

Google Map & Youtube API Note

Sunday, March 27, 2016 11:51 AM

Setup Genymotion and Android Studio

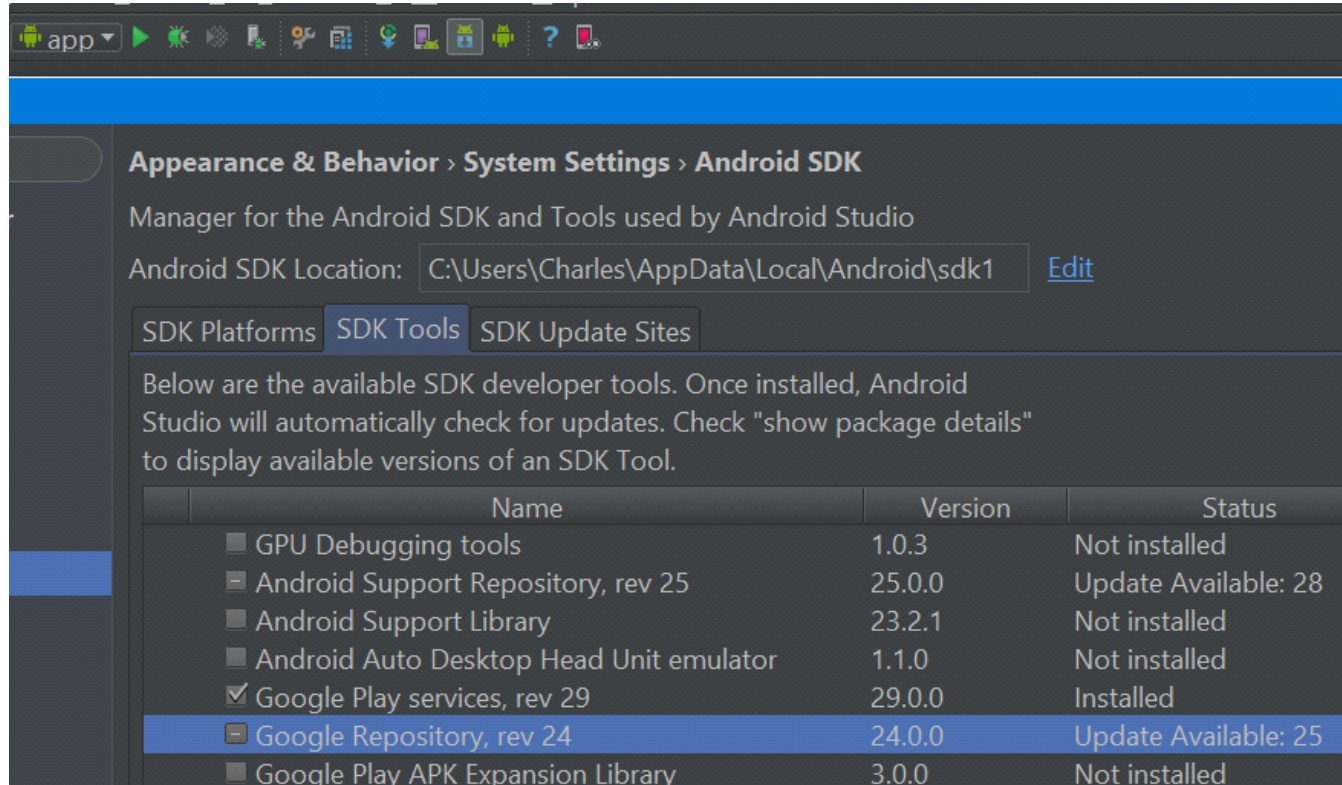
Follow the instruction and install Google Play and Google Services on Genymotion:

<http://stackoverflow.com/questions/17831990/how-do-you-install-google-frameworks-play-accounts-etc-on-a-genymotion-virt>

Then, login and open Play Store in the device and install

1. Youtube App, 2. update Google play services

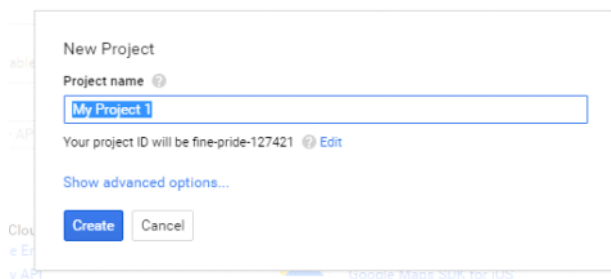
Download SDK Tools:

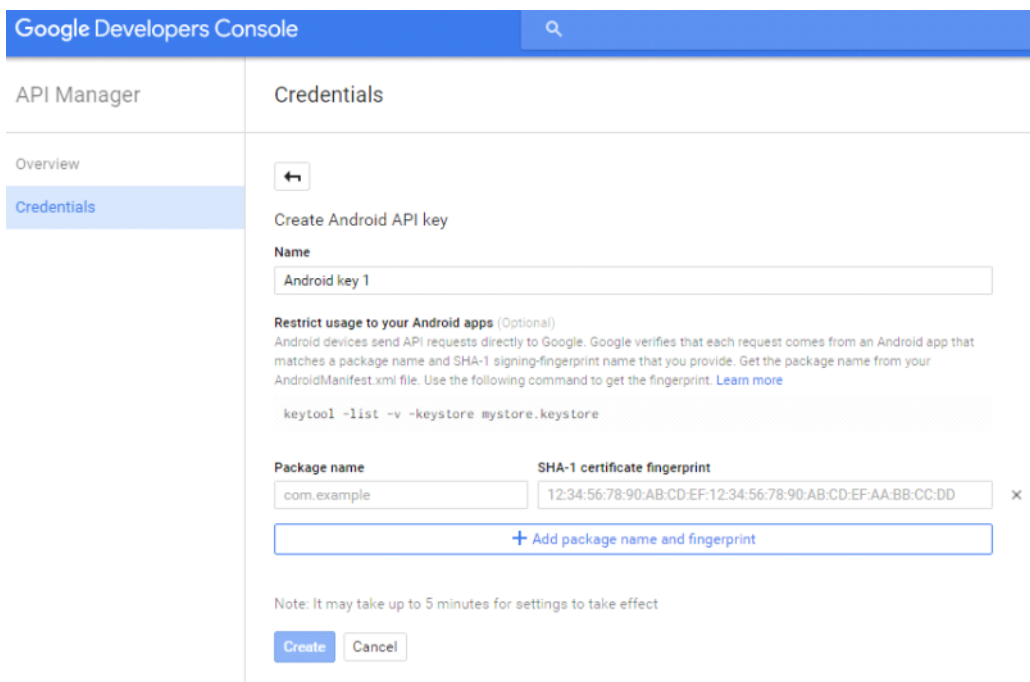
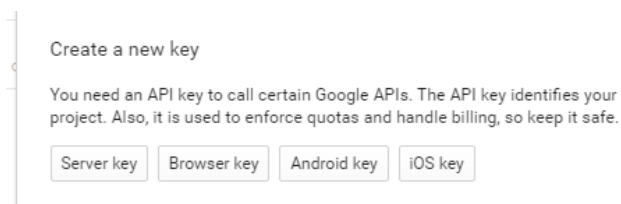
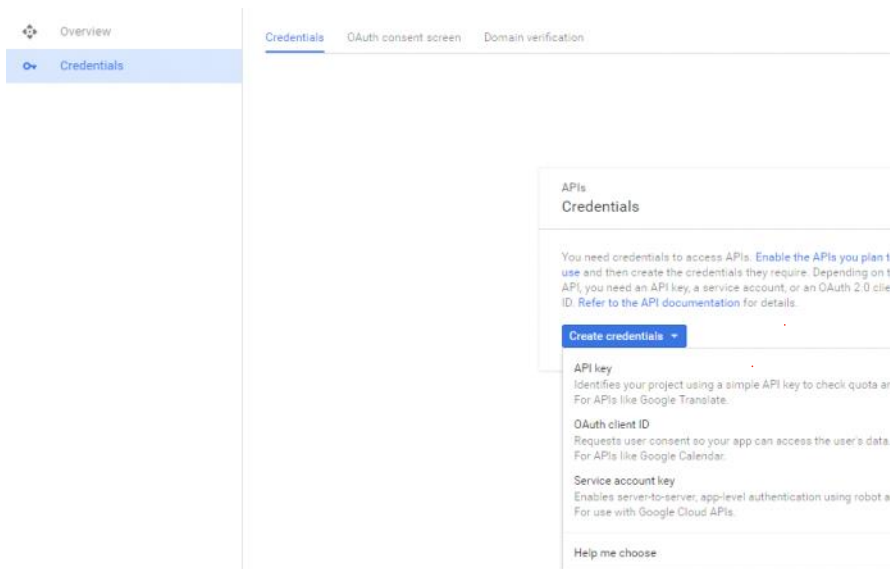


Get Api-key and enable services:

<https://developers.google.com/maps/documentation/android-api/>

Create a new project in console:





To get your own SHA1:

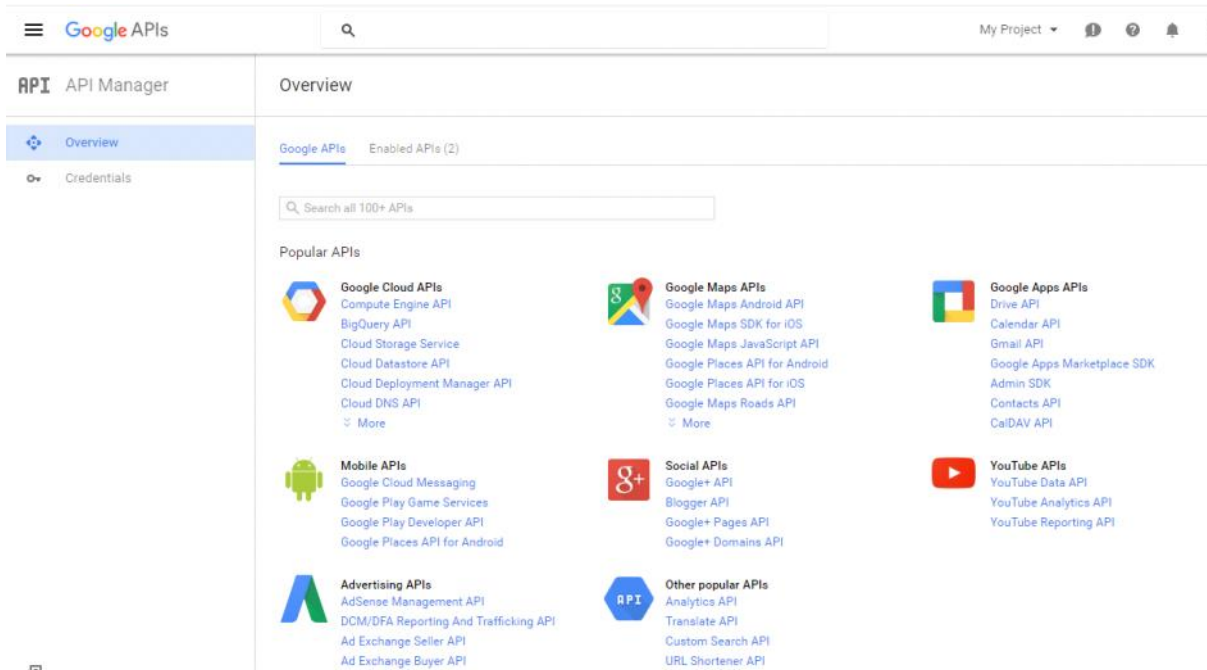
For Mac user:

```
keytool -list -alias androiddebugkey -keystore (path to your own debug keystore) -storepass android -keypass android
```

For windows user:

```
PS C:\Program Files\Java\jdk1.8.0_60\bin> ./keytool.exe -exportcert -alias androiddebugkey -keystore C:\Users\Charles\.android\debug.keystore -list -v
```

Enable services:



Generate google-services.json (optional)

<https://developers.google.com/mobile/add?platform=android&cntapi=signin&cnturl=https%2F%2Fdevelopers.google.com%2Fidentity%2Fsign-in%2Fandroid%2Fsign-in%3Fconfigured%3Dtrue&cntlbl=Continue%20Adding%20Sign-In>

Download and install configuration



Download google-services.json
for com.app.android.mapgeneral

The file contains configuration details, such as keys and identifiers, for the services you just enabled. After downloading, copy the google-services.json to the **app/** or **mobile/** module directory in your Android project.

Google Map Application

Add dependence to build.gradle(Module: app)

```
compile 'com.google.android.gms:play-services:8.3.0'
```

Add plugin to build.gradle(Module: app), at the top of the file.

```
apply plugin: 'com.google.gms.google-services'
```

Add classpath to build.gradle(Project)

```
classpath 'com.google.gms:google-services:2.0.0-alpha2'
```

Add your api-key to one xml file of your project:

```
<string name="google_maps_key" templateMergeStrategy = "preserve"
translatable = "false">your-api-key</string>
```

Add location access permission in AndroidManifest.xml

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

Add api-key to application in the AndroidManifest.xml

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="@string/google_maps_key" />
```

Layout file: fragment_google_map.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <fragment xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/googlemap"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
    />
</LinearLayout>
```

Implements:

```
public class FragmentGoogleMap extends Fragment implements
    OnMapReadyCallback,
    GoogleApiClient.ConnectionCallbacks,
    GoogleApiClient.OnConnectionFailedListener,
    LocationListener {
```

Initialize google Api client in onCreate() in Fragment:

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setRetainInstance(true);
    if(savedInstanceState==null){
        buildGoogleApiClient();
    }
}
```

```
private void buildGoogleApiClient() {
    if(mGoogleApiClient==null){
        mGoogleApiClient = new GoogleApiClient.Builder(getActivity())
            .addConnectionCallbacks(this)
            .addOnConnectionFailedListener(this)
            .addApi(LocationServices.API)
            .build();
    }
}
```

Callback for GoogleApiClient

```

@Override
public void onConnected(Bundle connectionHint) {
    LocationRequest mLocationRequest = createLocationRequest();
    LocationServices.FusedLocationApi.requestLocationUpdates(
        mGoogleApiClient, mLocationRequest, this);

    mLastLocation = LocationServices.FusedLocationApi.getLastLocation(
        mGoogleApiClient);
}

@Override
public void onConnectionFailed(ConnectionResult result) {
    // Refer to the javadoc for ConnectionResult to see what error codes might be returned in
    // onConnectionFailed.
    Log.i("onConnectionFailed ", "Connection failed: ConnectionResult.getErrorCode() = " + result.getErrorCode());
}

@Override
public void onConnectionSuspended(int cause) {
    mGoogleApiClient.connect();
}

```

In the Fragment

```

@Override
public void onStart() {
    super.onStart();
    mGoogleApiClient.connect();
}

@Override
public void onStop() {
    if (mGoogleApiClient.isConnected()) {
        mGoogleApiClient.disconnect();
    }
    super.onStop();
}

```

Set location update interval

```

/*
 * set location request: frequency, priority
 */
private LocationRequest createLocationRequest() {
    LocationRequest mLocationRequest = new LocationRequest();
    mLocationRequest.setInterval(2000);
    mLocationRequest.setFastestInterval(1000);
    mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    return mLocationRequest;
}

```

Callback for LocationListener

```
@Override
public void onLocationChanged(Location location) {

}
```

Get google map fragment and call `getMapAsync(OnMapReadyCallback)`

```
mMapFragment = (SupportMapFragment) getChildFragmentManager().findFragmentById(R.id.googlemap);
if (mMapFragment != null) {
    mMapFragment.getMapAsync(this);
}
```

Invoked when map is ready

```
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    mMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
    //mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
    //mMap.setMapType(GoogleMap.MAP_TYPE_NONE);
    mMap.setMyLocationEnabled(true);

    mMap.setOnMapClickListener((lat) -> {
        Toast.makeText(getContext(), "Latitude: " + lat.latitude + "\nLongitude: " + lat.longitude,
    });

    mMap.setOnMarkerClickListener((marker) -> {
        Toast.makeText(getContext(), marker.getTitle().toString(), Toast.LENGTH_SHORT).show();
        return true;
    });

    // Add a marker in Sydney and move the camera
    LatLng sydney = new LatLng(-34, 151);
    mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
}
```

Add and customize marker:

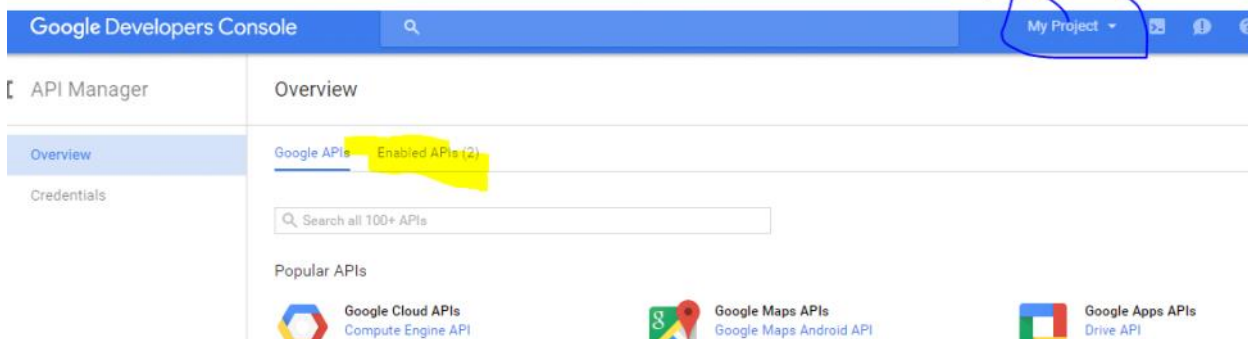
https://developers.google.com/maps/documentation/android-api/marker#customize_a_marker

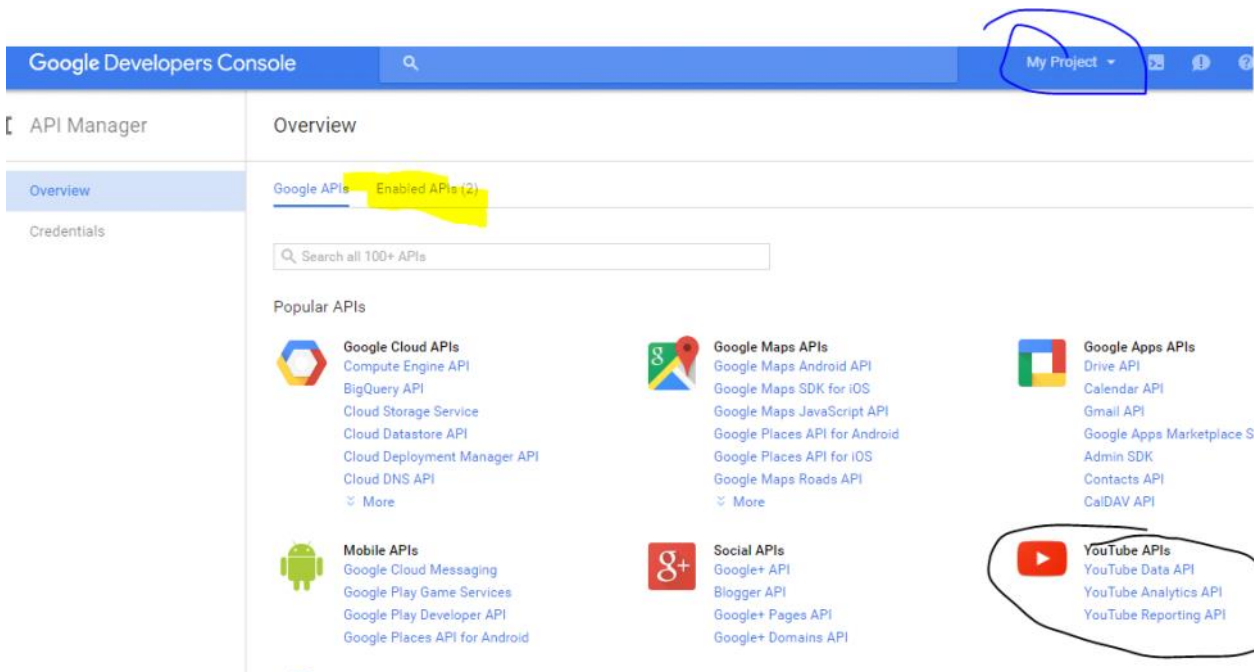
Add polyline:

<https://developers.google.com/android/reference/com/google/android/gms/maps/model/Polyline>

Youtube API

Enable api in google developer console:





Download youtube api:

<https://developers.google.com/youtube/android/player/downloads/>

Extract files and copy .jar file in the libs folder to
"Your_Project_Folder/app/libs/"

Check youtube service before you do anything with youtube api

```
//Check for youtube service issues
final YouTubeInitializationResult result = YouTubeApiServiceUtil.isYouTubeApiServiceAvailable(this);

if (result != YouTubeInitializationResult.SUCCESS) {
    //If there are any issues we can show an error dialog.
    result.getErrorDialog(this, 0).show();
} else{
    //do what you want to do.
}
```

Start to play a video in the youtube app on the device

```
//check if the youtube app is installed in the device
if (YouTubeIntents.canResolvePlayVideoIntent(this)) {
    //Opens the video in the YouTube app
    startActivity(YouTubeIntents.createPlayVideoIntent(this, "68A_HPYGdlk"));
    //startActivity(YouTubeIntents.createPlayVideoIntentWithOptions(this, "68A_HPYGdlk", 50000, true, true));
}
```

```
//Opens in the StandAlonePlayer, defaults to fullscreen
if (YouTubeIntents.canResolvePlayVideoIntent(this)) {
    //Opens in the StandAlonePlayer, defaults to fullscreen
    startActivity(YouTubeStandalonePlayer.createVideoIntent(this,
        getString(R.string.google_maps_key), "68A_HPYGdlk", 50000, true, true));
}
```


public static Intent **createVideoIntent** (Activity activity, String developerKey, String videoid, int timeMillis, boolean autoplay, boolean lightboxMode)

Creates an intent that will start a new standalone player activity that plays a single video.

Parameters

activity	The calling activity from which the standalone player will be started.
developerKey	A valid API key which is enabled to use the YouTube Data API v3 service. To generate a new key, visit the Google Developers Console .
videoid	The id of the video to be played.
timeMillis	The time, in milliseconds, where playback should start in the video.
autoplay	true to have the video start playback as soon as the standalone player loads, false to cue the video.
lightboxMode	true to have the video play in a dialog view above your current Activity, false to have the video play fullscreen.

Use YouTubePlayerView in activity

Add YouTubePlayerView:

```
<com.google.android.youtube.player.YouTubePlayerView
    android:id="@+id/youtubePlayerView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

Extends YouTubeBaseActivity:

```
public class YouTubeActivity extends YouTubeBaseActivity implements
    YouTubePlayer.OnInitializedListener{
```

Callback for OnInitializedListener

Success:

```
@Override
public void onInitializationSuccess(YouTubePlayer.Provider provider, YouTubePlayer youTubePlayer, boolean restored) {

    mPlayer=youTubePlayer;

    //This flag tells the player to switch to landscape when in fullscreen, it will also return to portrait
    //when leaving fullscreen
    mPlayer.setFullscreenControlFlags(YouTubePlayer.FULLSCREEN_FLAG_CONTROL_ORIENTATION);

    //This flag tells the player to automatically enter fullscreen when in landscape. Since we don't have
    //landscape layout for this activity, this is a good way to allow the user rotate the video player.
    mPlayer.addFullscreenControlFlag(YouTubePlayer.FULLSCREEN_FLAG_ALWAYS_FULLSCREEN_IN_LANDSCAPE);

    //This flag controls the system UI such as the status and navigation bar, hiding and showing them
    //alongside the player UI
    mPlayer.addFullscreenControlFlag(YouTubePlayer.FULLSCREEN_FLAG_CONTROL_SYSTEM_UI);

    if (mVideoId != null) {
        if (restored) {
            mPlayer.play();
        } else {
            mPlayer.loadVideo(mVideoId);
        }
    }
}
```



```

        mPlayer.play();
    } else {
        mPlayer.loadVideo(mVideoId);
    }
}
}

```

Failure:

```

@Override
public void onInitializationFailure(YouTubePlayer.Provider provider,
                                   YouTubeInitializationResult youTubeInitializationResult) {
    if (youTubeInitializationResult.isUserRecoverableError()) {
        youTubeInitializationResult.getErrorDialog(this, RECOVERY_DIALOG_REQUEST).show();
    } else {
        //Handle the failure
        Toast.makeText(this, "onInitializationFailure", Toast.LENGTH_LONG).show();
    }
}

```

Get YouTubePlayerView and initialize.

```

final YouTubePlayerView playerView = (YouTubePlayerView) findViewById(R.id.youtubePlayerView);
playerView.initialize(getString(R.string.google_maps_key), this);

```

Reference:

<https://developers.google.com/youtube/android/player/reference/com/google/android/youtube/player/YouTubePlayer#Summary>

Use YouTubePlayerSupportFragment (cannot use YouTubePlayerView in fragment!)

Add YouTubePlayerSupportFragment to the layout file

```

<fragment
    android:name="com.google.android.youtube.player.YouTubePlayerSupportFragment"
    android:id="@+id/moviePlayer"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

```

Get YouTubePlayer fragment after loading the layout in the Fragment:

```

YouTubePlayerSupportFragment playerFragment =
    (YouTubePlayerSupportFragment) getChildFragmentManager().findFragmentById(R.id.moviePlayer);
playerFragment.initialize(getString(R.string.google_maps_key), this);

```

Same thing, we need to implement YouTubePlayer.OnInitializedListener for Fragment

Get Thumbnails from YouTube

```

<com.google.android.youtube.player.YouTubeThumbnailView
    android:id="@+id/icon"
    android:layout_width="80dp"
    android:layout_height="100dp"
    android:layout_margin="5dp"
    android:background="@color/app_primary_dark"
    android:scaleType="centerCrop" />

```

```

thumb = (YouTubeThumbnailView) rootView.findViewById(R.id.icon);
thumb.setTag("68A_HPYGdlk");
thumb.initialize(getContext().getString(R.string.google_maps_key), this);

```

```
implements YouTubeThumbnailView.OnInitializedListener
```

Callback for YouTubeThumbnailView.OnInitializedListener:

```
@Override
public void onInitializationSuccess(YouTubeThumbnailView view, final YouTubeThumbnailLoader loader) {
    loader.setVideo((String) view.getTag());
    loader.setOnThumbnailLoadedListener(new YouTubeThumbnailLoader.OnThumbnailLoadedListener() {
        @Override
        public void onThumbnailLoaded(YouTubeThumbnailView youTubeThumbnailView, String s) {
            loader.release();
        }
    });
    @Override
    public void onThumbnailError(YouTubeThumbnailView youTubeThumbnailView,
                                YouTubeThumbnailLoader.ErrorReason errorReason) {
        Log.d("onInitialization Error ", "onThumbnailError");
    }
}

@Override
public void onInitializationFailure(YouTubeThumbnailView thumbnailView, YouTubeInitializationResult errorReason) {
    final String errorMessage = errorReason.toString();
    Toast.makeText(getContext(), errorMessage, Toast.LENGTH_LONG).show();
}
```