



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

# Lập trình Android

## Bài 23: *Location Services*

Phòng LT & Mạng

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





# Nội dung

---

1. Location Services
2. Location Services API
3. Lấy vị trí của thiết bị
4. Xử lý thông tin vị trí
5. Lấy vị trí của thiết bị theo định kỳ



# Location Services

- **Location Services** là dịch vụ cung cấp thông tin vị trí địa lý của thiết bị di động
- Location Services có nhiều ứng dụng trong nhiều lĩnh vực khác nhau
- Ví dụ:
  - Xác định vị trí gần nhất của ngân hàng (chi nhánh, máy ATM), nhà ga, trạm xe bus, nhà hàng, khách sạn, bệnh viện, đồn cảnh sát ...
  - Cung cấp thông tin vận tải từ điểm này đến một điểm khác
  - Các mạng xã hội dùng Location Services để xác định các sự kiện gần vị trí của người dùng, bạn bè, gia đình.





# Location Services API

---

- Google cung cấp **Fused Location API** để xác định vị trí của thiết bị
- Để sử dụng API này, cần khai báo sử dụng trong **Gradle** build file

```
dependencies {  
    implementation 'com.google.android.gms:play-services-location:11.8.0'  
    implementation 'com.google.android.gms:play-services-maps:11.8.0'  
}
```

- Và khai báo các permissions cần thiết trong file **AndroidManifest.xml**

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



# Lấy vị trí của thiết bị

- Để lấy vị trí của thiết bị, có thể thực hiện '***get last known location***'. Nếu thiết bị đã bật **GPS**, vị trí này chính là vị trí hiện tại:
- Tạo đối tượng **FusedLocationProviderClient**:

```
FusedLocationProviderClient locationClient = LocationServices.getFusedLocationProviderClient(context);
```

- Thực hiện lấy '*last location*':

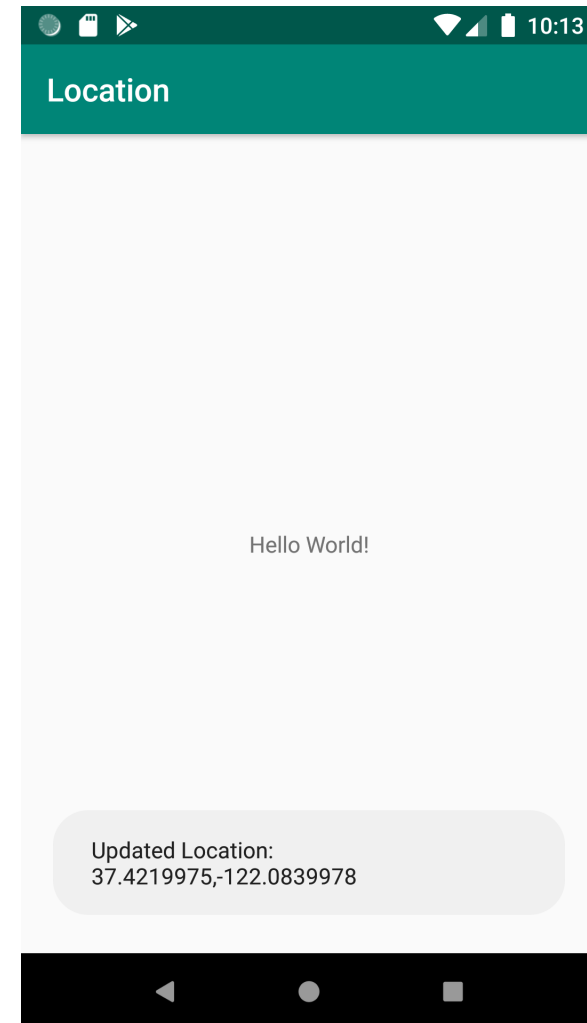
```
locationClient.getLastLocation()  
    .addOnSuccessListener(this, new OnSuccessListener<Location>() {  
        @Override  
        public void onSuccess(Location location) {  
            // Lấy last known location. Trong một số trường hợp location có thể bằng null  
            if (location != null) {  
                // Xử lý cập nhật vị trí  
                onLocationChanged(location);  
            }  
        }  
    });
```



# Xử lý thông tin Location

- Với location có được, có thể lấy các thông tin **latitude** và **longitude**

```
public void onLocationChanged(Location location) {  
    // Lấy thông tin latitude và longitude từ đối tượng location  
    String msg = "Updated Location: " +  
        Double.toString(location.getLatitude()) + "," +  
        Double.toString(location.getLongitude());  
    Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();  
    // Có thể tạo đối tượng LatLng để sử dụng với bản đồ Google Maps  
    LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());  
}
```





# Lấy vị trí của thiết bị theo định kỳ

- Lấy vị trí của thiết bị theo định kỳ, ví dụ: mỗi 10 giây, nhanh nhất là 2 giây

```
private long UPDATE_INTERVAL = 10 * 1000; // 10 seconds
private long FASTEST_INTERVAL = 3 * 1000; // 3 seconds
```

- Tạo đối tượng **LocationRequest** để bắt đầu nhận cập nhật thông tin vị trí

```
LocationRequest mLocationRequest = new LocationRequest();
mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
mLocationRequest.setInterval(UPDATE_INTERVAL);
mLocationRequest.setFastestInterval(FASTEST_INTERVAL);
```

- Tạo đối tượng **LocationSettingsRequest** sử dụng đối tượng **LocationRequest**

```
LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder();
builder.addLocationRequest(mLocationRequest);
LocationSettingsRequest locationSettingsRequest = builder.build();
```

- Kiểm tra cài đặt vị trí đã được bật chưa

```
SettingsClient settingsClient = LocationServices.getSettingsClient(this);
settingsClient.checkLocationSettings(locationSettingsRequest);
```

- Kiểm tra người dùng đã cho phép sử dụng dịch vụ **Location Services** chưa

```
ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
```





# Lấy vị trí của thiết bị theo định kỳ (2)

---

- Khởi tạo đối tượng **FusedLocationProviderClient**:

```
FusedLocationProviderClient locationClient = getFusedLocationProviderClient(this);
```

- Thực hiện lấy thông tin vị trí theo định kỳ:

```
locationClient.requestLocationUpdates(locationRequest, new LocationCallback() {  
    @Override  
    public void onLocationResult(LocationResult locationResult) {  
        onLocationChanged(locationResult.getLastLocation());  
    }  
},  
Looper.myLooper());
```





# Lấy vị trí của thiết bị theo định kỳ (3)

- Tổng hợp các bước thực hiện lấy vị trí của thiết bị theo định kỳ như sau:

```
protected void startLocationUpdates() {
    LocationRequest mLocationRequest = new LocationRequest();
    mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    mLocationRequest.setInterval(UPDATE_INTERVAL);
    mLocationRequest.setFastestInterval(FATEST_INTERVAL);

    LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder();
    builder.addLocationRequest(mLocationRequest);
    LocationSettingsRequest locationSettingsRequest = builder.build();

    SettingsClient settingsClient = LocationServices.getSettingsClient(this);
    settingsClient.checkLocationSettings(locationSettingsRequest);

    if (!isPermissionsAllowed()) return;

    FusedLocationProviderClient locationClient = getFusedLocationProviderClient(this);
    locationClient.requestLocationUpdates(mLocationRequest, new LocationCallback() {
        @Override
        public void onLocationResult(LocationResult locationResult) {
            onLocationChanged(locationResult.getLastLocation());
        }
    },
    Looper.myLooper());
}
```





# Kiểm thử trên Emulator

- Emulator → Settings (dấu 3 chấm) → Location

The screenshot displays the Android Studio interface. On the left, the 'Extended controls' panel for 'Nexus\_5\_API\_27:5554' is open, showing the 'Location' settings. The 'Coordinate system' is set to 'Decimal'. The 'Latitude' is 10.7624176 and the 'Longitude' is 106.6790081. The 'Currently reported location' box shows: Latitude: 10.7624, Longitude: 106.6790, Altitude: 0.0, Speed: 0.0, Heading: 0.0. The 'GPS data playback' section has a table with columns: Delay (sec), Latitude, Longitude, Elevation, Name, and Description. The 'SEND' button is visible. The emulator screen on the right shows the 'Location' app with the text 'Hello World!' and a notification 'Updated Location: 10.7624167,106.6790067'. The bottom navigation bar of the emulator is visible.

Extended controls - Nexus\_5\_API\_27:5554

Location

Cellular

Battery

Phone

Directional pad

Microphone

Fingerprint

Virtual sensors

Bug report

Snapshots

Record and Playback

Google Play

Settings

Help

GPS data point

Coordinate system: Decimal

Latitude: 10.7624176

Longitude: 106.6790081

Currently reported location

Latitude: 10.7624  
Longitude: 106.6790  
Altitude: 0.0  
Speed: 0.0  
Heading: 0.0

Altitude (meters): 0.0

Speed (knots): 0.0

SEND

GPS data playback

| Delay (sec) | Latitude | Longitude | Elevation | Name | Description |
|-------------|----------|-----------|-----------|------|-------------|
|-------------|----------|-----------|-----------|------|-------------|

Location

Hello World!

Updated Location:  
10.7624167,106.6790067

Speed 1X

LOAD GPX/KML

