

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/main"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:orientation="vertical"
9     android:background="@drawable/bgr"
10    tools:context=".MainActivity">
11
12
13    <View
14        android:layout_width="0dp"
15        android:layout_height="0dp"
16        android:layout_weight="1" />
17
18    <TextView
19        android:id="@+id/textView"
20        android:layout_width="wrap_content"
21        android:layout_height="wrap_content"
22        android:layout_gravity="center_horizontal"
23        android:background="@color/pink"
24        android:padding="10dp"
25        android:text="какая погода" />
26
27    <Button
28        android:id="@+id/btVar1"
29        android:layout_width="wrap_content"
30        android:layout_height="wrap_content"
31        android:layout_gravity="center"
32        android:text="узнать погоду"
33        android:background="@color/white"/>
34
35    <View
36        android:layout_width="0dp"
37        android:layout_height="0dp"
38        android:layout_weight="1" />
```

```

import android.annotation.SuppressLint
import android.content.pm.PackageManager
import android.location.Location
import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.android.volley.Request
import com.android.volley.toolbox.StringRequest
import com.android.volley.toolbox.Volley
import com.google.android.gms.location.FusedLocationProviderClient
import com.google.android.gms.location.LocationServices
import org.json.JSONObject

```

```

</> class MainActivity : AppCompatActivity() {
    private val apiKey = "32fc6ad2c543df4a49e6cb9778ad56d8"

    private lateinit var btVar1: Button
    private lateinit var textView: TextView
    private lateinit var fusedLocationClient: FusedLocationProviderClient
    private val LOCATION_PERMISSION_REQUEST_CODE = 1

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }
    }
}

```

```

        btVar1 = findViewById(R.id.btVar1)
        fusedLocationClient = LocationServices.getFusedLocationProviderClient(this)

        btVar1.setOnClickListener {
            checkForPermission()
        }
    }

    private fun checkForPermission() {
        if (ActivityCompat.checkSelfPermission(context, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
            ActivityCompat.checkSelfPermission(context, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(
                this,
                arrayOf(Manifest.permission.ACCESS_FINE_LOCATION, Manifest.permission.ACCESS_COARSE_LOCATION),
                LOCATION_PERMISSION_REQUEST_CODE
            )
        } else {
            obtainLocation()
        }
    }

    override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults)
        if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {
            if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                obtainLocation()
            } else {
                Toast.makeText(this, "Разрешение отклонено", Toast.LENGTH_SHORT).show()
            }
        }
    }

    @SuppressLint("MissingPermission")
    private fun obtainLocation() {
        fusedLocationClient.lastLocation.addOnSuccessListener { location: Location? ->
            if (location != null) {
                val weatherUrl = "https://api.openweathermap.org/data/2.5/weather?lat=${location.latitude}&lon=${location.longitude}&units=metric&"
                getTemp(weatherUrl)
            } else {
            }
        }
    }
}

```

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools">
4
5     <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
6     <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
7     <uses-permission android:name="android.permission.INTERNET"/>
8
9     <application
10         android:allowBackup="true"
11         android:dataExtractionRules="@xml/data_extraction_rules"
12         android:fullBackupContent="@xml/backup_rules"
13         android:icon="@mipmap/ic_launcher"
14         android:label="@string/app_name"
15         android:roundIcon="@mipmap/ic_launcher_round"
16         android:supportRtl="true"
17         android:theme="@style/Theme.AppCompat"
18         tools:targetApi="31">
19         <activity
20             android:name=".MainActivity"
21             android:exported="true">
22             <intent-filter>
23                 <action android:name="android.intent.action.MAIN" />
24
25                 <category android:name="android.intent.category.LAUNCHER" />
26             </intent-filter>
27         </activity>

```

Manifest

```
alias(libs.plugins.android.application)
alias(libs.plugins.kotlin.android)
}

android {
    namespace = "com.example.a11"
    compileSdk = 35

    defaultConfig {
        applicationId = "com.example.a11"
        minSdk = 24
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }

    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }

    kotlinOptions {
        jvmTarget = "1.8"
    }
}
```

## Buildgradle

<https://home.openweathermap.org/>

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.