



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh  
**TRUNG TÂM TIN HỌC**

# Lập trình Android

## Bài 22: *Notifications*

Phòng LT & Mạng

<http://csc.edu.vn/lap-trinh-va-csdl>





# Nội dung

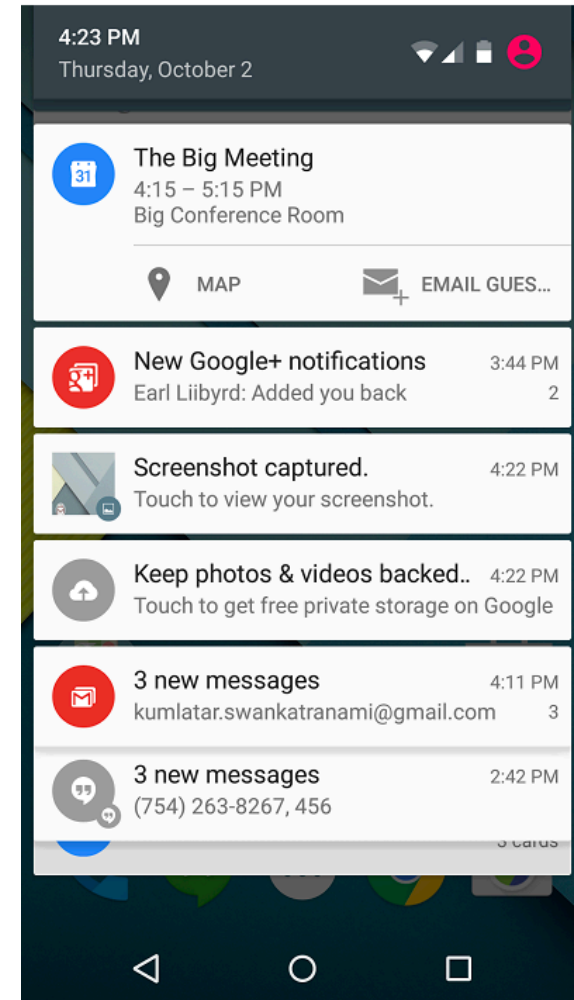
---

1. Notification
2. Các bước tạo một notification
3. Push notifications với Firebase Cloud Messaging



# Notifications

- **Notifications** là thông báo hiển thị đến người dùng bên ngoài phạm vi giao diện ứng dụng
- **Notifications** hiển thị trong vùng notification của hệ thống
- Một notification có những thành phần cơ bản nhất gồm:
  - Icon của ứng dụng
  - Tiêu đề và nội dung tin nhắn
  - Thời điểm hiển thị
- Ngoài ra, còn có thể thực hiện các action trên notification
- Từ API 26 (Android 8 – Oreo), **notification channels** được giới thiệu cho phép người dùng enable/disable các notifications trong **Settings** app





# Các bước tạo một notification

---

- Bước 1: Tạo notification channel
- Bước 2: Đăng ký notification channel
- Bước 3: Tạo một notification
- Bước 4: Gửi notification vừa tạo tới **NotificationManager** để hiển thị



# Bước 1: Tạo notification channel

- Đối với **API 26** trở lên, cần tạo **notification channel**, cung cấp các thông tin chuỗi **id**, chuỗi **name** và mức độ quan trọng (**importance level**). Có thể viết một class **NotificationTask** với phương thức **createNotificationChannel** như sau:

```
public static void createNotificationChannel(Context context) {  
    // 1 - Kiểm tra OS có API level lớn hơn hoặc 26  
    if (android.os.Build.VERSION.SDK_INT >= android.os.Build.VERSION_CODES.O) {  
        // 2 - Tạo ra notification channel với các thông tin cần thiết  
        int importance = NotificationManager.IMPORTANCE_DEFAULT;  
        NotificationChannel channel =  
            new NotificationChannel(CHANNEL_ID, CHANNEL_NAME, importance);  
        channel.setDescription("Reminders");  
        channel.setShowBadge(true);  
        // 3 – Tạo notification channel sử dụng NotificationManager  
        NotificationManager mNotificationManager = (NotificationManager)  
            context.getSystemService(Context.NOTIFICATION_SERVICE);  
        mNotificationManager.createNotificationChannel(channel);  
    }  
}
```



## Bước 2: Đăng ký notification channel

- Override **Application** class và thực hiện đăng ký notification channel trong phương thức **onCreate**

```
import android.app.Application;

public class App extends Application {

    @Override
    public void onCreate() {
        super.onCreate();
        NotificationTask.createNotificationChannel(this);
    }
}
```

App.java

← (1) Định nghĩa 1 class kế thừa từ class **Application**

(2) Khai báo **Application** class với thuộc tính **name** trong thẻ application của file **AndroidManifest.xml**

```
<manifest ...>

    <application
        ...
        android:name=".App">

    </application>

</manifest>
```

AndroidManifest.xml





## Bước 3, 4: Tạo một notification

- Trong class **NotificationTask**, tạo phương thức **createSampleNotification**:

```
public static void createSampleNotification(Context context, int notificationId, int icon, String title, String message) {
```

```
    NotificationCompat.Builder builder;
```

```
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
```

```
        // 1 – Tạo notification với thông tin channel id
```

```
        builder = new NotificationCompat.Builder(context, CHANNEL_ID)
```

```
            .setSmallIcon(icon)
```

```
            .setContentTitle(title)
```

```
            .setContentText(message);
```

```
    } else {
```

```
        // 2 – Tạo notification trong trường hợp OS có API level < 26
```

```
        builder = new NotificationCompat.Builder(context)
```

```
            .setSmallIcon(icon)
```

```
            .setContentTitle(title)
```

```
            .setContentText(message);
```

```
    }
```

```
    // 3 (Bước 4) – Gửi notification tới NotificationManager để hiển thị
```

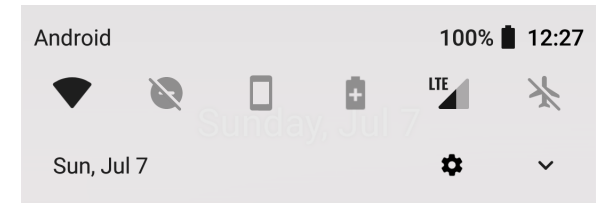
```
    NotificationManager mNotificationManager =
```

```
        (NotificationManager)
```

```
        context.getSystemService(Context.NOTIFICATION_SERVICE);
```

```
    mNotificationManager.notify(notificationId, builder.build());
```

```
}
```



Notifications • now

Hello

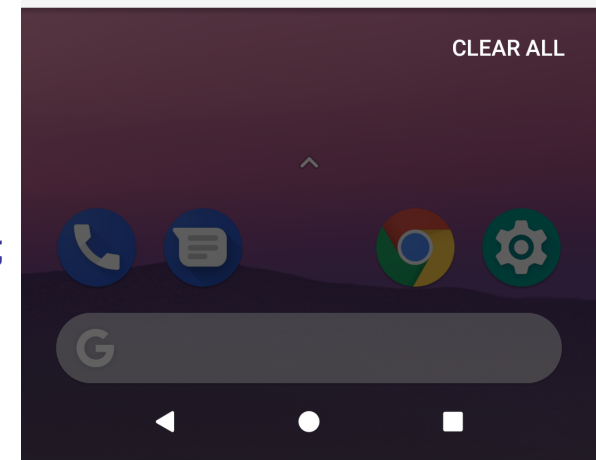
Hello World!

Android System

USB debugging connected

Tap to disable USB debugging.

Android System • USB charging this device





# Thêm action vào notification

---

- Để thực hiện mở **MainActivity** khi người dùng nhấn vào notification, cần khai báo một **PendingIntent**:

```
Intent intent = new Intent(context, MainActivity.class);  
intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);  
PendingIntent pendingIntent = PendingIntent.getActivity(context, REQUEST_CODE, intent, 0);
```

- Dùng đối tượng **NotificationCompat.Builder** để gắn **PendingIntent** vào notification bằng phương thức **setContentIntent**:

```
builder = new NotificationCompat.Builder(context, CHANNEL_ID)  
    .setSmallIcon(icon)  
    .setContentTitle(title)  
    .setContentText(message)  
    .setContentIntent(pendingIntent);
```

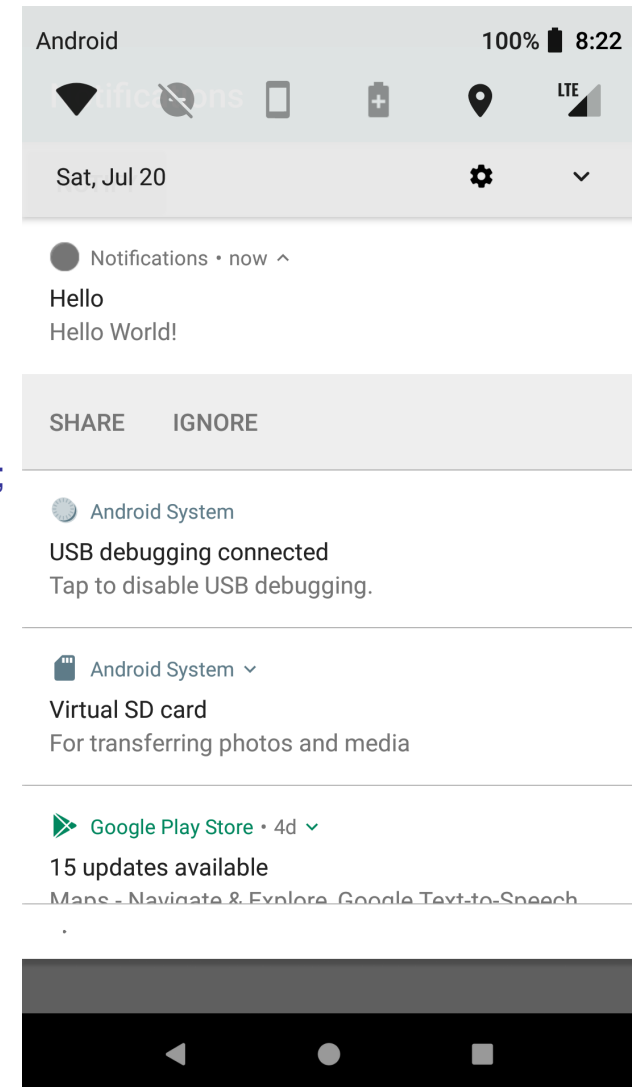




# Thêm action vào notification (2)

- Có thể thêm các action vào notification dưới dạng các button:

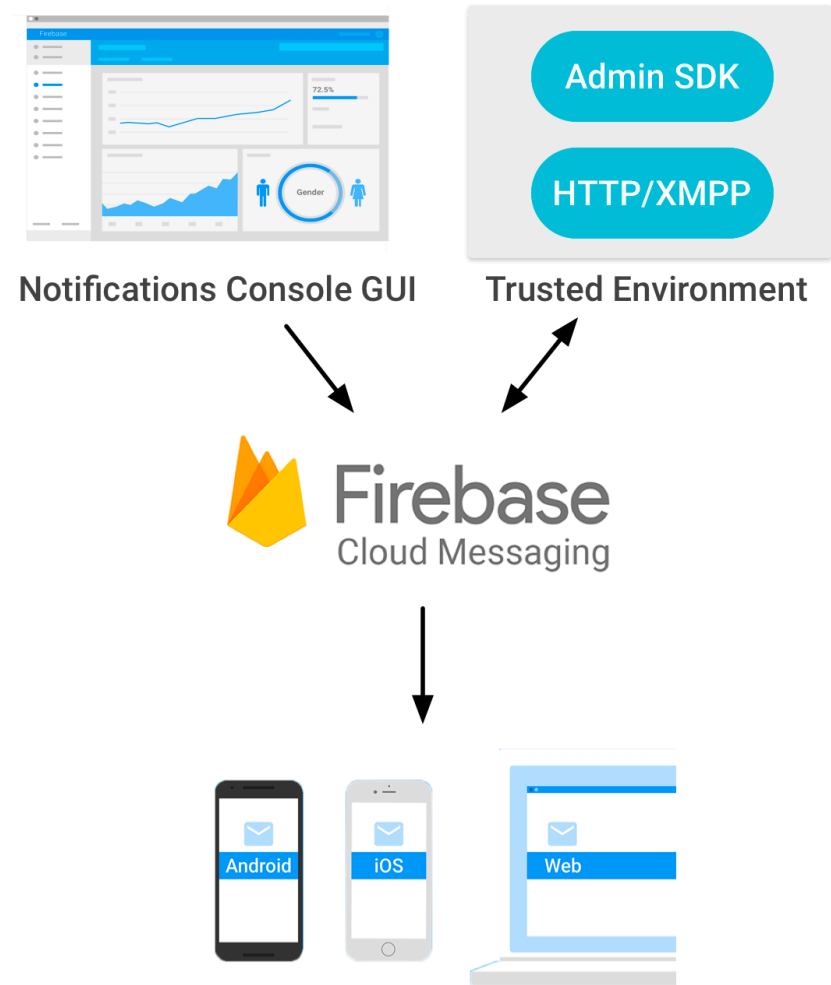
```
builder = new NotificationCompat.Builder(context, CHANNEL_ID)
    .setSmallIcon(icon)
    .setContentTitle(title)
    .setContentText(message)
    .setContentIntent(pendingIntent)
    .addAction(R.drawable.ic_share, "Share", sharePendingIntent)
    .addAction(R.drawable.ic_ignore, "Ignore", ignorePendingIntent);
```





# Firebase Cloud Messaging

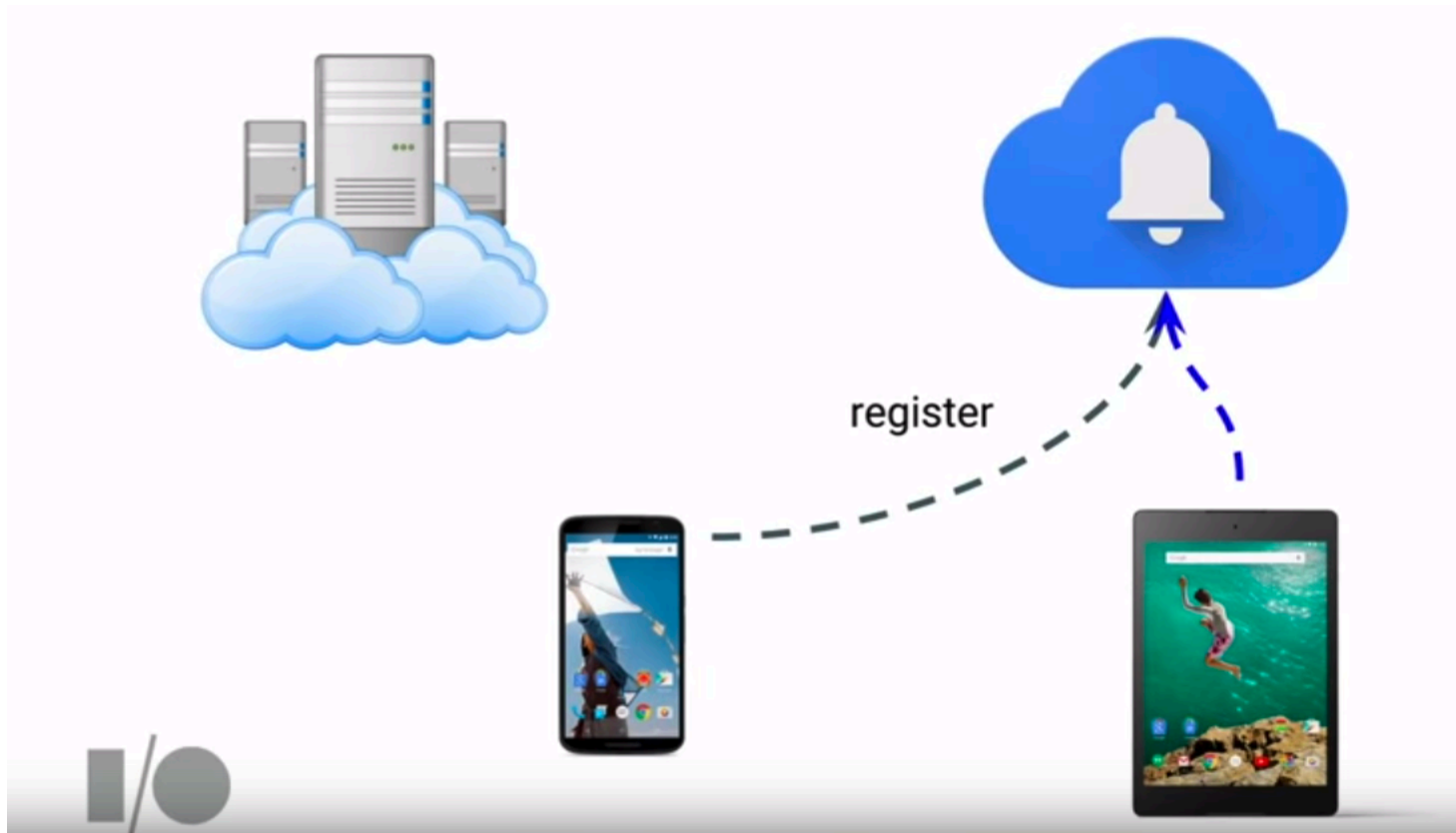
- Để hỗ trợ notification từ server về ứng dụng, cần sử dụng **Firebase Cloud Messaging (FCM)**
- Mọi notification gửi từ server đều phải thông qua **FCM** trước khi tới ứng dụng
- **FCM** ra đời để thay thế cho **Google Cloud Messaging (GCM)** đã bị khai tử từ tháng 4/2019





# FCM – Cơ chế hoạt động

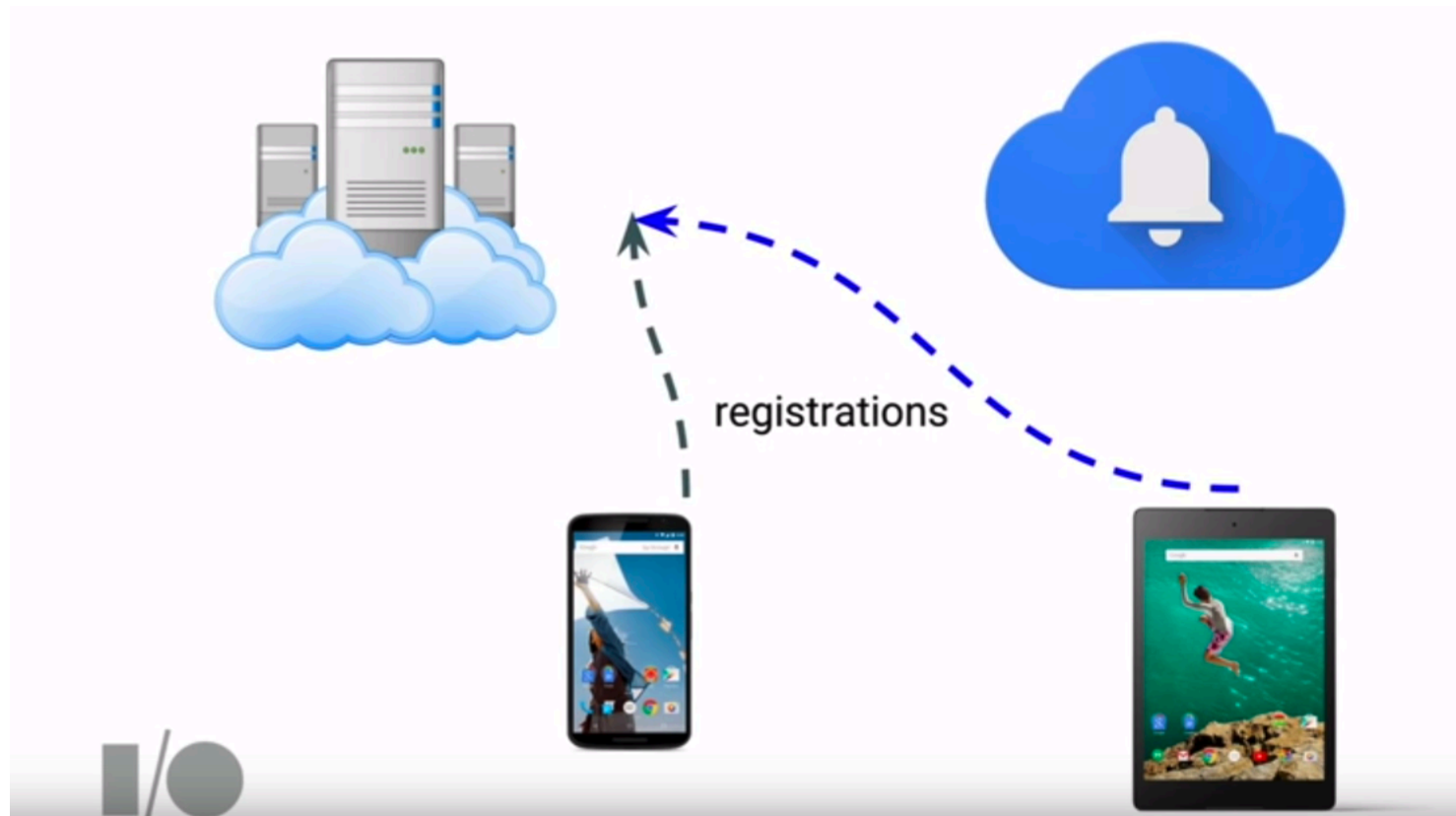
- **Bước 1:** App đăng ký với **FCM server** để nhận về một **token** tương ứng với thiết bị hiện tại





# FCM – Cơ chế hoạt động (2)

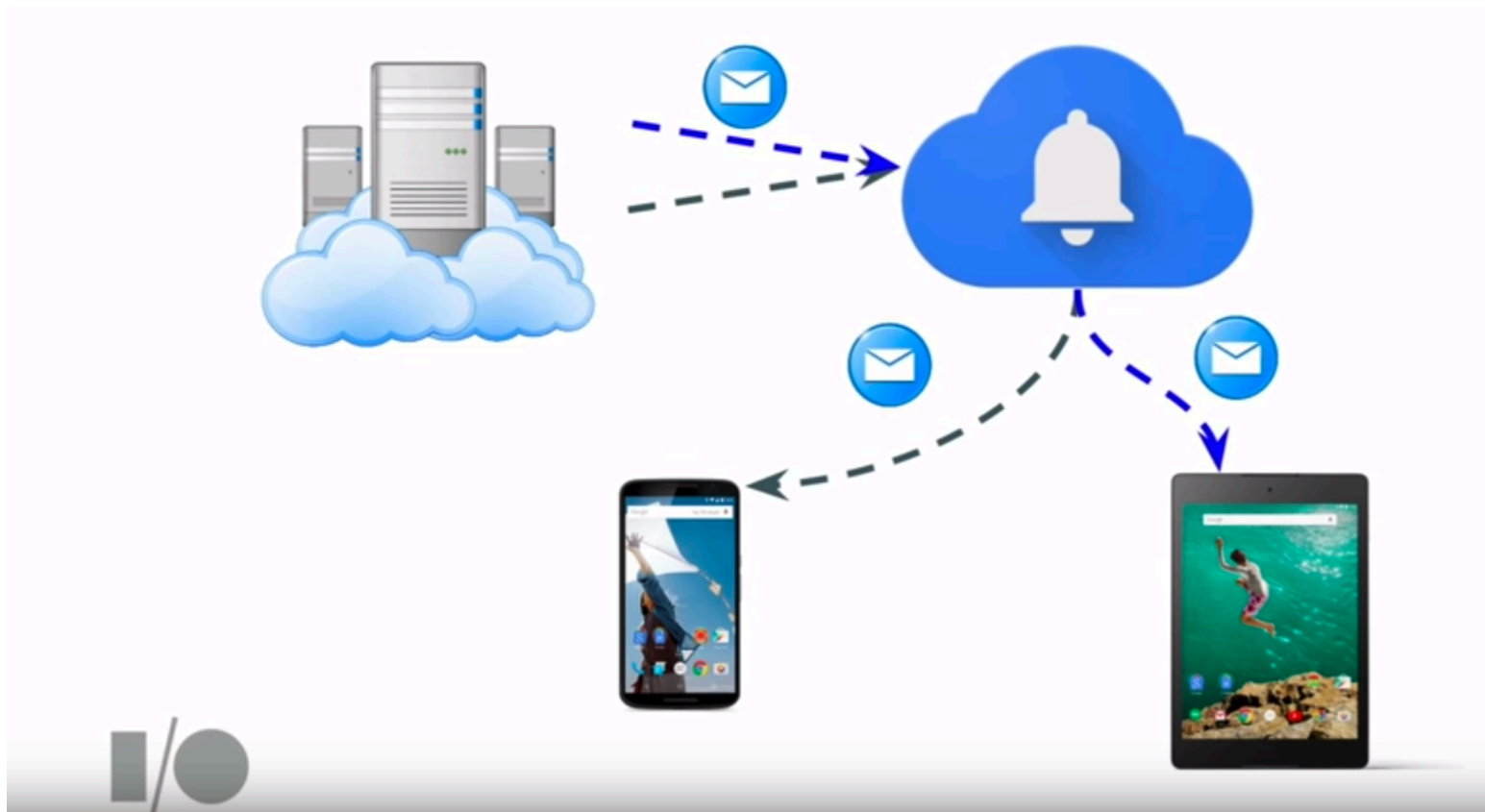
- **Bước 2:** Sau khi có **token** từ **FCM server**, **app** gửi thông tin **token** này về **Web server** để lưu tương ứng với thông tin người dùng app





# FCM – Cơ chế hoạt động

- **Bước 3:** Web server thực hiện **push notification** đến **app** bằng cách gửi thông điệp đến **FCM server**. **FCM server** có nhiệm vụ phân phát thông điệp đến app





# Cài đặt FCM

---

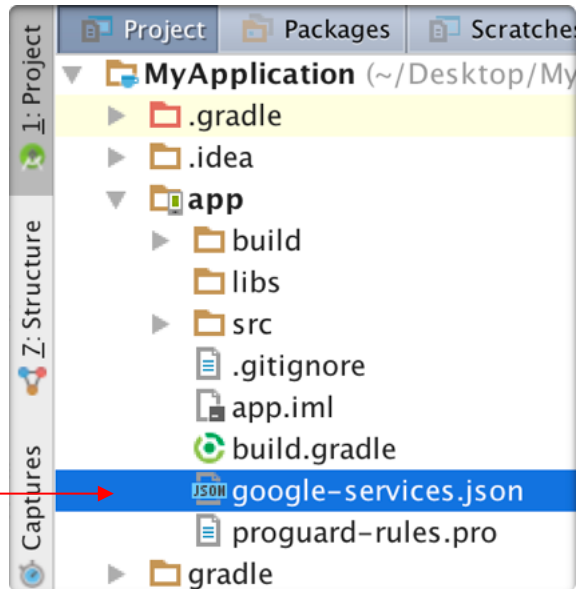
- **Bước 1:** Đăng ký/đăng nhập **Firebase developers console**
  - Tạo một **project Firebase** mới
  - Tạo một ứng dụng mới, cung cấp thông tin **SHA-1 key** dùng để ký (sign) trên ứng dụng
  - Tải về file cấu hình **google-services.json** và copy vào Android project
- **Bước 2:** Tích hợp **FCM** vào ứng dụng mobile
  - Đăng ký với FCM một **instance ID** và phát sinh một **token**
  - Gửi token lên Web Server
  - Đăng ký một **FCM Receiver** để xử lý các thông báo đến
  - Đăng ký một InstanceID Listener Service để xử lý cập nhật token
- **Bước 3:** Xây dựng Web server với các FCM endpoints
  - Endpoint cho phép đăng ký người dùng với token ở bước 2
  - Endpoint cho phép gửi một push notification tới một token đã đăng ký



# Đăng ký với Firebase developers console

- Truy cập <https://console.firebase.google.com>
- Tạo mới một **Firebase** project
  - (1) Tạo mới một Android app
  - (2) Tải về **google-services.json**
  - (3) Cấu hình **dependencies**

2



1

## Add Firebase to your Android app

1

### Register app

Android package name ?

edu.csc.notifications

App nickname (optional) ?

CSC Notifications

Debug signing certificate SHA-1 (optional) ?

00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

Required for Dynamic Links, Invites, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

Register app



# Cấu hình dependencies trong Gradle files

---

- Project-level **build.gradle** (<project>/build.gradle):

```
buildscript {  
    dependencies {  
        classpath 'com.google.gms:google-services:4.2.0'  
    }  
}
```

- App-level **build.gradle** (<project>/<app-module>/build.gradle):

```
dependencies {  
    implementation 'com.google.firebase:firebase-core:11.6.0'  
    implementation 'com.google.firebase:firebase-messaging:11.6.0'  
}  
  
apply plugin: 'com.google.gms.google-services'
```

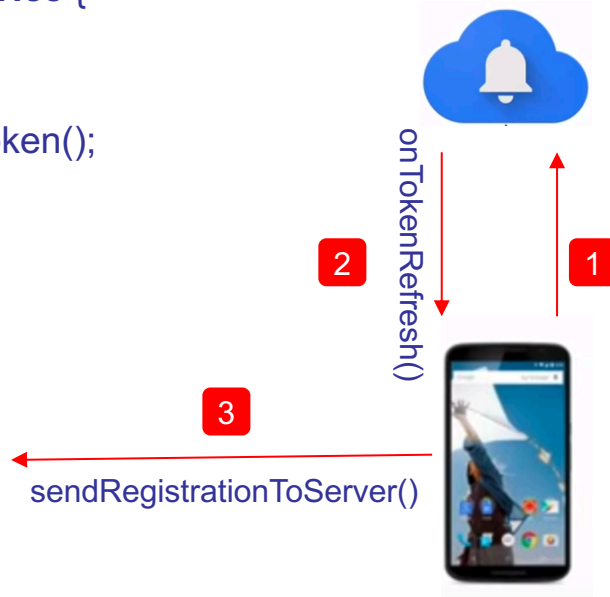




# FirebaseInstanceIdService

- Viết class **MyInstanceIdService** kế thừa từ **FirebaseInstanceIdService** và override phương thức **onTokenRefresh** → lấy **instance id token** và cập nhật lên Web server nếu có sự thay đổi token này

```
public class MyInstanceIdService extends FirebaseInstanceIdService {  
    @Override  
    public void onTokenRefresh() {  
        String refreshedToken = FirebaseInstanceId.getInstance().getToken();  
        Log.d("CSC", "CSC Refreshed token: " + refreshedToken);  
        sendRegistrationToServer(refreshedToken);  
    }  
    private void sendRegistrationToServer(String token) {  
    }  
}
```



- Đăng ký service này trong **AndroidManifest.xml**

```
<application ...  
    <service  
        android:name=".MyInstanceIdService" android:exported="false">  
        <intent-filter>  
            <action android:name="com.google.firebase.INSTANCE_ID_EVENT"/>  
        </intent-filter>  
    </service>  
</application>
```



# FirebaseMessagingService

- Cài đặt class **MyMessagingService** kế thừa từ **FirebaseMessagingService** và override phương thức **onMessageReceived** giúp xử lý các thông báo đến

```
public class MyMessagingService extends FirebaseMessagingService {  
  
    @Override  
    public void onMessageReceived(RemoteMessage remoteMessage) {  
        Map<String, String> data = remoteMessage.getData();  
        RemoteMessage.Notification notification = remoteMessage.getNotification();  
        // createNotification();  
    }  
}
```

- Khai báo service này trong **AndroidManifest.xml**

```
<application ...>  
    <service  
        android:name=".MyMessagingService" android:exported="false" >  
        <intent-filter>  
            <action android:name="com.google.firebase.MESSAGING_EVENT" />  
        </intent-filter>  
    </service>  
</application>
```





# Xử lý Message

- Một message gồm 2 loại payloads: **Notification** và **Data**. Tùy vào trạng thái của app và loại payload mà các payload này sẽ được xử lý ở đâu.

```
{
  "notification": {
    "title": "T3H",
    "body": "Hello World"
  },
  "data": {
    "title": "TTTH",
    "body": "Hello World"
  },
  "to": "device_token"
}
```

**@Override**

```
public void onMessageReceived(RemoteMessage msg) {
    Map<String, String> data = msg.getData();
    RemoteMessage.Notification notification
    = msg.getNotification();
}
```

App state	Notification	Data	Cả 2
Foreground	onMessageReceived	onMessageReceived	onMessageReceived
Background	System tray	onMessageReceived	Notification: system tray Data: nằm trong extras của intent



# Kiểm thử FCM push notification

- Sử dụng REST Client app để test notification với:
- FCM API endpoint: `https://fcm.googleapis.com/fcm/send`
- Method: POST
- Headers: Content-Type: `application/json` và Authorization: `key=<firebase_server_key>`
- Body: Chuỗi JSON

METHOD: POST

SCHEME :// HOST [ ":" PORT ] [ PATH [ "?" QUERY ] ]

https://fcm.googleapis.com/fcm/send length: 35 bytes Send

QUERY PARAMETERS

+ Add query parameter

HEADERS ? ↓

Form

☒ Content-Type : application/json x

☒ Authorization : key=AAAAr4b83rw:APA91bGN x ?

+ Add header ? Add authorization ?

BODY ? Text

```
1 {
2   "data": {
3     "title": "TTTH",
4     "body": "Hello World"
5   },
6   "to": "fPyUVN2KoVM:APA91bEB_S1oY6cQeA5f3YeJrJfDx32o6OobMJJaELZFpUcVJSV"
7 }
```





# Kiểm thử FCM push notification (2)

- Cách lấy server key:
  - Firebase Console → Project → Settings → Cloud Messaging → Server key

Project credentials

Key	Token
Server key	AAAAr4b83rw:APA91bGNurzKYyAl9kuFX90WIIWG1sWHWhU-o6S0qpW-I9PZsolJNMOjAiz3H5Vq0WsRdE6Lyvs6QpvcgRwP3ZdEh1aYRMSb9sMoX4maDprub3baXnlC3ioDa4IX9VEmurGF9IC8TQu3
Legacy server key ?	AlzaSyDScreuqVr1NffLmmDDpHTAVHsdhJjONOE
Sender ID ?	753883995836





# Kiểm thử FCM push notification (3)

- Kết quả kiểm thử:

