



UA.MASTER MÓVILES

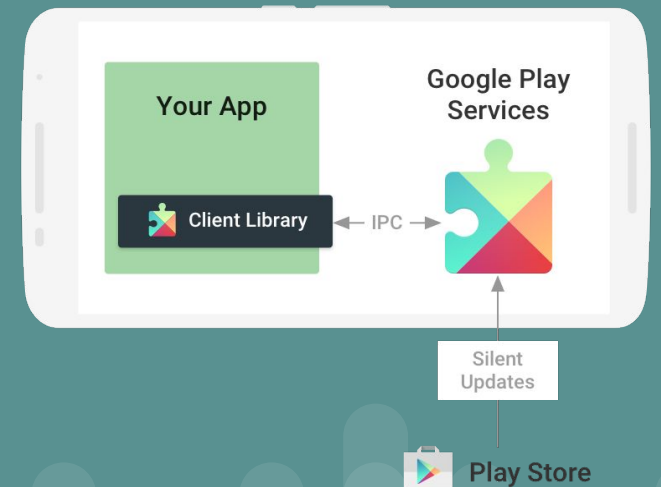
SERVICIOS DE LAS PLATAFORMAS MÓVILES

Servicios de Google Play



Puntos a tratar

- Visión general de los Servicios de Google Play
- Configuración
- Cómo acceder a las APIs
- APIs más comunes



<https://developers.google.com/android/guides/overview>

Visión general de los Servicios de Google Play

- **¿Qué son los servicios de Google Play?**

Son el nexo de unión entre Android y las APIs de Google


- **¿Cómo funcionan los servicios de Google Play?**

Es una librería creada por Google para que los desarrolladores pueden aprovechar las APIs y servicios de Google fácilmente.

Viene preinstalada en la mayoría de dispositivos Android con Google Play.

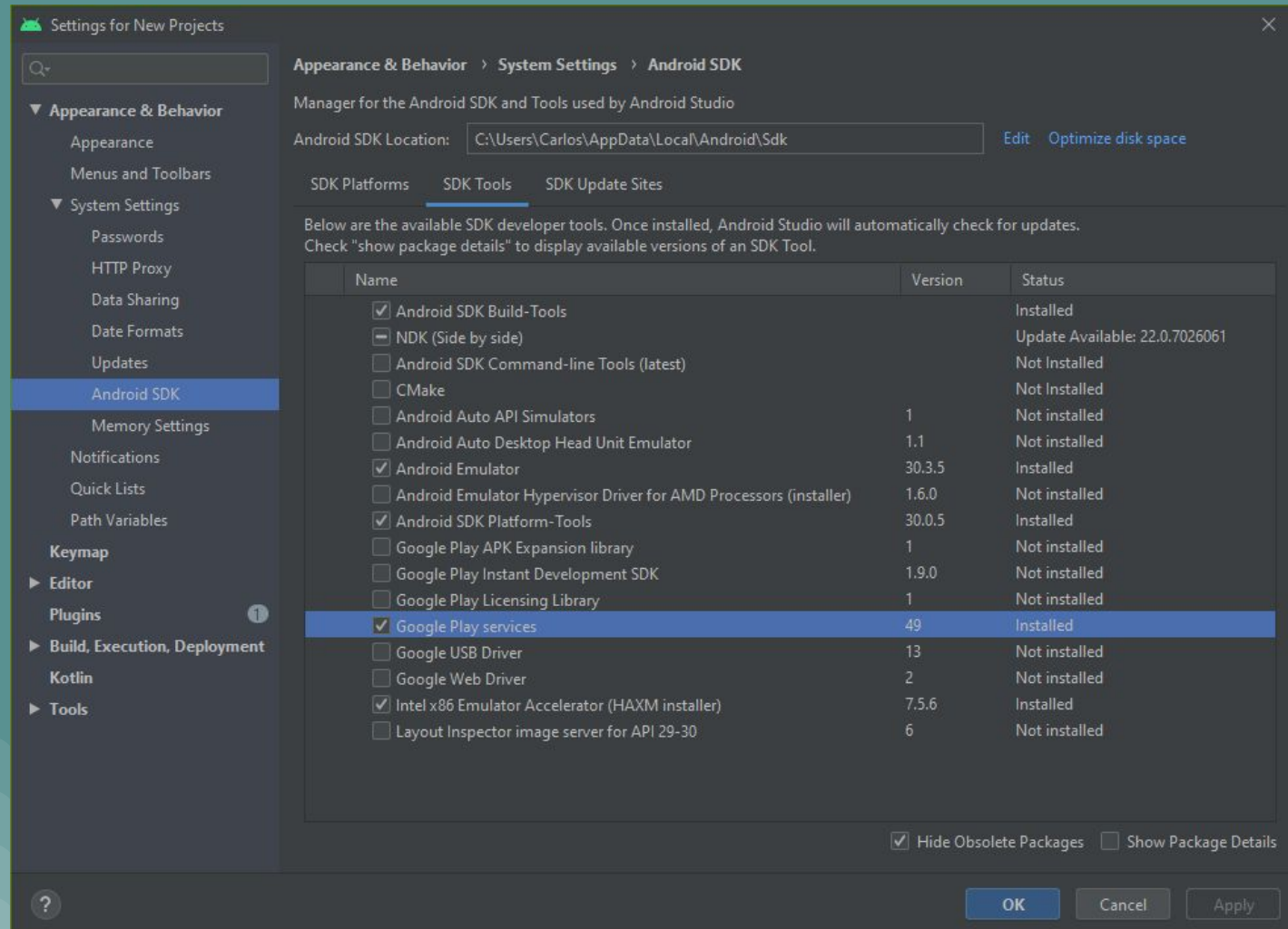
- **¿Para qué sirven los servicios de Google Play?**

Supone un modo estandarizado y abstracto de acceder a funciones avanzadas como ubicación, mapas o inicio de sesión sin necesidad de incluir el código en cada aplicación.



Configuración

Integración con Android Studio:



Configuración

Añadir los servicios de Google Play al proyecto:

- Asegurarnos que el fichero build.gradle a nivel de proyecto tiene una referencia al repositorio de `google()` o a maven `{url:"https://maven.google.com"}`

```
allprojects {  
    repositories {  
        google()  
        jcenter()  
  
        // If you are using a version of gradle lower than 4.1 you need to use:  
        // maven { url "https://maven.google.com" }  
    }  
}
```

- Añadiendo una nueva regla por cada una de las APIs que queramos utilizar dentro del fichero build.grade del módulo.

```
apply plugin: 'com.android.application'  
  
dependencies {  
    implementation "com.google.android.gms:play-services-..."  
}
```

Configuración

Añadir los servicios de Google Play al proyecto:

Servicios disponibles:

Google Mobile Ads	<code>com.google.android.gms:play-services-ads:xx.xx.xx</code>
Android Advertising ID (AAID)	<code>com.google.android.gms:play-services-ads-identifier:xx.xx.xx</code>
Lightweight version of Google Mobile Ads	<code>com.google.android.gms:play-services-ads-lite:xx.xx.xx</code>
AdSense for Search (AFS) Custom Search Ads (CSA)	<code>com.google.android.gms:play-services-afs-native:xx.xx.xx</code>
Google Analytics Services SDK for Android	<code>com.google.android.gms:play-services-analytics:xx.xx.xx</code>
Google Sign-In for Android	<code>com.google.android.gms:play-services-auth:xx.xx.xx</code>
SMS Retriever API	<code>com.google.android.gms:play-services-auth-api-phone:xx.xx.xx</code>
Google Awareness API	<code>com.google.android.gms:play-services-awareness:xx.xx.xx</code>
Utility classes	<code>com.google.android.gms:play-services-base:xx.xx.xx</code>
More utility classes	<code>com.google.android.gms:play-services-basement:xx.xx.xx</code>
Google Cast	<code>com.google.android.gms:play-services-cast:xx.xx.xx</code>
Google Cast Application Framework (CAF)	<code>com.google.android.gms:play-services-cast-framework:xx.xx.xx</code>
Android TV Receivers	<code>com.google.android.gms:play-services-cast-tv:xx.xx.xx</code>
Chromium network stack (Cronet)	<code>com.google.android.gms:play-services-cronet:xx.xx.xx</code>
Fast IDentity Online (FIDO) Authentication	<code>com.google.android.gms:play-services-fido:xx.xx.xx</code>
Google Fit API for Android	<code>com.google.android.gms:play-services-fitness:xx.xx.xx</code>
Google Play Games Services for Android	<code>com.google.android.gms:play-services-games:xx.xx.xx</code>

Configuración

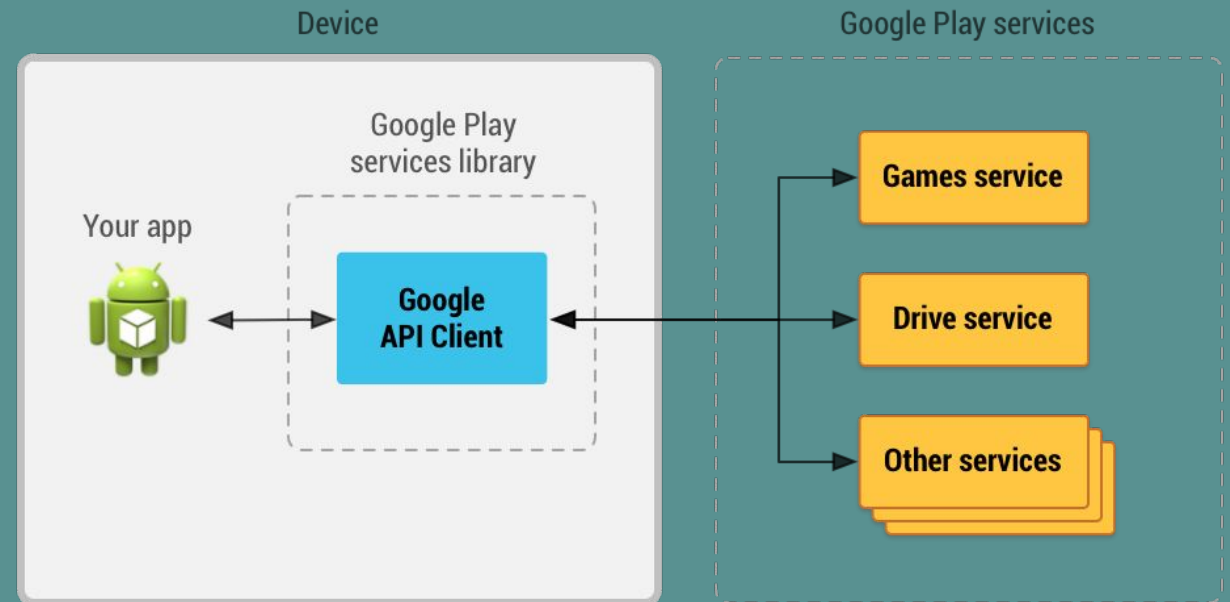
Añadir los servicios de Google Play al proyecto:

Servicios disponibles:

Google Play Instant	<code>com.google.android.gms:play-services-instantapps:xx.xx.xx</code>
Fused Location Provider for Android (includes activity recognition and geofencing)	<code>com.google.android.gms:play-services-location:xx.xx.xx</code>
Google Maps SDK for Android	<code>com.google.android.gms:play-services-maps:xx.xx.xx</code>
ML Kit barcode scanning	<code>com.google.android.gms:play-services-mlkit-barcode-scanning:xx.xx.xx</code>
ML Kit face detection	<code>com.google.android.gms:play-services-mlkit-face-detection:xx.xx.xx</code>
ML Kit image labeling	<code>com.google.android.gms:play-services-mlkit-image-labeling:xx.xx.xx</code>
ML Kit text recognition	<code>com.google.android.gms:play-services-mlkit-text-recognition:xx.xx.xx</code>
Nearby device discovery and connection	<code>com.google.android.gms:play-services-nearby:xx.xx.xx</code>
Show open-source licenses	<code>com.google.android.gms:play-services-oss-licenses:xx.xx.xx</code>
Screen lock quality check	<code>com.google.android.gms:play-services-password-complexity:xx.xx.xx</code>
reCAPTCHA Enterprise for app instrumentation	<code>com.google.android.gms:play-services-recaptcha:xx.xx.xx</code>
SafetyNet APIs	<code>com.google.android.gms:play-services-safetynet:xx.xx.xx</code>
Google Tag Manager	<code>com.google.android.gms:play-services-tagmanager:xx.xx.xx</code>
Tasks API on Android	<code>com.google.android.gms:play-services-tasks:xx.xx.xx</code>
Mobile Vision	<code>com.google.android.gms:play-services-vision:xx.xx.xx</code>
Google Pay for Payments on Android	<code>com.google.android.gms:play-services-wallet:xx.xx.xx</code>
Wearable Data Layer API	<code>com.google.android.gms:play-services-wearable:xx.xx.xx</code>

Cómo acceder a las APIs

- A través de una instancia de uno de los objetos cliente de la API, que son subclases de GoogleApi.
- Hay servicios que requieren un inicio de sesión por parte del usuario que autorice a nuestra aplicación, y otros que no.
- También hay servicios que requieren la identificación de la aplicación por medio del certificado de firma de la misma.



Cómo acceder a las APIs

Antes de utilizar la API para Android de Google Play Services es conveniente comprobar su disponibilidad:

```
boolean checkGooglePlayServices(Context context) {
    GoogleApiAvailability googleAPI = GoogleApiAvailability.getInstance();
    int result = googleAPI.isGooglePlayServicesAvailable(context);
    if(result != ConnectionResult.SUCCESS) {
        Log.d("GPS", "Google Play Services not available");
        if(googleAPI.isUserResolvableError(result)) {
            googleAPI.getErrorDialog(this, result, PLAY_SERVICES_RESOLUTION_REQUEST).show();
        }
        else {
            Log.d("GPS", "This device is not supported");
        }

        return false;
    }

    return true;
}
```

Cómo acceder a las APIs

Si el API no requiere autenticación, podemos usar directamente la API, obtenemos la instancia del objeto cliente pasándole el contexto o la activity actual:

```
FusedLocationProviderClient client = LocationServices.getFusedLocationProviderClient(this);

// Get the last known location
client.getLastLocation()
    .addOnCompleteListener(this, new OnCompleteListener<Location>() {
        @Override
        public void onComplete(@NonNull Task<Location> task) {
            // ...
        }
    });
```

Cómo acceder a las APIs

Si el API requiere autenticación, primero iniciamos sesión con el usuario y solicitamos permiso para acceder al servicio. Después obtenemos la instancia del objeto cliente pasándole el contexto o la activity actual:

```
// This account must have the necessary scopes to make the API call
// See https://developers.google.com/identity/sign-in/android/
GoogleSignInAccount account = GoogleSignIn.getLastSignedInAccount(this);

// Get the app's Drive folder
DriveResourceClient client = Drive.getDriveResourceClient(this, account);
client.getAppFolder().
    .addOnCompleteListener(this, new OnCompleteListener<DriveFolder>() {
        @Override
        public void onComplete(@NonNull Task<DriveFolder>() {
            // ...
        }
    });
```

APIs

Google Mobile Ads

SDK de anuncios para móviles.

com.google.android.gms:play-services-ads:xx.xx.xx

```
<manifest>
  <application>
    <meta-data
      android:name="com.google.android.gms.ads.APPLICATION_ID"
      android:value="ca-app-pub-xxxxxxxxxxxxxxxx~yyyyyyyyyy"/>
  </application>
</manifest>
```

Nota: En tus aplicaciones reales, es importante que se use un ID real de AdMob, no el que se indica anteriormente. Si solo quieres experimentar con el SDK, puedes usar el ID de aplicación de muestra que aparece más arriba.

Importante: Este paso es obligatorio a partir de la versión 17.0.0 del SDK de anuncios de Google para móviles. Si no se añade la etiqueta <meta-data> , se producirá un error y se mostrará el mensaje The Google Mobile Ads SDK was initialized incorrectly.

<https://developers.google.com/admob/android/quick-start>

APIs

Google Mobile Ads

SDK de anuncios para móviles

```
import com.google.android.gms.ads.MobileAds;

class MainActivity : AppCompatActivity() {
    ...
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        MobileAds.initialize(this) {}
    }
    ...
}
```

APIs

Lightweight version of Google Mobile Ads

Es una versión mucho más ligera del SDK de anuncios para móviles.

```
com.google.android.gms:play-services-ads-lite:xx.xx.xx
```

La biblioteca de cliente de Servicios de Google Play incluye el SDK estándar de anuncios de Google, pero no el SDK Lite, por lo que hay que comprobar que esté disponible. El SDK compara las versiones de la biblioteca de cliente y el APK de Servicios de Google Play, y usa la más reciente.

Nota: El SDK Lite solo debe usarse en aplicaciones distribuidas a través de **Google Play Store**.

<https://developers.google.com/admob/android/lite-sdk>

APIs

Android Advertising ID (AAID)

Biblioteca de ayuda para la recuperación del ID de la publicidad y la información relacionada, como la configuración del seguimiento de la publicidad limitada.

com.google.android.gms:play-services-ads-identifier:xx.xx.xx

```
String GoogleAAID;

class GetGoogleAAIDTask extends AsyncTask<String, Integer, String> {
    @Override
    protected String doInBackground(String... strings) {
        AdvertisingIdClient.Info adInfo = null;
        try {
            adInfo = AdvertisingIdClient.getAdvertisingIdInfo(MainActivity.this(getApplicationContext()));
            if (adInfo.isLimitAdTrackingEnabled()) // check if user has opted out of tracking
                return "Tracking limited";
        } catch (IOException | GooglePlayServicesNotAvailableException | GooglePlayServicesRepairableException e) {
            e.printStackTrace();
            return "Error getting tracker";
        }

        return adInfo.getId();
    }
    @Override
    protected void onPostExecute(String s) { GoogleAAID = s; }
}
```

<https://developers.google.com/android/reference/com/google/android/gms/ads/identifier/AdvertisingIdClient>

APIs

AdSense for Search (AFS) Custom Search Ads (CSA)

Monetiza los resultados de búsqueda en nuestra app

```
com.google.android.gms:play-services-afs-native:xx.xx.xx
```

<https://developers.google.com/custom-search-ads>

APIs

Google Analytics Services SDK for Android

Alternativa reducida a Firebase Analytics

`com.google.android.gms:play-services-analytics:xx.xx.xx`

Requiere añadir los siguientes permisos en el Manifest:

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

Requiere añadir el fichero `app/src/res/xml/global_tracker.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="ga_trackingId" translatable="false">${YOUR_TRACKING_ID}</string>
</resources>
```

<https://developers.google.com/analytics/devguides/collection/android/v4>

APIs

Google Analytics Services SDK for Android

Alternativa reducida a Firebase Analytics

`com.google.android.gms:play-services-analytics:xx.xx.xx`

<https://developers.google.com/analytics/devguides/collection/android/v4>

APIs

Google Analytics Services SDK for Android

Alternativa reducida a Firebase Analytics

```
import com.google.android.gms.analytics.GoogleAnalytics;
import com.google.android.gms.analytics.Tracker;

class AnalyticsApplication : Application() {
    companion object {
        private var sAnalytics: GoogleAnalytics? = null
        private var sTracker: Tracker? = null
    }

    override fun onCreate() {
        super.onCreate()
        sAnalytics = GoogleAnalytics.getInstance(this)
    }

    @Synchronized
    fun getDefaultTracker(): Tracker? {
        // To enable debug logging use: adb shell setprop log.tag.GAv4 DEBUG
        if (sTracker == null) {
            sTracker = sAnalytics.newTracker(R.xml.global_tracker)
        }
        return sTracker
    }
}
```

APIs

Google Sign-In for Android

Permite que los usuarios accedan a nuestras aplicaciones de forma rápida y segura mediante un sistema de registro que ya conocen, utilizan y en el que confían: su cuenta de Google.

`com.google.android.gms:play-services-auth:xx.xx.xx`

```
// Configure sign-in to request the user's ID, email address, and basic
// profile. ID and basic profile are included in DEFAULT_SIGN_IN.
val gso = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestEmail()
    .build()

// Build a GoogleSignInClient with the options specified by gso.
mGoogleSignInClient = GoogleSignIn.getClient(this, gso)

// Then, when the sign-in button is clicked, start the sign-in intent:
private fun signIn() {
    val signInIntent = mGoogleSignInClient.signInIntent()
    startActivityForResult(signInIntent, RC_SIGN_IN)
}
```

<https://developers.google.com/identity/sign-in/android>

APIs

Google Sign-In for Android

Permite que los usuarios accedan a nuestras aplicaciones de forma rápida y segura mediante un sistema de registro que ya utilizan y en el que confían: su cuenta de Google.

```
// Finally, handle the activity result:

fun onActivityResult(requestCode:Int, resultCode:Int, data:Intent) {
    super.onActivityResult(requestCode, resultCode, data)
    // Result returned from launching the Intent from GoogleSignInClient.getSignInIntent(...);
    if (requestCode == RC_SIGN_IN) {
        // The Task returned from this call is always completed, no need to attach a listener.
        val task = GoogleSignIn.getSignedInAccountFromIntent(data)
        handleSignInResult(task)
    }
}

private fun handleSignInResult(completedTask:Task<GoogleSignInAccount>) {
    try {
        val account = completedTask.getResult(ApiException::class.java)
        // Signed in successfully, show authenticated UI.
        updateUI(account)
    } catch (e:ApiException) {
        // The ApiException status code indicates the detailed failure reason.
        // Please refer to the GoogleSignInStatusCodes class reference for more information.
        Log.w(TAG, "signInResult:failed code=" + e.getStatusCode())
        updateUI(null)
    }
}
```

APIs

SMS Retriever API

Verifica automáticamente números de teléfono.

Se deben implementar las partes cliente y servidor del flujo de verificación.

`com.google.android.gms:play-services-auth-api-phone:xx.xx.xx`

<https://developers.google.com/identity/sms-retriever/overview>

APIs

Google Awareness API

Permite que las aplicaciones reaccionen de manera inteligente a la situación actual del usuario. Se pueden establecer alarmas que incluyan combinaciones de los siguientes eventos:

- Hora
- Ubicación
- Actividad (andando, corriendo, en bici)
- Balizas cercanas (estamos cerca de un determinado lugar)
- Localización (longitud/latitud)
- Lugares (deprecated) Nuevo API: [Places SDK for Android](#)
- Auriculares (el usuario ha enchufado los auriculares)
- Clima (deprecated)

`com.google.android.gms:play-services-awareness:xx.xx.xx`

<https://developers.google.com/awareness>

APIs

Google Cast SDK

Permite extender aplicaciones (Android, iOS o Chrome) para controlar transmisiones de video y audio a un televisor o sistema de sonido. Su aplicación se convierte en el control remoto para reproducir, pausar, buscar, rebobinar, detener y controlar los medios.

`com.google.android.gms:play-services-cast:xx.xx.xx`

<https://developers.google.com/cast>

`com.google.android.gms:play-services-cast-framework:xx.xx.xx`

<https://developers.google.com/cast/docs/developers>

Android TV Receiver Overview

Permite que las aplicaciones se comuniquen con las aplicaciones de Android TV a través del protocolo Cast.

`com.google.android.gms:play-services-cast-tv:xx.xx.xx`

https://developers.google.com/cast/docs/android_tv_receiver

APIs

Cronet: Chromium network stack

Cronet es la pila de red de Chromium que está disponible para las apps de Android como una biblioteca. Cronet aprovecha las múltiples tecnologías que reducen la latencia y aumentan el rendimiento de las solicitudes de red que tu app necesita para funcionar.

`com.google.android.gms:play-services-cronet:xx.xx.xx`

<https://developer.android.com/guide/topics/connectivity/cronet>

APIs

FIDO: Fast IDentity Online Authentication

Es un conjunto de estándares para una autenticación rápida, simple y sólida desarrollados por FIDO Alliance, una asociación de la industria con representantes de una variedad de organizaciones, incluidas Google, Microsoft, Mozilla y Yubico.

Los estándares permiten la autenticación multifactor, sin contraseña y resistente al phishing. Mejoran la experiencia de usuario en línea al hacer que la autenticación sólida sea más fácil de implementar y usar.

`com.google.android.gms:play-services-fido:xx.xx.xx`

<https://developers.google.com/identity/fido>

APIs

Google Fit API

Incluye soporte para sensores en el dispositivo móvil y sensores Bluetooth de baja energía emparejados con el dispositivo. Google Fit permite a los desarrolladores implementar soporte para otros sensores y exponerlos para otras aplicaciones Android. Los sensores compatibles con Google Fit están disponibles para las aplicaciones de Android como fuentes de datos.

`com.google.android.gms:play-services-fitness:xx.xx.xx`

<https://developers.google.com/fit/android>

APIs

Google Play Games Services

Proporciona servicios de juegos de Google Play multiplataforma que le permiten integrar fácilmente funciones de juegos como logros, tablas de clasificación y partidas guardadas localmente.

`com.google.android.gms:play-services-games:xx.xx.xx`

<https://developers.google.com/games/services/android/quickstart>

APIs

Google Play Instant

Permite iniciar aplicaciones y juegos nativos en dispositivos que ejecutan Android 5.0 (nivel de API 21) o versiones posteriores sin que estén instalados.

Tu app o juego debe tener menos de 15 MB para poder habilitar Google Play Instant.

`com.google.android.gms:play-services-instantapps:xx.xx.xx`

<https://developer.android.com/topic/google-play-instant/overview>

APIs

Fused Location Provider

Proporciona una mejor localización del usuario combinando diferentes sensores para ahorrar energía.

Ejemplo: Para seguir el movimiento de un individuo que camina hacia un edificio, entra en él y sale. En este escenario, cuando la persona está en el exterior, el servicio GPS funcionará bien. Sin embargo, cuando entra en el edificio, la señal GPS puede caer. Este API maneja este escenario rastreando la ubicación en el exterior a través del GPS y cuando la señal del GPS cae dentro del edificio, utiliza el Wi-Fi. Durante el proceso de cambio de GPS a Wi-Fi, para proporcionar una transición más suave, se utilizan los sensores internos del dispositivo para predecir el movimiento.

`com.google.android.gms:play-services-location:xx.xx.xx`

<https://developer.android.com/training/location>

APIs

Fused Location Provider

Proporciona una mejor localización del usuario combinando diferentes sensores para ahorrar energía.

```
FusedLocationProviderClient client = LocationServices.getFusedLocationProviderClient(this);

// Get the last known location
client.getLastLocation()
    .addOnCompleteListener(this, new OnCompleteListener<Location>() {
        @Override
        public void onComplete(@NonNull Task<Location> task) {
            // ...
        }
    });
```

APIs

Google Maps

Permite añadir a tu aplicación mapas basados en los datos de Google Maps. La API gestiona automáticamente el acceso a los servidores de Google Maps, la descarga de datos, la visualización de mapas y la respuesta a los gestos del mapa. También puedes utilizar las llamadas a la API para añadir marcadores, polígonos y superposiciones a un mapa básico, y para cambiar la vista del usuario de una zona del mapa en particular.

La API permite añadir estos gráficos a un mapa:

- Iconos anclados a posiciones específicas en el mapa (Marcadores).
- Conjuntos de segmentos de líneas abiertos (Polilíneas) o cerrados (Polígonos).
- Imágenes ancladas en posiciones específicas del mapa (Ground Overlays).
- Conjuntos de imágenes que se muestran sobre los mosaicos del mapa base (Tile Overlays).

`com.google.android.gms:play-services-maps:xx.xx.xx`

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

APIs

ML Kit: Barcode scanning

Permite reconocer y decodificar códigos de barras.

Los siguientes formatos están soportados:

- Code 128
- Code 39
- Code 93
- Codabar
- EAN-13
- EAN-8
- ITF
- UPC-A
- UPC-E
- QR Code
- PDF417
- Aztec
- Data Matrix



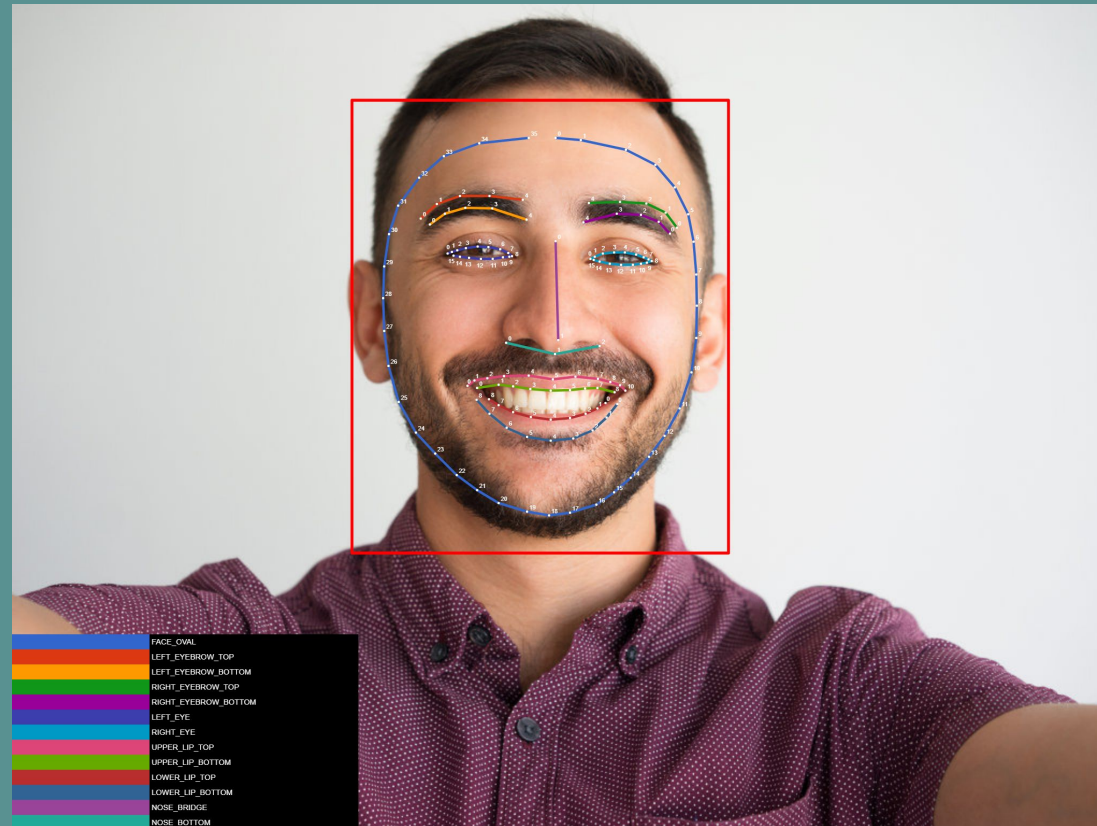
com.google.android.gms:play-services-mlkit-barcode-scanning:xx.xx.xx

<https://developers.google.com/ml-kit/vision/barcode-scanning/android>

APIs

ML Kit: Face Detection

Permite detectar caras en imágenes y vídeo extrayendo características.



com.google.android.gms:play-services-mlkit-face-detection:xx.xx.xx

<https://developers.google.com/ml-kit/vision/face-detection/android>

APIs

ML Kit: Image Labeling

Permite detectar y extraer información sobre entidades en una imagen a través de un amplio grupo de categorías (400 en 2020). El modelo de etiquetado de imágenes por defecto puede identificar objetos generales, lugares, actividades, especies animales, productos, etc.

Permite especificar el modelo de clasificación (TensorFlow) desde diferentes fuentes para cumplir con los requisitos específicos del usuario.

Ejemplo:



1. Estadio (Confianza: 0.9205354)
2. Deportes (Confianza: 0.7531109)
3. Evento (Confianza: 0,66905296)
4. Ocio (Confianza: 0,59904146)
5. Fútbol (Confianza: 0,56384534)
6. Red (Confianza: 0,54679185)
7. Planta (Confianza: 0,524364)

com.google.android.gms:play-services-mlkit-image-labeling:xx.xx.xx

<https://developers.google.com/ml-kit/vision/image-labeling>

APIs

ML Kit: Text Recognition

Permite reconocer texto en imágenes.



La API de reconocimiento de texto utiliza un modelo desagregado que debe ser descargado. Se tiene la opción de realizar la descarga cuando se instala la app o cuando se lanza por primera vez.

`com.google.android.gms:play-services-mlkit-text-recognition:xx.xx.xx`

<https://developers.google.com/ml-kit/vision/text-recognition/android>

APIs

Nearby device discovery and connection

Permite descubrir dispositivos cercanos y establecer comunicación con ellos. Utiliza tecnologías como Bluetooth, Wi-Fi, IP y audio.

Incluye las APIs:

- Nearby Connection: <https://developers.google.com/nearby/connections/overview>
Permite que las aplicaciones descubran, se conecten e intercambien datos fácilmente con los dispositivos cercanos en tiempo real, independientemente de la conectividad de la red.
- Nearby Messages: <https://developers.google.com/nearby/messages/overview>
Permite pasar pequeñas cargas binarias entre dispositivos Android e iOS conectados a Internet.
- Fast Pair: <https://developers.google.com/nearby/fast-pair/spec>
Utiliza la tecnología Bluetooth Low Energy (BLE) para descubrir dispositivos Bluetooth cercanos sin necesidad de utilizar una gran cantidad de batería.

`com.google.android.gms:play-services-nearby:xx.xx.xx`

<https://developers.google.com/nearby>

APIs

Show open-source licenses

Incluye un conjunto de herramientas diseñadas para dar a los desarrolladores una forma sencilla de mostrar las licencias de código abierto de las bibliotecas que utilizada en sus aplicaciones.

Incluye un plugin de Gradle que recoge las licencias de las bibliotecas incluidas, tal y como se declaran en los archivos POM, y una actividad que se puede utilizarse para mostrar estas licencias.

```
com.google.android.gms:play-services-oss-licenses:xx.xx.xx
```

<https://developers.google.com/android/guides/opensource>

APIs

Screen lock quality check (Enterprise API*)

A partir de Android 10, las aplicaciones con funciones críticas que requieren un bloqueo de pantalla pueden consultar la complejidad del bloqueo de pantalla de un dispositivo o perfil de trabajo. Las aplicaciones que necesitan un bloqueo de pantalla más fuerte pueden dirigir al usuario a los ajustes de bloqueo de pantalla del sistema, lo que les permite actualizar sus ajustes de seguridad.

`com.google.android.gms:play-services-password-complexity:xx.xx.xx`

https://developer.android.com/work/versions/android-10#screen_lock_quality_check

(*)Android Enterprise es una iniciativa liderada por Google para permitir el uso de dispositivos y aplicaciones Android en el lugar de trabajo. El programa ofrece APIs y otras herramientas para que los desarrolladores integren la compatibilidad con Android en sus soluciones de gestión de la movilidad empresarial.

<https://developers.google.com/android/work/overview>

APIs

reCAPTCHA (Enterprise API*)

Devuelve una puntuación basada en la interacción con su sitio web, siendo 1.0 una probable buena interacción y 0.0 una probable acción abusiva.

Basándose en la puntuación de reCAPTCHA Enterprise, se pueden tomar medidas. Por ejemplo, con una puntuación baja, puede exigir una autenticación de dos factores, o una verificación por correo electrónico, para permitir que el usuario continúe.

`com.google.android.gms:play-services-recaptcha:xx.xx.xx`

<https://cloud.google.com/recaptcha-enterprise>

APIs

SafetyNet

Proporciona un conjunto de servicios que ayudan a proteger tu app contra amenazas de seguridad, lo que incluye la manipulación del dispositivo, URL incorrectas, apps potencialmente dañinas y usuarios falsos.

`com.google.android.gms:play-services-safetynet:xx.xx.xx`

<https://developer.android.com/training/safetynet>

APIs

Tag Manager

Permite implementar y gestionar etiquetas de medición en sus aplicaciones móviles sin tener que reconstruir y volver a enviar los binarios de la aplicación al store. Los desarrolladores que trabajan con Firebase Analytics pueden añadir fácilmente Google Tag Manager para ayudar a gestionar y realizar cambios en la implementación, incluso después de que la aplicación se haya enviado.

Los desarrolladores pueden registrar eventos importantes y decidir más tarde qué etiquetas deben activarse.

`com.google.android.gms:play-services-tagmanager:xx.xx.xx`

<https://developers.google.com/tag-manager/android/v5>

APIs

Tasks

A partir de la versión 9.0.0 de Google Play Services, se puede utilizar una API de tareas y una serie de métodos que devuelven tareas que representan llamadas a métodos asíncronos, similar a PendingResult en versiones anteriores de Google Play Services.

`com.google.android.gms:play-services-tasks:xx.xx.xx`

<https://developers.google.com/android/guides/tasks>

APIs

Mobile Vision (deprecated. Use ML Kit)

Proporciona un marco para encontrar objetos (caras, códigos de barras y texto) en fotos y vídeos. El marco incluye detectores, que localizan y describen objetos visuales y una API basada en eventos que rastrea la posición de esos objetos en el vídeo.

`com.google.android.gms:play-services-vision:xx.xx.xx`

<https://developers.google.com/vision>

APIs

Google Pay for Payments

Permite un pago rápido y sencillo dentro de tu aplicación, y te da un cómodo acceso a cientos de millones de tarjetas guardadas en las cuentas de Google de todo el mundo.

El flujo de pago es el siguiente:



1. El usuario toca el botón de pago de Google Pay y ve una hoja de pago con una lista de métodos de pago admitidos.
2. El usuario selecciona un método de pago y Google Pay devuelve de forma segura un token de pago para ese método a tu aplicación.
3. Tu aplicación envía el token de pago, junto con los detalles de la compra, a su backend.
4. Para ejecutar el pago, el backend procesa la compra y envía el token de pago al proveedor de servicios de pago.

`com.google.android.gms:play-services-wallet:xx.xx.xx`

<https://developers.google.com/pay/api/android/overview>

APIs

Wearable Data Layer

La API de Wearable Data Layer permite a nuestra aplicación comunicarse con Wear OS mediante un conjunto de objetos de datos que el sistema puede enviar y sincronizar, junto con objetos de escucha que informan a las apps sobre ciertos eventos.

`com.google.android.gms:play-services-wearable:xx.xx.xx`

<https://developer.android.com/training/wearables/data-layer>

Wear OS:

Wear OS (también conocido como Android Wear) es el sistema operativo para dispositivos corporales (wearables) basado en Android que Google presentó en marzo de 2014. El sistema está pensado para ser utilizado en relojes inteligentes (smartwatches), pulseras inteligentes (smartbands), y cualquier otro dispositivo wearable que pueda surgir en el futuro.

<https://developer.android.com/wear>