

# MusicPlayerAppの概要(2023/04/26)

---

## このアプリの概要

今回作成した「MusicPlayerApp」は、iTunesを経由することなく、iPhoneなどのアプリ専用フォルダに音楽ファイルを入れることで音楽再生することができる。また、別途作成した音楽サーバからサーバに保存されている音楽ファイルをダウンロードし、再生することもできる。さらに、アプリ側からMP3に変換したいYouTubeのURLをサーバに転送すると、サーバ側でYoutubeの動画を音楽ファイル(MP3)に変換し、それをダウンロード、再生することもできる。

## 作成した背景

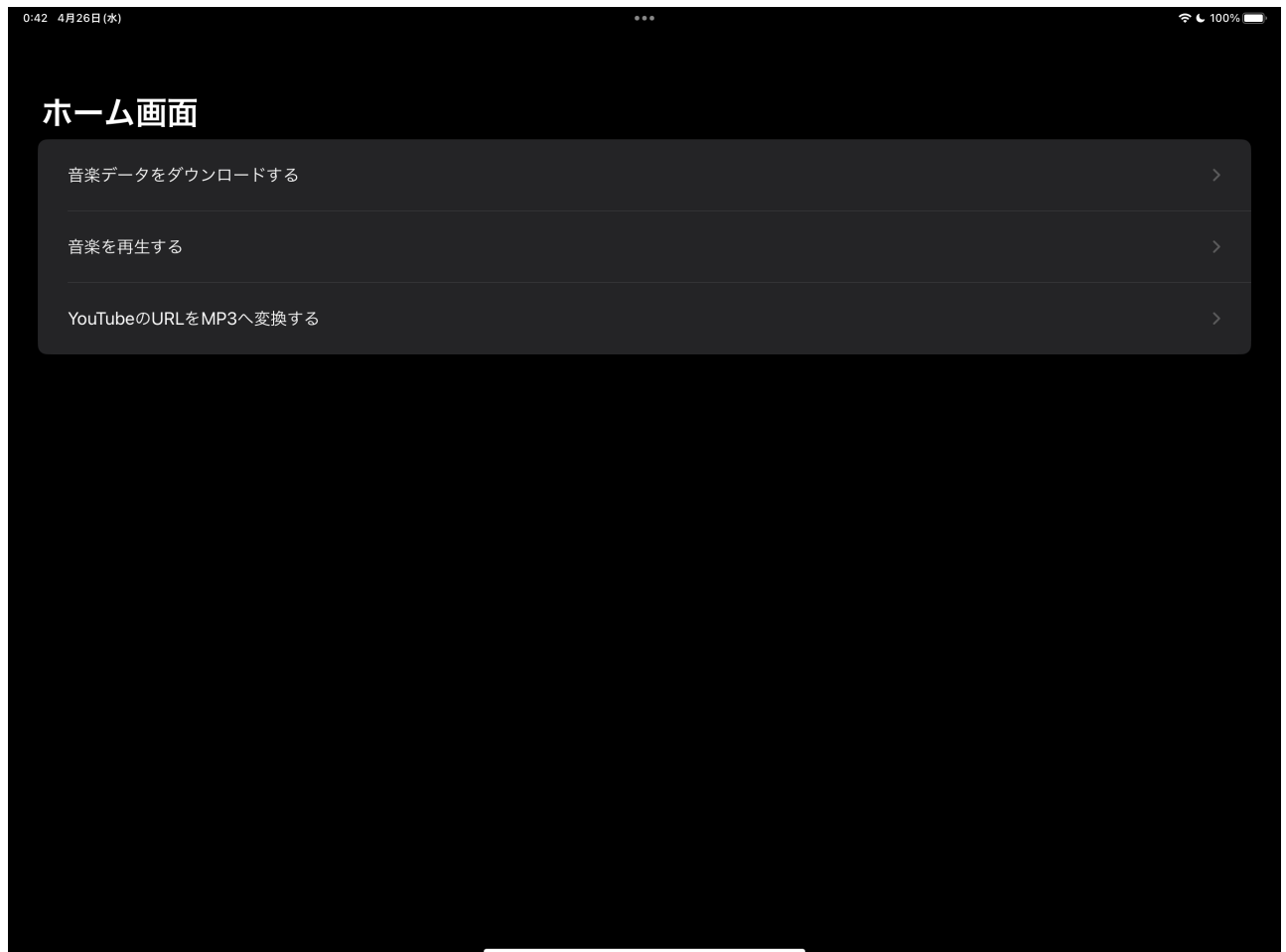
iTunesを経由して音楽ファイルをiPhoneなどに入れることが面倒だったので、それを回避するために作成した。また、アプリ側から間接的ではあるがYouTubeの動画をMP3に変換した音楽ファイルを扱えるようにできると便利だと思い、その機能も実装した。あと、Webサーバの勉強の延長線も理由である。

## 制作期間

2023/04/12~2023/04/22

## 実際のアプリ画面

- ホーム画面



- 音楽選択画面



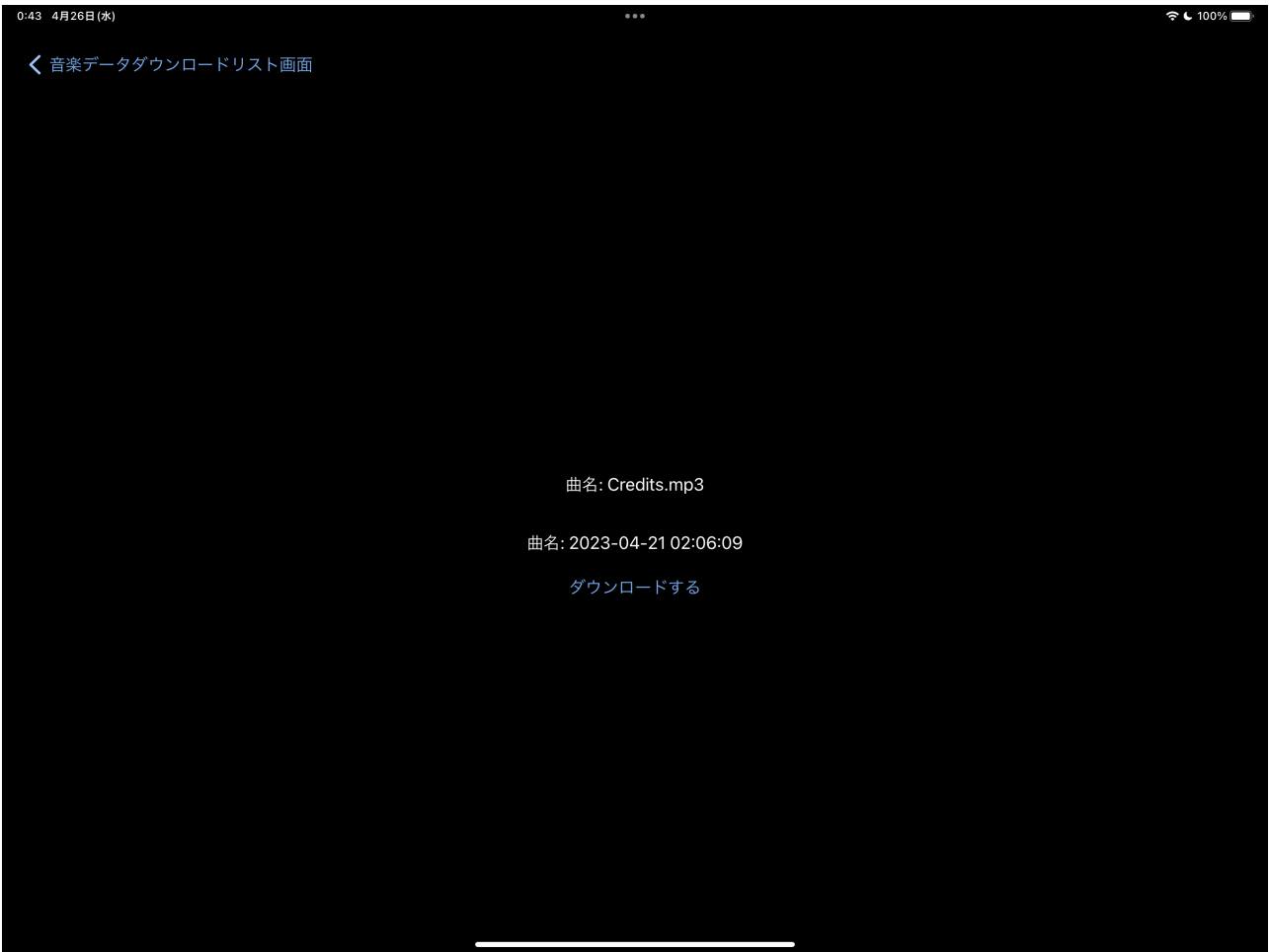
- 音楽再生画面



- 音楽ダウンロード選択画面



- 音楽ダウンロード画面



- サーバへ音楽ファイルの変換要求画面

0:43 4月26日 (水) 100%

< ホーム画面

MP3へ変換したいYouTubeの動画URLを入力

変換したいYouTubeのURL

変換後の音楽ファイル名を入力

変換後の音楽ファイル名

サーバへ送信

## 懸念事項

Swiftでアプリ開発は日が立って浅いため、プログラムの良くない所はあると思うが、やさしい目で見てほしい。

## 以下は、設計時の仕様メモ

### 各システムの概要

#### スマホアプリ側

##### 動作環境

- 使用デバイス  
iPad Pro 12.9 5th, iPhone XS Max
- プログラミング言語  
Swift(SwiftUI)

#### サーバ側

##### 動作環境

- 使用デバイス  
サーバ： MINISFORUM ミニPC U820 Core i5 8279U Ubuntu22.04

- プログラミング言語  
JavaScript(Node.js)

## スマホアプリ側の設計について

### プロジェクト名

MusicApp

### 必要な機能

- デバイス内ファイル（アプリ内Documents）から音楽ファイルを読み出し、再生する
- 再生する音楽の制御（アプリ内の音量調整, 再生タイミングの制御など）
- 音楽サーバからサーバ内の各音楽データの情報を取得し、指定した音楽データをダウンロード, アプリ内ファイルに保存する

### 現時点で実現可能な機能について

現時点(2024/04/14)で実現可能なことは、

- ファイル内音楽データの再生
- 音楽データの再生制御
- サーバから音楽ファイルのダウンロードし、ストリーミング再生, ファイル内保存, 再生

### データの送受信について

#### 音楽ファイルの情報取得について

音楽ファイルのの情報取得は音楽サーバにアクセスし、JSON形式で音楽ファイル情報を取得

JSONの型は、（例）{"MusicName": "kimitonara.mp3", "RegistratinData": "2023-04-14 00:00:00"}

#### 音楽ファイルのデータの受信について

音楽ファイルのデータの受信は「音楽ファイルの情報取得について」とは別のURLでアクセスし、  
URLSession.shared.downloadtask()で取得

### 進行度

- 2023/04/14  
アプリ内Documentsにある音楽ファイルの読み込み&再生ができた サーバから音楽データリスト情報を取得  
や音楽データを受信, 保存ができた
- 2023/04/18  
アプリ側から指定した音楽ファイル名をサーバに送信し、その音楽ファイルをダウンロードできた（再生可）  
ファイル内の音楽データでmp3以外のファイルは読み取れないようにできた（拡張子判別）
- 2023/04/19  
音楽再生画面でスライドバーの表示ができた
- 2023/04/21  
サーバへYouTubeの動画をmp3へ変換するリクエストはできた。（エラー処理は未完成）

- 2023/04/28  
バックグラウンド再生が可能になった

## サーバ側の設計について

### 必要な機能について

- アクセスしてきたアプリに対して、音楽データの情報をJSON形式で送信する
- アプリが指定した音楽データをアプリに対して、送信する

### データベースについて

音楽データが保存されているディレクトリ内の音楽データの情報を保存する 具体的には、

- 登録番号
- 登録日付
- 音楽ファイルの名前(JSON)
- 音楽ファイルのパス(JSON)  
を保存する  
また、一部のデータはJSON形式で保存する

### データベースのアクセス方法

host: 'localhost', user: 'EC2\_Ubuntu', password: 'Kenji0510!', database: 'MusicDataInfo' table:  
'MusicInfoTable'

### データベースの構成

以下は表'MusicInfoTable'の例

Number	RegistrationDate	MusicInfo
1	2023-04-14 00:00:00	{"MusicName": "kimitonara.mp3", "FilePath": "/home/kenji/WorkSpace/FileServer/public/music/kimitonara.mp3"}
2	2023-04-14 00:00:00	{"MusicName": "Track5.mp3", "FilePath": "/home/kenji/WorkSpace/FileServer/public/music/Track5.mp3" }

※ MusicInfoはJSON形式で保存

### 進行度

- 2023/04/14  
public/music以下に存在する音楽データの情報をデータベースに保存できた 音楽データ情報のデータベースから存在する音楽データの情報リストを取得し、スマホアプリに送信することができた
- 2023/04/18  
アプリ側から指定された音楽ファイル名をデータベースに問い合わせ、その音楽ファイルのパスを取得。そのパスをもとにファイルから音楽ファイルを読み出し、アプリへ送信できた
- 2023/04/19  
fileWatchApp.jsを追加作成 /public/musicに新規追加された音楽ファイル情報（音楽ファイル名、音楽ファイ

ルの絶対パス)を追加された時点でデータベースに自動登録することができた

- 2023/04/20  
指定されたYoutubeの動画をm4aでダウンロードし、それをmp3へ変換する機能をつくれた。
- 2023/04/21  
アプリからのYouTubeの動画をmp3へ変換するリクエストを