

# Android et Innovation

Adnane Cabani

D1 272

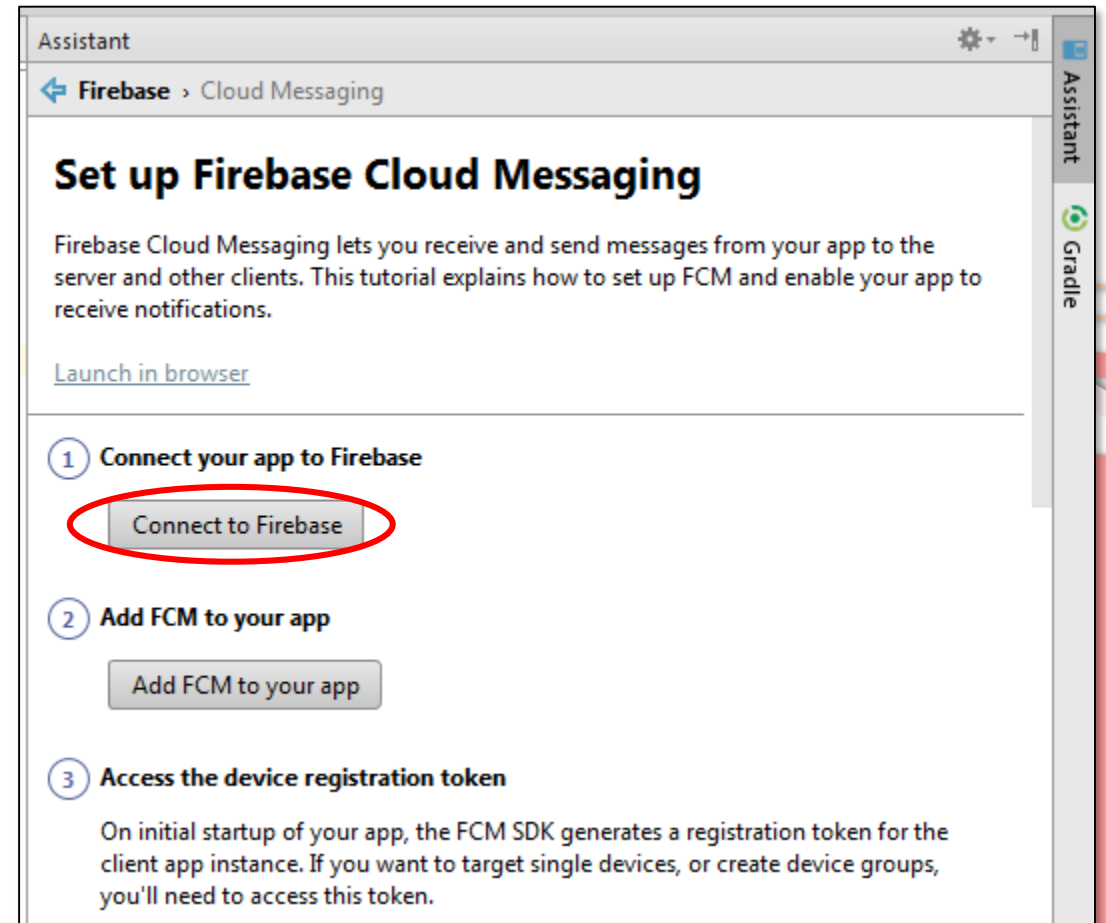
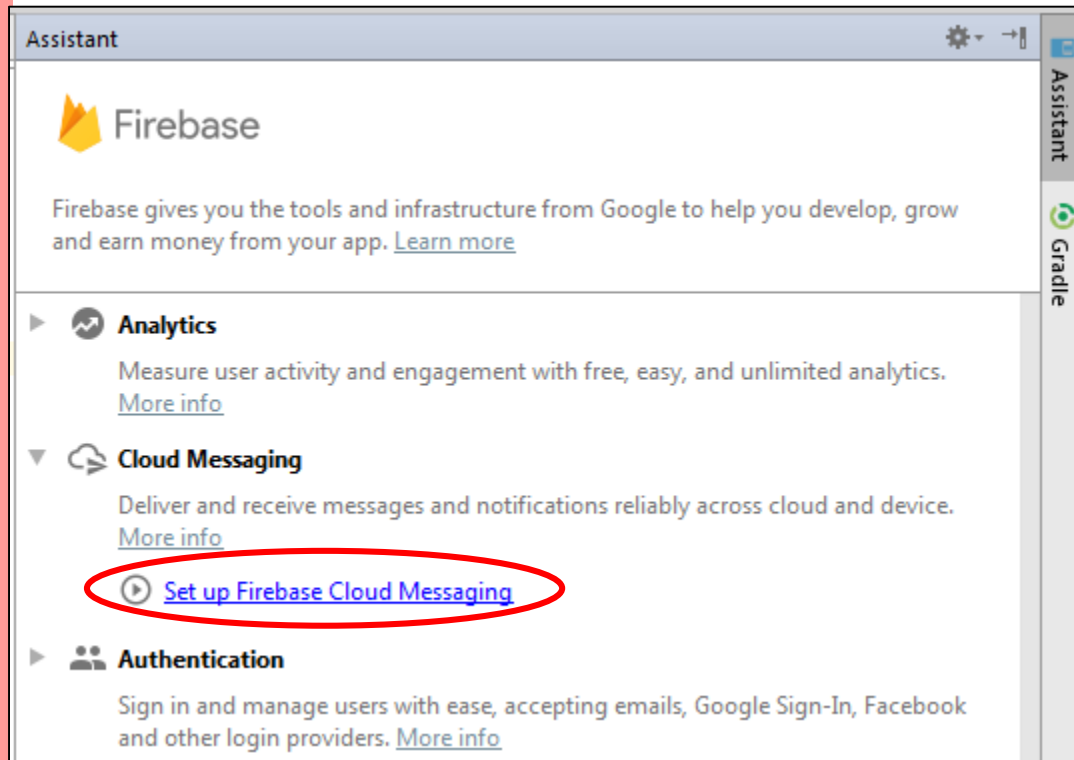
[cabani@esigelec.fr](mailto:cabani@esigelec.fr)

# Firestore Cloud Messaging

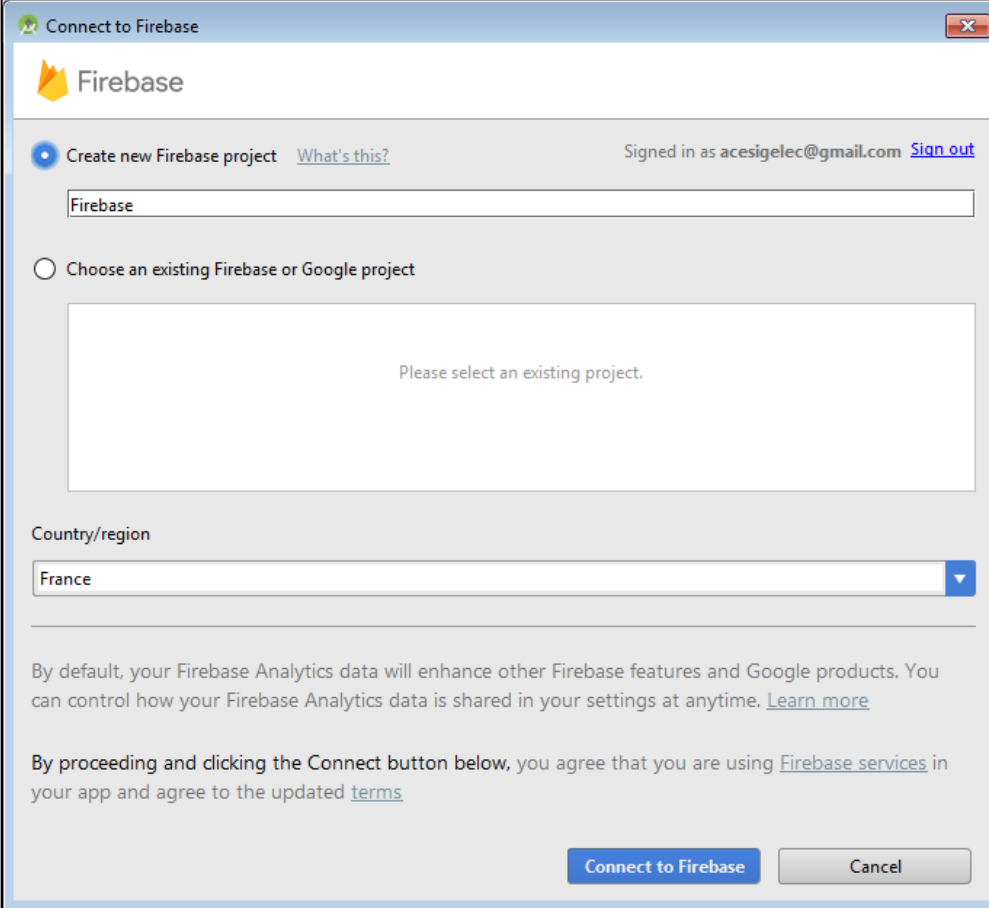
# Firestore Cloud Messaging pour Android

- Permet d'envoyer des données de votre serveur vers vos clients,
- Et de recevoir des messages des clients dans la même connexion
- Utilisée pour les notifications de type *Push*
- Taille maximale du message (données utiles)= 4ko

# Utilisation de l'assistant Android Studio




# Utilisation de l'assistant Android Studio



The screenshot shows the 'Connect to Firebase' dialog box. At the top, it has a title bar 'Connect to Firebase' and a close button. Below the title bar is the Firebase logo and the word 'Firebase'. The main content area has two radio buttons: 'Create new Firebase project' (selected) and 'Choose an existing Firebase or Google project'. The 'Create new Firebase project' option has a text input field containing 'Firebase'. The 'Choose an existing Firebase or Google project' option has a large empty box with the text 'Please select an existing project.' below it. Below these options is a 'Country/region' dropdown menu with 'France' selected. At the bottom, there is a paragraph of text about Firebase Analytics data sharing, followed by a paragraph about agreeing to terms. At the very bottom are two buttons: 'Connect to Firebase' and 'Cancel'.

Connect to Firebase

 Firebase

☒ Create new Firebase project [What's this?](#) Signed in as acesigelec@gmail.com [Sign out](#)

☐ Choose an existing Firebase or Google project

Please select an existing project.

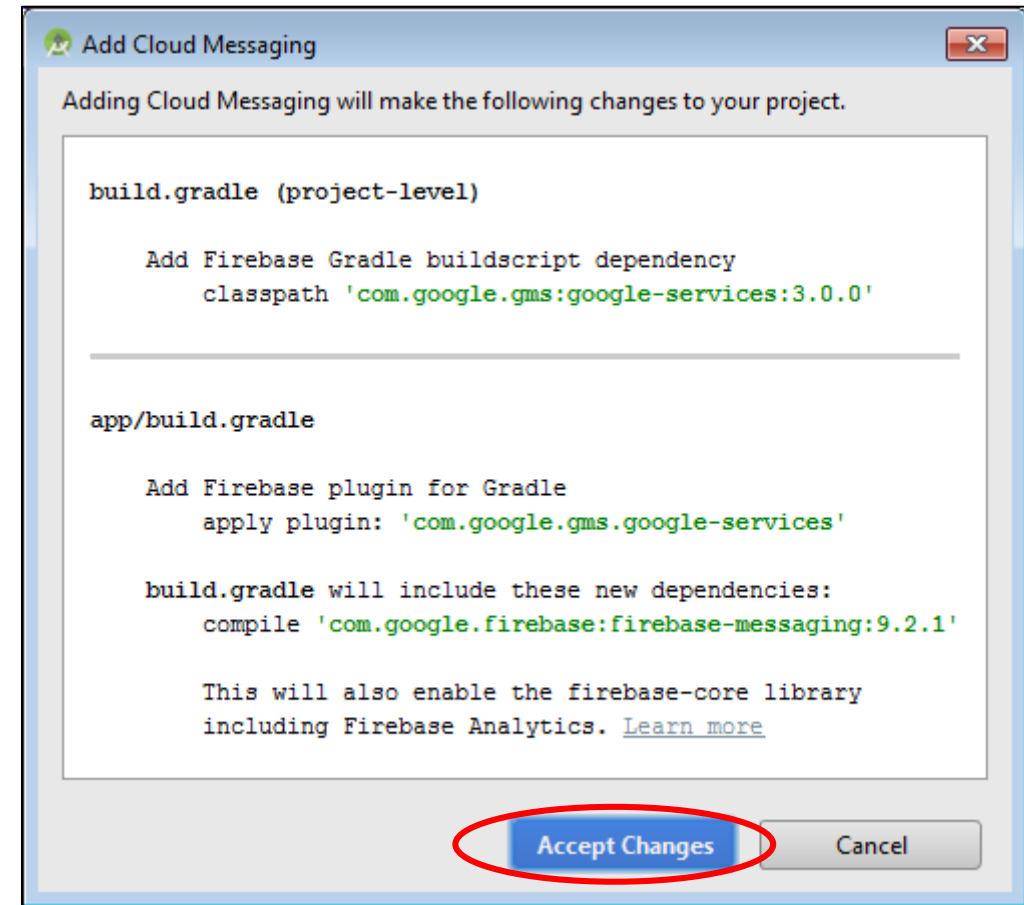
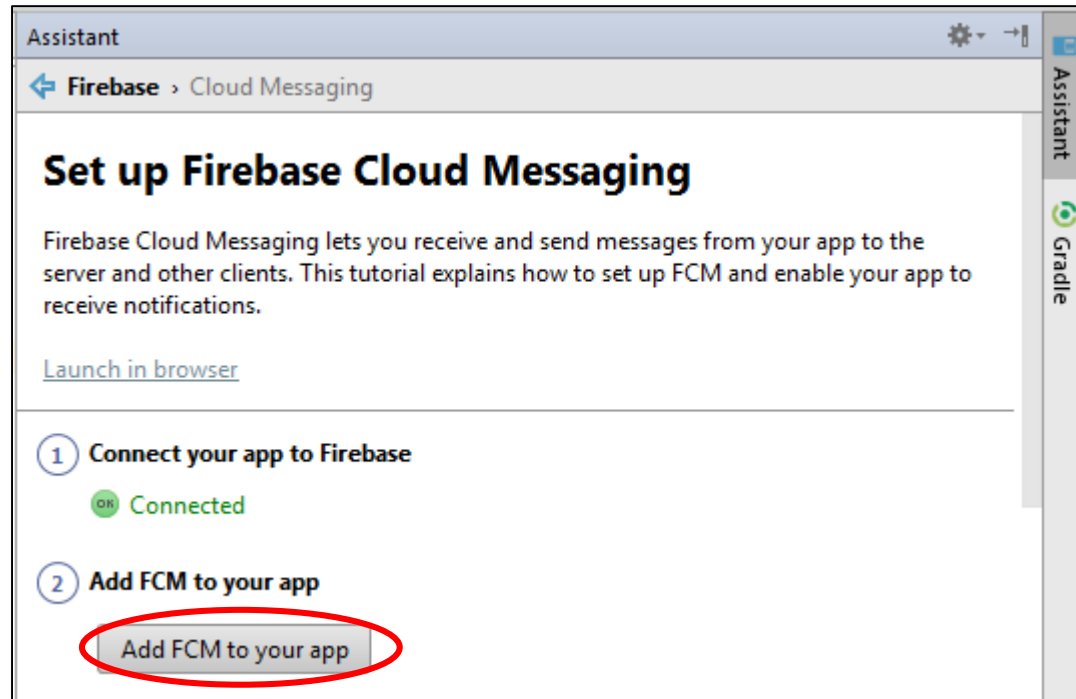
Country/region

By default, your Firebase Analytics data will enhance other Firebase features and Google products. You can control how your Firebase Analytics data is shared in your settings at anytime. [Learn more](#)

By proceeding and clicking the Connect button below, you agree that you are using [Firebase services](#) in your app and agree to the updated [terms](#)

[Connect to Firebase](#) [Cancel](#)

# Utilisation de l'assistant Android Studio



# Récupération du jeton

- Créer une classe qui hérite de *FirebaseInstanceIdService* et redéfinir la méthode *onTokenRefresh()*
- *onTokenRefresh()* est appelé chaque fois qu'un nouveau jeton est généré
- *FirebaseInstanceId.getToken()* retourne null si le jeton n'est pas encore généré
- Envoyer le jeton au serveur d'application

# Récupération du jeton

```
package com.cabani.firebase;

import android.util.Log;

import com.google.firebase.iid.FirebaseInstanceId;
import com.google.firebase.iid.FirebaseInstanceIdService;

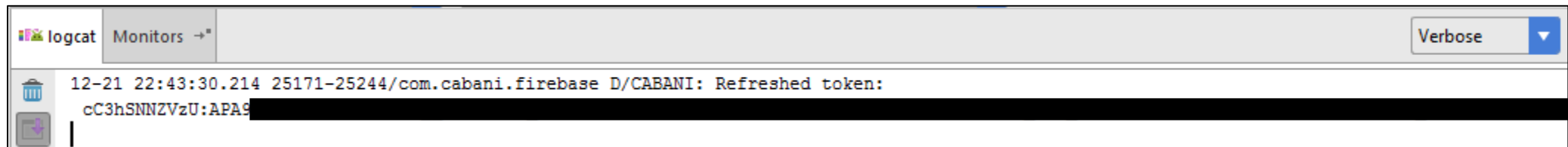
public class MyFirebaseInstanceIdService extends FirebaseInstanceIdService {

    @Override
    public void onTokenRefresh() {

        String refreshedToken = FirebaseInstanceId.getInstance().getToken();
        Log.d("CABANI", "Refreshed token: " + refreshedToken);

    }

}
```





```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.cabani.firebase">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Firebase"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <service
            android:name=".MyFirebaseInstanceIdService">
            <intent-filter>
                <action android:name="com.google.firebase.INSTANCE_ID_EVENT"/>
            </intent-filter>
        </service>

    </application>
</manifest>
```

# La notification

- Pour effectuer un traitement en arrière plan lors de la réception d'une notification, il faut créer un nouveau service.

```
package com.cabani.firebase;

import android.util.Log;

import com.google.firebase.messaging.FirebaseMessagingService;
import com.google.firebase.messaging.RemoteMessage;

public class MyFirebaseMessagingService extends FirebaseMessagingService {

    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        Log.d("CABANI", "From: " + remoteMessage.getFrom());
        Log.d("CABANI", "Notification Message Body: " + remoteMessage.getNotification().getBody());
    }
}
```

```
<service
    android:name=".MyFirebaseMessagingService">
    <intent-filter>
        <action android:name="com.google.firebase.MESSAGING_EVENT"/>
    </intent-filter>
</service>
```

AndroidManifest.xml

# Tester à partir de la console

- Pour tester l'application, aller sur la console Firebase :  
<https://console.firebase.google.com/>
- Sélectionner votre projet
- Sur le menu de gauche, cliquer sur *Notifications*
- Cliquer sur *Envoyer votre premier message*
- Remplir le formulaire
- Cliquer sur *Envoyer le message* puis *envoyer*



# Tester en ligne de commande

- Récupérer la clé du serveur

Console Firebase > Paramètres de l'application > Onglet *Cloud messaging*

- Vérifier la validité de la clé

```
curl
  -H "Authorization: key=YOUR_SERVER_KEY"
  -H Content-Type:"application/json"
  https://fcm.googleapis.com/fcm/send
  -d "{\"registration_ids\": [\"ABC\"]}"
```

- Erreur 401 : La clé n'est pas valide

- Sinon : {

```
  "multicast_id":8825458097146608191,
  "success":0,
  "failure":1,
  "canonical_ids":0,
  "results":[{"error":"InvalidRegistration"}]
}
```



# Tester en ligne de commande

- Envoyer un message

```
curl -X POST
  --header "Authorization: key=YOUR_SERVER_KEY"
  --header "Content-Type: application/json"
  https://fcm.googleapis.com/fcm/send
  -d '{"to":"'DEVICE_REGISTRATION_TOKEN',"priority":"'high',"notification":{"body": "TEST CABANI"}}'
```



# Script php

```
<?php
```

```
define( 'API_ACCESS_KEY', 'AAA [REDACTED]
```

```
$registrationIds = array (  
    "dVpIWqoqfu4:APA91bGAokXA0 [REDACTED]  
    "eslLgz_6Jzc:APA91bGuamnGa [REDACTED]  
    "cSyUNuv7ksg:APA91bFZSZ8-K [REDACTED]
```

```
);
```

```
$msg = array(  
    'body' => 'Body Of Notification',  
    'title' => 'Title Of Notification',  
    'icon' => 'myicon',  
    'sound' => 'mySound'
```

```
);
```

```
$fields = array(  
    'registration_ids' => $registrationIds,  
    'notification' => $msg
```

```
);
```

# Script php

```
$headers = array(
    'Authorization: key=' . API_ACCESS_KEY,
    'Content-Type: application/json'
);

$ch = curl_init();
curl_setopt( $ch, CURLOPT_URL, 'https://fcm.googleapis.com/fcm/send' );
curl_setopt( $ch, CURLOPT_POST, true );
curl_setopt( $ch, CURLOPT_HTTPHEADER, $headers );
curl_setopt( $ch, CURLOPT_RETURNTRANSFER, true );
curl_setopt( $ch, CURLOPT_SSL_VERIFYPEER, false );
curl_setopt( $ch, CURLOPT_POSTFIELDS, json_encode( $fields ) );
$result = curl_exec($ch );
curl_close( $ch );
```

?>