であったかを監視するウェブアプリケーション

使用技術

- Python
 - 今後の機能展開としてデータの分析を見込んでいるため、仕様に大幅な機能変更を必要としないため
 - ライブラリが豊富であり、一気通貫で書くことができるため
- Docker
 - 開発作業が一つではないため、ライブラリやその他のディストリビューションのバージョン管理などの手間を省くため
 - デプロイ後の管理作業もローカル環境で差異なく行うため
- FastAPI
 - Pythonを用いたフレームワークで軽量かつレスポンスの確認作業などがミスなく行えるため
- Streamlit
 - 本ウェブアプリケーションの主眼がプレイリスト情報の適切管理のため、フロントエンド技術に関して必要以上の機能を必要としないため
- MySQL
 - 本ウェブアプリケーションは一度登録したら読み取りが原則の作業となるため、読み取り操作に関して高いパフォーマンスを持つとされるため

その他、枝葉のフレームワークなどに関しては適宜選定理由を記述する

ページ遷移

- ホーム
 - 本ウェブアプリケーションの説明など
- ユーザ
 - ユーザが登録するプレイリスト一覧
 - ユーザが登録するプレイリストのうち削除されたものの一覧
 - プレイリスト登録画面
- ユーザ登録
 - ユーザ登録画面
 - 必要情報
 - ユーザ名
 - Emailアドレス
 - パスワード
- プレイリスト確認

- ユーザ登録不要で使えるプレイリストに含まれる楽曲を一覧で表示する機能

プロジェクトのディレクトリ構成

```
.
├── .github
│   ├── ISSUE_TEMPLATE
│   └── workflows
├── Document
├── src
│   ├── .dockervenv
│   ├── .venv
│   ├── Nginx
│   │   └── public
│   ├── api
│   │   ├── cruds
│   │   ├── haven
│   │   ├── lib
│   │   ├── models
│   │   ├── routers
│   │   └── schemas
│   ├── cron
│   │   └── log
│   ├── st_server
│   │   └── public
│   │       └── streamlit1
│   │           ├── img
│   │           ├── libs
│   │           └── views
└── tests
```

各ディレクトリと必要機能について

/.dockervenv

- dockerにて仮想環境を作成する際に生成されるディレクトリ
- 基本リポジトリにはコミットしないものとする

/api

- FastAPIによるバックエンド部分の処理を担うコンテナに関する記述
- **main.py**
 - エントリーポイント
- **db.py**
 - DB接続
- **Dockerfile**
 - コンテナ設定
- **mysql.conf**
 - 日本語記述有効化のマウント用

- `create_table.py`
 - MySQLDBのマイグレーションを行う
- 子階層は以下の通り
 - 機能としては命名規約通り
 - `/cruds`
 - `/lib`
 - `/models`
 - `/routers`
 - `/schemas`

`/cron`

- 日々の監視作業を担うコンテナに関する記述
- ログ記述に関する部分と情報取得に関する部分
- `main.py`
 - 監視作業と実行ログ・エラーログを吐き出す
 - 情報取得し、増減を適切に処理する
- `Dockerfile`
 - コンテナの設定を記述
- 子階層は以下の通り
 - `/log`

`/Nginx`

- プロキシサーバーの役割を担うコンテナに関する記述
- フロントエンド部分のマスク
- 子階層は以下の通り
 - `/public` 表示テスト
- `default.conf`
 - Nginxの設定ファイルを記述
 - Dockerコンテナ上の設定ファイルをマウントする
 - プロキシサーバーとしての記述を行う
- `Dockerfile`
 - Nginxコンテナに関する記述

`/st_server`

- フロントエンド部分を担うコンテナに関する記述
- 原則`Streamlit`で記述する
- 子階層は以下の通り
 - `/public/streamlit1`
 - `/img/` : イメージを格納
 - `/libs` : バックエンドとの関連付けを一括電管理
 - `/views` : 各ページ遷移
 - `sample_1.py` : エントリーポイント
- `Dockerfile`
 - コンテナに関する記述

tests

- 単体テスト
 - `test_main.py`: メイン部分の処理のテスト
 - `test.sqlite3`: サンプルDB