

Android GPS

Для работы с местоположением необходимо разрешение

ACCESS_FINE_LOCATION

ACCESS_COARSE_LOCATION

Следует отметить, что данное разрешение относится к группе Dangerous

Основным классом для работы с местоположением является класс

LocationManager

Получить экземпляр данного класса можно используя метод **getSystemService**

```
LocationManager locationManager =  
    (LocationManager) getSystemService(LOCATION_SERVICE);
```

Для запроса местоположения используется метод **LocationManager**-а:

```
void requestLocationUpdates(  
    String provider,  
    long minTime,  
    float minDistance,  
    LocationListener listener  
)
```

Параметры:

String provider - тип поставщика координат, могут быть указаны:

- **LocationManager.GPS_PROVIDER**
- **LocationManager.NETWORK_PROVIDER**

long minTime - минимальный интервал обновления в миллисекундах (0 - автоматически)

float minDistance – минимальное расстояние, при смещении на которое произойдет повторный запрос местоположения (0 – автоматически)

LocationListener listener – экземпляр класса **LocationListener** который будет выполнять обработку принятого местоположения

Пример (делаем сразу 2 запроса):

```
@SuppressWarnings("MissingPermission")
private void requestPosition() {
    if (!checkPermission()) return;
    locationManager.requestLocationUpdates(
        locationManager.GPS_PROVIDER,
        1000 * 10,
        10,
        locationManager
    );
    locationManager.requestLocationUpdates(
        locationManager.NETWORK_PROVIDER,
        1000 * 10,
        10,
        locationManager
    );
}
```

Метод **checkPermission()** проверяет наличие необходимых разрешений:

```
boolean checkPermission() {
    return ActivityCompat.checkSelfPermission(
        this,
        Manifest.permission.ACCESS_FINE_LOCATION
    ) == PackageManager.PERMISSION_GRANTED
    || ActivityCompat.checkSelfPermission(
        this,
        Manifest.permission.ACCESS_COARSE_LOCATION
    ) == PackageManager.PERMISSION_GRANTED;
}
```

Проверить доступность провайдера можно так:

```
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)
```

Рассмотрим реализацию интерфейса `LocationListener`

```
LocationListener locationListener = new LocationListener() {

    @Override
    public void onLocationChanged(Location location) {
        showLocation(location);
    }

    @Override
    public void onProviderDisabled(String provider) {}

    @Override @SuppressWarnings("MissingPermission")
    public void onProviderEnabled(String provider) {
        if (!checkPermission()) return;
        Location last =
            locationManager.getLastKnownLocation(provider);
        showLocation(last);
    }

    @Override
    public void onStatusChanged(String provider, int status,
        Bundle extras) {}
};
```

Метод **`onLocationChanged`** вызывается при получении нового местоположения от `LocationManager` и принимает экземпляр класса `Location`, для вывода которого в данном примере реализован метод **`showLocation()`**:

```
void showLocation(Location location) {
    StringBuilder builder = new StringBuilder();
    builder.append("latitude:");
    builder.append(location.getLatitude());
    builder.append(" , ");
    builder.append("longitude:");
    builder.append(location.getLongitude());
    builder.append(" , ");
    Toast.makeText(this, builder.toString(), Toast.LENGTH_LONG)
        .show();
}
```

Метод **`onProviderEnabled`** вызывается, когда использование провайдера становится возможным, в этот момент можно получить последние координаты и вывести их:

```
locationManager.getLastKnownLocation(provider)
```

Метод **onStatusChanged** вызывается, если изменился статус указанного провайдера.

В поле `status` могут быть значения

- `OUT_OF_SERVICE` (данные будут недоступны долгое время),
- `TEMPORARILY_UNAVAILABLE` (данные временно недоступны),
- `AVAILABLE` (все ок, данные доступны).

В этом методе мы просто выводим новый статус на экран.

ВНИМАНИЕ:

Примененная в примерах аннотация `@SuppressWarnings("MissingPermission")` подавляет ошибку связанную с отсутствием прав, но проверка прав вынесена в метод `checkPermission()`, поэтому проблемы не будет.

Важно отметить, что части кода, отвечающие за получение `RuntimePermission` упущены