

原 最简单的基于FFmpeg的移动端例子附件：IOS自带播放器

2015年08月01日 17:03:47 阅读数：16967

=====

最简单的基于FFmpeg的移动端例子系列文章列表：

[最简单的基于FFmpeg的移动端例子：Android HelloWorld](#)

[最简单的基于FFmpeg的移动端例子：Android 视频解码器](#)

[最简单的基于FFmpeg的移动端例子：Android 视频解码器-单个库版](#)

[最简单的基于FFmpeg的移动端例子：Android 推流器](#)

[最简单的基于FFmpeg的移动端例子：Android 视频转码器](#)

[最简单的基于FFmpeg的移动端例子附件：Android 自带播放器](#)

[最简单的基于FFmpeg的移动端例子附件：SDL Android HelloWorld](#)

[最简单的基于FFmpeg的移动端例子：IOS HelloWorld](#)

[最简单的基于FFmpeg的移动端例子：IOS 视频解码器](#)

[最简单的基于FFmpeg的移动端例子：IOS 推流器](#)

[最简单的基于FFmpeg的移动端例子：IOS 视频转码器](#)

[最简单的基于FFmpeg的移动端例子附件：IOS自带播放器](#)

[最简单的基于FFmpeg的移动端例子：Windows Phone HelloWorld](#)

=====

前几篇文章记录的都是IOS下基于FFmpeg的多媒体程序。本文记录一个IOS下自带的视频播放器。相比于移植FFmpeg来说，IOS自带的多媒体框架播放视频要简单很多——只需要调用几个接口就可以了。

□

源代码

项目的源代码位于ViewController.m文件中，内容如下所示。

```
[objc]  📄  📁

1.  /**
2.   * 最简单的IOS视频播放器
3.   * Simplest IOS Player
4.   *
5.   * 雷霄骅 Lei Xiaohua
6.   * leixiaohua1020@126.com
7.   * 中国传媒大学/数字电视技术
8.   * Communication University of China / Digital TV Technology
9.   * http://blog.csdn.net/leixiaohua1020
10.  *
11.  * 本程序是IOS上最简单的视频播放器。
12.  * This software is the simplest Video Player on IOS.
13.  *
14.  */
15.
16. #import "ViewController.h"
17. #import <MediaPlayer/MediaPlayer.h>
18.
19. @interface ViewController ()
20.
21. @property (nonatomic, strong) MPMoviePlayerController *moviePlayer; //视频播放控制器
22.
23. @end
24.
25. @implementation ViewController
26.
27. - (void)viewDidLoad {
28.     [super viewDidLoad];
29.     // Do any additional setup after loading the view, typically from a nib.
30. }
```

```

30.
31.     [self.moviePlayer play];
32.
33.     [self addNotification];
34.
35. }
36.
37. -(void)dealloc{
38.     //移除所有通知监控
39.     [[NSNotificationCenter defaultCenter] removeObserver:self];
40. }
41.
42.
43. -(MPMoviePlayerController *)moviePlayer{
44.     if (!_moviePlayer) {
45.         NSString *urlStr=[[NSBundle mainBundle] resourcePath] stringByAppendingPathComponent:@"resource.bundle/war3end.mp4"];
46.         NSURL *url=[NSURL URLWithString:urlStr];
47.         _moviePlayer=[MPMoviePlayerController alloc] initWithContentURL:url;
48.         _moviePlayer.view.frame=self.view.bounds;
49.         _moviePlayer.view.autoresizingMask=UIViewAutoresizingFlexibleWidth|UIViewAutoresizingFlexibleHeight;
50.         [self.view addSubview:_moviePlayer.view];
51.     }
52.     return _moviePlayer;
53. }
54.
55. -(void)addNotification{
56.     NSNotificationCenter *notificationCenter=[NSNotificationCenter defaultCenter];
57.     [notificationCenter addObserver:self selector:@selector(mediaPlayerPlaybackStateChange:) name:MPMoviePlayerPlaybackStateDidChange
Notification object:self.moviePlayer];
58.     [notificationCenter addObserver:self selector:@selector(mediaPlayerPlaybackFinished:) name:MPMoviePlayerPlaybackDidFinishNotifica
tion object:self.moviePlayer];
59.
60. }
61.
62. -(void)mediaPlayerPlaybackStateChange:(NSNotification *)notification{
63.     switch (self.moviePlayer.playbackState) {
64.         case MPMoviePlaybackStatePlaying:
65.             NSLog(@"正在播放...");
66.             break;
67.         case MPMoviePlaybackStatePaused:
68.             NSLog(@"暂停播放.");
69.             break;
70.         case MPMoviePlaybackStateStopped:
71.             NSLog(@"停止播放.");
72.             break;
73.         default:
74.             NSLog(@"播放状态:%li",self.moviePlayer.playbackState);
75.             break;
76.     }
77. }
78.
79. -(void)mediaPlayerPlaybackFinished:(NSNotification *)notification{
80.     NSLog(@"播放完成.%li",self.moviePlayer.playbackState);
81. }
82.
83.
84.
85.
86. - (void)didReceiveMemoryWarning {
87.     [super didReceiveMemoryWarning];
88.     // Dispose of any resources that can be recreated.
89. }
90.
91. @end

```

运行结果

App在手机上运行后的结果如下图所示。程序运行后，会自动播放resource.bundle中的war3end.mp4文件。

□

下载

simplest ffmpeg mobile

项目主页

Github：https://github.com/leixiaohua1020/simplest_ffmpeg_mobile

开源中国：https://git.oschina.net/leixiaohua1020/simplest_ffmpeg_mobile

SourceForge：<https://sourceforge.net/projects/simplestffmpegmobile/>

CSDN工程下载地址：<http://download.csdn.net/detail/leixiaohua1020/8924391>

本解决方案包含了使用FFmpeg在移动端处理多媒体的各种例子：

[Android]

simplest_android_player: 基于安卓接口的视频播放器

simplest_ffmpeg_android_helloworld: 安卓平台下基于FFmpeg的HelloWorld程序

simplest_ffmpeg_android_decoder: 安卓平台下最简单的基于FFmpeg的视频解码器

simplest_ffmpeg_android_decoder_onelib: 安卓平台下最简单的基于FFmpeg的视频解码器-单库版

simplest_ffmpeg_android_streamer: 安卓平台下最简单的基于FFmpeg的推流器

simplest_ffmpeg_android_transcoder: 安卓平台下移植的FFmpeg命令行工具

simplest_sdl_android_helloworld: 移植SDL到安卓平台的最简单程序

[IOS]

simplest_ios_player: 基于IOS接口的视频播放器

simplest_ffmpeg_ios_helloworld: IOS平台下基于FFmpeg的HelloWorld程序

simplest_ffmpeg_ios_decoder: IOS平台下最简单的基于FFmpeg的视频解码器

simplest_ffmpeg_ios_streamer: IOS平台下最简单的基于FFmpeg的推流器

simplest_ffmpeg_ios_transcoder: IOS平台下移植的ffmpeg.c命令行工具

simplest_sdl_ios_helloworld: 移植SDL到IOS平台的最简单程序

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/leixiaohua1020/article/details/47072861>

文章标签：[IOS](#) [视频](#) [解码](#) [播放器](#)

个人分类：[IOS多媒体](#) [FFMPEG](#)

所属专栏：[FFmpeg](#)

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com