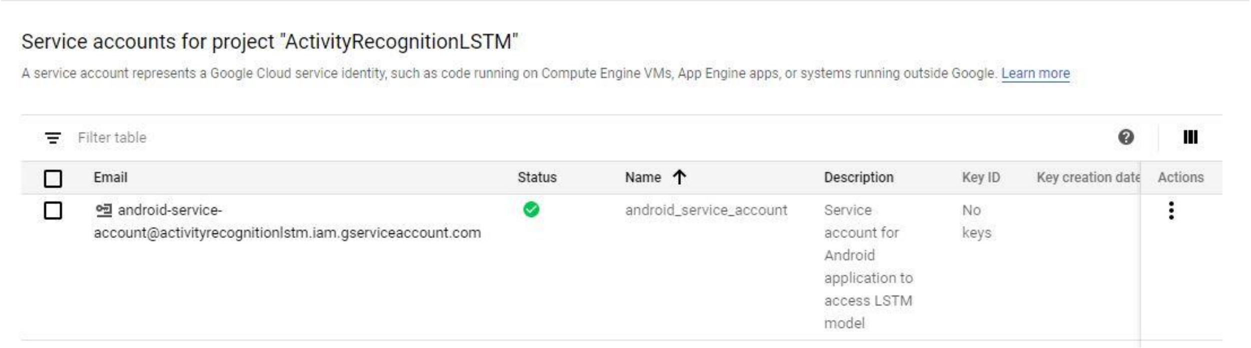
Allowing Android Apps to Access Google Cloud API Services

1. Create a new service account (https://cloud.google.com/iam/docs/creating-managing-service-accounts#iam-service-accounts-create-console)

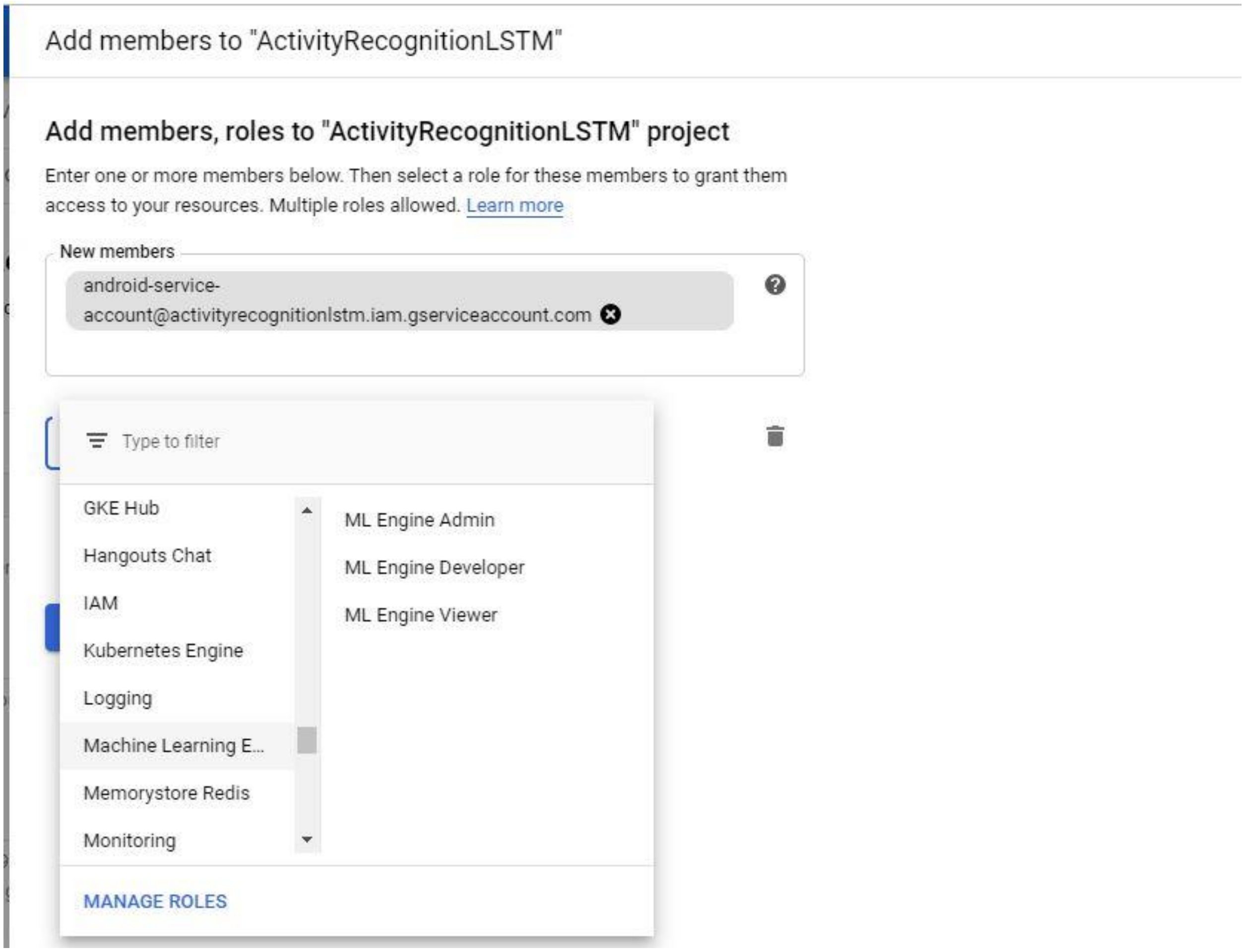


Service accounts for project "ActivityRecognitionLSTM"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. Learn more

□	Email	Status	Name ↑	Description	Key ID	Key creation date	Actions
□	android-service-account@activityrecognitionlstm.iam.gserviceaccount.com	✓	android_service_account	Service account for Android application to access LSTM model	No keys		⋮

2. During or after creating a service account give the account the correct role (roles/ml.developer). See: https://cloud.google.com/ml-engine/docs/access-control

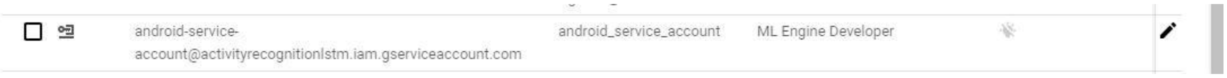


Add members to "ActivityRecognitionLSTM"

Add members, roles to "ActivityRecognitionLSTM" project

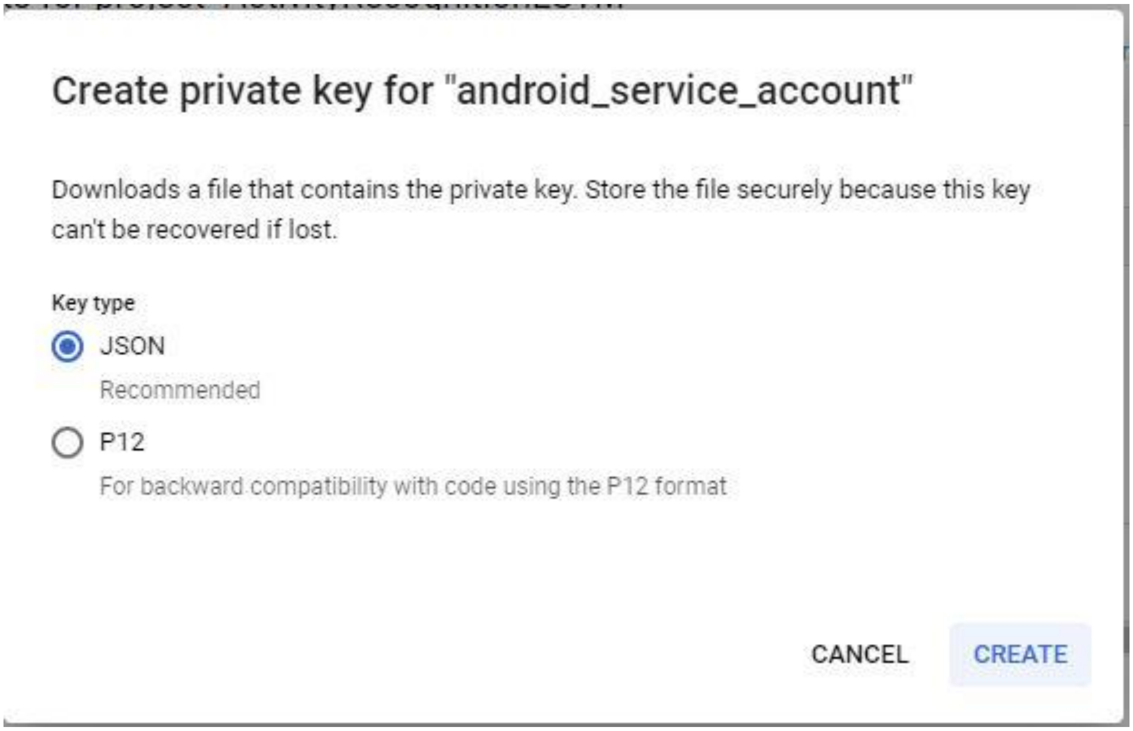
Enter one or more members below. Then select a role for these members to grant them access to your resources. Multiple roles allowed. Learn more

New members: android-service-account@activityrecognitionlstm.iam.gserviceaccount.com



□	android-service-account@activityrecognitionlstm.iam.gserviceaccount.com	android_service_account	ML Engine Developer

3. Create a secret key (.JSON file) and store it locally



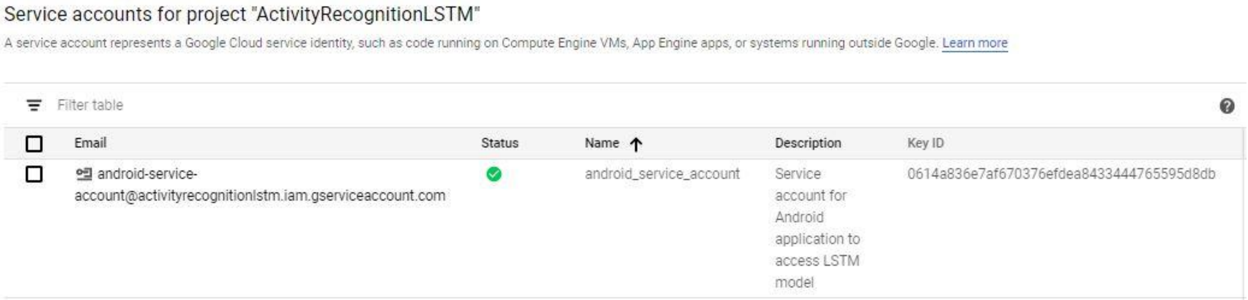
Create private key for "android_service_account"

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type

● JSON
Recommended

○ P12
For backward compatibility with code using the P12 format



Service accounts for project "ActivityRecognitionLSTM"

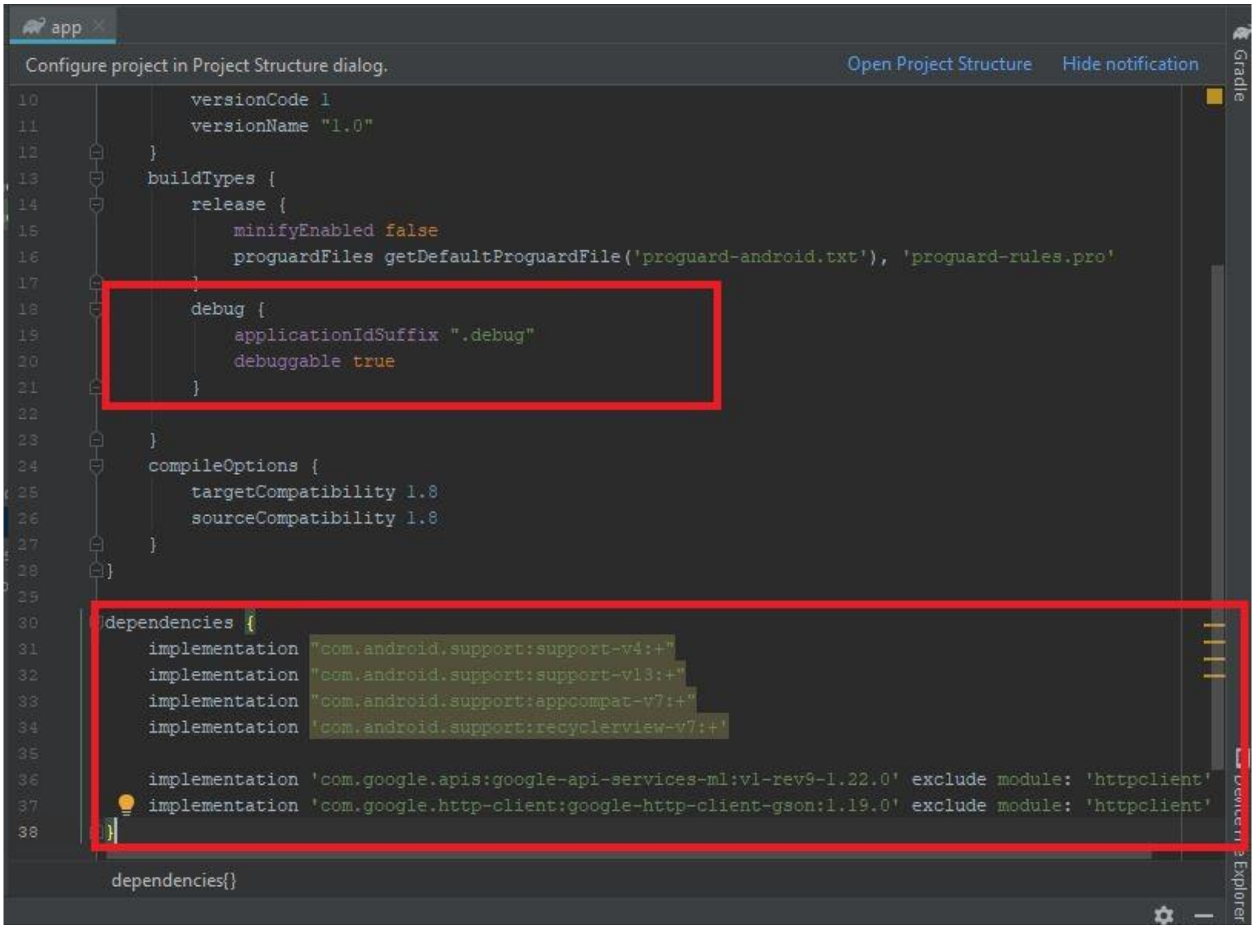
A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. Learn more

□	Email	Status	Name ↑	Description	Key ID
□	android-service-account@activityrecognitionlstm.iam.gserviceaccount.com	✓	android_service_account	Service account for Android application to access LSTM model	0614a836e7af670376efdea8433444765595d8db

4. Clone the Github repository at this URL for the next steps:

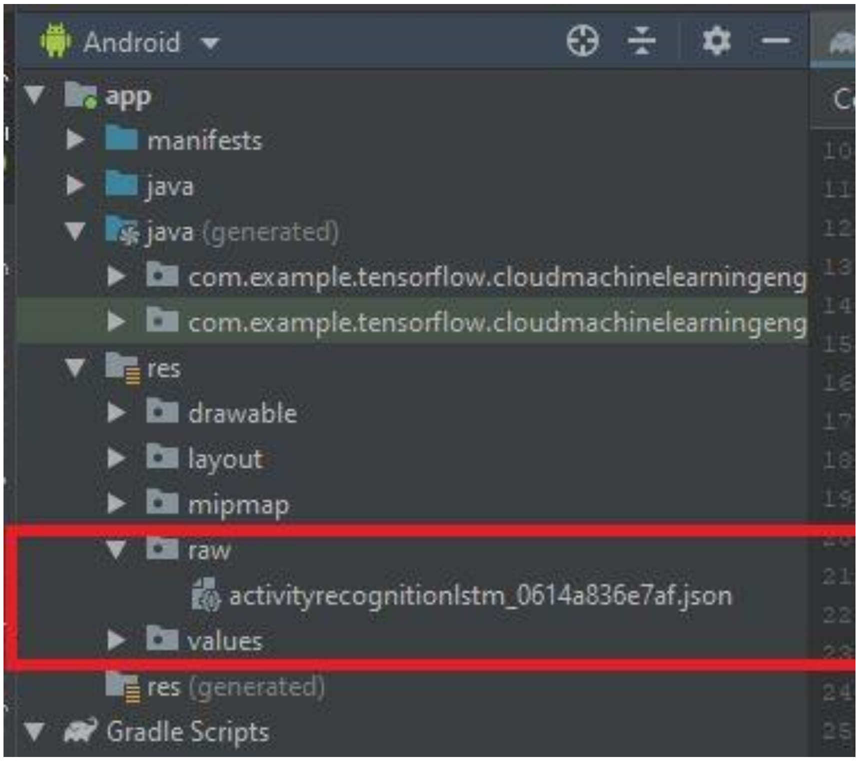
https://github.com/googlesamples/android-TensorFlowCloudMachineLearningEngineStylizer)

5. Update Gradle file to make it de-bugable and swap ‘implementation’ for ‘compile’ in the Gradle dependencies to get rid of error messages (dependencies are a bit outdated, but still appear to be working).

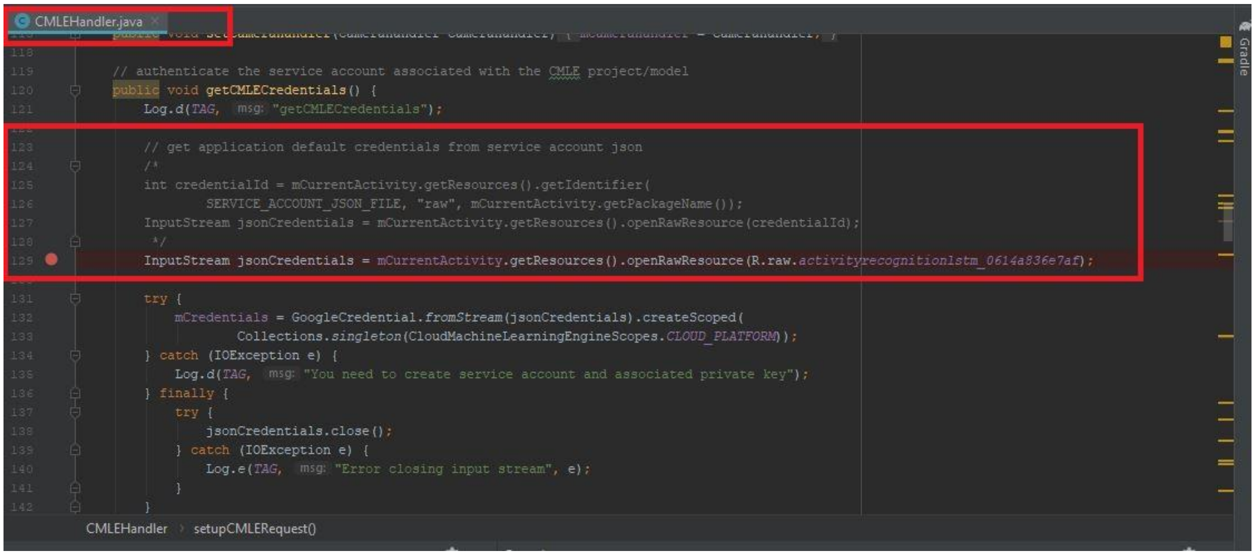


```
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
        debug {
            applicationIdSuffix ".debug"
            debuggable true
        }

    }
    compileOptions {
        targetCompatibility 1.8
        sourceCompatibility 1.8
    }
}

dependencies {
    implementation "com.android.support:support-v4:+"
    implementation "com.android.support:support-v13:+"
    implementation "com.android.support:appcompat-v7:+"
    implementation 'com.android.support:recyclerview-v7:+'

    implementation 'com.google.apis:google-api-services-ml:v1-rev9-1.22.0' exclude module: 'httpclient'
    implementation 'com.google.http-client:google-http-client-gson:1.19.0' exclude module: 'httpclient'
}
```

6. Create a ‘raw’ folder in the res directory of the project and copy the secret key created in step 3 into it.

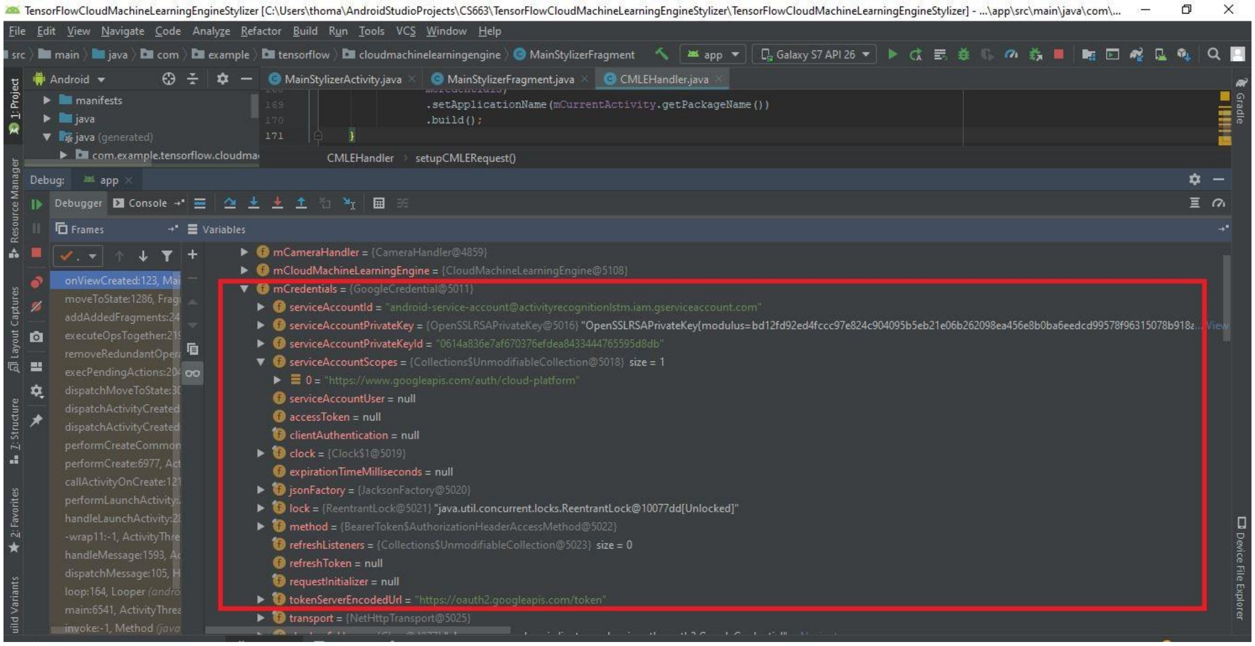

Android
app
manifests
java
java (generated)
com.example.tensorflow.cloudmachinelearningeng
com.example.tensorflow.cloudmachinelearningeng
res
drawable
layout
mipmap
raw
activityrecognitionlstm_0614a836e7af.json
values
res (generated)
Gradle Scripts

7. Open the CMLEHandler class file, comment out the lines to get the secret key file inputstream, and add the single line of code to get the jsonCredentials inputstream (see image below).



```java
    // authenticate the service account associated with the CMLE project/model
    public void getCMLECredentials() {
        Log.d(TAG,  msg: "getCMLECredentials");

        // get application default credentials from service account json
        /*
        int credentialId = mCurrentActivity.getResources().getIdentifier(
                SERVICE_ACCOUNT_JSON_FILE, "raw", mCurrentActivity.getPackageName());
        InputStream jsonCredentials = mCurrentActivity.getResources().openRawResource(credentialId);
         */
        InputStream jsonCredentials = mCurrentActivity.getResources().openRawResource(R.raw.activityrecognitionlstm_0614a836e7af);

        try {
            mCredentials = GoogleCredential.fromStream(jsonCredentials).createScoped(
                    Collections.singleton(CloudMachineLearningEngineScopes.CLOUD_PLATFORM));
        } catch (IOException e) {
            Log.d(TAG,  msg: "You need to create service account and associated private key");
        } finally {
            try {
                jsonCredentials.close();
            } catch (IOException e) {
                Log.e(TAG,  msg: "Error closing input stream", e);
            }
        }
```

8. Test app in debug mode to verify that you can connect with the secret key.


TensorFlowCloudMachineLearningEngineStylizer [C:\Users\thoma\AndroidStudioProjects\CS663\TensorFlowCloudMachineLearningEngineStylizer\TensorFlowCloudMachineLearningEngineStylizer] - ...\app\src\main\java\com\...
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
src 〉 main 〉 java 〉 com 〉 example 〉 tensorflow 〉 cloudmachinelearningengine 〉 MainStylizerFragment
app
Galaxy S7 API 26
Android
manifests
java
java (generated)
com.example.tensorflow.cloudma
MainStylizerActivity.java
MainStylizerFragment.java
CMLEHandler.java
CMLEHandler 〉 setupCMLERequest()
Debug:
app
Debugger
Console
Frames
Variables
1: Project
Resource Manager
Layout Captures
7: Structure
2: Favorites
uild Variants
Gradle
Device File Explorer

```
                .setApplicationName(mCurrentActivity.getPackageName())
                .build();
    }
```

```
onViewCreated:123, Mai
moveToState:1286, Frag
addAddedFragments:24
executeOpsTogether:21
removeRedundantOpera
execPendingActions:20
dispatchMoveToState:3
dispatchActivityCreated
dispatchActivityCreated
performCreateCommon
performCreate:6977, Act
callActivityOnCreate:12
performLaunchActivity:
handleLaunchActivity:2
-wrap11:-1, ActivityThre
handleMessage:1593, Ac
dispatchMessage:105, H
loop:164, Looper (andro
main:6541, ActivityThrea
invoke:-1, Method (java
```

```
▶ mCameraHandler = {CameraHandler@4859}
▶ mCloudMachineLearningEngine = {CloudMachineLearningEngine@5108}
▼ mCredentials = {GoogleCredential@5011}
    ▶ serviceAccountId = "android-service-account@activityrecognitionlstm.iam.gserviceaccount.com"
    ▶ serviceAccountPrivateKey = {OpenSSLRSAPrivateKey@5016} "OpenSSLRSAPrivateKey{modulus=bd12fd92ed4fccc97e824c904095b5eb21e06b262098ea456e8b0ba6eedcd99578f96315078b918a... View
    ▶ serviceAccountPrivateKeyId = "0614a836e7af670376efdea8433444765595d8db"
    ▼ serviceAccountScopes = {Collections$UnmodifiableCollection@5018}  size = 1
        ▶ 0 = "https://www.googleapis.com/auth/cloud-platform"
      serviceAccountUser = null
      accessToken = null
      clientAuthentication = null
    ▶ clock = {Clock$1@5019}
      expirationTimeMilliseconds = null
    ▶ jsonFactory = {JacksonFactory@5020}
    ▶ lock = {ReentrantLock@5021} "java.util.concurrent.locks.ReentrantLock@10077dd[Unlocked]"
    ▶ method = {BearerToken$AuthorizationHeaderAccessMethod@5022}
      refreshListeners = {Collections$UnmodifiableCollection@5023}  size = 0
      refreshToken = null
      requestInitializer = null
    ▶ tokenServerEncodedUrl = "https://oauth2.googleapis.com/token"
    ▶ transport = {NetHttpTransport@5025}
```

9.