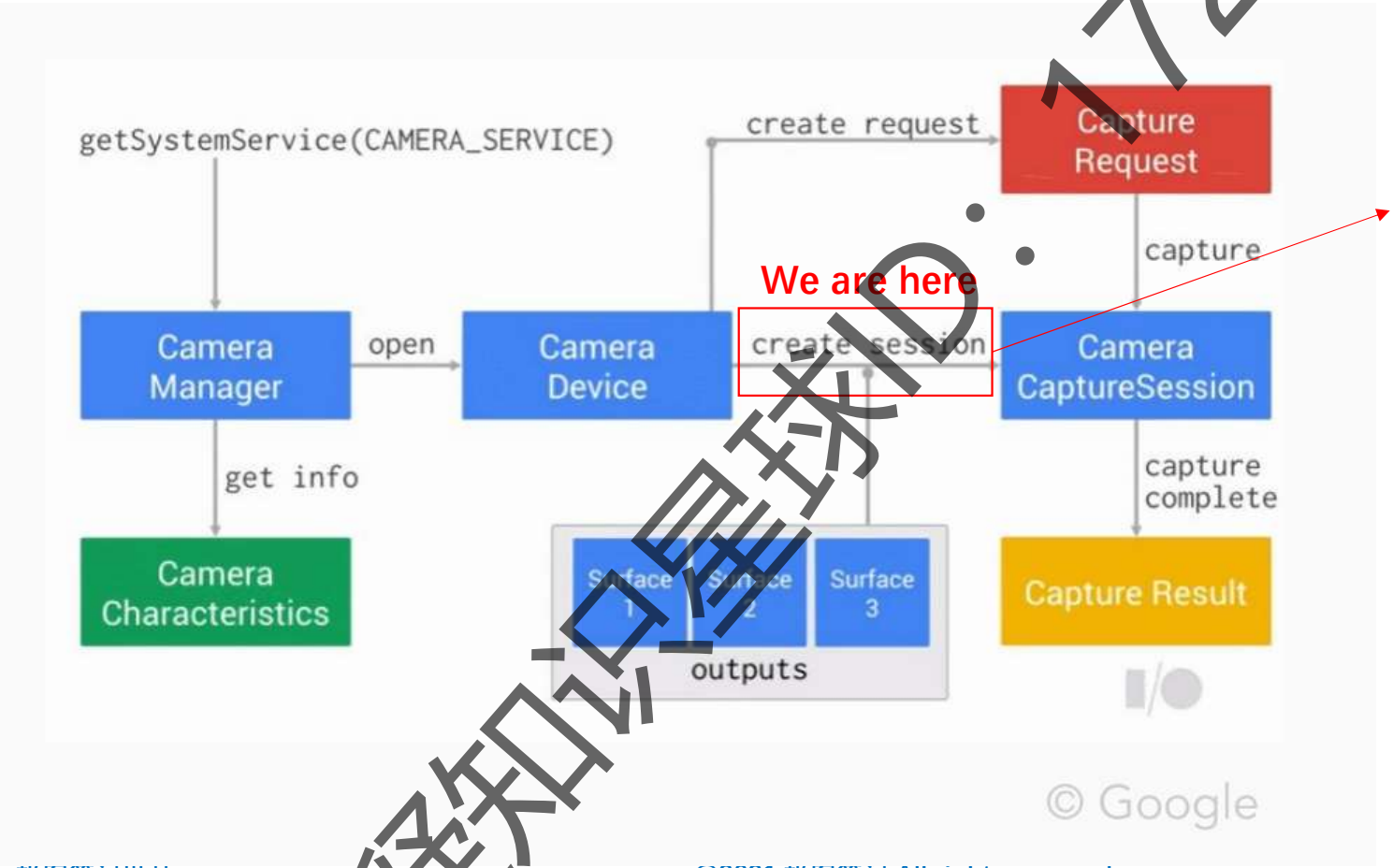
Android Camera2 API专题

第18讲

Reprocessable Capture Session详解二

课程体系



- StreamConfigurationMap
- OutputConfiguration
- **InputConfiguration**
- SessionConfiguration
- createCaptureSession

Agenda

- Reprocessable Architecture
- Reprocessable Flow
- Reprocessing guaranteed stream configurations
- **Reprocessable 相关的APIs**
  - **InputConfiguration**
  - **CameraDevice**
  - **CameraCaptureSession**
  - **ImageWriter**

InputConfiguration简介

- InputConfiguration用于创建Reprocessable capture session.
- **如何使用** InputConfiguration
  - **createReprocessableCaptureSession**(InputConfiguration, List<Surface> outputs, callback, handler)
  - **createReprocessableCaptureSessionByConfigurations**(InputConfiguration, List<OutputConfiguration> outputs, callback, handler)
  - **createCaptureSession** (SessionConfiguration config)
- 如何判断**是否支持** Reprocessable
  - Capabilities中必须一种Reprocessable capability，才能支持配置InputConfiguration
    - PRIVATE_REPROCESSING
    - YUV_REPROCESSING
- 如何获取**支持的Size和Format**
  - StreamConfigurationMap#getinputformats
  - StreamConfigurationMap# getInputSizes

InputConfiguration APIs

API	Description	API
InputConfiguration(int width, int height, int format)	根据width，height和format创建InputConfiguration，这里的format必须来自StreamConfigurationMap#getinputsizes	API 23
InputConfiguration(Collection<MultiResolutionStreamInfo> multiResolutionInputs, int format)	根据multiResolutionInputs和format创建InputConfiguration，这里的format必须来自MultiResolutionStreamConfigurationMap#getInputFormats()，表明从API 31开始Reprocessing也支持多分辨率输入	**API 31**
getFormat()	获取当前InputConfiguration的format	API 23
getHeight()	获取当前InputConfiguration的height	API 23
getWidth()	获取当前InputConfiguration的width	API 23
isMultiResolution()	是否为多分辨率的InputConfiguration，多分辨率的InputConfiguration意味着从它创建的Reprocessable Camera Capture Session允许输入不同大小的图像。	API 31

CameraDevice与Reprocessing相关的APIs

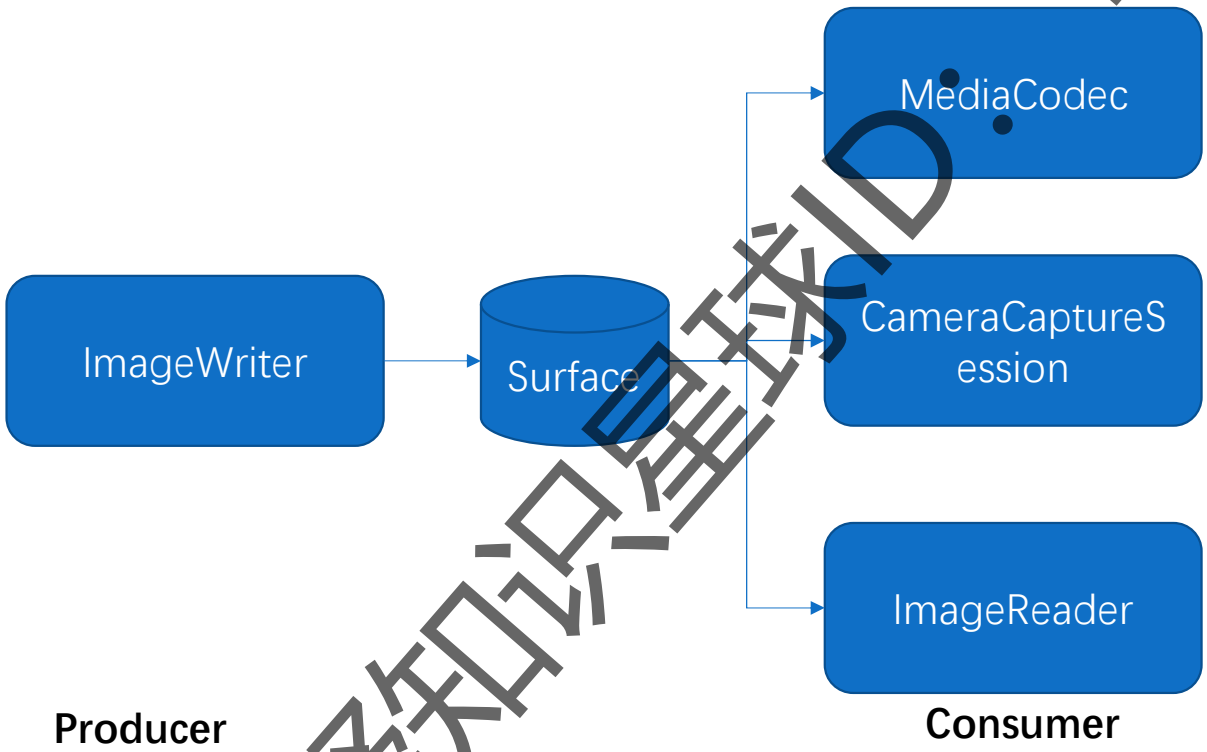
API	Description	API
createReprocessCaptureRequest(TotalCaptureResult inputResult)	根据TotalCaptureResult创建用于处理Reprocessing的CaptureRequest.Builder，这里的TotalCaptureResult与Input image必须匹配，通常通过Timestamp进行匹配。	API 23
createReprocessableCaptureSession (InputConfiguration inputConfig, List<Surface> outputs, CameraCaptureSession.StateCallback callback, Handler handler)	deprecated in API level 30.创建Reprocessable CaptureSession	API 23
createReprocessableCaptureSessionByConfigurations (InputConfiguration inputConfig, List<OutputConfiguration> outputs, CameraCaptureSession.StateCallback callback, Handler handler)	deprecated in API level 30.创建Reprocessable CaptureSession	API 24
createCaptureSession(SessionConfiguration config)	根据SessionConfiguration来创建Reprocessable CaptureSession	API 28

CameraCaptureSession与Reprocessing相关的APIs

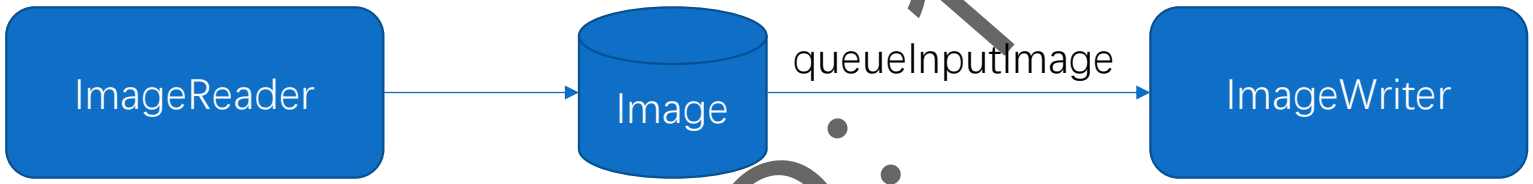
API	Description	API
isReprocessable	判断是否可以通过该CameraCaptureSession submit Reprocessable capture requests	API 23
getInputSurface	获取Input Surface，注意这里只支持一个Input Surface，也没有必要支持多个Input Surfaces	API 23

ImageWriter简介

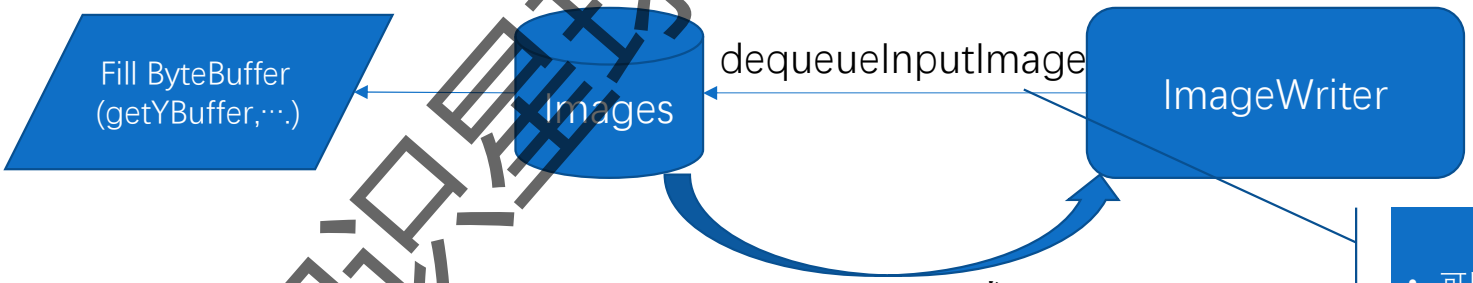
- App通过ImageWriter可以向Surface送一块Buffer



ImageWriter queueInputImage Flow



方式一：直接将Image queue给ImageWriter



queueInputImage(Image) 或 Image#close()

方式二：先dequeue一张Image，填好后再 queueInputImage给ImageWriter

- 可以多Dequeue几张，queue进来的顺序无严格要求

ImageWriter APIs

API	Description	API
newInstance(Surface surface, int maxImages, int format)	根据format和maxImages创建ImageWriter	API 23
newInstance(Surface surface, int maxImages)	根据maxImages创建ImageWriter	API 23
setOnImageReleasedListener(ImageWriter.OnImageReleasedListener listener, Handler handler)	注册OnImageReleasedListener，当Consumer将Image归还给ImageWriter时会调用	
dequeueInputImage()	从ImageWriter中Dequeue一张Image来填	API 23
getMaxImages()	能从ImageWriter中dequeue出来的最大Image数量。dequeueInputImage的数量超出maxImages时会发生IllegalStateException.	API 23
queueInputImage(Image image)	向ImageWriter Queue一张Image	API 23
getFormat()	获取ImageWriter的Buffer Format。	API 23
close()	释放该ImageWriter的所有资源，close后再使用该ImageWriter会发生IllegalStateException	API 23

创建ImageWriter

- newInstance(Surface surface, int maxImages)
  - maxImages决定能从ImageWriter dequeue的最大Image数量，该值要设置适度，太大会占用更多内存
  - ImageWriter中的Image format和size由Surface决定
- newInstance(Surface surface, int maxImages, int format)
  - 这里指定的format会覆盖Surface中的format
  - 比如Surface来自SurfaceTexture默认是PixelFormat#RGBA_8888，这里的format是ImageFormat#PRIVATE，则Surface会被覆盖为ImageFormat#PRIVATE
  - 注意：如果Surface的format与输入format有冲突，请确保Consumer能吃下输入format

dequeueInputImage

- 从ImageWriter dequeue一张Image出来填写
- 填好后，通过queueInputImage(Image) 或 Image#close() 还给ImageWriter
- 无Image可用时（都在Consumer那边未Release），该方法会Block住
- 每当Consumer归还一张Buffer给ImageWriter时，OnImageReleasedListener#onImageReleased会被调用
- ImageFormat#PRIVATE
  - < Android P，这种format是不允许dequeueInputImage的，会发生IllegalStateException
  - >= Android P，这种format允许dequeueInputImage的，App可以通过Image#getHardwareBuffer访问

queueInputImage

- 向ImageWriter queue一张图
- Image来源
  - **ImageReader**
    - Image 属性(size, format, strides, etc.) 必须跟ImageWriter dequeue出来的Image属性完全一致.
    - 如果ImageWriter没有free的Image可用，该方法会被block住
  - **ImageWriter**
    - App负责向Image填数据
- Timestamp
  - Image中的timestamp需要Producer来填写
- 调用频率
  - Producer可以收到OnImageReleasedListener#onImageReleased后才queue下一张Image
- 调用该方法后，输入Image就不能再使用了（可以看成调用了Image.close方法）

Thanks