

[首页](#) [资讯](#) [文章](#) [频道 ▾](#) [资源](#) [小组](#) [❤ 相亲](#)[频道 ▾](#) [➔ 登录](#) [👤 注册](#) [?](#)

加入伯乐在线专栏作者

扩大知名度，还能得 赞赏

[首页](#)[所有文章](#)[业界动态](#)[iOS开发](#)[Swift开发](#)[产品推广](#)[产品设计](#)[更多频道 ▾](#)

- 导航条 -

[伯乐在线](#) > [iOS - 伯乐在线](#) > [所有文章](#) > [iOS开发](#) > iOS 三种录制视频方式

iOS 三种录制视频方式

2016/05/15 · [iOS开发](#) · [视频](#)

分享到：

3 本文作者：[伯乐在线 - deargjn](#)。未经作者许可，禁止转载！欢迎加入伯乐在线 [专栏作者](#)。

随着每一代 iPhone 处理能力和相机硬件配置的提高，使用它来捕获视频也变得更加有意思。它们小巧，轻便，低调，而且与专业摄像机之间的差距已经变得非常小，小到在某些情况下，iPhone 可以真正替代它们。

这篇文章讨论了关于如何配置视频捕获管线 (pipeline) 和最大限度地利用硬件性能的一些不同选择。这里有个使用了不同管线的样例 app，可以在 [GitHub](#) 查看。

UIImagePickerController

目前，将视频捕获集成到你的应用中的最简单的方法是使用 UIImagePickerController。这是一个封装了完整视频捕获管线和相机 UI 的 view controller。

在实例化相机之前，首先要检查设备是否支持相机录制：

```
Objective-C
1 if ([UIImagePickerController
2     isSourceTypeAvailable:UIImagePickerControllerSourceTypeCamera]) {
3     NSArray *availableMediaTypes = [UIImagePickerController
4         availableMediaTypesForSourceType:UIImagePickerControllerSourceTypeCamera];
5     if ([availableMediaTypes containsObject:(NSString *)kUTTypeMovie]) {
6         // 支持视频录制
7     }
8 }
```

然后创建一个 UIImagePickerController 对象，设置好代理便于进一步处理录制好的视频 (比如存到相册) 以及对于用户关闭相机作出响应：

首页 资讯 文章 频道 ▾ 资源 小组 ♡ 相亲 频道 ▾ 登录 注册 ?

```
1 UIImagePickerController *camera = [UIImagePickerController new];
2 camera.sourceType = UIImagePickerControllerSourceTypeCamera;
3 camera.mediaTypes = @[(NSString *)kUTTypeMovie];
4 camera.delegate = self;
```

这是你实现一个功能完善的摄像机所需要写的所有代码。

相机配置

UIImagePickerController 提供了额外的配置选项。

通过设置 cameraDevice 属性可以选择一个特定的相机。这是一个 UIImagePickerControllerCameraDevice 枚举，默认情况下是 UIImagePickerControllerCameraDeviceRear，你也可以把它设置为 UIImagePickerControllerCameraDeviceFront。每次都应事先确认你想要设置的相机是可用的：

Objective-C

```
1 UIImagePickerController *camera = ...
2 if ([UIImagePickerController isCameraDeviceAvailable:UIImagePickerControllerCameraDeviceFront])
3     [camera setCameraDevice:UIImagePickerControllerCameraDeviceFront];
4 }
```

videoQuality 属性用于控制录制视频的质量。它允许你设置一个特定的编码预设，从而改变视频的比特率和分辨率。以下是六种预设：

Objective-C

```
1 enum {
2     UIImagePickerControllerQualityTypeHigh           = 0,
3     UIImagePickerControllerQualityTypeMedium        = 1, // default value
4     UIImagePickerControllerQualityTypeLow           = 2,
5     UIImagePickerControllerQualityType640x480       = 3,
6     UIImagePickerControllerQualityTypeIFrame1280x720 = 4,
7     UIImagePickerControllerQualityTypeIFrame960x540  = 5
8 };
9 typedef NSUInteger UIImagePickerControllerQualityType;
```

前三种为相对预设 (low, medium, high)。这些预设的编码配置会因设备不同而不同。如果选择 high，那么你选定的相机会提供给你该设备所能支持的最高画质。后面三种是特定分辨率的预设 (640×480 VGA, 960×540 iFrame, 和 1280×720 iFrame)。

自定义 UI

就像上面提到的，UIImagePickerController 自带一套相机 UI，可以直接使用。然而，你也可以自定义相机的控件，通过隐藏默认控件，然后创建带有控件的自定义视图，并覆盖在相机预览图层上面：

Objective-C

```
1 UIView *cameraOverlay = ...
2 picker.showsCameraControls = NO;
3 picker.cameraOverlayView = cameraOverlay;
```

然后你需要将你覆盖层上的控件关联上 UIImagePickerController 的控制方法 (比如，startVideoCapture 和 stopVideoCapture)。

AVFoundation

[首页](#) [资讯](#) [文章](#) [频道 ▾](#) [资源](#) [小组](#) [❤ 相亲](#)[频道 ▾](#)[➔ 登录](#)[👤 注册](#)[?](#)

要使用 AVFoundation。

AVFoundation 中关于视频捕获的主要的类是 AVCaptureSession。它负责调配影音输入与输出之间的数据流：

- AVCaptureSession setup

使用一个 capture session，你需要先实例化，添加输入与输出，接着启动从输入到输出之间的数据流：

```
Objective-C
1 AVCaptureSession *captureSession = [AVCaptureSession new];
2 AVCaptureDeviceInput *cameraDeviceInput = ...
3 AVCaptureDeviceInput *micDeviceInput = ...
4 AVCaptureMovieFileOutput *movieFileOutput = ...
5
6 if ([captureSession canAddInput:cameraDeviceInput]) {
7
8     [captureSession addInput:cameraDeviceInput];
9
10 }
11
12 if ([captureSession canAddInput:micDeviceInput]) {
13
14     [captureSession addInput:micDeviceInput];
15
16 }
17
18 if ([captureSession canAddOutput:movieFileOutput]) {
19
20     [captureSession addOutput:movieFileOutput];
21
22 }
23
24 [captureSession startRunning];
```

(为了简单起见，调度队列 (dispatch queue) 的相关代码已经从上面那段代码中省略了。所有对 capture session 的调用都是阻塞的，因此建议将它们分配到后台串行队列中。)

capture session 可以通过一个 sessionPreset

来进一步配置，这可以用来指定输出质量的等级。有 11 种不同的预设模式：

```
Objective-C
1 NSString *const AVCaptureSessionPresetPhoto;
2 NSString *const AVCaptureSessionPresetHigh;
3 NSString *const AVCaptureSessionPresetMedium;
4 NSString *const AVCaptureSessionPresetLow;
5 NSString *const AVCaptureSessionPreset352x288;
6 NSString *const AVCaptureSessionPreset640x480;
7 NSString *const AVCaptureSessionPreset1280x720;
8 NSString *const AVCaptureSessionPreset1920x1080;
9 NSString *const AVCaptureSessionPresetiFrame960x540;
10 NSString *const AVCaptureSessionPresetiFrame1280x720;
11 NSString *const AVCaptureSessionPresetInputPriority;
```

第一个代表高像素图片输出。接下来的九个和之前我们在设置 UIImagePickerControllerController 的 videoQuality 时看到过的 UIImagePickerControllerControllerQualityType 选项非常相似，不同的是，这里有一些额外可用于 capture session 的预设。最后一个 (AVCaptureSessionPresetInputPriority) 代表 capture session 不去控制音频与视频输出设置。而是通过已连接的捕获设备的 activeFormat 来反过来控制 capture session 的输出质量等级。在下一节，我们将会看到更多关于设备和设备格式的细节。

- 输入

AVCaptureDeviceInput 连接上 capture session。

我们可以使用 [AVCaptureDevice devices] 来寻找可用的捕获设备。以 iPhone 6 为例：

Objective-C

```
1 (
2
3     “<AVCaptureFigVideoDevice: 0x136514db0 [Back Camera][com.apple.avfoundation.avcapture
4
5     “<AVCaptureFigVideoDevice: 0x13660be80 [Front Camera][com.apple.avfoundation.avcaptu
6
7     “<AVCaptureFigAudioDevice: 0x174265e80 [iPhone Microphone][com.apple.avfoundation.avc
8
9 )
```

• 视频输入

配置相机输入，需要实例化一个 AVCaptureDeviceInput 对象，参数是你期望的相机设备，然后把它添加到 capture session：

Objective-C

```
1 AVCaptureSession *captureSession = ...
2 AVCaptureDevice *cameraDevice = [AVCaptureDevice defaultDeviceWithMediaType:AVMediaTypeVi
3 NSError *error;
4
5 AVCaptureDeviceInput *cameraDeviceInput = [[AVCaptureDeviceInput alloc] initWithDevice:cc
6
7 if ([captureSession canAddInput:input]) {
8     [captureSession addInput:cameraDeviceInput];
9 }
```

如果上面提到的 capture session 预设列表里能满足你的需求，那你就不需要做更多的事情了。如果不够，比如你想要高的帧率，你将需要配置具体的设备格式。一个视频捕获设备有许多设备格式，每个都带有特定的属性和功能。下面是对于 iPhone6 的后置摄像头的一些例子 (一共有 22 种可用格式)：

Objective-C

	格式	分辨率	FPS	HRSI	FOV	VIS	最大放大比例	Upscales	AF	ISO	SS
1	420v	1280x720	5 - 240	1280x720	54.626	YES	49.12	1.09	1	29.0 - 928	0.000
2											
3	420f	1280x720	5 - 240	1280x720	54.626	YES	49.12	1.09	1	29.0 - 928	0.000
4											
5	420v	1920x1080	2 - 30	3264x1836	58.040	YES	95.62	1.55	2	29.0 - 464	0.000
6											
7	420f	1920x1080	2 - 30	3264x1836	58.040	YES	95.62	1.55	2	29.0 - 464	0.000
8											
9	420v	1920x1080	2 - 60	3264x1836	58.040	YES	95.62	1.55	2	29.0 - 464	0.000
10											
11	420f	1920x1080	2 - 60	3264x1836	58.040	YES	95.62	1.55	2	29.0 - 464	0.000
12											
13	420v	1920x1080	2 - 60	3264x1836	58.040	YES	95.62	1.55	2	29.0 - 464	0.000
14											
15	格式 = 像素格式										
16	FPS = 支持帧数范围										
17	HRSI = 高像素静态图片尺寸										
18	FOV = 视角										
19	VIS = 该格式支持视频防抖										
20	Upscales = 加入数字 upscaling 时的放大比例										
21	AF = 自动对焦系统 (1 是反差对焦, 2 是相位对焦)										
22	ISO = 支持感光度范围										
23	SS = 支持曝光时间范围										
24	HDR = 支持高动态范围图像										

[首页](#) [资讯](#) [文章](#) [频道 ▾](#) [资源](#) [小组](#) [❤ 相亲](#)[频道 ▾](#)[🔑 登录](#)[👤 注册](#)[?](#)

—— 1080P 格式。另外右定安捕获 1920X1080 的力班半的视频时站，定个又付 240 帧每秒的。

配置一个具体设备格式，你首先需要调用 `lockForConfiguration` 来获取设备的配置属性的独占访问权限。接着你简单地使用 `setActiveFormat` 来设置设备的捕获格式。这将会自动把 `capture session` 的预设置为 `AVCaptureSessionPresetInputPriority`。

一旦你设置了预想的设备格式，你就可以在这种设备格式的约束参数范围内进行进一步的配置了。

对于视频捕获的对焦，曝光和白平衡的设置，与图像捕获时一样，具体可参考第 21 期“iOS 上的相机捕捉”。除了那些，这里还有一些视频特有的配置选项。

你可以用捕获设备的 `activeVideoMinFrameDuration` 和 `activeVideoMaxFrameDuration` 属性设置帧速率，一帧的时长是帧速率的倒数。设置帧速率之前，要先确认它是否在设备格式所支持的范围内，然后锁住捕获设备来进行配置。为了确保帧速率恒定，可以将最小与最大的帧时长设置成一样的值：

```
Objective-C
1 NSError *error;
2
3 CMTime frameDuration = CMTimeMake(1, 60);
4 NSArray *supportedFrameRateRanges = [device.activeFormat videoSupportedFrameRateRanges];
5 BOOL frameRateSupported = NO;
6
7 for (AVFrameRateRange *range in supportedFrameRateRanges) {
8     if (CMTimeCompareInline(frameDuration, >=, range.minFrameDuration) &&
9         CMTimeCompareInline(frameDuration, <=, range.maxFrameDuration)) {
10         frameRateSupported = YES;
11     }
12 }
13
14 if (frameRateSupported && [device lockForConfiguration:&error]) {
15     [device setActiveVideoMaxFrameDuration:frameDuration];
16     [device setActiveVideoMinFrameDuration:frameDuration];
17     [device unlockForConfiguration];
18 }
```

视频防抖是在 iOS 6 和 iPhone 4S 发布时引入的功能。到了 iPhone 6，增加了更强劲和流畅的防抖模式，被称为影院级的视频防抖动。相关的 API 也有所改动（目前为止并没有在文档中反映出来，不过可以查看头文件）。防抖并不是在捕获设备上配置的，而是在 `AVCaptureConnection` 上设置。由于不是所有的设备格式都支持全部的防抖模式，所以在实际应用中应事先确认具体的防抖模式是否支持：

```
Objective-C
1 AVCaptureDevice *device = ...;
2
3 AVCaptureConnection *connection = ...;
4
5 AVCaptureVideoStabilizationMode stabilizationMode = AVCaptureVideoStabilizationModeCinematic;
6
7 if ([device.activeFormat isVideoStabilizationModeSupported:stabilizationMode]) {
8
9     [connection setPreferredVideoStabilizationMode:stabilizationMode];
10 }
11 }
```

iPhone 6 的另一个新特性就是视频 HDR（高动态范围图像），它是“高动态范围的视频流，与传统的将不同曝光度的静态图像合成一张高动态范围图像的方法完全不同”，它是内建在传感器中的。有两种方法可以配置视频 HDR：直接将 `capture device` 的 `videoHDEnabled` 设置为启用或禁用，或者使用 `automaticallyAdjustsVideoHDEnabled` 属性来留给系统处理。

技术参考：iPhone 6 和 iPhone Plus 的新 AV Foundation 相机特性

[首页](#) [资讯](#) [文章](#) [频道 ▾](#) [资源](#) [小组](#) [❤ 相亲](#)[频道 ▾](#)[🔑 登录](#)[👤 注册](#)[?](#)

之前展示的捕获设备列表里面只有一个音频设备，你可能觉得奇怪，毕竟 iPhone 6 有 3 个麦克风。然而因为有时会放在一起使用，便于优化性能，因此可能被当做一个设备来使用。例如在 iPhone 5 及以上的手机录制视频时，会同时使用前置和后置麦克风，用于定向降噪。

Technical Q&A: AVAudioSession – Microphone Selection

大多数情况下，设置成默认的麦克风配置即可。后置麦克风会自动搭配后置摄像头使用（前置麦克风则用于降噪），前置麦克风和前置摄像头也是一样。

然而想要访问和配置单独的麦克风也是可行的。例如，当用户正在使用后置摄像头捕获场景的时候，使用前置麦克风来录制解说也应是可能的。这就要依赖于 AVAudioSession。为了变更要访问的音频，audio session 首先需要设置为支持这样做的类别。然后我们需要遍历 audio session 的输入端口和端口数据来源，来找到我们想要的麦克风：

Objective-C

```
1 // 配置 audio session
2 AVAudioSession *audioSession = [AVAudioSession sharedInstance];
3 [audioSession setCategory:AVAudioSessionCategoryPlayAndRecord error:nil];
4 [audioSession setActive:YES error:nil];
5
6 // 寻找期望的输入端口
7 NSArray* inputs = [audioSession availableInputs];
8 AVAudioSessionPortDescription *builtInMic = nil;
9
10 for (AVAudioSessionPortDescription* port in inputs) {
11     if ([port.portType isEqualToString:AVAudioSessionPortBuiltInMic]) {
12         builtInMic = port;
13         break;
14     }
15 }
16
17 // 寻找期望的麦克风
18
19 for (AVAudioSessionDataSourceDescription* source in builtInMic.dataSources) {
20
21     if ([source.orientation isEqualToString:AVAudioSessionOrientationFront]) {
22         [builtInMic setPreferredDataSource:source error:nil];
23         [audioSession setPreferredInput:builtInMic error:&error];
24         break;
25     }
26 }
```

除了设置非默认的麦克风配置，你也可以使用 AVAudioSession 来配置其他音频设置，比如音频增益和采样率等。

- 访问权限

有件事你需要记住，访问相机和麦克风需要先获得用户授权。当你给视频或音频创建第一个 AVCaptureDeviceInput 对象时，iOS 会自动弹出一对话框，请求用户授权，但你最好还是自己实现下。之后你就可以在还没有被授权的时候，使用相同的代码来提示用户进行授权。当用户未授权时，对于录制视频或音频的尝试，得到的将是黑色画面和无声。

- 输出

输入配置完了，现在把我们的注意力转向 capture session 的输出。

AVCaptureMovieFileOutput

将视频写入文件，最简单的选择就是使用 AVCaptureMovieFileOutput 对象。把它作为输出添加到 capture

[首页](#) [资讯](#) [文章](#) [频道 ▾](#) [资源](#) [小组](#) [❤ 相亲](#)[频道 ▾](#)[➔ 登录](#)[👤 注册](#)[?](#)

Objective-C

```
1 AVCaptureMovieFileOutput *movieFileOutput = [AVCaptureMovieFileOutput new];
2
3 if([captureSession canAddOutput:movieFileOutput]){
4     [captureSession addOutput:movieFileOutput];
5 }
6
7 // 开始录制
8
9 // 开始录制
10
11 NSURL *outputURL = ...
12
13 [movieFileOutput startRecordingToOutputFileURL:outputURL recordingDelegate:self];
```

当实际的录制开始或停止时，想要接收回调的话就必须需要一个录制代理。当录制停止时，输出通常还在写入数据，等它完成之后会调用代理方法。

AVCaptureMovieFileOutput 有一些其他的配置选项，比如在某段时间后，在达到某个指定的文件尺寸时，或者当设备的最小磁盘剩余空间达到某个阈值时停止录制。如果你还需要更多设置，比如自定义视频音频的压缩率，或者你想要在写入文件之前，处理视频音频的样本，那么你需要一些更复杂的操作。

AVCaptureDataOutput 和 AVAssetWriter

如果你想要对影音输出有更多的操作，你可以使用 AVCaptureVideoDataOutput 和 AVCaptureAudioDataOutput 而不是我们上节讨论的 AVCaptureMovieFileOutput。

这些输出将会各自捕获视频和音频的样本缓存，接着发送到它们的代理。代理要么对采样缓冲进行处理（比如给视频加滤镜），要么保持原样传送。使用 AVAssetWriter 对象可以将样本缓存写入文件：

Using an AVAssetWriter

配置一个 asset writer 需要定义一个输出 URL 和文件格式，并添加一个或多个输入来接收采样的缓冲。我们还需要将输入的 expectsMediaInRealTime 属性设置为 YES，因为它们需要从 capture session 实时获得数据。

Objective-C

```
1 NSURL *url = ...;
2
3 AVAssetWriter *assetWriter = [AVAssetWriter assetWriterWithURL:url fileType:AVFileTypeMPEG4];
4
5 AVAssetWriterInput *videoInput = [[AVAssetWriterInput alloc] initWithMediaType:AVMediaTypeVideo];
6
7 videoInput.expectsMediaDataInRealTime = YES;
8
9 AVAssetWriterInput *audioInput = [[AVAssetWriterInput alloc] initWithMediaType:AVMediaTypeAudio];
10
11 audioInput.expectsMediaDataInRealTime = YES;
12
13 if ([assetWriter canAddInput:videoInput]) {
14     [assetWriter addInput:videoInput];
15 }
16
17 if ([assetWriter canAddInput:audioInput]) {
18     [assetWriter addInput:audioInput];
19 }
20
21 }
```

[首页](#) [资讯](#) [文章](#) [频道 ▾](#) [资源](#) [小组](#) [❤ 相亲](#)[频道 ▾](#)[🔑 登录](#)[👤 注册](#)

在上面的示例代码中，我们将 asset writer 的 outputSettings 设置为 nil。这就意味着附加上来的样本不会再被重新编码。如果你确实想要重新编码这些样本，那么需要提供一个包含具体输出参数的字典。关于音频输出设置的键值被定义在这里，关于视频输出设置的键值定义在这里。

为了更简单点，AVCaptureVideoDataOutput 和 AVCaptureAudioDataOutput 分别带有 recommendedVideoSettingsForAssetWriterWithOutputFileType: 和 recommendedAudioSettingsForAssetWriterWithOutputFileType: 方法，可以生成与 asset writer 兼容的带有全部键值对的字典。所以你可以通过在这个字典里调整你想要重写的属性，来简单地定义你自己的输出设置。比如，增加视频比特率来提高视频质量等。

或者，你也可以使用 AVOutputSettingsAssistant 来配置输出设置的字典，但是从我的经验来看，使用上面的方法会更好，它们会提供更实用的输出设置，比如视频比特率。另外，AVOutputSettingsAssistant 似乎存在一些缺点，例如，当你改变希望的视频的帧速率时，视频的比特率并不会改变。

实时预览

当使用 AVFoundation 来做图像捕获时，我们必须提供一套自定义的用户界面。其中一个关键的相机交互组件是实时预览图。最简单的实现方式是通过把 AVCaptureVideoPreviewLayer 对象作为一个 sublayer 加到相机图层上去：

Objective-C

```
1 AVCaptureSession *captureSession = ...;
2 AVCaptureVideoPreviewLayer *previewLayer = [AVCaptureVideoPreviewLayer layerWithSession:captureSession];
3 UIView *cameraView = ...;
4 previewLayer.frame = cameraView.bounds;
5 [cameraView.layer addSublayer:previewLayer];
```

如果你想要更进一步操作，比如，在实时预览图加滤镜，你需要将 AVCaptureVideoDataOutput 对象加到 capture session，并且使用 OpenGL 展示画面，具体可查看该文“iOS 上的相机捕捉”

总结

有许多不同的方法可以给 iOS 上的视频捕获配置管线，从最直接的 UIImagePickerController，到精密配合的 AVCaptureSession 与 AVAssetWriter。如何抉择取决于你的项目要求，比如期望的视频质量和压缩率，或者是你想要展示给用户的相机控件。

加入伯乐在线专栏作者。扩大知名度，还能得赞赏！详见《[招募专栏作者](#)》

打赏支持作者写出更多好文章，谢谢！

¥ [打赏作者](#)

👍 1 赞

🔖 5 收藏

💬 评论

关于作者：dearqjn

[首页](#) [资讯](#) [文章](#) [频道 ▾](#) [资源](#) [小组](#) [❤ 相亲](#)[频道 ▾](#) [登录](#) [注册](#) [?](#)

ios职业开发者一枚，喜交好友，热爱运动，为人友善，让我们一起探索IT的魅力，早日登顶 [个人主页](#) · [我的文章](#) · [1](#) · [🔍](#) [🔔](#)

加入伯乐在线专栏作者
扩大知名度，还能得 **赞赏**



相关文章

- [在 iOS 上捕获视频](#)
- [终极指南：如何为iOS8应用制作预览视频](#)
- [iOS视频边下边播-缓存播放数据流](#)
- [学习Swift开发的免费视频教程列表](#)
- [iPhone和iPad上最棒的免费视频播放器](#)
- [结合 Reveal 谈谈 MBProgressHUD 的用法](#)
- [iOS 上的相机捕捉](#)
- [iOS开发之窥探UICollectionViewController\(一\)：Ready Your CollectionViewController](#)
- [滑动的维度](#)
- [iOS9 每日学习第2天：UI Testing](#)

可能感兴趣的话题

- [各位程序员们，会不会介意女朋友工资高低 · 278](#)
- [webmap example -Kenneth's 2016/5/1 Holiday](#)
- [大家五一都想着去干啥? · 31](#)
- [酷狗音乐（繁星网）PHP岗位笔试题 · 6](#)
- [男朋友天天加班到九点好奇想知道各位一般几点下班 · 73](#)
- [唯品会2017校园招聘笔试题\(程序错误问题\) · 17](#)
- [为什么技术出身要比非技术出身人员的创业成功率要低得多 · 12](#)
- [补缴社保这事儿，谁遇到过，据说很麻烦 · 10](#)
- [一个30岁女生，是继续在岗位上为别人工作？还是自己出来干点什么? · 78](#)
- [经常被人问起：你的职业生涯规划是什么 · 10](#)

« 动画篇-碎片动画

[国内 90%以上的 iOS 开发者，对 APNs 的认识都是错的 »](#)

[登录后评论](#)[新用户注册](#)[直接登录](#)

[首页](#) [资讯](#) [文章](#) [频道 ▾](#) [资源](#) [小组](#) [❤ 相亲](#)[频道 ▾](#)[🔑 登录](#)[👤 注册](#)[搜索](#)[iOS小组话题](#)[我有新话题 💬](#)[iOS程序媛求指点：该继续做ios还是...](#)[阳光](#) 发起 • 4 回复[iOS如何实现闹钟的周期提醒功能？](#)[海市蜃楼](#) 发起 • 1 回复[如何看待一个本科不是计算机专业的妹...](#)[Jc 鑫16](#) 发起 • 238 回复[iOS项目求救](#)[晴天](#) 发起 • 12 回复[DTCoreText 编译问题](#)[Y小鬼CQ](#) 发起[怎么在storyboard中关掉autoresizing](#)[晴天](#) 发起 • 2 回复

- [本月热门iOS文章](#)
- [年度热门](#)
- [热门标签](#)

[0 Github 上的 iOS 开源项目](#)[1 工厂模式三部曲：工厂方法模式](#)

[首页](#) [资讯](#) [文章](#) [频道 ▾](#) [资源](#) [小组](#) [❤ 相亲](#)[频道 ▾](#)[🔑 登录](#)[👤 注册](#)[?](#)[3 工厂模式三部曲：抽象工厂模式](#)[4 iOS开发经验总结](#)[5 详解 Objective-C 中的 Runtime](#)[6 MVVM架构的一次实践，重写iOS...](#)[7 移动开发必读书单](#)[8 iOS开发大神必备的Xcode插件](#)[9 你真的会用 SDWebImage?](#)[业界热点资讯](#)[更多 >](#)[Java版权案：法官和陪审团貌似并不理解 API](#)
2 天前 · 👍 14 · 💬 1[Google宣布6月弃用SSLv3和RC4：Gmail与SMTP服...](#)
2 小时前 · 👍 2[Facebook 内部是如何使用 JavaScript 和 GraphQ...](#)
6 小时前 · 👍 3[Java版权案：安卓之父作证时被甲骨文律师问了4个多小时](#)
2 天前 · 👍 8[Java版权案：甲骨文联席CEO否认收购Sun是为了提起诉讼](#)

首页 资讯 文章 频道 资源 小组 相亲

频道 登录 注册 ?

iOS工具资源

更多资源 »



[JBChartView: iOS图表库, 包含折线图和直方图](#)
[iOS, 图表](#)



[PNChart: 一个简单漂亮的iOS图表库](#)
[iOS, 图表](#)



[JTChartView: 完全可定制的轻量级图表解决方案](#)
[iOS, 图表](#)



[ios-charts: 一个强大的iOS图表框架](#)
[iOS, 图表](#)



[SDURLCache: 为iPhone/iPad应用提供本地磁盘缓存的U...](#)
[iOS, 缓存](#)



最新评论



Re: [小结 iOS 中的 copy](#)
不错



Re: [小结 iOS 中的 copy](#)
哪里啊, 还是菜鸟一只, 需要学习的东西实在太多了

[首页](#) [资讯](#) [文章](#) [频道 ▾](#) [资源](#) [小组](#) [❤ 相亲](#)[频道 ▾](#)[🔑 登录](#)[👤 注册](#)

Re: [AFNetworking 3.0.4 使用](#)

楼主有没有https的请求



Re: [AFNetworking 3.0.4 使用](#)

拜谢楼主，一鞠躬



Re: [小结 iOS 中的 copy](#)

大二就对iOS有如此的见解 着实让人佩服



Re: [iOS，如何模仿一个App](#)

请问这个抓包必须是苹果手机才可以么？



Re: [如何打造一个高性能Hybrid App](#)

当把UIWebView换成WKWebView时无法拦截ajax请求，怎么处理？



Re: [iOS中集成ijkplayer视频直播框架](#)

谢谢 很详细

关于iOS频道

iOS频道分享iOS和Swift开发，应用设计和推广，iOS相关的行业动态。

快速链接

[问题反馈与求助 »](#)

[iOS工具资源 »](#)

[iOS技术话题 »](#)

关注我们

新浪微博: [@iOS大全](#)

RSS: [订阅地址](#)

推荐微信号



iOS大全



UI设计达人



程序员的那些事

合作联系

Email: bd@Jobbole.com

QQ: 2302462408 (加好友请注明来意)

更多频道

[小组](#) – 好的话题、有启发的回复、值得信赖的圈子

