


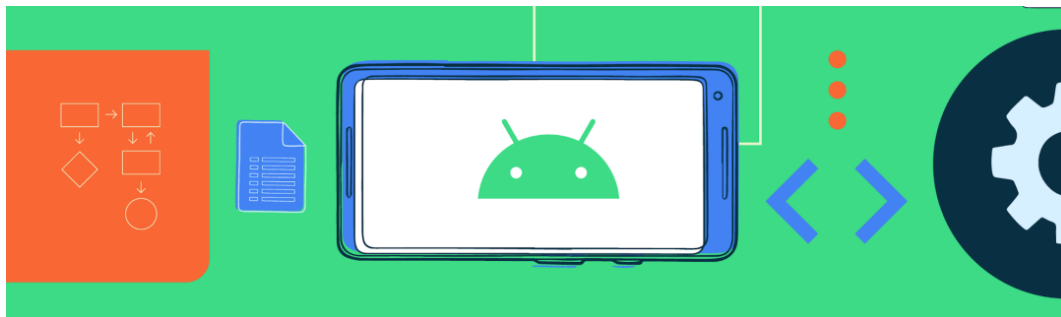
Android Jetpack CameraX 库 Beta 版正式发布！



谷歌开发者 
已认证的官方帐号

已关注

5 人赞同了该文章



CameraX 是一个 Jetpack 支持库，旨在帮助您简化相机应用的开发工作。它提供一致且易于使用的 API 界面，适用于大多数 Android 设备，并可向后兼容至 Android 5.0 (API 级别 21)。

CameraX 的 Beta 版本正式发布，我们向为此作出贡献的全体开发者社区成员致谢，这是我们共同努力的结果。这次的版本将为开发者呈现一个基础的相机 API，未来会一如既往地提供与兼容性有关或其他 bug 的修复程序和解决方法，助力您在多台设备上打造一致的开发者体验。

除了在自动化测试农场扩大测试范围，有很多应用也在生产环境中使用了 CameraX 库，这使得我们有信心向各位开发者推荐 CameraX，推荐大家开始尝试并将基于 CameraX 的应用发布到应用商店。我们也预想到可能会出现一些 bug，但是我们会及时处理各种问题，来帮助大家实现最佳的相机使用体验。如果您需要帮助，可以通过[Google Groups 讨论群组](#)或者[新建一个 issue](#)来向我们反馈。

本文我们会向大家简要介绍 Android Jetpack 组件中的 CameraX 库，以及 Beta 版本中的更新。Beta 版本意味着该版本的 Jetpack 库已经可以面向生产环境，但仍会有一些非关键的 bug 的存在。

请参阅[官方文档](#)以及[官方示例代码](#)以获取最新资源，并加入我们的[线上开发者社区](#)。

CameraX logo

CameraX 概览

首先，让我们通过视频快速了解一下[CameraX 是什么](#)，如果您已经熟悉 CameraX，那么可以跳过本节。

<https://v.qq.com/x/page/m08905yyh0l.html>

[v.qq.com](#)



作为 Jetpack 库，CameraX 不仅仅兼容 API level 21 及以上的 Android 设备，并且与各种硬件设备相兼容，开发者们无需关心具体的设备形态、相机配置或者设备的实现细节。

生命周期感知能力

CameraX 的一个核心属性是它可感知生命周期。相较于手动打开或关闭相机，我们将相机的调用交给具有生命周期的对象，当这些对象 (如 Activity、Fragment 等) 启动或者停止的时候，同时会启动或者停止相机。

CameraX 用例

Beta 版本的 CameraX 所提供的用例主要包含:

- Preview: 用于显示一个取景器 (viewfinder)，来预览当前相机所拍到的图像；
- ImageAnalysis: 用于解析相机采集到的数据流；
- ImageCapture: 用于拍摄高质量照片。

△ 人像模式

Beta 版本里的改变

和我们上次聊 CameraX 时相比，Beta 版本有一些新的变化，这些变化可以在[官方文档](#)、[官方示例代码](#)或者[线上开发者社区](#)里找到。

初始化

开发者需要通过[ProcessCameraProvider](#)来初始化 CameraX，代码示例如下：

```
val cameraProviderFuture: ListenableFuture<ProcessCameraProvider> =
    ProcessCameraProvider.getInstance(context)
// 当 future 的值就绪以后，会触发该监听器
cameraProviderFuture.addListener(Runnable {
    // 现在可以保证这里的 cameraProvider 是可用的
    val cameraProvider = cameraProviderFuture.get()
    ...
}, executor)
```

相机选取

在本用例中，选择所需的相机是通过 CameraSelector 实现的。首先使用一组可选的约束参数创建一个 CameraSelector 对象，然后 CameraX 会选择最符合约束条件的可用相机：

```
val cameraSelector = CameraSelector.Builder()
    .requireLensFacing(CameraSelector.LENS_FACING_BACK)
    .build()
```

如果您希望能够启用前置或者后置相机，可以使用这两个静态字段
CameraSelector.DEFAULT_FRONT_CAMERA 或者 CameraSelector.DEFAULT_BACK_CAMERA。

请在 CameraSelector 文档查阅更多资料:

<https://developer.android.google.cn/reference/androidx/camera/core/Camera...>

developer.android.google.cn



相机预览

这并不属于 Beta 版本推出的内容，[PreviewView](#) 已经于 alpha08 版本中加入，通过 PreviewView 来实现相机预览的功能是我们推荐的使用方式。要使用它，首先将它添加到您的 XML 布局文件中:

```
<androidx.camera.view.PreviewView
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

然后确保您的 Preview 用例使用的是 PreviewView 的 surface 对象:

```
val preview: Preview = ...
val viewFinder: PreviewView = ... // 比如 findViewById()
preview.setSurfaceProvider(viewFinder.previewSurfaceProvider)
```

如果您想使用自己的 Surface 对象 (比如通过 TextureView 得到)，那么您需要自行实现 [surface provider](#)，同时确保能够正确处理尺寸和设备朝向，有时这两项处理起来会很棘手。

相机控制

您可以使用 [CameraProcessProvider.bindToLifecycle\(\)](#) 返回的相机对象，来查询和修改相机的某些功能，如对焦、变焦和闪光灯。您可以通过 CameraInfo 对象获取内含相机当前状态的 LiveData 对象，如以下代码示例所示:

```
val camera = cameraProvider.bindToLifecycle(...)
val zoomState: LiveData<ZoomState> = camera.cameraInfo.zoomState
val torchState: LiveData<Int> = camera.cameraInfo.torchState
```

反之，您可以通过相应的 CameraControl 对象来控制这些值:

```
val camera = cameraProvider.bindToLifecycle(...)
camera.cameraControl.enableTorch(true)
camera.cameraControl.setLinearZoom(0.5f)
camera.cameraControl.startFocusAndMetering(...)
```

下一步

相机开发并非易事，我们会从以下几个方面提高 CameraX 易用性:

- 继续 CameraX 测试套件的开发;
- 在自动化测试环境中增加新的设备。

开始使用 CameraX Beta 并和我们一起改善用户在 Android 上的相机体验吧。



详细了解 CameraX，请参阅以下资源：

- 官方文档

<http://developer.android.google.cn/camerax>

developer.android.google.cn



- 官方示例代码

[android/camera-samples](https://github.com/android/camera-samples)

github.com



- Google Groups 讨论群租

<https://groups.google.com/a/android.com/forum/#!forum/camerax-developers>

groups.google.com



我们也在去年的 Android 开发者峰会中与大家分享了 CameraX 库的介绍和用法，请查看下面的视频回顾：

- 腾讯视频链接

CameraX 的最新进展 | ADS 中文字幕
视频_腾讯视频

v.qq.com



- Bilibili 视频链接

CameraX 的最新进展 |
AndroidDevSummit 中文字幕视频_...

www.bilibili.com



[点击这里](#)了解更多 CameraX 相关内容

发布于 04-01

[Android](#) [Android 开发](#) [Android 应用](#)

▲ 赞同 5 ▼

● 添加评论

➦ 分享

♥ 喜欢

★ 收藏

📄 申请转载

...

推荐阅读

Android Q Beta 正式发布 | 扁平形，安于内

多动行业在 2019 年创新不断，随着 5G 时代的到来与折叠屏技术的成熟，智能设备正在迈向未来新时代，而 Android 更是处在颠覆创新的风口浪尖。通过与生态圈伙伴们的深度合作，我们从软件到...

谷歌开发者



觉得 Android 的分享功能很难用？它帮你解决这个问题

少数派

Android Q Beta 2 已上线！我们来看看都做了哪些更新

我们在几周前发布了 Android Q 的首个 Beta 版，让大家尝鲜体验了下个版本的 Android 系统。除用户隐私方面的特性更新之外，Android Q 还为开发者们带去了许多新功能，如折叠屏增强项、新网...

谷歌开发者



Android开发之简单登录界面

没想好昵称呵

还没有评论

写下你的评论...