

iOS中支持的音频播放类型

- 播放音效 (System Sound Service)
- 播放本地音频 (AVFoundataion)
- 播放流媒体音频 (AVFoundataion)

iOS支持的音频格式

音频格式	硬件解码	软件解码
AAC	YES	YES
ALAC	YES	YES
HE-AAC	YES	
iLBC		YES
IMA4		YES
Linea PCM		YES
MP3	YES	YES
μ-law and a-law		YES
CAF	YES	YES

注意：硬件解码器一次只能对一个音频文件解码。在实际应用开发中通常使用非压缩音频格式(AIFF)或者CAF音频格式，从而减低系统在音频解码上的消耗，大致省电的目的

关于音效的使用

- 音效又称短音频，是一个声音文件，在应用程序中起到点缀效果，**用于提升应用程序的整体用户体验**
- 通常短音效在程序中的**播放时长为1~2秒**，对于某些特殊动画效果的点缀音效可以略长
- 由于音效的播放相对比较频繁，为了不影响应用程序的性能，在实际应用中通常使用非压缩的音频格式（AIFF）或者CAF音频格式，从而减低系统在音频解码上的消耗，达到省电的目的
- 使用系统声音服务播放音效，需要导入**AudioToolbox**框架
- 系统声音服务**通过SystemSoundID来播放声音文件**，对于同一个声音文件，可以创建多个SystemSoundID
- 系统声音服务是一套**C语言的框架**
- **为了提高应用程序性能，避免声音文件被重复加载，通常采用单例模式处理系统声音的播放**

加载和播放音效代码

// 加载音效

```
SystemSoundID soundID;
```

```
NSString *path = [_soundBundle pathForResource:name ofType:nil];
```

```
NSURL *url = [NSURL fileURLWithPath:path];
```

```
AudioServicesCreateSystemSoundID((__bridge CFURLRef)(url), &soundID);
```

```
return soundID;
```

// 播放音效

```
AudioServicesPlaySystemSound(soundID);
```

// 播放音效同时振动（当用户静音时，可以通过振动提示用户注意重要操作）

```
AudioServicesPlayAlertSound(soundID);
```

使用AVAudioPlayer播放本地音乐

- 使用AVAudioPlayer可以播放本地音乐，具备以下特点：
 1. 能够预先缓存，提高播放的流畅性（prepareToPlay）
 2. 播放起始时间（playAtTime）
 3. 播放控制（play/pause/stop）
 4. 循环播放次数（numberOfLoops）
 5. 播放音量（volume）
 6. 设置播放速度（enableRate/rate）
 7. 音量监控（meteringEnabled/peakPowerForChannel/averagePowerForChannel）
- 除上述功能外，还可以通过AVAudioPlayer的代理方法监控音乐的播放进度

“流”媒体Stream 网络的音频和视频MediaPlayer



之所以设置三个缓冲区,是为了解决磁盘缓存跟不上播放速度

此方式会每次都从服务器重新下载播放.

应用程序的实际应用场景!

实例化AVAudioPlayer

```
NSError *error = nil;
AVAudioPlayer *player = [[AVAudioPlayer alloc] initWithContentsOfURL:url
error:&error];
if (error) {
    NSLog(@"加载音乐播放器失败 - %@", error.localizedDescription);
    return nil;
}
return player;
```

AVAudioPlayer播放时的音量监控

应用场景：诵读类App的播放时长截取

// 1. 加载朗诵音频

```
_player = [SoundTool audioPlayerWithName:@"上01—东.mp3"];
```

```
_player.numberOfLoops = 0;
```

```
[_player prepareToPlay];
```

```
[_player setDelegate:self];
```

// 2. 允许音量测量

```
[_player setMeteringEnabled:YES];
```

// 3. 实例化音频信息数组

```
_soundInfo = [NSMutableArray array];
```

// 4. 启动时钟

```
_timer = [NSTimer scheduledTimerWithTimeInterval:0.1f target:self  
selector:@selector(timerFired:) userInfo:nil repeats:YES];
```

// 5. 开始播放

```
[_player play];
```


时钟处理方法

// 1. 更新播放器音量

```
[_player updateMeters];
```

// 2. 判断是否需要记录

```
if ([_player averagePowerForChannel:0] > -50.0) {
```

```
    if (_shouldRecord) {
```

```
        NSLog(@"%f", [_player currentTime]);
```

```
        if (![_soundInfo containsObject:@([_player currentTime]))] {
```

```
            [_soundInfo addObject:@([_player currentTime])];
```

```
        }
```

```
    }
```

```
    _shouldRecord = NO;
```

```
} else {
```

```
    _shouldRecord = YES;
```

```
}
```

关于AVAudioSession

- AVAudioSession是一个单例，无需实例化即可直接使用。
AVAudioSession在各种音频环境中起着非常重要的作用
- 针对不同的音频应用场景，需要设置不同的音频会话分类

AVAudioSession的类别 (1)

- **AVAudioSessionCategoryAmbient**
 - 混音播放，例如雨声、汽车引擎等，可与其他音乐一起播放
- **AVAudioSessionCategorySoloAmbient**
 - 后台播放，其他音乐将被停止
- **AVAudioSessionCategoryPlayback**
 - 独占音乐播放
- **AVAudioSessionCategoryRecord**
 - 录制音频
- **AVAudioSessionCategoryPlayAndRecord**
 - 播放和录制音频
- **AVAudioSessionCategoryAudioProcessing**
 - 使用硬件解码器处理音频，该音频会话使用期间，不能播放或录音

AVAudioSession的类别 (2)

类别	输入	输出	与iPOD	遵从静音
AVAudioSessionCategoryAmbient	No	Yes	Yes	Yes
AVAudioSessionCategorySoloAmbi	No	Yes	No	Yes
AVAudioSessionCategoryPlayback	No	Yes	No	No
AVAudioSessionCategoryRecord	Yes	No	No	No
AVAudioSessionCategoryPlayAndR	Yes	Yes	No	No

后台播放音乐（1）——设置后台任务

```
+ (UIBackgroundTaskIdentifier)backgroundPlayerID:(UIBackgroundTaskIdentifier)backTaskId
{
    // 1. 设置并激活音频会话类别
    AVAudioSession *session = [AVAudioSession sharedInstance];
    [session AVAudioSessionCategoryPlayback error:nil];
    [session setActive:YES error:nil];
    // 2. 允许应用程序接收远程控制
    [[UIApplication sharedApplication] beginReceivingRemoteControlEvents];
    // 3. 设置后台任务ID
    UIBackgroundTaskIdentifier newTaskId = UIBackgroundTaskInvalid;
    newTaskId = [[UIApplication sharedApplication] beginBackgroundTaskWithExpirationHandler:nil];
    if (newTaskId != UIBackgroundTaskInvalid && backTaskId != UIBackgroundTaskInvalid) {
        [[UIApplication sharedApplication] endBackgroundTask:backTaskId];
    }
    return newTaskId;
}
```

后台播放音乐（2）——允许后台播放

► Required device capabilities	↕	Array	(1 item)
► Supported interface orientations	↕	Array	(3 items)
▼ Required background modes	↕	Array	(1 item)
Item 0		String	App plays audio or streams audio/video using AirPlay
► Supported interface orientations (...)	↕	Array	(4 items)

后台播放音乐（3）——记录后台任务代号

// 后台播放任务Id

UIBackgroundTaskIdentifier _bgTaskId;

// 设置音频会话，允许后台播放

_bgTaskId = [SoundTool backgroundPlayerID:_bgTaskId];

音频录音机参数字典

```
NSMutableDictionary *setting = [[NSMutableDictionary alloc] init];
```

```
// 音频格式
```

```
[setting setValue:[NSNumber numberWithInt:kAudioFormatAppleIMA4]  
forKey:AVFormatIDKey];
```

```
// 音频采样率
```

```
[setting setValue:[NSNumber numberWithFloat:8000.0] forKey:AVSampleRateKey];
```

```
// 音频通道数
```

```
[setting setValue:[NSNumber numberWithInt:1] forKey:AVNumberOfChannelsKey];
```

```
// 线性音频的位深度
```

```
[setting setValue:[NSNumber numberWithInt:8] forKey:AVLinearPCMBitDepthKey];
```

```
return setting;
```