MegaPi

Welcome to MegaPi, the dynamic backend API powering the music sim web application. Engineered to revolutionize the way users explore music, MegaPi delivers personalized song recommendations with unparalleled accuracy. At its core, MegaPi harnesses the power of the Milvus vector database and sophisticated single feature vectors, crafted using an Essentia CNN specifically trained for music genre classification. Coupled with the robust MinIO object storage system, MegaPi provides a seamless and innovative solution for navigating through an extensive music library, ensuring every user experience is both unique and engaging.

Runs in a Docker container

```
docker compose up -d
```

```
docker ps --format 'table
{{.ID}}\t{{.Image}}\t{{.Command}}\t{{.Status}}\t{{.Names}}'
```

```
NAMES                           PORTS

megapi-megapi-1                 8000:8000
megapi-minio-1                  9000:9001
megapi-postgre-1                5432:5432
```

Project layout

```
mkdocs.yml    # The configuration file.
docs/
    index.md  # The documentation homepage.
    ...       # Other markdown pages, images and other files.
```

```
.
├── app.py
├── docker-compose.yaml
├── Dockerfile
├── requirements.txt
├── mkdocs.yml
├── README.md
│
├── core
│   ├── data
│   │   ├── mtg_jamendo_genre.json
│   │   └── music.db
│   ├── __init__.py
│   ├── database.py
│   ├── extract_openl3_embeddings.py
│   ├── config.py
│   └── extract_openl3_embeddings.py
│
├── docs
│   ├── endpoints
│   ├── index.md
│   └── services
│
├── gui
│   └── templates
│       └── index.html
│
├── models
│   ├── favorites.py
│   ├── __init__.py
│   ├── milvus.py
│   ├── minio.py
│   ├── music.py
│   ├── openl3.py
│   ├── spotinite.py
│   ├── uploaded.py
│   └── users.py
│
├── routes
│   ├── auth.py
│   ├── favorites.py
│   ├── __init__.py
│   ├── lyrics.py
│   ├── milvus.py
│   ├── minio.py
│   ├── monitoring.py
│   ├── music.py
│   ├── openl3.py
│   ├── spotinite.py
│   └── uploaded.py
│
├── services
│   ├── auth.py
│   ├── favorites.py
│   ├── __init__.py
│   ├── lyrics.py
│   ├── milvus.py
│   ├── minio.py
│   ├── monitoring.py
│   ├── openl3.py
│   ├── spotinite.py
│   └── uploaded.py
│
├── site
│   └── ...
│
└── tests
    ├── __init__.py
    ├── test_auth.py
    ├── test_files.py
    ├── test_milvus.py
    └── test_minio.py
```

MkDocs Commands

- `mkdocs new [dir-name]` - Create a new project.
- `mkdocs serve` - Start the live-reloading docs server.
- `mkdocs build` - Build the documentation site.
- `mkdocs -h` - Print help message and exit.