Camera API - Android

- Rohit Vobbilisetty

What is the Camera API

- Capture Photos
- Capture Videos

Before starting

- Declare the permissions in the Android Manifest

Camera Usage:

```
<uses-permission android:name="android.permission.CAMERA" />
```

Camera Features:

```
<uses-feature android:name="android.hardware.camera" />
```

Camera Auto-Focus:

```
<uses-feature android:name="android.hardware.camera.autofocus" />
```

Write to External Storage:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>
```

Camera API

Overview

Primary API for controlling device cameras. Used to capture pictures and videos.

Package: android.hardware.Camera

Methods:

Camera.open() – Obtain an instance
startPreview() – Starts the Camera preview
takePicture() – Takes a picture
stopPreview() – Stops the Camera preview
release() – Releases the camera
getParameters() – Zoom, Image Quality, Location Information

Camera API – Capture Photo

- Detect and Access Camera
- Create a Preview Class
- Build a Preview Layout
- Setup Listeners for Capture
- Capture and Save Files
- Release the Camera

Camera API – Capture Photo

Detect and Access Camera

Check for an existing Camera and obtain reference.

Detect

```
if (context.getPackageManager().hasSystemFeature(PackageManager.FEATURE_CAMERA)){
            // this device has a camera
            return true;
} else {
            // no camera on this device
            return false;
}
```

Access

Use camera.open() to obtain a reference to the Camera. An exception implies Camera is in use or does not exist.

```
Camera c = null;
        try {
            c = Camera.open(); // attempt to get a Camera instance
        }
        catch (Exception e){
            // Camera is not available (in use or does not exist)
        }
```

Use **Camera.getParameters()** to obtain the Camera capabilities.

Camera API – Capture Photo

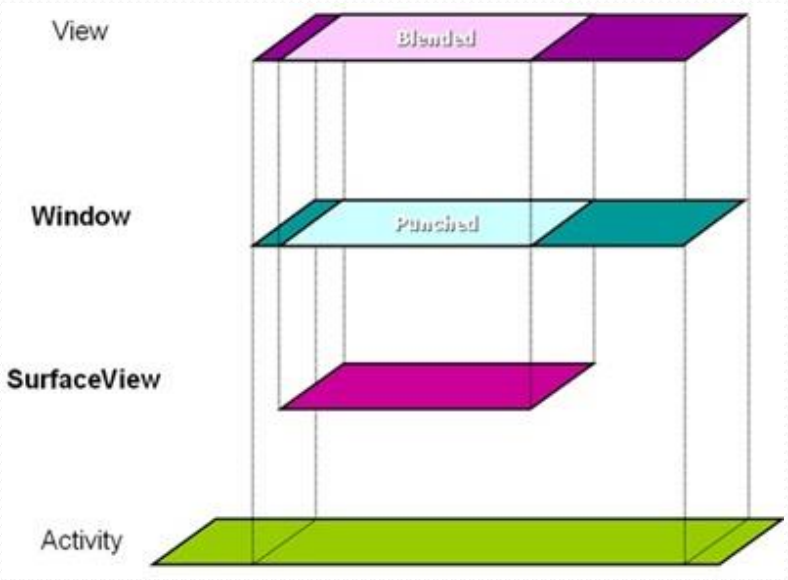
Create a Preview Class and Preview Layout

Surface View (android.view.SurfaceView)

- Provides a dedicated surface
- Is Z- ordered, so useful for overlaying buttons over this surface

SurfaceHolder.Callback (interface) (android.view.SurfaceHolder.Callback)

- Receives information about changes to the surface (SurfaceView).
- Methods:
  surfaceCreated()
  surfaceChanged()
  surfaceDestroyed()



Camera API – Capture Photo

Create a Preview Class and Preview Layout

A Preview Class is a SurfaceView that can display live image data coming directly from the Camera.

Create a class that extends SurfaceView and implements Surfaceholder.Callback..

Override the methods surfaceCreated(), surfaceChanged() and surfaceDestroyed()

Preview Layout:

Relative Layout with a Frame Layout to display the Camera Preview, and a Button which will trigger the capture.

Camera API – Capture Photo

Setup Listeners for Capture

Attach an OnClick Listener to the button.
This listener should call takePicture().

NOTE: The method takePicture() requires the instance of PictureCallback as an argument.

Camera API – Capture Photo

Capture and Save Files

Create an instance of PictureCallback() and override the method onPictureTaken(). This method includes the logic to save the image to a file.

Once the picture is taken, the method onPictureTaken() is called.

Camera API – Capture Photo

Release the Camera

Release the Camera by calling Camera.release(), once the application does not require it or on application exit.

Camera API – Capture Photo

Orientation and Rotation

The Camera Preview's Orientation depends on the Orientation and Rotation of the device.

Orientation (Portrait OR Landscape)

getResources().getConfiguration().orientation

Rotation

activity.getWindowManager().getDefaultDisplay().getRotation();

Returns: ROTATION_0, ROTATION_90, ROTATION_180, ROTATION_270

Set the Camera Preview Orientation using Camera.setDisplayOrientation(angle).

Camera API – Capture Photo

Orientation and Rotation

Orientation: Landscape

Rotation: ROTATION_270

Fix:

Set the Camera Preview Orientation using Camera.setDisplayOrientation()


```
If(Orientation=Landscape &
Rotation=ROTATION_270){
Camera.setDisplayOrientation(180).
}
```

Camera API – Capture Photo

Diagram

Device Camera

Activity

new
CameraPreview(Camera)
Button Event Handler

Camera Preview

surfaceCreated()
surfaceChanged()
surfaceDestroyed()

Layout
FrameLayout and
Button

Additional Features

- Zoom (API level 8)
- GPS Data
- Video Snapshot (API level 14)
- Face Detection (API level 14)

Demo

References

- Android Developer – Camera API
  http://developer.android.com/guide/topics/media/camera.html
- StackOverflow
- Vogella.com – Camera API
  http://www.vogella.com/articles/AndroidCamera/article.html

Questions ??

Thank You