

# 为什么不推荐使用 PHPicker

RayJiang97 知识小集 今天



作者 | RayJiang97

来源 | 掘金，点击“阅读原文”查看作者更多文章

WWDC 20 过去已经有好几个月了，iOS 14 正式版也发布了，这篇文章写的有点晚了，因为有些 API 没有彻底弄懂，所以一直拖到了现在（奇怪我怎么感觉去年也说过一样的话: doge）其实过了这么多个月，大家应该或多或少都看过一些别人写文

章，介绍相册的变化，介绍 PHPicker，但是有一些点没讲清楚，比如怎么用 PHPicker 获取视频？PHPicker 有什么不足？那么下面让我们一起来看看什么是 PHPicker 以及 iOS 14 相册有什么新的变化。

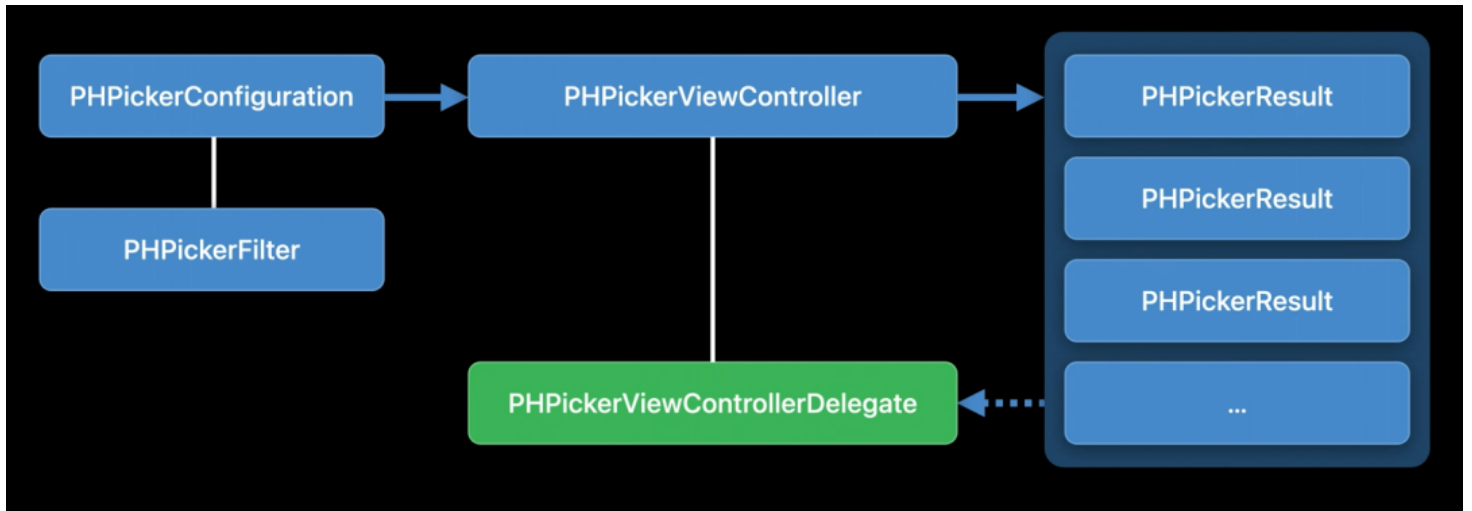
# PHPicker

---

iOS 14 中系统新增了一个图片选择器 PHPicker，官方建议使用 PHPicker 来替代原有的 API 进行图片选择，下面我们来看看 PHPicker 的优点：

- 支持多选
- 支持搜索
- 独立的进程
- 内置隐私
  - 不需要直接访问用户相册
  - 不会弹出访问相册提示
  - 仅提供用户选择的照片和视频（App 无法获取其他照片）

# 如何调用 PHPicker



我们先来看下 PHPicker 的流程图，首先声明 PHPickerConfiguration，进行配置，再传给 PHPickerViewController，完成调用环节，代码如下：

```
var config = PHPickerConfiguration()  
// 可选的资源数量，0表示不设限制，默认为1  
config.selectionLimit = 0  
// 可选的资源类型  
// 只显示图片（注：images 包含 livePhotos）  
config.filter = .images  
// 显示 Live Photos 和视频（注：livePhotos  
config.filter = .any(of: [.livePhotos, .  
// 如果要获取视频，最好设置该属性，避免系统对视频  
config.preferredAssetRepresentationMode  
  
let picker = PHPickerViewController(conf
```

```
picker.delegate = self
present(picker, animated: true, completi
```

## 处理 **PHPicker** 的回调

PHPicker 的代理方法只有一个，声明如下：

```
@available(iOS 14, *)
public protocol PHPickerViewControllDe
    func picker(_ picker: PHPickerViewCo
}
```

注意：取消选择也会触发代理方法，会返回空的 results。

## 如何获取照片

PHPicker 获取图片的方法还是比较简单的，代码如下：

```
func picker(_ picker: PHPickerViewContro
    // 首先需要 dismiss picker
    picker.dismiss(animated: true, compl
    for result in results {
```

```

// 判断类型是否为 UIImage
if result.itemProvider.canLoadOb
    // 确认类型后，调用 loadObject
    result.itemProvider.loadObje
    // 回调结果是在异步线程，展示
    if let image = data as?
        DispatchQueue.main.a
            self.showImage(i
        }
    }
}
}
}
}
}
}
}
}

```

## 如何获取视频

其他文章中都没有介绍 PHPicker 如何获取视频，其实获取视频的方法在官方的 Demo 以及视频中都没有介绍，这也是我迟迟没有写文章的原因，因为之前我也不知道怎么获取，那么下面让我们一起来看下怎么获取视频。

```

func picker(_ picker: PHPickerViewContro
    // 首先需要 dismiss picker
    picker.dismiss(animated: true, compl

```

```

for result in results {
    if result.itemProvider.canLoadOb
        // 判断类型是否为 UIImage
        ...
    } else {
        // 类型为 Video
        // 调用 loadFileRepresentatic
        // 这里 Type Identifier 我们用
        result.itemProvider.loadFile
            if let error = error {
                print(error)
                return
            }
        // 系统会将视频文件存放到 tm
        // 我们必须在这个回调结束前，
        // 所以一定要确定拷贝结束后，
        // 另外不用担心视频过大而导致
        guard let url = url else
        let fileName = "\ (Int(Da
        let newUrl = URL(fileURL
        try? FileManager.default
        DispatchQueue.main.async
            self.playVideo(newUr
        }
    }
}
}
}
}

```

注意：如果你遇到了部分资源可以加载，而部分资源无法加载的话，那么有可能是设备没有连接到 iCloud，只能加载本地资源，而无法加载 iCloud 上的资源。

## 被废弃的 **API**

有新的 API 出现，也会有一些 API 被废弃，在 UIImagePickerController 中有三个 sourceType，现在有两个被废弃，只留下 camera。

```
public enum SourceType : Int {  
    @available(iOS, introduced: 2, depre  
    case photoLibrary = 0  
    case camera = 1  
    @available(iOS, introduced: 2, depre  
    case savedPhotosAlbum = 2  
}
```

另外 AssetsLibrary 早在几年前被废弃，如果还在使用 AssetsLibrary 请尽快使用新的 API。

## **PHPicker** 的缺点

为什么不推荐使用 PHPicker，虽然说 PHPicker 有一些优点，但同时也有一些缺点：

- 加载 iCloud 资源时没有进度回调
- 不支持图片编辑（比如选择头像要将图片裁剪成正方形）

## 有没有其他的解决方案？

有的，如果你不能接受 PHPicker 的缺点，同时又想保护用户的隐私，那么可以考虑使用我和我的小伙伴做的第三方图片选择框架 AnyImageKit，目前有 Picker、Editor、Capture 三个模块，支持图片/视频选择、编辑、拍摄功能，支持 SPM、CocoaPods 方式引入。

接下来我会结合 AnyImageKit 中的案例一起来介绍 iOS 14 相册的改动。

## 新增权限

---



iOS 14 中相册新增了一个“Limited Photos Library”模式，在授权时多了一个“选择照片”的选项。点击之后系统会弹出 UIImagePickerController 用户可以选择指定的照片让 App 读取。



当用户选择了 limited 模式后，系统将在 App 每次启动后首次触发相册时弹出提示，允许用户修改需要授权给 App 的照片。



当然这个弹窗是可以关闭的，如果你希望手动控制 PHPickerController 弹出的时机也是有办法的。

我们需要在 Info.plist 中添加 PHPhotoLibraryPreventAutomaticLimitedAccessAlert 字段，并设置为 YES，设置后系统将不再弹出访问提示。

然后我们可以在合适的时机调用以下这个 API 来推出 PHPickerController。

```
let viewController = self
PHPhotoLibrary.shared().presentLimitedLi
```

我们可以看到，当用户选择 limited 模式后，底部出现了一段提示：“无法查看相册全部照片，点击选择更多照片”。当点击这个提示后，将会推出 PHPickerController，此时用户可以修改授权给 App 的照片。同时我们会监听相册的变化，当用户修改授权的照片后，会立即刷新相册，用户可以继续进行选择照片的流程。

## 监听相册变化

配合手动调用 PHPickerController，我们还需要监听用户添加/删除了哪些照片。

注意：这组 API 并不是新出的，从 iOS 8 开始就支持了。

```
let viewController = self
// 开始监听
```

```
PHPhotoLibrary.shared().register(viewCon  
// 结束监听  
PHPhotoLibrary.shared().unregisterChange
```

处理监听回调：

```
/// 回调方法  
func photoLibraryDidChange(_ changeInsta  
    // Your code  
}
```

由于这是一组旧的 API，所以就不介绍细节了（比如判断是新增还是删除），感兴趣的朋友可以去了解一下。

## 新增的 API

### PHAccessLevel

在 iOS 14 中新增了权限等级枚举 PHAccessLevel，有两个 case，分别是“只读”和“读写”。

```
public enum PHAccessLevel : Int {  
    case addOnly = 1  
    case readWrite = 2  
}
```

对应新增了一组获取/查看权限的 API:

```
let level: PHAccessLevel = .readWrite  
// 获取权限  
PHPhotoLibrary.requestAuthorization(for:  
    // Your code  
)  
// 查看权限  
let status: PHAuthorizationStatus = PHPh
```

## PHAuthorizationStatus

PHAuthorizationStatus 新增了一个 case limited。

```
public enum PHAuthorizationStatus : Int  
    case notDetermined = 0  
    case restricted = 1  
    case denied = 2  
    case authorized = 3
```

```
@available(iOS 14, *)  
case limited = 4  
}
```

当用户在授权时选择了“选择照片”的选项时：

- 使用新 API 将会返回 limited case
- 使用旧 API 将会返回 authorized case

注意：limited case 仅在 PHAccessLevel = .readWrite 时会返回。

## 总结

---

新出的 PHPicker 个人觉得一般，如果对 Picker 要求不多的朋友可以考虑使用。然后是新出的“Limited Photos Library”模式，这个非常棒，如果有自定义 Picker 的朋友建议跟进一下。如果没有自定义 Picker 的朋友可以考虑使用我们做的第三方图片选择框架 AnyImageKit。

以上就是 iOS 14 相册的改动以及 PHPicker 的全部内容，如有错误欢迎指出。

## 参考资料

---

[1] <https://developer.apple.com/videos/play/wwdc2020/10652/>

[2] <https://developer.apple.com/videos/play/wwdc2020/10641/>

[3] <https://developer.apple.com/forums/thread/652695>

阅读原文