package

android.hardware.camera2

Added in API level 21

The android.hardware.camera2 package provides an interface to individual camera devices connected to an Android device. It replaces the deprecated Camera (/reference/android/hardware/Camera.html) class.

This package models a camera device as a pipeline, which takes in input requests for capturing a single frame, captures the single image per the request, and then outputs one capture result metadata packet, plus a set of output image buffers for the request. The requests are processed in-order, and multiple requests can be in flight at once. Since the camera device is a pipeline with multiple stages, having multiple requests in flight is required to maintain full framerate on most Android devices.

To enumerate, query, and open available camera devices, obtain a CameraManager (/reference/android/hardware/camera2/CameraManager.html) instance.

Individual CameraDevices (/reference/android/hardware/camera2/CameraDevice.html) provide a set of static property information that describes the hardware device and the available settings and output parameters for the device. This information is provided through the CameraCharacteristics (/reference/android/hardware/camera2/CameraCharacteristics.html) object, and is available through getCameraCharacteristics(String) (/reference/android/hardware/camera2/CameraManager.html#getCameraCharacteristics(java.lang.String))

To capture or stream images from a camera device, the application must first create a camera capture session (/reference/android/hardware/camera2/CameraCaptureSession.html) with a set of output Surfaces for use with the camera device, with createCaptureSession(List, CameraCaptureSession.StateCallback, Handler) (/reference/android/hardware/camera2/CameraDevice.html#createCaptureSession(java.util.List<android.view.Surface>, android.hardware.camera2.CameraCaptureSession.StateCallback, android.os.Handler)). Each Surface has to be pre-configured with an appropriate size and format (/reference/android/hardware/camera2/params/StreamConfigurationMap.html) (if applicable) to match the sizes and formats available from the camera device. A target Surface can be obtained from a variety of classes, including SurfaceView (/reference/android/view/SurfaceView.html), SurfaceTexture (/reference/android/graphics/SurfaceTexture.html) via Surface(SurfaceTexture) (/reference/android/view/Surface.html#Surface(android.graphics.SurfaceTexture)), MediaCodec (/reference/android/media/MediaCodec.html), MediaRecorder (/reference/android/media/MediaRecorder.html), Allocation (/reference/android/renderscript/Allocation.html), and ImageReader (/reference/android/media/ImageReader.html).

Generally, camera preview images are sent to SurfaceView

(/reference/android/view/SurfaceView.html) or TextureView (/reference/android/view/TextureView.html) (via its SurfaceTexture (/reference/android/graphics/SurfaceTexture.html)). Capture of JPEG images or RAW buffers for DngCreator (/reference/android/hardware/camera2/DngCreator.html) can be done with ImageReader (/reference/android/media/ImageReader.html) with the JPEG (/reference/android/graphics/ImageFormat.html#JPEG) and RAW_SENSOR (/reference/android/graphics/ImageFormat.html#RAW_SENSOR) formats. Application-driven processing of camera data in RenderScript, OpenGL ES, or directly in managed or native code is best done through Allocation (/reference/android/renderscript/Allocation.html) with a YUV Type (/reference/android/renderscript/Type.html), SurfaceTexture (/reference/android/graphics/SurfaceTexture.html), and ImageReader (/reference/android/media/ImageReader.html) with a YUV_420_888 (/reference/android/graphics/ImageFormat.html#YUV_420_888) format, respectively.

The application then needs to construct a CaptureRequest (/reference/android/hardware/camera2/CaptureRequest.html), which defines all the capture parameters needed by a camera device to capture a single image. The request also lists which of the configured output Surfaces should be used as targets for this capture. The CameraDevice has a factory method (/reference/android/hardware/camera2/CameraDevice.html#createCaptureRequest(int)) for creating a request builder (/reference/android/hardware/camera2/CaptureRequest.Builder.html) for a given use case, which is optimized for the Android device the application is running on.

Once the request has been set up, it can be handed to the active capture session either for a one-shot capture (/reference/android/hardware/camera2/CameraCaptureSession.html#capture(android.hardware.camera2.CaptureRequest, android.hardware.camera2.CameraCaptureSession.CaptureCallback, android.os.Handler)) or for an endlessly repeating (/reference/android/hardware/camera2/CameraCaptureSession.html#setRepeatingRequest(android.hardware.camera2.CaptureRequest, android.hardware.camera2.CameraCaptureSession.CaptureCallback, android.os.Handler)) use. Both methods also have a variant that accepts a list of requests to use as a burst capture / repeating burst. Repeating requests have a lower priority than captures, so a request submitted through capture() while there's a repeating request configured will be captured before any new instances of the currently repeating (burst) capture will begin capture.

After processing a request, the camera device will produce a TotalCaptureResult (/reference/android/hardware/camera2/TotalCaptureResult.html) object, which contains information about the state of the camera device at time of capture, and the final settings used. These may vary somewhat from the request, if rounding or resolving contradictory parameters was necessary. The camera device will also send a frame of image data into each of the output Surfaces included in the request. These are produced asynchronously relative to the output CaptureResult, sometimes substantially later.

Classes

CameraCaptureSession	A configured capture session for a `CameraDevice`, used for capturing images from the camera.
CameraCaptureSession.CaptureCallback	A callback object for tracking the progress of a `CaptureRequest` (/reference/android/hardware/camera2/CaptureRequest.html) submitted to the camera device.
CameraCaptureSession.StateCallback	A callback object for receiving updates about the state of a camera capture session.
CameraCharacteristics	The properties describing a `CameraDevice` (/reference/android/hardware/camera2/CameraDevice.html).
CameraCharacteristics.Key<T>	A `Key` is used to do camera characteristics field lookups with `get(CameraCharacteristics.Key)`.
CameraDevice	The CameraDevice class is a representation of a single camera connected to an Android device, allowing for fine-grain control of image capture and post-processing at high frame rates.
CameraDevice.StateCallback	A callback objects for receiving updates about the state of a camera device.
CameraManager	A system service manager for detecting, characterizing, and connecting to `CameraDevices` (/reference/android/hardware/camera2/CameraDevice.html).
CameraManager.AvailabilityCallback	A callback for camera devices becoming available or unavailable to open.
CameraMetadata<TKey>	The base class for camera controls and information.
CaptureFailure	A report of failed capture for a single image capture from the image sensor.
CaptureRequest	An immutable package of settings and outputs needed to capture a single image from the camera device.
CaptureRequest.Builder	A builder for capture requests.
CaptureRequest.Key<T>	A `Key` is used to do capture request field lookups with `get(CaptureResult.Key)` or to set fields with `set(Key, Object)`.
CaptureResult	The subset of the results of a single image capture from the image sensor.
CaptureResult.Key<T>	A `Key` is used to do capture result field lookups with `get(CaptureResult.Key)`.
DngCreator	The `DngCreator` class provides functions to write raw pixel data as a DNG file.
TotalCaptureResult	The total assembled results of a single image

capture from the image sensor.

Exceptions

CameraAccessException	`CameraAccessException` is thrown if a camera device could not be queried or opened by the `CameraManager` (`/reference/android/hardware/camera2/CameraManager.html`), or if the connection to an opened `CameraDevice` (`/reference/android/hardware/camera2/CameraDevice.html`) is no longer valid.