**Документация для API обработки и анализа аудиофайлов**

Это API позволяет загружать аудиофайлы, анализировать их с использованием различных сервисов, синтезировать текстовую информацию в речь и предоставлять результаты для скачивания.

**Структура API**

1. **Базовый URL**: /
2. **Подключенные модули**:
   * /acrcloud — работа с ACRCloud для идентификации аудио.
   * /acrcloud\_json — извлечение информации о треке.
   * /gigachat — работа с GigaChat.
   * /yandex — работа с Яндекс.Облаком (синтез речи).

**Эндпоинты**

**1. Главная страница — GET /**

Возвращает HTML-страницу с формой для загрузки и обработки аудиофайлов.

**2. Загрузка аудиофайла — POST /upload/**

**Описание**: Загрузка аудиофайла на сервер.

**Параметры:**

* file (UploadFile): Аудиофайл для загрузки.

**Ответ:**

json

Копировать код

{

"message": "Файл загружен",

"file\_path": "uploads/<имя\_файла>",

"play\_url": "/play\_upload/<имя\_файла>"

}

**3. Воспроизведение загруженного аудиофайла — GET /play\_upload/{filename}**

**Описание**: Воспроизведение ранее загруженного файла.

**Параметры:**

* filename (str): Имя файла, загруженного ранее.

**Ответ:**

Возвращает аудиофайл для воспроизведения.

**Ошибки:**

* 404 File Not Found: Если файл отсутствует.

**4. Обработка аудиофайла — POST /process\_audio/**

**Описание**: Анализ аудиофайла, генерация истории о треке и синтез текста в речь.

**Параметры (в теле запроса):**

* file (UploadFile): Аудиофайл для анализа.
* acr\_access\_key (str): Ключ доступа к ACRCloud.
* acr\_access\_secret (str): Секретный ключ доступа к ACRCloud.
* end\_point\_acrcloud (str): URL конечной точки ACRCloud.
* oauth\_token (str): OAuth-токен для Яндекс.Облака.
* folder\_id (str): Идентификатор каталога в Яндекс.Облаке.
* end\_point\_yandex\_iam\_token (str): URL для получения IAM-токена Яндекс.Облака.
* end\_point\_yandex\_text\_to\_speech (str): URL для синтеза текста в речь.
* auth\_token (str): Токен доступа для GigaChat.
* end\_point\_gigachat\_token (str): URL конечной точки GigaChat.

**Ответ:**

json

Копировать код

{

"message": "Аудио успешно обработано.",

"full\_text": "<полный текст истории>",

"output\_file": "output/<имя\_файла>"

}

**Ошибки:**

* 400 Bad Request: Ошибка загрузки файла.
* 500 Internal Server Error: Ошибка анализа или синтеза.

**5. Скачивание обработанного файла — GET /download/**

**Описание**: Скачивание сгенерированного аудиофайла.

**Параметры (в query string):**

* file\_path (str): Путь к файлу для скачивания.

**Ответ:**

Возвращает файл для скачивания.

**Ошибки:**

* 404 File Not Found: Если файл отсутствует.

**Используемые библиотеки и компоненты**

* **FastAPI**: Для создания API.
* **Jinja2**: Для рендеринга HTML-шаблонов.
* **StaticFiles**: Для обслуживания статических файлов.
* **ACRCloud**: Для идентификации аудио.
* **GigaChat**: Для генерации текстовой информации о треке.
* **Яндекс.Облако**: Для синтеза текста в речь.

**Пример сценария использования**

1. **Загрузка аудиофайла**:

bash

Копировать код

curl -X POST -F "file=@audio.mp3" http://localhost:8000/upload/

Ответ:

json

Копировать код

{

"message": "Файл загружен",

"file\_path": "uploads/audio.mp3",

"play\_url": "/play\_upload/audio.mp3"

}

1. **Анализ и синтез речи**:

bash

Копировать код

curl -X POST -F "file=@audio.mp3" \

-F "acr\_access\_key=your\_acr\_key" \

-F "acr\_access\_secret=your\_acr\_secret" \

-F "oauth\_token=your\_oauth\_token" \

-F "folder\_id=your\_folder\_id" \

-F "auth\_token=your\_gigachat\_token" \

http://localhost:8000/process\_audio/

Ответ:

json

Копировать код

{

"message": "Аудио успешно обработано.",

"full\_text": "Полная информация о треке.",

"output\_file": "output/audio.wav"

}

1. **Скачивание результата**:

bash

Копировать код

curl -O "http://localhost:8000/download/?file\_path=output/audio.wav"