内容

[1 概要 2](#_Toc43053297)

[1.1 リソース 2](#_Toc43053298)

[2 開発環境 2](#_Toc43053299)

[2.1 アンドロイドスタジオ（Android Studio 4.0） 2](#_Toc43053300)

[2.2 Google Cloud Pllatform 2](#_Toc43053301)

[3 アンドロイドAPPの作成 2](#_Toc43053302)

[3.1 「Empty Activity」でプロジェクトを作成します。 2](#_Toc43053303)

[4 認証 2](#_Toc43053304)

[4.1 Firebase を Android プロジェクトに追加します。 2](#_Toc43053305)

[4.2 Firebaseログイン方法 2](#_Toc43053306)

[4.3 FirebaseUI の依存関係を追加 3](#_Toc43053307)

[4.4 APIの使い方 4](#_Toc43053308)

[5 音声の再生 4](#_Toc43053309)

[5.1 コントルール画面 4](#_Toc43053310)

[5.2 サービスで音声再生 5](#_Toc43053311)

[6 素材データの保存 5](#_Toc43053312)

[6.1 FirebaseのStorage 5](#_Toc43053313)

[6.2 APIの使い方 5](#_Toc43053314)

[7 状態遷移 7](#_Toc43053315)

# 概要

ユーザーに寄添う放送システムを開発します。

## リソース

・プロジェクトURL

<https://github.com/tofumiyoshi/mybroadcast>

・アンドロイドAPK

<https://github.com/tofumiyoshi/coronafighter/blob/master/coronafighter/app/release/app-release.apk>

# 開発環境

## アンドロイドスタジオ（Android Studio 4.0）

下記のURLを参照し、インストールします。

<https://developer.android.com/studio/install?authuser=1&hl=ja>

## Google Cloud Pllatform

下記のURLを参照し、アカウントを作成します。

<https://cloud.google.com/docs/?hl=ja>

# アンドロイドAPPの作成

## 「Empty Activity」でプロジェクトを作成します。

* 1. プロジェクト名に「MyBroadcast」を入力します。

# 認証

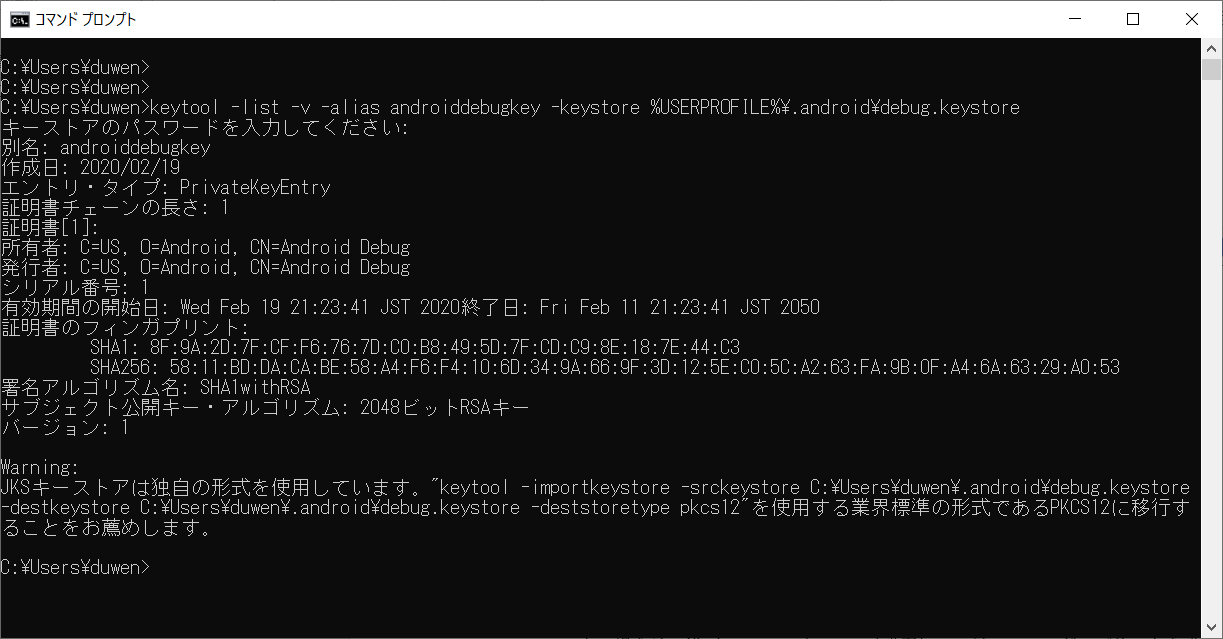
## Firebase を Android プロジェクトに追加します。

## Firebaseログイン方法

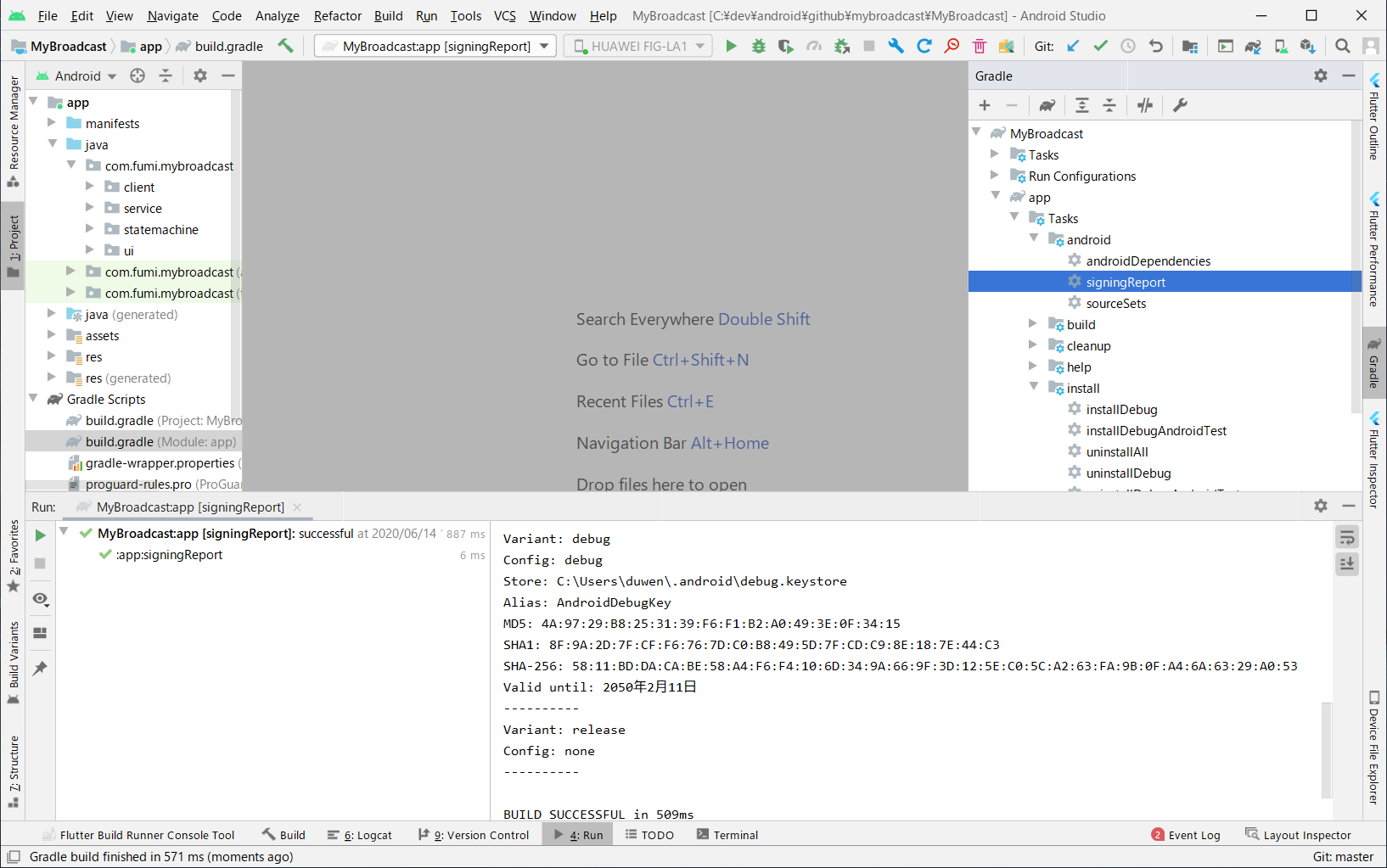
Firebase コンソールで [Authentication] セクションを開き、サポートするログイン方法を有効にします。

Google ログインをサポートし、アプリの SHA-1 フィンガープリントを指定していない場合は、Firebase コンソールの [設定] ページでフィンガープリントを設定します。

keytool -list -v -alias androiddebugkey -keystore %USERPROFILE%\.android\debug.keystore



或いは

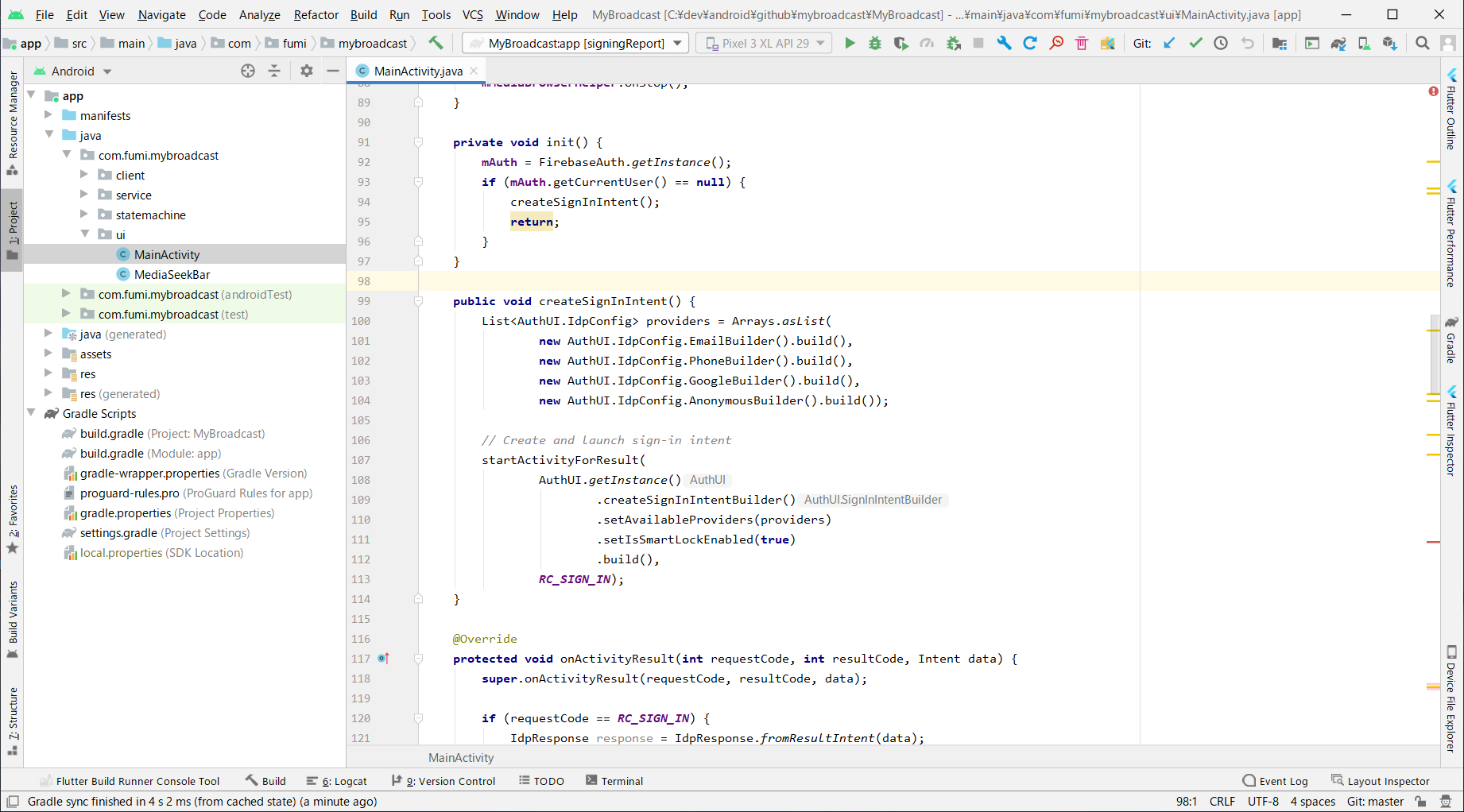


## FirebaseUI の依存関係を追加

アプリレベルの build.gradle ファイルに FirebaseUI の依存関係を追加します。

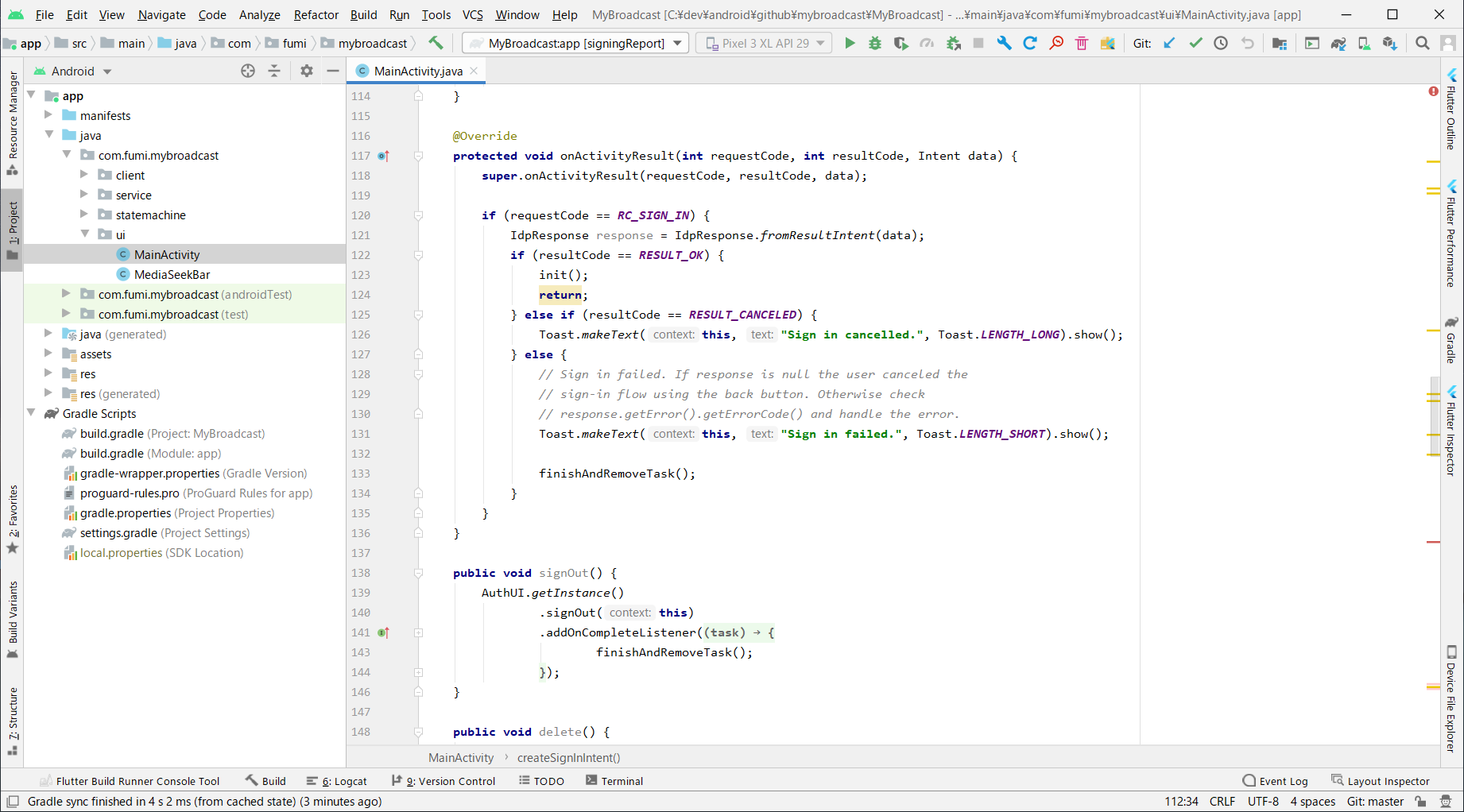
implementation **'com.google.firebase:firebase-auth:19.3.1'**implementation **'com.firebaseui:firebase-ui-auth:4.3.1'**

## APIの使い方



ログインをする

未ログインの場合、ログインメソッドを呼びだす



ログイン成功の場合、初期化処理続行

# 音声の再生

## Google Cloud Platform Storageから再生

## メディア・ファイルをGoogle Cloud Platform Storageにアップロードする。



## メディアファイルの参照を取得する。

StorageReference mStorageRef =

FirebaseStorage.getInstance().getReference();

public static StorageReference getMusicStorageRef(String mediaId) {

if (musicFileName.containsKey(mediaId)) {

StringBuffer path = new StringBuffer();

path.append(mediaId).append("/")

.append(musicFileName.get(mediaId));

return mStorageRef.child(path.toString());

}

return null;

}

## URLからメディアを再生する。

StorageReference ref = MusicLibrary.getMusicStorageRef(message);

ref.getDownloadUrl().addOnCompleteListener(

new OnCompleteListener<Uri>() {

@Override

public void onComplete(@NonNull Task<Uri> task) {

Uri uri = task.getResult();

try {

playMusic(uri.toString(), mediaPlayer);

} catch (IOException ex) {

Log.d(TAG, ex.getMessage(), ex);

}

});

private void playMusic(String url, MediaPlayer mediaPlayer)

throws IOException {

mediaPlayer.setAudioStreamType(

AudioManager.STREAM\_MUSIC);

mediaPlayer.setDataSource(url);

mediaPlayer.prepare();

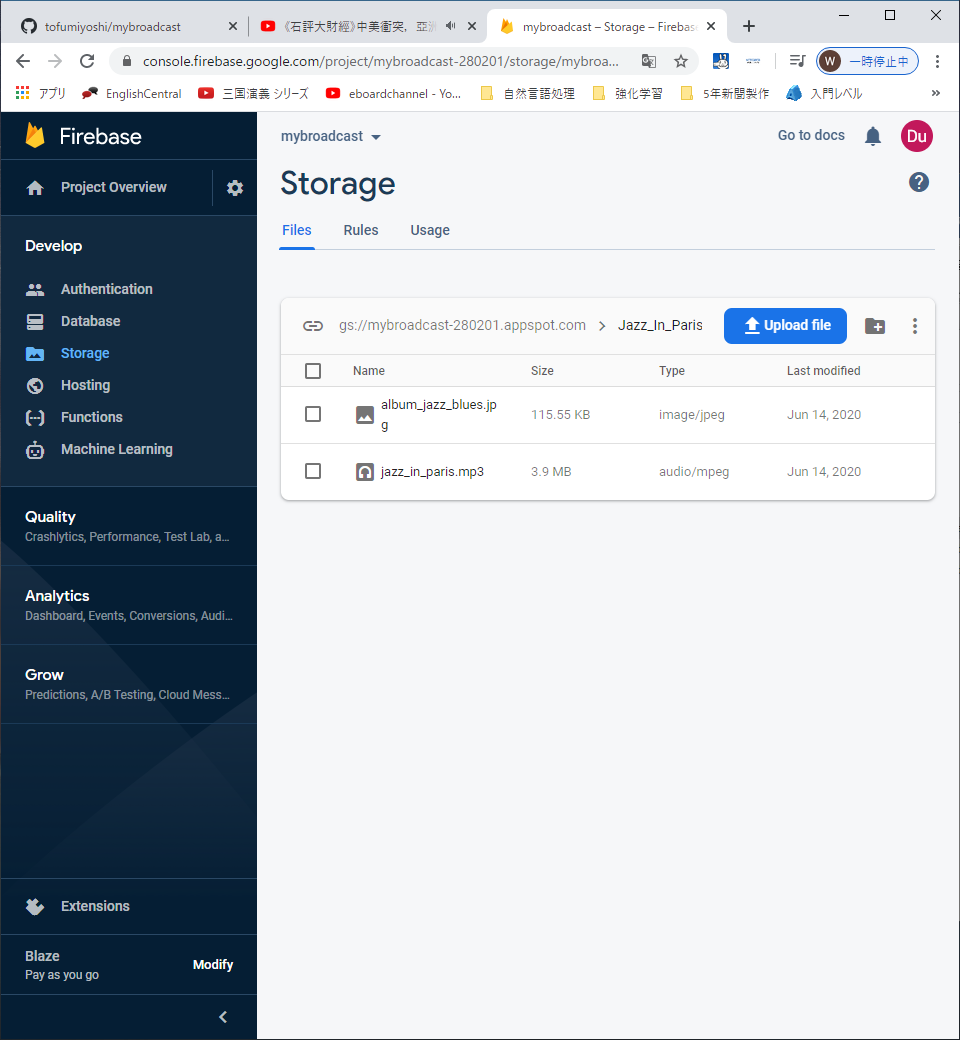
mediaPlayer.start();

}

## サービスで音声再生

# 素材データの保存

## FirebaseのStorage



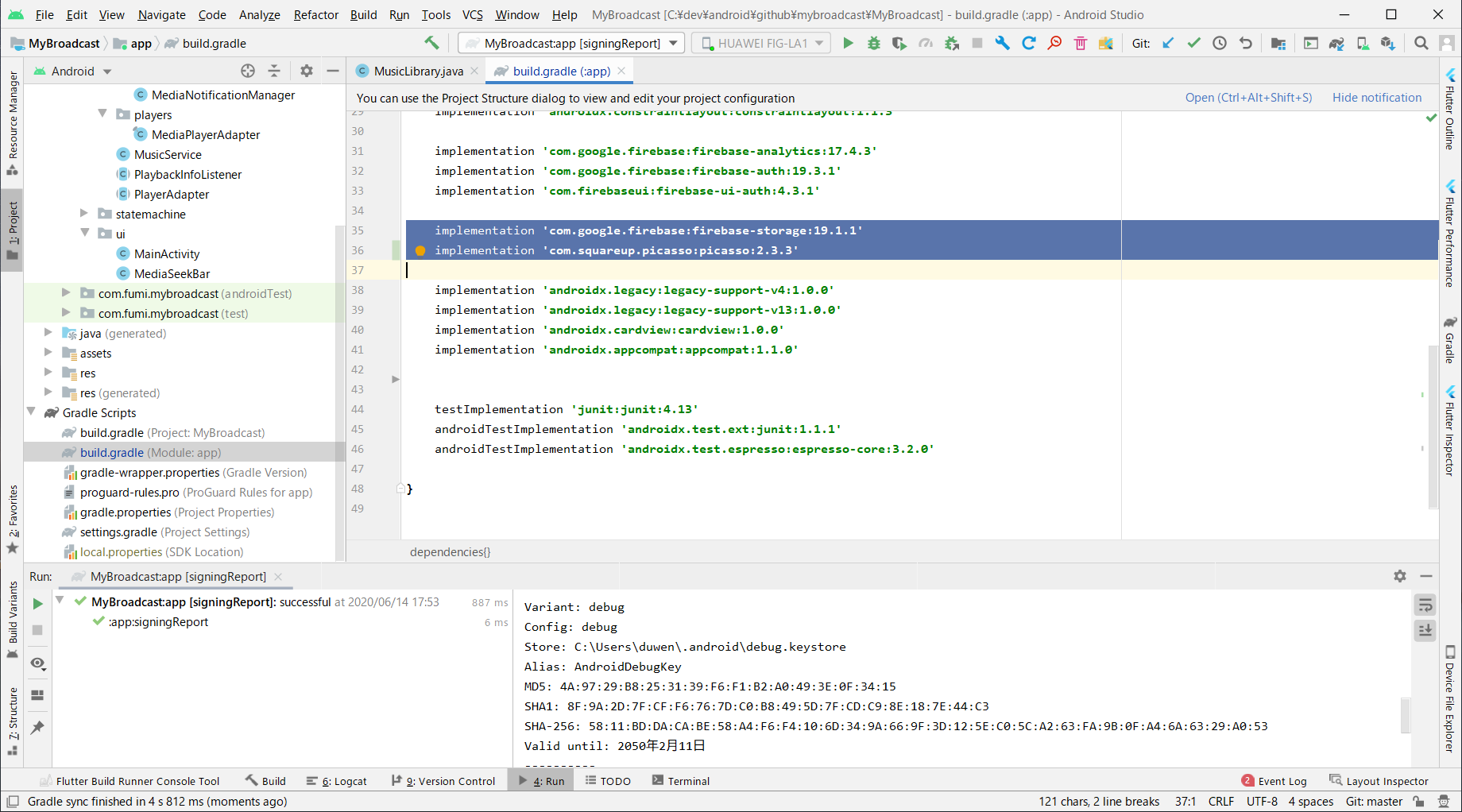
アルバム・アートワーク

メディアデータ

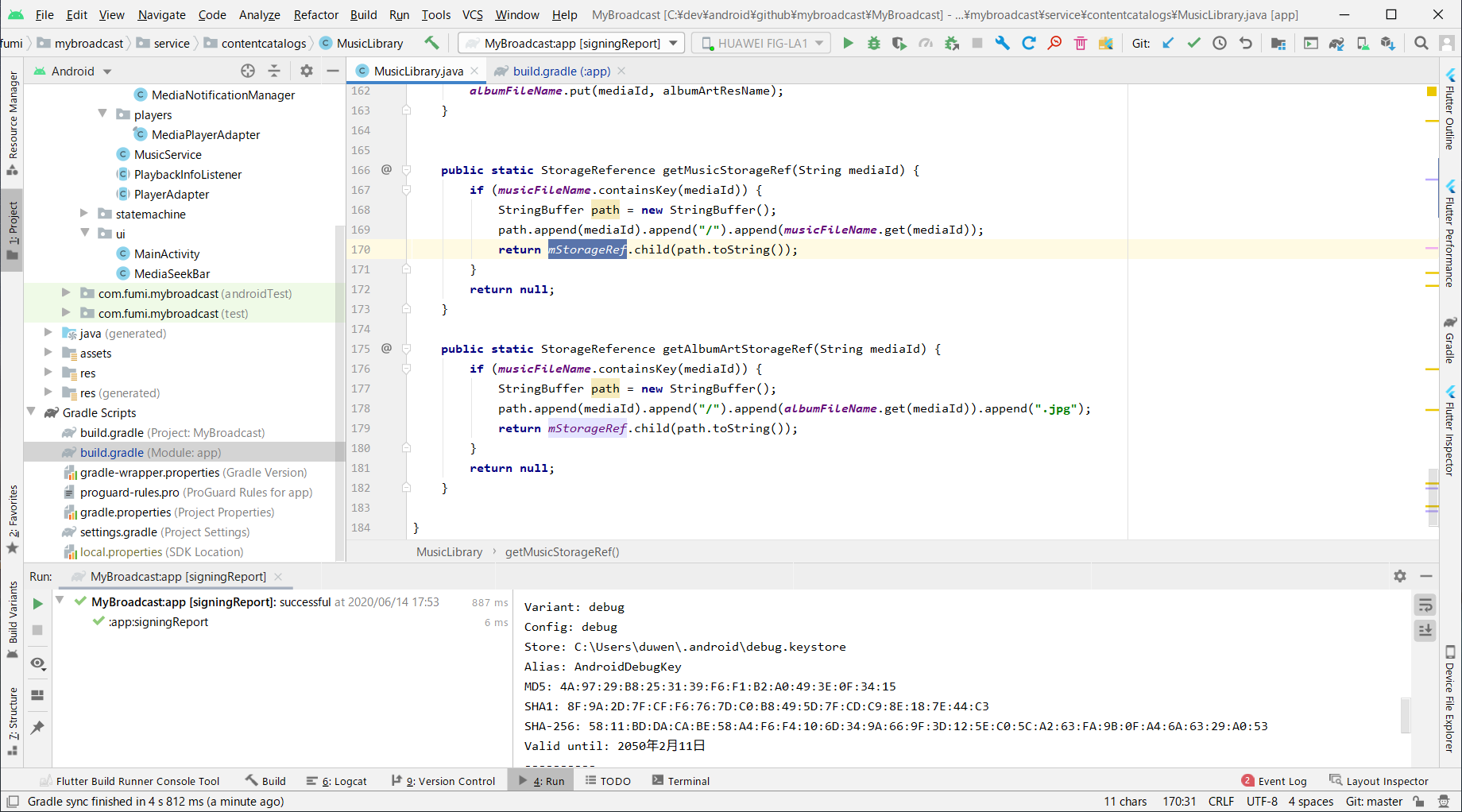
メディアID

## APIの使い方

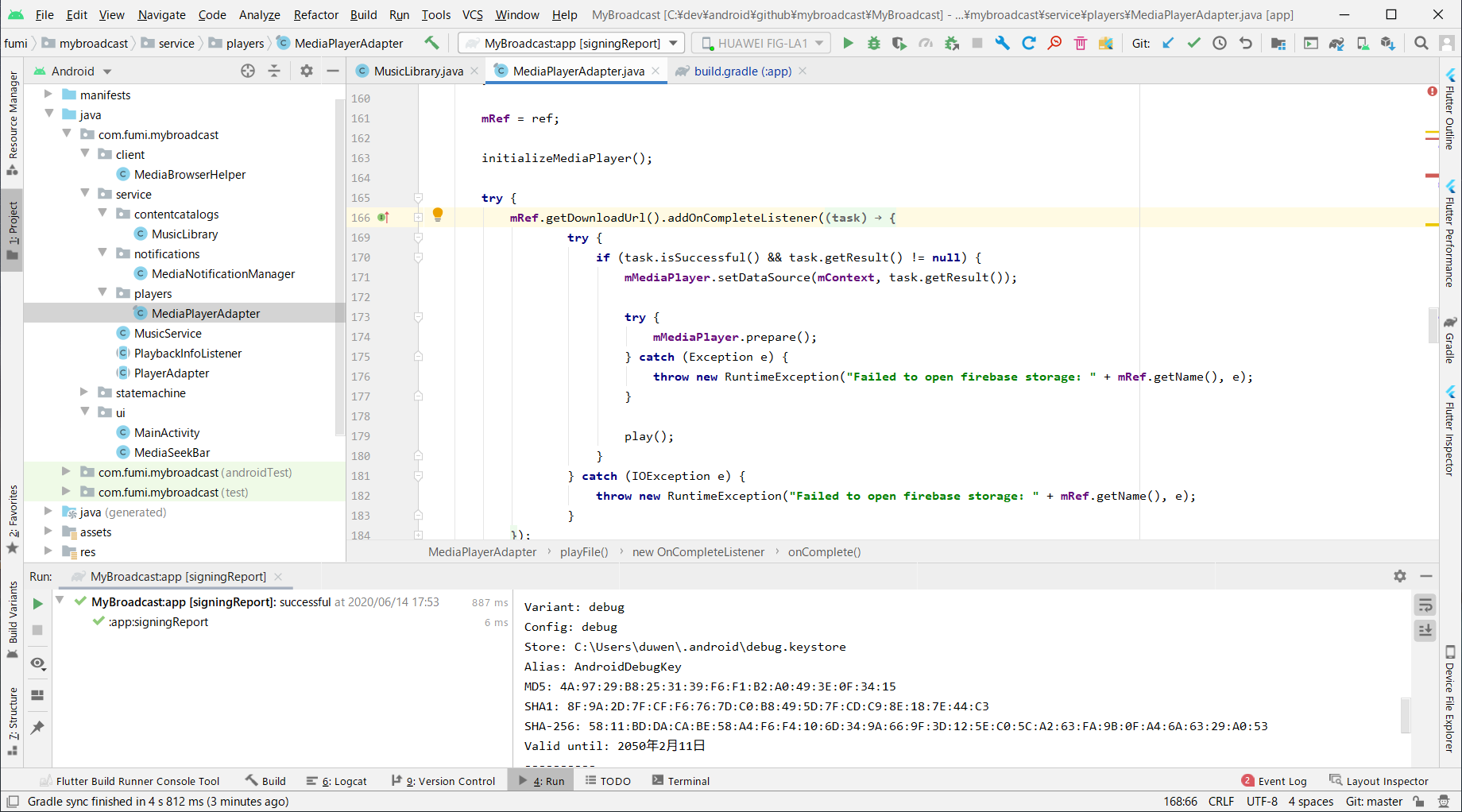
・依存関係の設定



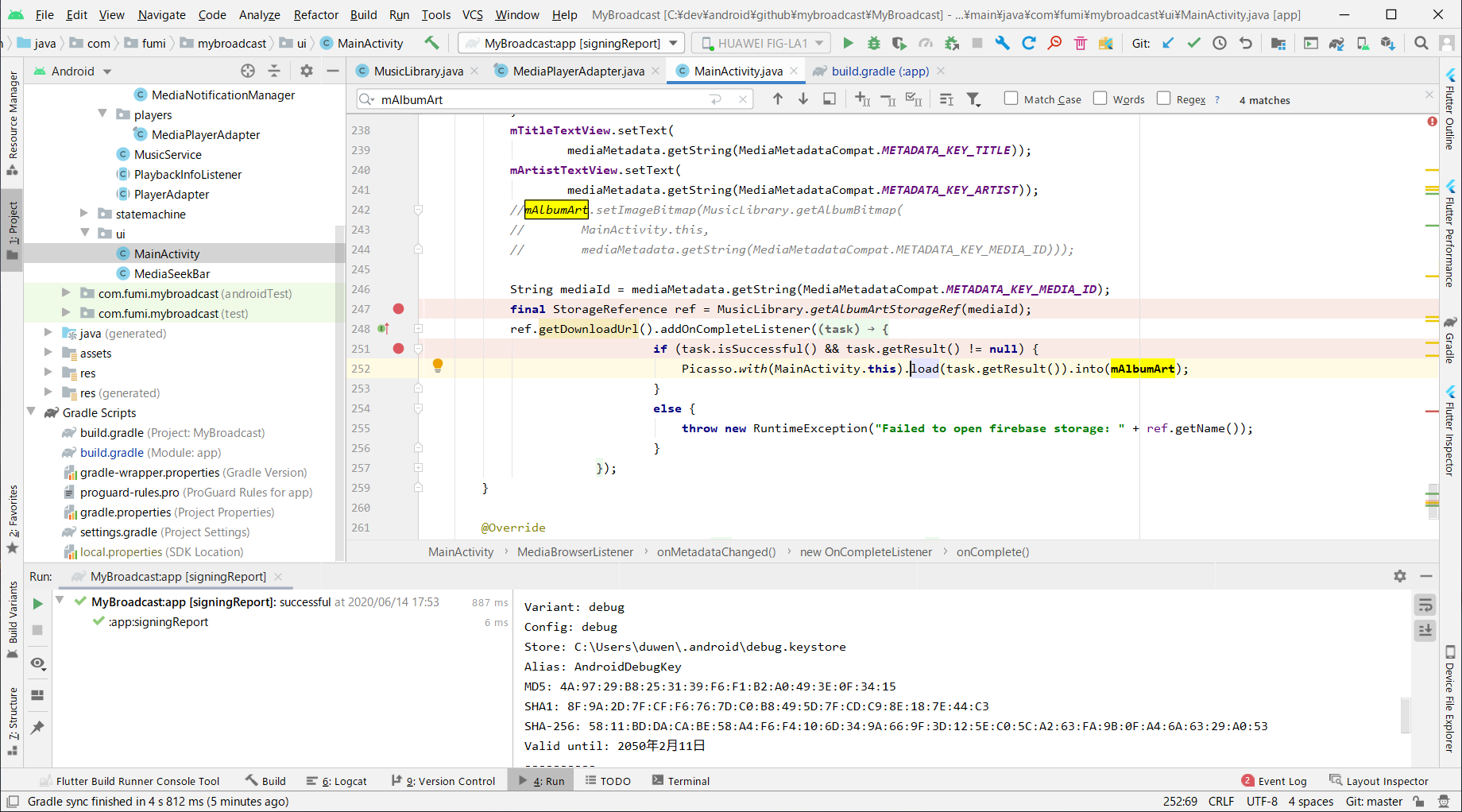
・参照取得



・URIでメディア再生



・URIでアルバム画像を表示

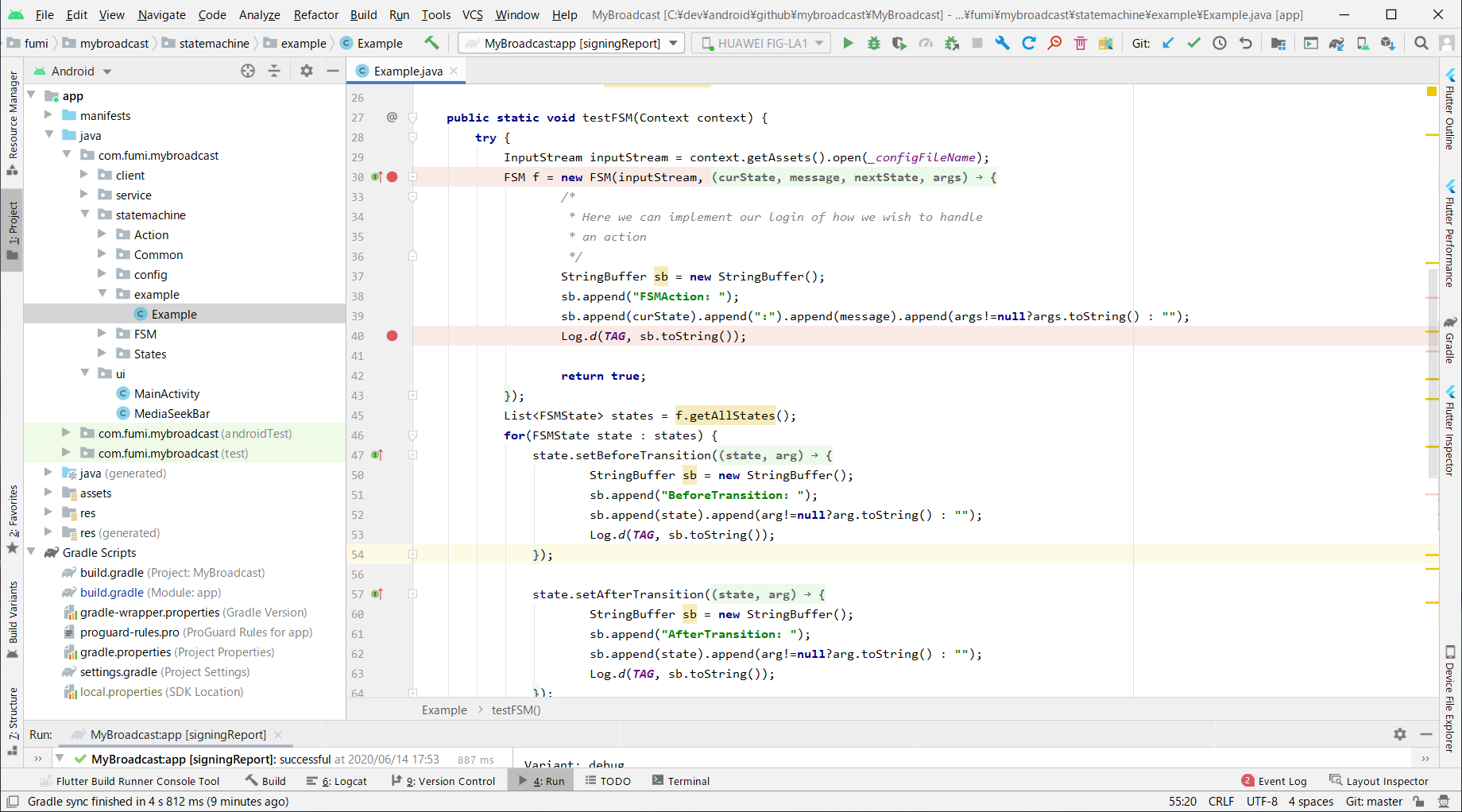


# 状態遷移

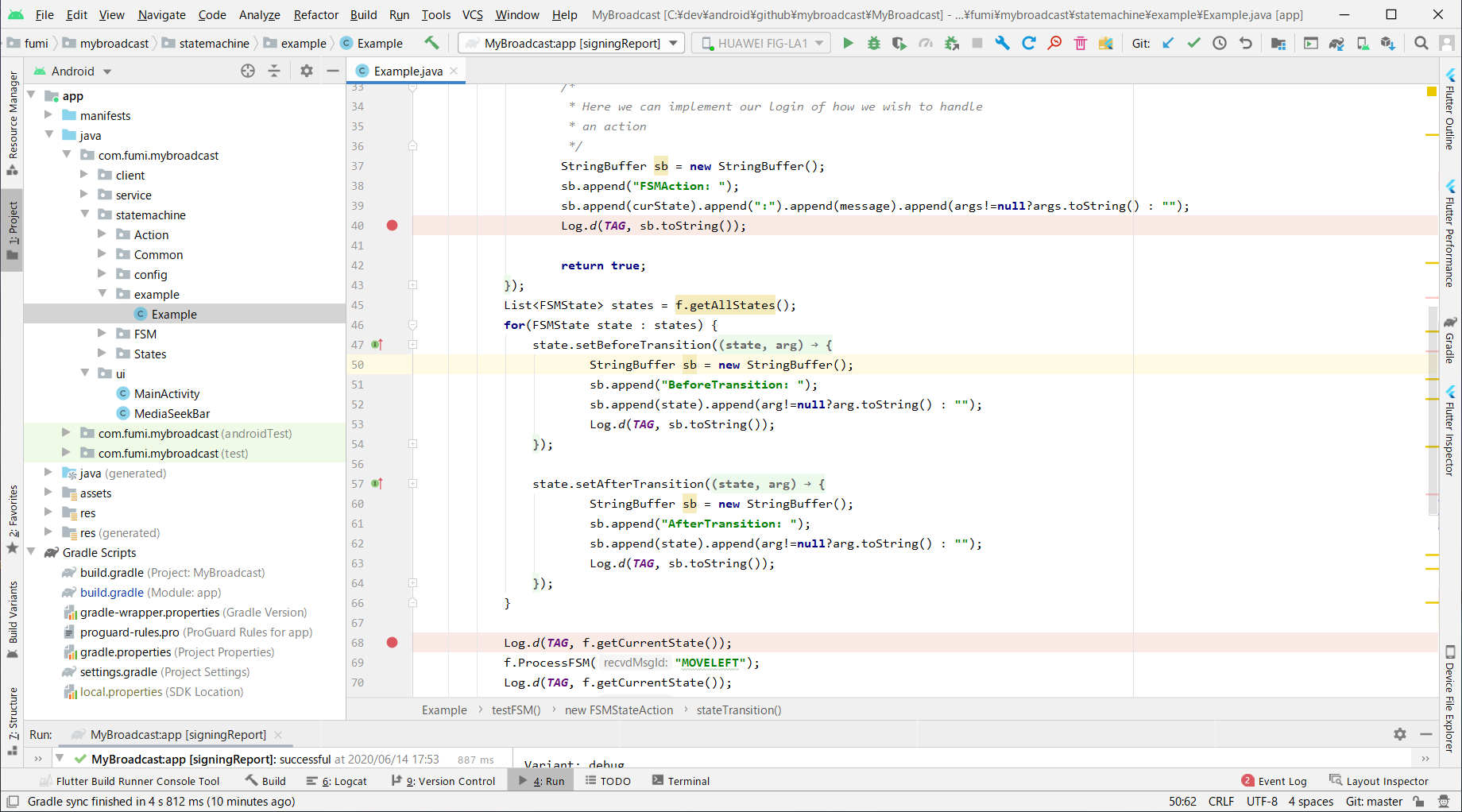
・easy-fsmからソースコードを利用

<https://github.com/koushikr/easy-fsm.git>

・使い方

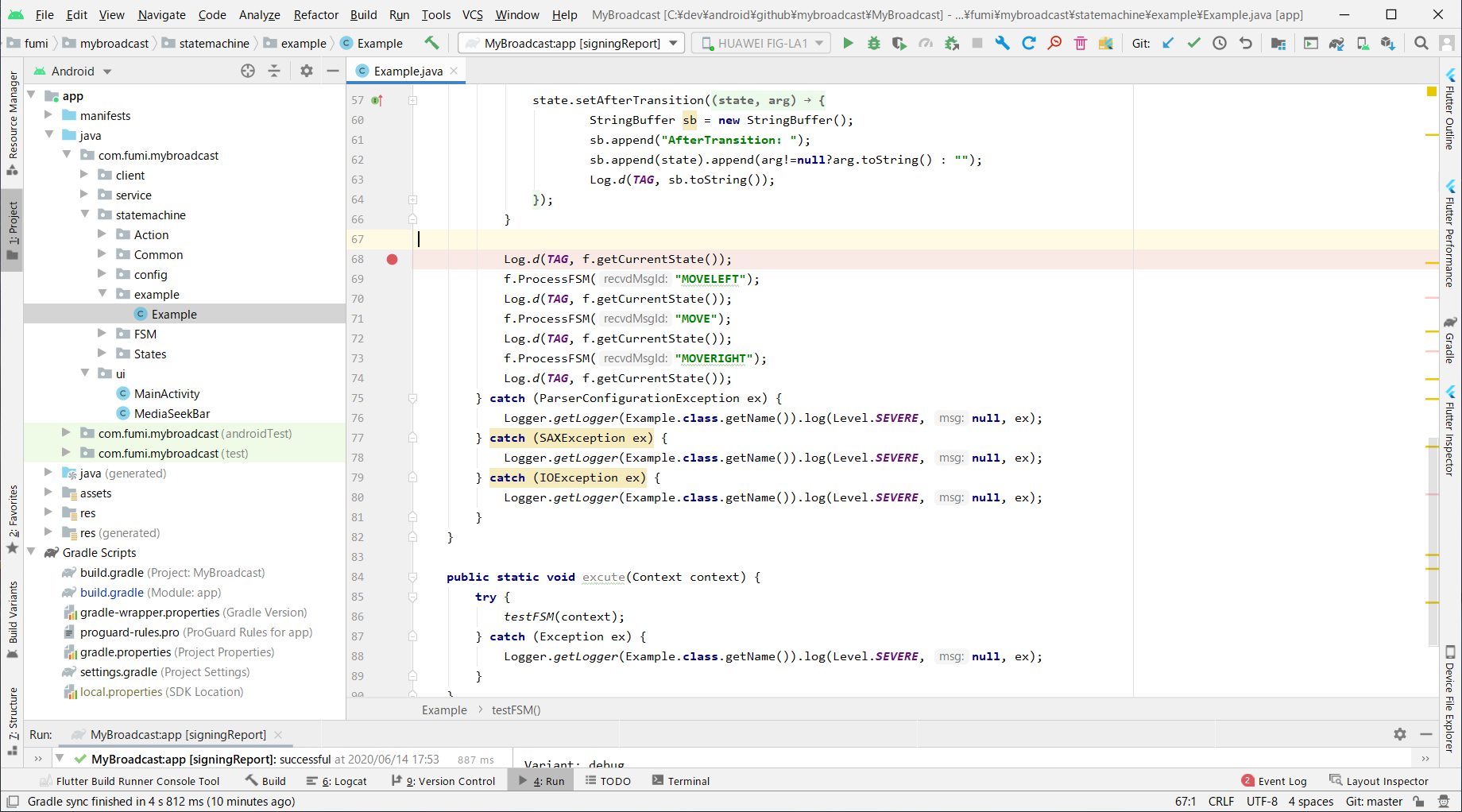


状態遷移時実行する処理



遷移後実行する処理

遷移前実行する処理



状態遷移させます