

## 위치 정보 활용하기

1. 사용자 위치 얻기
2. 구글 지도 활용하기
3. 구글 지도 앱 만들기

# 1. 사용자 위치 얻기

## ■ 위치 접근 권한

- 사용자의 위치를 추적하기 위한 3가지 권한
  - `android.permission.ACCESS_COARSE_LOCATION`: 와이파이나 모바일 데이터(또는 둘 다)를 사용해 기기의 위치에 접근하는 권한. 도시에서 1블록 정도의 오차 수준.
  - `android.permission.ACCESS_FINE_LOCATION`: 위성, 와이파이, 모바일 데이터 등 이용할 수 있는 위치 제공자를 사용해 최대한 정 확한 위치에 접근하는 권한.
  - `android.permission.ACCESS_BACKGROUND_LOCATION`: 안드로이드 10(API 레벨 29) 이상에서 백그라운드 상태에서 위치에 접근 하는 권한

### • 서비스에서 위치 접근

```
<service
    android:name="MyNavigationService"
    android:foregroundServiceType="location"
    (... 생략 ...)
</service>
```

# 1. 사용자 위치 얻기

## ■ 플랫폼 API의 위치 매니저

- 사용자의 위치를 얻을 때는 LocationManager라는 시스템 서비스를 이용

• 위치 매니저 사용

```
val manager = getSystemService(LOCATION_SERVICE) as LocationManager
```

- 위치 제공자 지정하기
  - GPS: GPS 위성을 이용
  - Network: 이동 통신망을 이용
  - Wifi: 와이파이를 이용
  - Passive: 다른 앱에서 이용한 마지막 위치 정보를 이용

# 1. 사용자 위치 얻기

- 모든 위치 제공자 알아보기
  - 현재 기기에 어떤 위치 제공자가 있는지를 알고 싶다면 LocationManager의 allProviders 프로퍼티를 이용
- 지금 사용할 수 있는 위치 제공자를 알아보려면 getProviders() 함수를 이용

## • 모든 위치 제공자 알아보기

```
var result = "All Providers : "  
val providers = manager.allProviders  
for (provider in providers) {  
    result += "$provider. "  
}  
Log.d("kkang", result) // All Providers : passive, gps, network,
```

## • 지금 사용할 수 있는 위치 제공자 알아보기

```
result = "Enabled Providers : "  
val enabledProviders = manager.getProviders(true)  
for (provider in enabledProviders) {  
    result += "$provider, "  
}  
Log.d("kkang", result) // Enabled Providers : passive, gps, network,
```

# 1. 사용자 위치 얻기

## ■ 위치 정보 얻기

- LocationManager의 getLastKnownLocation() 함수를 이용
- Location은 위치의 정확도, 위도, 경도, 획득 시간 등의 데이터를 포함
  - getAccuracy(): 정확도
  - getLatitude(): 위도
  - getLongitude(): 경도
  - getTime(): 획득 시간

• 위치 한번만 가져오기

```
if (ContextCompat.checkSelfPermission(
    this, Manifest.permission.ACCESS_FINE_LOCATION
) == PackageManager.PERMISSION_GRANTED
){
    val location: Location? =
        manager.getLastKnownLocation(LocationManager.GPS_PROVIDER)
    location?.let {
        val latitude = location.latitude
        val longitude = location.longitude
        val accuracy = location.accuracy
        val time = location.time

        Log.d("kkang", "$latitude, $longitude, $accuracy, $time")
    }
}
```

# 1. 사용자 위치 얻기

- 계속 위치를 가져와야 한다면 LocationListener를 이용
  - onLocationChanged(): 새로운 위치를 가져오면 호출
  - onProviderEnabled(): 위치 제공자가 이용할 수 있는 상황이면 호출
  - onProviderDisabled(): 위치 제공자가 이용할 수 없는 상황이면 호출

## • 위치 계속 가져오기

```
val listener: LocationListener = object : LocationListener {  
    override fun onLocationChanged(location: Location) {  
        Log.d("kkang", "${location.latitude}, ${location.longitude},  
            ${location.accuracy}")  
    }  
    override fun onProviderDisabled(provider: String) {  
        super.onProviderDisabled(provider)  
    }  
    override fun onProviderEnabled(provider: String) {  
        super.onProviderEnabled(provider)  
    }  
}  
manager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 10_000L, 10f, listener)  
(... 생략 ...)  
manager.removeUpdates(listener)
```

# 1. 사용자 위치 얻기

## ■ 구글 Play 서비스의 위치 라이브러리

- 위치 제공자를 지정할 때 고려할 사항
  - 전력을 적게 소비하는가?
  - 정확도는 높은가?
  - API가 간단한가?
  - 부가 기능을 제공하는가?
  - 대부분 안드로이드 기기를 지원하는가?
- 구글에서는 최적의 알고리즘으로 위치 제공자를 지정할 수 있도록 Fused Location Provider라는 라이브러리를 제공

• play 서비스 사용 선언

```
implementation 'com.google.android.gms:play-services:12.0.1'
```

# 1. 사용자 위치 얻기

- Fused Location Provider에서 핵심 클래스
  - FusedLocationProviderClient: 위치 정보를 얻음
  - GoogleApiClient: 위치 제공자 준비 등 다양한 콜백을 제공
- GoogleApi Client에서는 GoogleApiClient.ConnectionCallbacks와 GoogleApiClient.OnConnectionFailedListener 인터페이스를 구현한 객체를 지정

## • GoogleApiClient 초기화

```
val connectionCallback=object: GoogleApiClient.ConnectionCallbacks {  
    override fun onConnected(p0: Bundle?) {  
        // 위치 제공자를 사용할 수 있을 때  
        // 위치 획득  
    }  
    override fun onConnectionSuspended(p0: Int) {  
        // 위치 제공자를 사용할 수 없을 때  
    }  
}  
  
val onConnectionFailedCallback = object : GoogleApiClient  
    .OnConnectionFailedListener {  
    override fun onConnectionFailed(p0: ConnectionResult) {  
        // 사용할 수 있는 위치 제공자가 없을 때  
    }  
}  
  
val apiClient: GoogleApiClient = GoogleApiClient.Builder(this)  
    .addApi(LocationServices.API)  
    .addConnectionCallbacks(connectionCallback)  
    .addOnConnectionFailedListener(onConnectionFailedCallback)  
    .build()
```



# 1. 사용자 위치 얻기

---

- FusedLocationProviderClient는 초기화

• FusedLocationProviderClient 초기화

```
val providerClient: FusedLocationProviderClient =  
    LocationServices.getFusedLocationProviderClient(this)
```

- GoogleApiClient 객체에 위치 제공자를 요청

• 위치 제공자 요청

```
apiClient.connect()
```

# 1. 사용자 위치 얻기

- onConnected() 함수에서 FusedLocationProviderClient의 getLastLocation() 함수 호출

## • 사용자 위치 얻기

```
override fun onConnected(p0: Bundle?) {  
    if (ContextCompat.checkSelfPermission(  
        this@FusedActivity,  
        Manifest.permission.ACCESS_FINE_LOCATION  
    ) == PackageManager.PERMISSION_GRANTED  
    ) {  
        providerClient.getLastLocation().addOnSuccessListener(  
            this@FusedActivity,  
            object : OnSuccessListener<Location> {  
                override fun onSuccess(location: Location?) {  
                    val latitude = location?.latitude  
                    val longitude = location?.longitude  
                }  
            })  
    }  
}
```

## 2. 구글 지도 활용하기

### ■ 지도 사용 설정하기

- 빌드 그래들의 dependencies 항목

• 구글 지도 사용 선언

```
implementation 'com.google.android.gms:play-services:12.0.1'
```

- 퍼미션을 등록

• 퍼미션 등록

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

## 2. 구글 지도 활용하기

- 구글 지도 API를 이용하는 키를 등록

### • 구글 지도 API 키 등록

```
<application ... 생략 ... >
    <uses-library android:name="org.apache.http.legacy" android:required="false" />
    <meta-data android:name="com.google.android.maps.v2.API_KEY"
        android:value="###구글 지도 API 키 등록###" />
    <meta-data android:name="com.google.android.gms.version"
        android:value="@integer/google_play_services_version" />
</application>
```

여기에 구글 지도의 API 키를 등록합니다.

## 2. 구글 지도 활용하기

### ■ 구글 개발자 콘솔에서 지도 API 키 얻기

- 구글 개발자 콘솔(console.cloud.google.com)에 접속해 프로젝트를 생성하고 사용자 인증 정보를 만들면 지도 API 키를 발급
- 구글 개발자 콘솔에서 얻은 지도 API 키를 매니페스트 파일에 등록
- 레이아웃 XML

#### • 지도 프래그먼트 등록

```
<fragment
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/mapView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="com.google.android.gms.maps.SupportMapFragment"
/>
```

## 2. 구글 지도 활용하기

### ■ 지도 제어하기

- 지도의 중심 이동하기
- 지도를 출력하는 뷰 객체를 얻어야 함.

#### • 지도 뷰 객체 얻기

```
class MapActivity : AppCompatActivity(), OnMapReadyCallback {  
    lateinit var binding: ActivityMapBinding  
    var googleMap: GoogleMap? = null  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        binding = ActivityMapBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
        (supportFragmentManager.findFragmentById(R.id.mapView) as  
            SupportMapFragment?)!!.getMapAsync(this)  
    }  
    override fun onMapReady(p0: GoogleMap?) {  
        googleMap = p0  
    }  
}
```



## 2. 구글 지도 활용하기

- 지도의 중심을 이동

- 지도의 중심 이동

```
val latLng = LatLng(37.566610, 126.978403)
val position = CameraPosition.Builder()
    .target(latLng)
    .zoom(16f)
    .build()
googleMap?.moveCamera(CameraUpdateFactory.newCameraPosition(position))
```

- 마커 표시하기

- 마커 표시하기

```
// 마커 옵션
val markerOptions = MarkerOptions()
markerOptions.icon(BitmapDescriptorFactory.fromResource(R.drawable.ic_marker))
markerOptions.position(latLng)
markerOptions.title("서울시청")
markerOptions.snippet("Tel:01-120")

// 마커 표시하기
googleMap?.addMarker(markerOptions)
```

## 2. 구글 지도 활용하기

- 지도에서 사용자 이벤트 처리
  - GoogleMap.OnMapClickListener: 지도 클릭 이벤트
  - GoogleMap.OnMapLongClickListener: 지도 롱 클릭 이벤트
  - GoogleMap.OnMarkerClickListener: 마커 클릭 이벤트
  - GoogleMap.OnMarkerDragListener: 마커 드래그 이벤트
  - GoogleMap.OnInfoWindowClickListener: 정보 창 클릭 이벤트
  - GoogleMap.OnCameraIdleListener: 지도 화면 변경 이벤트

### • 지도 이벤트 핸들러

```
googleMap?.setOnMapClickListener { latLng ->
    Log.d("kkang", "click : ${latLng.latitude}, ${latLng.longitude}")
}
googleMap?.setOnMapLongClickListener { latLng ->
    Log.d("kkang", "long click : ${latLng.latitude}, ${latLng.longitude}")
}
googleMap?.setOnCameraIdleListener {
    val position = googleMap!!.cameraPosition
    val zoom = position.zoom
    val latitude = position.target.latitude
    val longitude = position.target.longitude
    Log.d("kkang", "user change : $zoom, $latitude, $longitude")
}
googleMap?.setOnMarkerClickListener { marker ->
    true
}
googleMap?.setOnInfoWindowClickListener { marker ->
}
```



# 실습 : 구글 지도 앱 만들기

---

## ■ 1단계. 모듈 생성하고 빌드 그래들 수정하기

- Ch19\_Map 라는이름으로새로운모듈을만듭니다.
- 라이브러리를추가

## ■ 2단계. SHA1 지문 얻기

- SHA1 지문이라는 인증 정보가 필요

## ■ 3단계. 구글 콘솔에서 지도 API 키 발급받기

- [console.cloud.google.com](https://console.cloud.google.com)
- 구글 개발자 콘솔에 접속해 지도 API 키를 발급 4단계. 매니페스트 설정하기
- 매니페스트 파일을 열고 퍼미션과 부가 정보를 선언

## ■ 5단계. 메인 레이아웃 XML 작성하기

- 화면을 구성하는 레이아웃 XML 파일에 구글 지도를 출력하는 프래그먼트를 작성

## ■ 6단계. 메인 액티비티 작성하기

- MainActivity.kt 파일을 열고 작성

## ■ 7단계. 앱 실행하기

