

# Андройд – Лекц 14

Ж.Золжаргал  
([j.zoljargal@must.edu.mn](mailto:j.zoljargal@must.edu.mn))



# Using location-based services

- LBS нь ялгаатай технологиуд төхөөрөмжийн одоогын байршлыг тодорхойлоход ашиглагддаг
- Үүнд үндсэн 2 бүрэлдхүүн байна
  - LocationManager – Байршилд суурилсан үйлчилгээ. (LBService)
  - LocationProviders – Төхөөрөмжийн одоогын байршлыг тодорхойлох ялгаатай технологиуд



# Using location-based services

- Location Manager-г ашигласанаар юу хийж болох
  - Одоогын байршлыг гаргаж авна
  - Хөдөлгөөнийг мөшгих (Track movement)
  - Тодорхой муж руу орох болон гарах үед илрүүлэх
  - Боломжтой Location provider-ийг хайх



# Using location-based services



- Идэвхитэй GPS provider-ын мэдээллийг авах

```
LocationManager locationManager;
```

```
@Override  
public void onCreate(Bundle icle) {  
    super.onCreate(icle);  
    setContentView(R.layout.main);
```

```
    String location_context = Context.LOCATION_SERVICE;  
    locationManager = (LocationManager) getSystemService(location_context);  
    testProviders();  
}
```

```
public void testProviders() {}
```

# Using location-based services

```
public void testProviders() {
    TextView tv = (TextView)findViewById(R.id.myTextView);
    StringBuilder sb = new StringBuilder("Enabled Providers:");

    List<String> providers = locationManager.getProviders(true);

    for (String provider : providers) {

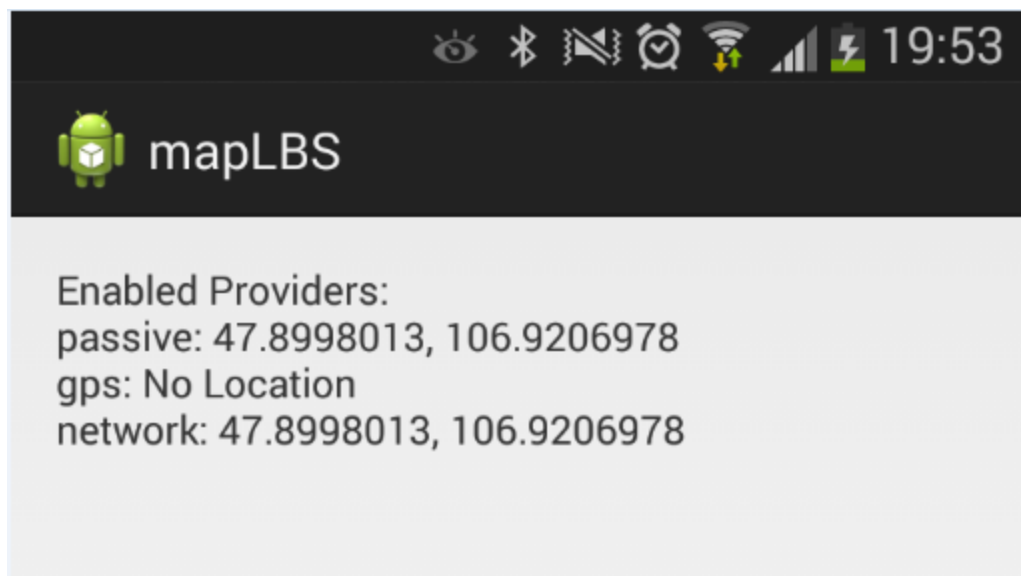
        locationManager.requestLocationUpdates(provider, 1000, 0,
            new LocationListener() {
                public void onLocationChanged(Location location) {}
                public void onProviderDisabled(String provider){}
                public void onProviderEnabled(String provider){}
                public void onStatusChanged(String provider, int status,
                    Bundle extras){}
            });

        sb.append("\n").append(provider).append(": ");

        Location location = locationManager.getLastKnownLocation(provider);
        if (location != null) {
            double lat = location.getLatitude();
            double lng = location.getLongitude();
            sb.append(lat).append(", ").append(lng);
        } else {
            sb.append("No Location");
        }
    }
    tv.setText(sb);
}
```

# Using location-based services

- Үр дүн



# Selecting a location provider

- Тухайн төхөөрөмж дээр Андроид нь одоогын байрлалыг тодорхойлоход хэд хэдэн технологи байж болно
- Тэдгээр нь эрчим хүч зарцуулалт, мөнгө зарцуулалт, нарвийвчлал, хурд, чиглэлийн мэдээлэл зэрэг нь харилцан адилгүй
- Тухайн provider-н нэрийг дамжуулж шаардлагатай provider-г үүсгэнэ

```
String providerName = LocationManager.GPS_PROVIDER;  
LocationProvider gpsProvider;  
gpsProvider = locationManager.getProvider(providerName);
```

# Finding the Available Providers

- LocationManager класс нь provider-н нэрийг агуулсан тэмдэгт мөрөн тогтмолуудтай
- Түгээмэл хэрэглэгддэг 2 нь
  - LocationManager.GPS\_PROVIDER
  - LocationManager.NETWORK\_PROVIDER
- Бүх provider-н мэдээллийг авах / зөвхөн идэвхитэй

```
boolean enabledOnly = true;  
List<String> providers = locationManager.getProviders(enabledOnly);
```



# Finding Providers Based on Requirement Criteria

- Ихэнх тохиолдол, програмист ямар Location Provider-г ашиглаж байршилаа тодорхойлохыг заах нь тохиромжгүй
- Програмистын тодорхойлсон шаардлагын дагуу Андроид нь хамгийн тохиромжтой технологийг сонгодог
- Үүнд Criteria классыг ашиглах бөгөөд нарийвчлал (fine, coarse), эрчим хүч зарцуулалт (low, medium, high), үнэ болон далайн түвшинээс дээших өндөр, хурд, чиглэл зэрэг шаардлагтай эсэхийг тодорхойлох

# Finding Providers Based on Requirement Criteria

- Доорх байдлаар шаардлагаа тодорхойлно

```
Criteria criteria = new Criteria();  
criteria.setAccuracy(Criteria.ACCURACY_COARSE);  
criteria.setPowerRequirement(Criteria.POWER_LOW);  
criteria.setAltitudeRequired(false);  
criteria.setBearingRequired(false);  
criteria.setSpeedRequired(false);  
criteria.setCostAllowed(true);
```

- Шаардлагад нийцсэн Provider олдсон тохиолдолд `getBestProvider` функц хамгийн сайн тохирсоныг нь буцаах бөгөөд эсвэл `getProvider` функцээр бүгдийг нь ашиглаж болно.

# Finding Providers Based on Requirement Criteria

- Хэрэв 1-с олон Location provider олдож байвал хамгийн нарийвчлал сайтайг нь сонгодог

```
String bestProvider = locationManager.getBestProvider(criteria, true);
```

- Хэрэв тохирох Location provider олдохгүй бол олдох хүртэл доорх дарааллаар шаардлагыг сулруулна.

- Эрчим хүч зарцуулалт
- Нарийвчлал
- Чиглэл, хурд, өндөр зэргийг буцаах чадвар

# Find Your Location

- Location-based service-н зорилго нь тухайн төхөөрөмжны физик байрлалыг тодорхойлох
- Үүнийг Location manager систем сервисийг ашиглаж хийнэ

```
String serviceString = Context.LOCATION_SERVICE;  
LocationManager locationManager;  
locationManager = (LocationManager) getSystemService(serviceString);
```

- Үүнийг ашиглахын өмнө LBS техник хангамж руу хандах uses-permission авах шаардлагатай

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

# Find Your Location

- Мөн provider бүрээр хамгийн сүүлд тодорхойлсон байрлал руу хандах боломжтой

```
String provider = LocationManager.GPS_PROVIDER;  
Location location = locationManager.getLastKnownLocation(provider);
```

- Энэ нь уртраг, өргөрөг, чиглэл, өндөр, хурд болон байрлалын хугацааг агуулна



# Tracking Movement

- Ихэнх байршил тодорхойлдог апп-ууд хэрэглэгчийн хөдөлгөөнийг тусгадаг
- Location Manager нь Location Provider-с шинэ байршил тодорхойлохдоо хөдөлгөөнийг өөр дээрээ тооцдоггүй.
- requestLocationUpdates функцыг ашиглан одоогын байршлын өөрчлөлтүүдийг авна.



# Tracking Movement

```
String provider = LocationManager.GPS_PROVIDER;
```

```
int t = 5000; // milliseconds
```

```
int distance = 5; // meters
```

```
LocationListener myLocationListener = new LocationListener() {  
  
    public void onLocationChanged(Location location) {  
        // Update application based on new location.  
    }  
  
    public void onProviderDisabled(String provider){  
        // Update application if provider disabled.  
    }  
  
    public void onProviderEnabled(String provider){  
        // Update application if provider enabled.  
    }  
  
    public void onStatusChanged(String provider, int status,  
                                Bundle extras){  
        // Update application if provider hardware status changed.  
    }  
};
```

```
locationManager.requestLocationUpdates(provider, t, distance,  
                                        myLocationListener);
```



# Tracking Movement

- Хугацаа болон зай хэтэрвэл Location Listener нь onLocationChanged үзэгдлийг ажлуулдаг
- Байршил шинэчлэлтийг дараах байдлаар зогсооно

```
locationManager.removeUpdates(myLocationListener);
```



# Tracking Movement

```
@Override
public void onCreate(Bundle icle) {
    super.onCreate(icle);
    setContentView(R.layout.main);

    LocationManager locationManager;
    String context = Context.LOCATION_SERVICE;
    locationManager = (LocationManager) getSystemService(context);

    Criteria criteria = new Criteria();
    criteria.setAccuracy(Criteria.ACCURACY_FINE);
    criteria.setAltitudeRequired(false);
    criteria.setBearingRequired(false);
    criteria.setCostAllowed(true);
    criteria.setPowerRequirement(Criteria.POWER_LOW);

    String provider = locationManager.getBestProvider(criteria, true);

    Location location = locationManager.getLastKnownLocation(provider);

    updateWithNewLocation(location);

    locationManager.requestLocationUpdates(provider, 2000, 10,
                                            locationListener);
}
```



# Tracking Movement

- 2 секунд тутамд 10 метрээс илүү хөдөлгөөн хийсэн өөрчлөлтийг чагнах

```
private final LocationListener locationListener = new LocationListener() {  
    public void onLocationChanged(Location location) {  
        updateWithNewLocation(location);  
    }  
  
    public void onProviderDisabled(String provider){  
        updateWithNewLocation(null);  
    }  
  
    public void onProviderEnabled(String provider){ }  
    public void onStatusChanged(String provider, int status,  
                                Bundle extras){ }  
};
```



# Tracking Movement

- updateWithNewLocation()

```
private void updateWithNewLocation(Location location) {  
    String latLongString;  
    TextView myLocationText;  
    myLocationText = (TextView)findViewById(R.id.myLocationText);  
    if (location != null) {  
        double lat = location.getLatitude();  
        double lng = location.getLongitude();  
  
        latLongString = "Lat:" + lat + "\nLong:" + lng;  
    } else {  
        latLongString = "No location found";  
    }  
    myLocationText.setText("Your Current Position is:\n" +  
                           latLongString);  
}
```



# Using Proximity Alerts

- Хэрэглэгч тодорхой байрлалд очих болон байрлалаас холдоход апп хариу үйлдэл үзүүлэхэд ихэвчлэн хэрэглэгддэг.
- Өдөөлт бий болоход Intent шиддэг бөгөөд ихэвчлэн BroadcastIntent ашиглагддаг ба PendingIntent тодорхойлох шаардлагатай

```
Intent intent = new Intent(MY_ACTIVITY);  
PendingIntent pendingIntent = PendingIntent.getBroadcast(this, -1, intent, 0);
```

# Using Proximity Alerts

- Proximity Alert тохируулахад бүрхэх талбайн
    - Төв цэгийг Уртраг, Өргөрөгөөр
    - Тухайн цэгийг тойрох радиус
- Болон тухайн Alert-г дуусах хугацааг тохируулна





# Using Proximity Alerts

- Location Manager заасан мужид орох болон гарахыг илрүүлэх үед  
LocationManager.KEY\_PROXIMITY\_ENTERING  
түлхүүр дээр true эсвэл false утга дамжуулдаг

```
public class ProximityIntentReceiver extends BroadcastReceiver {  
  
    @Override  
    public void onReceive (Context context, Intent intent) {  
        String key = LocationManager.KEY_PROXIMITY_ENTERING;  
  
        Boolean entering = intent.getBooleanExtra(key, false);  
        [ ... perform proximity alert actions ... ]  
    }  
  
}
```

# Using Proximity Alerts

- Хүлээн авах BroadcastReceiver–ээ бүртгүүлэх

```
IntentFilter filter = new IntentFilter(TREASURE_PROXIMITY_ALERT);  
registerReceiver(new ProximityIntentReceiver(), filter);
```







# Using the Geocoder

- Дээрх 2 функц аль аль нь Хаягын жагсаалтыг буцаах ба тэдгээр нь тухайн хаягын боломжит утгууд байна



# Reverse Geocoding

- Geocoder заасан координатад ямар нэгэн хаяг тодорхойлж чадахгүй бол null утга буцаана

```
location = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
```

```
double latitude = location.getLatitude();  
double longitude = location.getLongitude();
```

```
Geocoder gc = new Geocoder(this, Locale.getDefault());
```

```
List<Address> addresses = null;
```

```
try {
```

```
    addresses = gc.getFromLocation(latitude, longitude, 10);
```

```
} catch (IOException e) {}
```

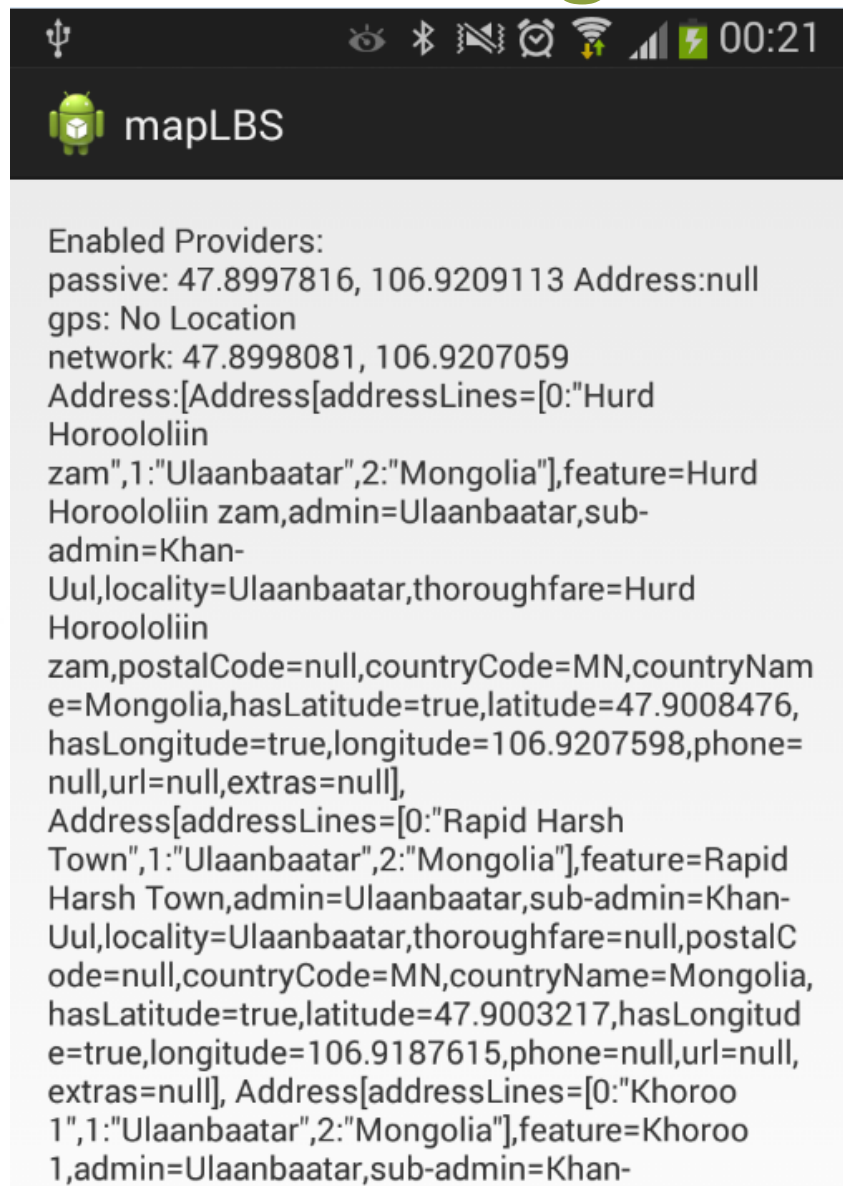
# Forward Geocoding

- Заасан хаягнаас уртраг/өргөрөг тодорхойлох

```
Geocoder fwdGeocoder = new Geocoder(this, Locale.US);  
String streetAddress = "160 Riverside Drive, New York, New York";  
  
List<Address> locations = null;  
try {  
    locations = fwdGeocoder.getFromLocationName(streetAddress, 10);  
} catch (IOException e) {}
```

# Forward Geocoding

- Үр дүн



# Forward Geocoding

- Заасан хаягнаас уртраг/өргөрөг тодорхойлох

```
Geocoder fwdGeocoder = new Geocoder(this, Locale.US);  
String streetAddress = "160 Riverside Drive, New York, New York";  
  
List<Address> locations = null;  
try {  
    locations = fwdGeocoder.getFromLocationName(streetAddress, 10);  
} catch (IOException e) {}
```