

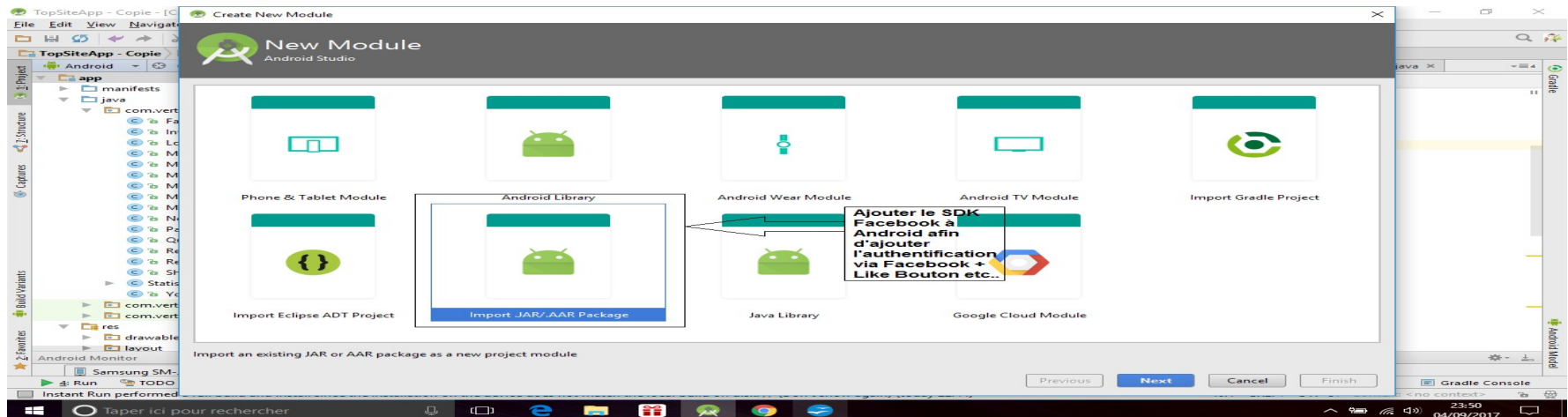
# Intégrer Data API V3 YouTube dans votre Application Android

- ▶ 1) Importation du YouTube Android Player à votre projet Android!

Télécharger le fichier .JAR à l'adresse suivante:

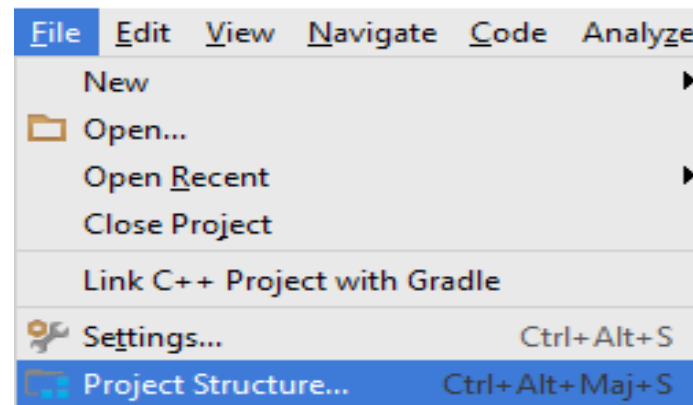
<https://developers.google.com/youtube/android/player/downloads/YouTubeAndroidPlayerApi-1.2.2.zip>

- ▶ Dézipper le contenu et dans votre projet Android en sélectionnant le dossier app + clic droit dans votre arborescence de votre projet à gauche! Faites New → Module → Import an existing JAR package → Sélectionner le fichier .JAR du YouTube Android Player télécharger normalement appelé : YouTubeAndroidPlayerApi-1.2.2.zip

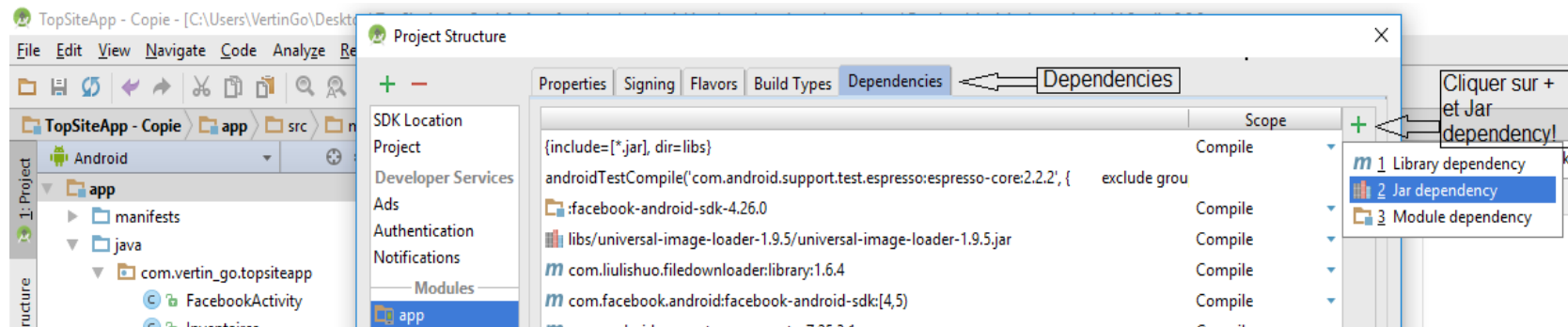


# Intégrer Data API V3 You Tube dans votre Application Android

- Ajouter également les dépendances à votre projet en cliquant sur File → Project Structure:



- Et ensuite ajouter les dépendances à votre projet comme illustrer ci-dessous!



# Intégrer Data API V3 You Tube dans votre Application Android

- Récupérer votre clé API sur Google Cloud Platform!(Créer un Projet si ce n'est pas déjà fait!)

The screenshot shows the Google Cloud Platform console dashboard. The browser address bar displays the URL: <https://console.cloud.google.com/home/dashboard?project=delta-vial-177705&hl=fr>. The dashboard includes a navigation sidebar on the left with categories like 'Accueil', 'API et services', 'Facturation', 'PRODUITS', 'Cloud Launcher', 'Assistance', 'IAM et administration', and 'App Engine'. A blue arrow points from the 'API et services' menu item to a text box that says 'Aller sur API et Services!'. The main content area is titled 'TABLEAU DE BORD' and 'ACTIVITÉ'. It features several widgets: 'Informations sur le projet' (Project Information) showing details for 'TopSiteApp' (ID: delta-vial-177705, Number: 482574690279), 'Compute Engine' (Processeur (%)), 'État de Google Cloud Platform' (Functionnement normal de tous les services), 'Error Reporting' (Aucun signe d'erreur), 'Ressources' (Compute Engine, Cloud Storage), and 'Actualités'. A link 'Accéder aux paramètres du projet' is visible under the project information.

# Intégrer Data API V3 You Tube dans votre Application Android

- Récupérer votre clé API sur Google Cloud Platform!(Créer un Projet si ce n'est pas déjà fait!)

! Hurricane Irma: We do not anticipate any service degradation for any GCP regions from Irma. [IGNORER](#)

Google Cloud Platform TopSiteApp

API API et services

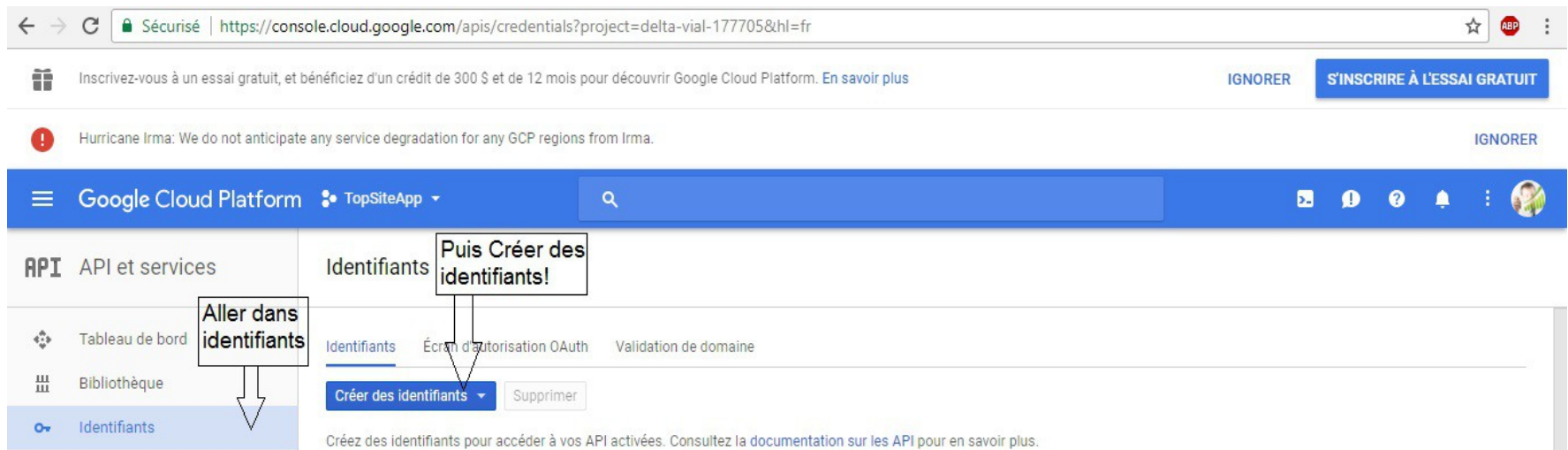
Tableau de bord [+ ACTIVER LES API ET LES SERVICES](#)

Google Cloud SQL	—	—	—	—	—	Désactiver
Google Cloud Storage	—	—	—	—	—	Désactiver
Google Cloud Storage JSON API	—	—	—	—	—	Désactiver
Google Cloud Testing API	—	—	—	—	—	Désactiver
Google Service Management API	—	—	—	—	—	Désactiver
Google+ API	—	—	—	—	—	Désactiver
Identity Toolkit API	—	—	—	—	—	Désactiver ⚙
Legacy Cloud Source Repositories API Private API ?	—	—	—	—	—	Désactiver
Mobile Crash and Performance Reporting API	—	—	—	—	—	Désactiver
Stackdriver Debugger API	—	—	—	—	—	Désactiver
Stackdriver Logging API	—	—	—	—	—	Désactiver
Stackdriver Monitoring API	—	—	—	—	—	Désactiver
Stackdriver Trace API	—	—	—	—	—	Désactiver
Token Service API	—	—	—	—	—	Désactiver
YouTube Data API v3	—	—	—	—	—	Désactiver

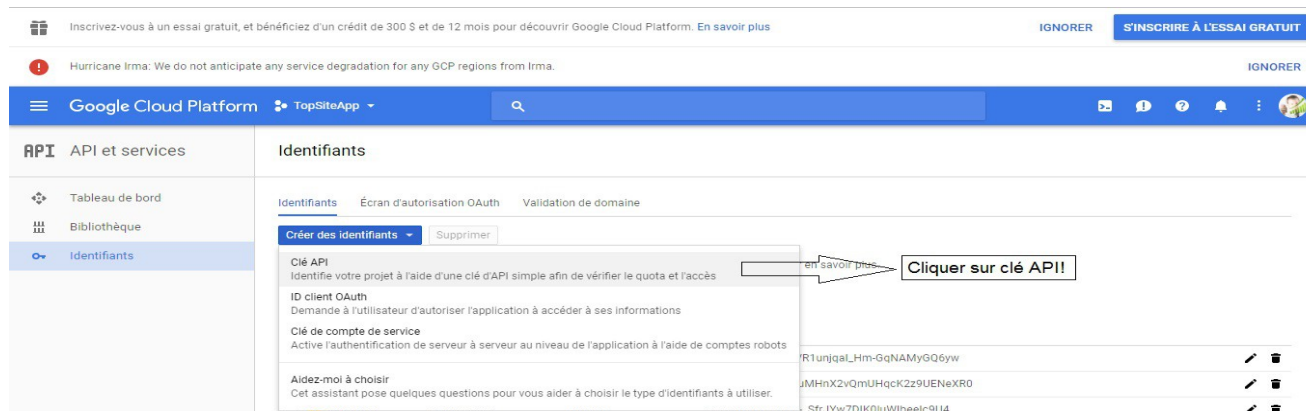
Activer YouTube Data API v3

# Intégrer Data API V3 You Tube dans votre Application Android

- Rendez-vous dans le menu Identifiants puis Créer des identifiants!



- Sélectionner dans le sous-menu Créer des identifiants → Clé Api



# Intégrer Data API V3 You Tube dans votre Application Android

- ▶ Cliquer sur restreindre la clé!

## Clé API créée

Utilisez cette clé dans votre application sous la forme du paramètre `key=API_KEY`.

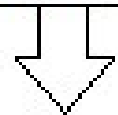
Votre clé API

AIzaSyCeHo3HfcxfHwgOTtqDr2yQPxkUoNOVeS8



 Ajoutez des restrictions pour votre clé afin d'éviter toute utilisation non autorisée en production.

Cliquer sur  
restreindre  
la clé!



FERMER

RESTREINDRE LA CLÉ



# Intégrer Data API V3 You Tube dans votre Application Android

- Choisir Application Android, indiquer nom de package et empreinte de certification SHA-1 (Indication sur la diapo suivante), puis enregistrer!

The screenshot shows the 'Clé API' (API Key) configuration page in the Google Cloud console. At the top, there are navigation links: a back arrow, 'Clé API', 'REGÉNÉRER LA CLÉ' (Regenerate the key), and 'SUPPRIMER' (Delete). Below this is a text input field containing 'Clé API 7'. The main section is titled 'Restriction de clé' (Key restriction) and explains that a key restriction allows specifying websites, IP addresses, or applications that can use the key. There are four radio button options: 'Aucun' (None), 'Référénts HTTP (sites Web)' (Referenced HTTP (websites)), 'Adresses IP (serveurs Web, tâches Cron, etc.)' (IP addresses (web servers, cron jobs, etc.)), and 'Applications Android' (Android applications), which is selected. A callout box points to the 'Applications Android' option with the text 'Choisir application Android!'. Below the radio buttons is a section titled 'Limiter l'utilisation de vos applications Android (Facultatif)' (Limit the use of your Android applications (Optional)). It instructs the user to add the package name and the SHA-1 certificate fingerprint to limit the use of their Android applications. It provides a terminal command: `$ keytool -list -v -keystore mystore.keystore`. Below the command, there are two input fields: 'Nom du package' (Package name) with the value 'com.example' and 'Empreinte du certificat SHA-1' (SHA-1 certificate fingerprint) with the value '12:34:56:78:90:AB:CD:EF:12:34:56:78:90:AB:CD:EF:AA:BB:CC:DD'. A callout box points to the package name field with the text 'Indiquer le nom de package de votre projet!' (Indicate the package name of your project!). Another callout box points to the SHA-1 fingerprint field with the text 'Pour l'empreinte du certificat SHA-1 indication à suivre sur la diapo suivante!' (For the SHA-1 certificate fingerprint, follow the indication on the next slide!). Below these fields is a button labeled '+ Ajouter nom du package et empreinte' (Add package name and fingerprint). At the bottom, there are two buttons: 'Enregistrer' (Save) and 'Annuler' (Cancel). A remark at the bottom states: 'Remarque : L'application de ce paramètre peut prendre jusqu'à cinq minutes.' (Note: Applying this parameter may take up to five minutes.).

← Clé API    REGÉNÉRER LA CLÉ    SUPPRIMER

Clé API 7

**Restriction de clé**  
Une restriction de clé vous permet de spécifier les sites Web, les adresses IP ou les applications pouvant utiliser cette clé. [En savoir plus](#)

☐ Aucun  
☐ Référénts HTTP (sites Web)  
☐ Adresses IP (serveurs Web, tâches Cron, etc.)  
☒ Applications Android    Choisir application Android!  
☐ Applications iOS

**Limiter l'utilisation de vos applications Android (Facultatif)**  
Ajoutez le nom du package et l'empreinte de certificat de signature SHA-1 pour limiter l'utilisation de vos applications Android.  
Pour connaître le nom du package, reportez-vous au fichier AndroidManifest.xml. Si vous souhaitez obtenir l'empreinte, utilisez la commande suivante :

```
$ keytool -list -v -keystore mystore.keystore
```

Nom du package    Empreinte du certificat SHA-1

com.example    12:34:56:78:90:AB:CD:EF:12:34:56:78:90:AB:CD:EF:AA:BB:CC:DD

+ Ajouter nom du package et empreinte

Indiquer le nom de package de votre projet!    Pour l'empreinte du certificat SHA-1 indication à suivre sur la diapo suivante!

Remarque : L'application de ce paramètre peut prendre jusqu'à cinq minutes.

Enregistrer    Annuler

# Intégrer Data API V3 You Tube dans votre Application Android

- ▶ Ouvrir une invite de commande dans le dossier suivant:  
C:\Program Files\Android\Android Studio\jre\jre\bin\keytool
- ▶ Chemin vers le fichier debug.keystore:  
C:\Users\Username\.android\debug.keystore
- ▶ Mot de passe demandé : android

```
Sélection C:\Windows\system32\cmd.exe

C:\Program Files\Android\Android Studio\jre\jre\bin> keytool -list -v -keystore C:\Users\VertinGo\.android\debug.keystore
Entrez le mot de passe du fichier de clés :

Type de fichier de clés : JKS
Fournisseur de fichier de clés : SUN
Mot de passe demandé: android

Votre fichier de clés d'accès contient 1 entrée

Nom d'alias : androiddebugkey
Date de création : 20 août 2017
Type d'entrée : PrivateKeyEntry
Longueur de chaîne du certificat : 1
Certificat[1]:
Propriétaire : C=US, O=Android, CN=Android Debug
Emetteur : C=US, O=Android, CN=Android Debug
Numéro de série : 1
Valide du : Sun Aug 20 06:31:07 CEST 2017 au : Tue Aug 13 06:31:07 CEST 2047
Empreintes du certificat :
MD5: 4B:5D:15:11:BD:74:73:2F:1B:E3:BD:E6:36:36:3A:65
SHA1 : 47:20:BB:45:8B:22:45:75:83:4F:F9:6F:16:DC:90:F2:93:4B:DB:A1
SHA256 : 3E:2D:90:42:97:82:2E:8D:B7:8C:AE:36:99:AA:96:67:E3:49:0C:47:C2:A1:A0:0B:3A:73:51:83:6B:D0:D5:06
Nom de l'algorithme de signature : SHA1withRSA
Version : 1

*****
*****
```



# Intégrer Data API V3 You Tube dans votre Application Android

- ▶ Exemple d'activité Android YouTube!(Intégration d'une vidéo YouTube!)
- ▶ Pour commencer déclaration de la class config qui contiendra votre clé d'API YouTube

```
package com.nom_de_votre_package;

/**
 * Created by VertinGo on 07/09/2017.
 */

public final class Config {

    private Config() {

    }

    public static final String YOUTUBE_API_KEY = "votre_clé_d_api";

}
```

# Intégrer Data API V3 You Tube dans votre Application Android

- ▶ Ajouter dans votre fichier: build.gradle la dépendance suivante:

```
compile files('libs/YouTubeAndroidPlayerApi.jar')
```

- ▶ Ajouter dans votre AndroidManifest.xml:

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

- ▶ Ajouter les ressources strings suivantes dans res/strings.xml

```
<string name="app_name">VideoTube</string>
<string name="player_error">Error initializing YouTube player: %s</string>
<string name="seek_to">Jump To</string>
<string name="seek_to_hint">Seconds</string>
```

- ▶ Ajouter le layout suivant : activity\_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

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

</RelativeLayout>
```

# Intégrer Data API V3 You Tube dans votre Application Android

## ► Ajout de la class MainActivity

```
import android.content.Intent;
import android.os.Bundle;
import android.widget.Toast;

import com.google.android.youtube.player.YouTubeBaseActivity;
import com.google.android.youtube.player.YouTubeInitializationResult;
import com.google.android.youtube.player.YouTubePlayer;
import com.google.android.youtube.player.YouTubePlayer.Provider;
import com.google.android.youtube.player.YouTubePlayerView;

public class MainActivity extends YouTubeBaseActivity implements YouTubePlayer.OnInitializedListener {

    private static final int RECOVERY_REQUEST = 1;
    private YouTubePlayerView youtubeView;
    private MyPlayerStateChangeListener playerStateChangeListener;
    private MyPlaybackEventListener playbackEventListener;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        youtubeView = (YouTubePlayerView) findViewById(R.id.youtube_view);
        youtubeView.initialize(Config.YOUTUBE_API_KEY, this);

        playerStateChangeListener = new MyPlayerStateChangeListener();
        playbackEventListener = new MyPlaybackEventListener();
    }
}
```

# Intégrer Data API V3 You Tube dans votre Application Android

## ► Ajouter les méthodes suivante à la class MainActivity

```
private void showMessage(String message) {
    Toast.makeText(this, message, Toast.LENGTH_LONG).show();
}

@Override
public void onInitializationSuccess(Provider provider, YouTubePlayer player, boolean wasRestored) {
    player.setPlayerStateChangeListener(playerStateChangeListener);
    player.setPlaybackEventListener(playbackEventListener);

    if (!wasRestored) {
        player.cueVideo("FzbLp1FKX8Q"); // Plays https://www.youtube.com/watch?v=FzbLp1FKX8Q
    }
}

@Override
public void onInitializationFailure(Provider provider, YouTubeInitializationResult errorReason) {
    if (errorReason.isUserRecoverableError()) {
        errorReason.getErrorDialog(this, RECOVERY_REQUEST).show();
    } else {
        String error = String.format(getString(R.string.player_error), errorReason.toString());
        Toast.makeText(this, error, Toast.LENGTH_LONG).show();
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == RECOVERY_REQUEST) {
        // Retry initialization if user performed a recovery action
        getYouTubePlayerProvider().initialize(Config.YOUTUBE_API_KEY, this);
    }
}

protected Provider getYouTubePlayerProvider() {
    return youTubeView;
}
```

# Intégrer Data API V3 You Tube dans votre Application Android

## ► Ajout de la class suivante dans la class MainActivity

```
private final class MyPlaybackEventListener implements YouTubePlayer.PlaybackEventListener {  
  
    @Override  
    public void onPlaying() {  
        // Called when playback starts, either due to user action or call to play().  
        showMessage("Playing");  
    }  
  
    @Override  
    public void onPaused() {  
        // Called when playback is paused, either due to user action or call to pause().  
        showMessage("Paused");  
    }  
  
    @Override  
    public void onStopped() {  
        // Called when playback stops for a reason other than being paused.  
        showMessage("Stopped");  
    }  
  
    @Override  
    public void onBuffering(boolean b) {  
        // Called when buffering starts or ends.  
    }  
  
    @Override  
    public void onSeekTo(int i) {  
        // Called when a jump in playback position occurs, either  
        // due to user scrubbing or call to seekRelativeMillis() or seekToMillis()  
    }  
}
```

# Intégrer Data API V3 You Tube dans votre Application Android

## ► Ajout de la class suivante dans le MainActivity

```
private final class MyPlayerStateChangeListener implements YouTubePlayer.PlayerStateChangeListener {

    @Override
    public void onLoading() {
        // Called when the player is loading a video
        // At this point, it's not ready to accept commands affecting playback such as play() or pause()
    }

    @Override
    public void onLoaded(String s) {
        // Called when a video is done loading.
        // Playback methods such as play(), pause() or seekToMillis(int) may be called after this callback.
    }

    @Override
    public void onAdStarted() {
        // Called when playback of an advertisement starts.
    }

    @Override
    public void onVideoStarted() {
        // Called when playback of the video starts.
    }

    @Override
    public void onVideoEnded() {
        // Called when the video reaches its end.
    }

    @Override
    public void onError(YouTubePlayer.ErrorReason errorReason) {
        // Called when an error occurs.
    }
}
```



# Intégrer Data API V3 You Tube dans votre Application Android

- ▶ Capture d'écran de ce que vous devriez obtenir!(Sans les boutons sociaux!)



# Intégrer Data API V3 You Tube dans votre Application Android

- ▶ Je m'abonne à la chaîne You Tube Vertin Go Website pour être constamment informé des derniers tutoriels techniques Vertin Go Website!

