Как вы помните, нам потребуются координаты пользователя, чтобы узнать погоду в том месте, где он находится. В этом нам поможет API для работы с геолокацией в Android.

Работа с LocationManager

Процесс получения координат в Android достаточно прост:

1. Получаем LocationManager.
2. Подписываемся на обновления координат.
3. Когда координаты изменяются, LocationManager дёргает наш коллбэк.

LocationManager работает с так называемыми **провайдерами**, источниками, из которых система получает геоданные.

Существуют такие провайдеры:

* GPS\_PROVIDER — определяет координаты, используя спутники **GPS**.
* NETWORK\_PROVIDER — определяет координаты, используя **мобильную** и **Wi-Fi** сеть.
* PASSIVE\_PROVIDER — по сути не получает координаты сам, а лишь использует данные, полученные **другими приложениями**. Если никакие другие приложения при этом не получают обновлений координат, то и мы не получим.
* FUSED\_PROVIDER — специальный провайдер, который получает данные, используя **все другие** провайдеры сразу.

Получаем разрешение

Чтобы работать с геолокацией, нам потребуется соответствующее разрешение:

* ACCESS\_FINE\_LOCATION — если нам нужны точные координаты (получаемые по GPS).
* ACCESS\_COARSE\_LOCATION — если нам нужны примерные координаты (получаемые, например, по сети).

Добавьте разрешение в манифест:

Поскольку гео разрешения считаются **опасными**, на новых версиях Android придётся запросить **Runtime Permission**.

В MainActivity добавьте код:

/\*\*

\* Проверяем, есть ли разрешение и запрашиваем его, если нет

\*/

private void checkAndRequestGeoPermission() {

int permissionCheck = ContextCompat.checkSelfPermission(this,

Manifest.permission.ACCESS\_FINE\_LOCATION);

if (permissionCheck != PackageManager.PERMISSION\_GRANTED) {

ActivityCompat.requestPermissions(this,

new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION},

REQUEST\_CODE\_LOCATION\_PERMISSION);

} else {

setupLocation();

}

}

/\*\*

\* Подписываемся на обновления гео

\*/

private void setupLocation() {

}

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

super.onRequestPermissionsResult(requestCode, permissions, grantResults);

if (requestCode == REQUEST\_CODE\_LOCATION\_PERMISSION) {

if (grantResults.length > 0

&& grantResults[0] == PackageManager.PERMISSION\_GRANTED) {

setupLocation();

} else {

// Нет гео

// Попробуем показать ещё раз

checkAndRequestGeoPermission();

}

}

}

Этот код должен быть вам знаком, так как ранее мы уже запрашивали рантайм разрешения ранее: проверяем, есть ли разрешение, если есть — начинаем работать с гео, если нет — запрашиваем. В коллбэке результата запроса разрешения мы опять проверяем, получили ли разрешение, если нет — запрашиваем ещё раз, если да — начинаем работать с гео.

Вызовите метод checkAndRequestGeoPermission() в конце onCreate():

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

checkAndRequestGeoPermission();

}

Запрашиваем обновления гео

Итак, теперь начнём работу с геолокацией. Создайте приватное поле LocationManager:

public class MainActivity extends AppCompatActivity {

private static final int REQUEST\_CODE\_LOCATION\_PERMISSION = 0;

private LocationManager locationManager;

Так же создадим LocationListener, методы которого будут вызываться, когда происходят какие-то изменения, касающиеся гео:

@Override

public void onLocationChanged(Location location) {

Log.v(TAG, "Location changed: " + location);

}

@Override

public void onStatusChanged(String provider, int status, Bundle extras) {

Log.v(TAG, "Status changed: " + provider + ", status: " + status);

}

@Override

public void onProviderEnabled(String provider) {

Log.v(TAG, "Provider enabled: " + provider);

}

@Override

public void onProviderDisabled(String provider) {

Log.v(TAG, "Provider disabled: " + provider);

}

};

У него четыре метода:

* onLocationChanged() — вызывается, когда изменяется геопозиция.
* onStatusChanged() — вызывается, когда изменяется статус провайдера.
* onProviderEnabled() — вызывается, когда один из провайдеров стал доступен.
* onProviderDisabled() — вызывается, когда один из провайдеров стал недоступен.

Прежде чем запросить обновления гео, нужно решить, какой провайдер мы будем использовать. Некоторые провайдеры могут быть недоступны (например, если их отключит пользователь), поэтому обратимся к LocationManager за помощью — он подскажет нам, какой провайдер использовать.

В методе setupLocation() добавьте код:

Criteria criteria = new Criteria();

При помощи класса Criteria мы можем указать параметры, по которым LocationManager подберёт подходящий провайдер. Нам достаточно дефолтных параметров, поэтому не будем настраивать критерии.

Теперь получим провайдер:

**Первый параметр** — критерий, который мы будем использовать. **Второй параметр** — должен ли метод производить подбор только среди включенных провайдеров. Нас не интересуют отключенные, поэтому передадим true.

Итак, этот метод найдёт доступные провайдеры, соответствующие переданным критериям, и вернёт лучший, сортируя по таким параметрам, как энергопотребление, точность и т.д.

Если же найти подходящий провайдер не выйдет, метод вернёт null.

Теперь подпишемся на обновления геолокации:

locationManager.requestLocationUpdates(

bestProvider,

0,

0,

locationListener

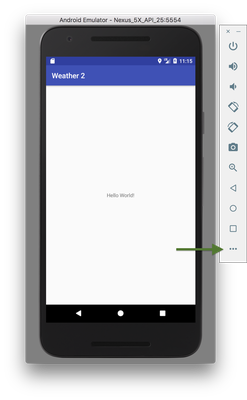
);

}

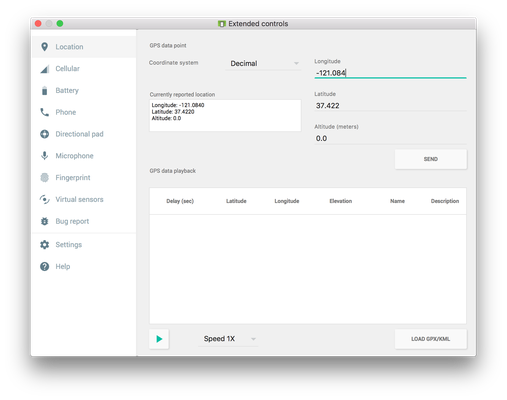
* **Первый параметр** — провайдер, на обновления которого мы подписываемся.
* **Второй параметр** — минимальное время в миллисекундах, спустя которое мы будем получать обновления. Это значит, что если мы передали, скажем, 10 секунд, а геопозиция изменится через 1 секунду, то мы получим обновление не ранее, чем через 10 секунд.
* **Третий параметр** — минимальное расстояние в метрах, спустя которое мы получим обновление.
* **Четвёртый параметр** — listener, который будет вызываться при обновлении.

Запустите приложение. После запуска вы не получите обновлений, пока геопозиция не изменится. Давайте изменим её.

Нажмите на эту кнопку:

[](https://skillberg.com/media/uploads/2017/11/17/54f2f4fe25b2436a8e32a1f6a9e1906a.png)Кнопка настроек эмулятора

Откроется окно:

[](https://skillberg.com/media/uploads/2017/11/17/702dd8296de446ffa51f94948c0145eb.png)Настройка геопозиции эмулятора

Измените координаты и нажмите кнопку **SEND**, после чего вы увидите в логах:

11-14 23:13:41.305 5474-5474/com.skillberg.weather2 V/Weather/MainActivity: Location changed: Location[gps 37.421998,-121.084000 acc=20 et=+2h21m49s724ms alt=0.0 {Bundle[mParcelledData.dataSize=40]}]

Получение последней доступной геопозиции

А что делать, если нам нужно узнать геопозицию при запуске, ведь мы не получим информации, если пользователь не будет двигаться?

Мы можем получить последнюю известную геопозицию, используя метод LocationManager.getLastKnownLocation(). Мы передаём в этот метод предпочитаемый провайдер, а получаем назад геопозицию, если, конечно, она доступна. В противном случае получим null.

Измените метод setupLocation():

private void setupLocation() {

// Получаем LocationManager

locationManager = (LocationManager) getSystemService(LOCATION\_SERVICE);

// Получаем лучший провайдер

Criteria criteria = new Criteria();

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

Log.v(TAG, "Best provider: " + bestProvider);

if (bestProvider != null) {

// Получаем последнюю доступную позицию

Location lastKnownLocation = locationManager.getLastKnownLocation(bestProvider);

Log.v(TAG, "Last location: " + lastKnownLocation);

// Подписываемся на обновления

locationManager.requestLocationUpdates(

bestProvider, // провайдер

0, // мин. время

0, // мин. расстояние

locationListener

);

}

}

В логах увидим:

11-14 23:20:45.532 5586-5586/com.skillberg.weather2 V/Weather/MainActivity: Last location: Location[gps 37.421998,-121.084000 acc=20 et=+2h21m49s724ms alt=0.0 {Bundle[mParcelledData.dataSize=40]}]

Отлично!

Отписка от изменений геопозиции

Мы должны обязательно отписаться от обновлений в тот момент, когда перестаём нуждаться в них — например, когда Activity закрывается.

В нашем случае мы подписываемся на обновления в onCreate(), значит будет логичным отписаться в onDestroy():

@Override

protected void onDestroy() {

if (locationManager != null) {

locationManager.removeUpdates(locationListener);

}

super.onDestroy();

}

**Обратите внимание:** в общем случае лучше подписываться в onStart() и отписываться в onStop().

Другие способы получения геопозиции

Есть и иные способы получения гео — например, с использованием **Google Play Services**. но об этом в другой раз.

В следующем уроке мы поговорим о сервисах.

ПРИМЕРЫ КОДА

[ИСХОДНЫЙ КОД УРОКА](https://github.com/Skillberg/WeatherAndroid2/tree/lesson-33)