

Android - GPS

- This application allows the user to see their current location based on latitude and longitude
- The latitude and longitude gets updated on change of location
- `LocationListener` is used as callback whenever there are location events
- The permission `android.permission.ACCESS_FINE_LOCATION` has to be declared in `AndroidManifest.xml`
- `LocationManager` is used to access the GPS information on the device
- The location information can be obtained in 3 ways :
 - `NETWORK_PROVIDER`
 - `GPS_PROVIDER` (used in the example)
 - `PASSIVE_PROVIDER`
- **Override** `onLocationChanged()`
 - Called when the location has changed
 - To update the change in latitude and longitude

- The other callback methods are:
 - `onProviderDisabled()`
 - Called when the provide is disabled by the user
 - `onProviderEnabled()`
 - Called when the provider is enabled by the user
 - `onStatusChanged()`
 - Called when the provider status changes
- It may take a while for GPS information to be obtained
- Good practice to display the last known location whenever available
- `getLastKnownLocation()`
 - Gets the last known location
 - Returns `Location` object from which longitude and latitude from previous session can be obtained using `getLongitude()` and `getLatitude()`

Using the “geo fix” command in the emulator console

- To send mock location data from the command line.
- Launch your application in the Android emulator and open a terminal/console in your SDK's /tools directory.
- Connect to the emulator console. Use the command:
telnet localhost <console-port>
telnet localhost 5554
- Send the location data:
geo fix longitude latitude altitude
- This command accepts a longitude and latitude in decimal degrees, and an optional altitude in meters. For example:
geo fix -121.45356 46.51119 4392

Three ways to emulate GPS using DDMS

- Switch to the DDMS (Dalvik Debug Monitor Service) perspective.
- Using the Manual tab under Location Controls, manually send individual longitude/latitude coordinates to the device.
- Use a GPX file describing a route for playback to the device.
- Use a KML file describing individual place marks for sequenced playback to the device.

Accessed in Android Studio with:
Tools -> Android -> Android Device Monitor

The screenshot displays the Android Studio interface. On the left, a virtual Android device is shown with a GPS application. The application's status bar at the top indicates a location icon, 3G signal, and the time 12:25. The app's main screen is titled "GPS" and displays the following data:

- Longitude: -122.084095
- Latitude: 37.422005
- Status: -

On the right, the Android Device Monitor is open, showing various tabs. The "System Information" tab is selected, displaying the following sections:

- Telephony Status:** Voice: home, Speed: Full, Data: home, Latency: None.
- Telephony Actions:** Incoming number: (empty field), Voice (selected), SMS, Message: (empty field), Call, Hang Up.
- Location Controls:** Manual (selected), GPX, KML, Decimal (selected), Sexagesimal, Longitude: -122.084095, Latitude: 37.422006, Send.

References

- [LocationManager](#)
- [LocationListener](#)

Exercise

- Add another label that displays the accuracy of the Location object.