

Practice 11. Solunin

Регистрируемся в openWeatherMap и получаем наш API ключ

Создаем макет

Activity_main

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical"
    android:padding="20dp"
    android:background="@drawable/fon">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Какая погода у вас?"
        android:textColor="@color/black"
        android:textSize="30sp"/>

    <Button
        android:id="@+id/btVar1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Нажми на меня!" />

</LinearLayout>
```

MainActivity

```
package com.example.myapplication11

import android.Manifest
import android.annotation.SuppressLint
import android.app.DownloadManager
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.JSONException
import org.json.JSONObject

class MainActivity : AppCompatActivity() {
```

```

// Ваш ключ API для доступа к OpenWeatherMap
var api_key = "626efee14f966bc74fd2ad04452ec919"

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
    }

    textView = findViewById(R.id.textView)
    btVar1 = findViewById(R.id.btVar1)

    // Создаем экземпляр клиента для получения местоположения
    fusedLocationClient =
LocationServices.getFusedLocationProviderClient(this)

    // Проверяем разрешения на доступ к местоположению
    btVar1.setOnClickListener {
        checkForPermission()
    }
}

// Метод для проверки наличия разрешений
private fun checkForPermission() {
    if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
        // Запрашиваем разрешения, если они не предоставлены
        ActivityCompat.requestPermissions(this,
            arrayOf(Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION),
                LOCATION_PERMISSION_REQUEST_CODE)
    } else {
        // Если разрешения уже предоставлены, получаем местоположение
        obtainLocation()
    }
}

// Метод, который обрабатывает результат запроса разрешений
override fun onRequestPermissionsResult(requestCode: Int, permissions:
Array<out String>, grantResults: IntArray) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults)
    // Проверяем код запроса разрешений
    if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {
        // Если разрешение было предоставлено
        if (grantResults.isNotEmpty() && grantResults[0] ==

```

```

PackageManager.PERMISSION_GRANTED) {
    // Получаем местоположение
    obtainLocation()
} else {
    Toast.makeText(this, "Разрешение отклонено",
Toast.LENGTH_SHORT).show()
}
}

}

@SuppressLint("MissingPermission")
private fun obtainLocation() {
    // Получаем последнее известное местоположение
    fusedLocationClient.lastLocation
        .addOnSuccessListener { location: Location? ->
            // Проверяем, что местоположение не равно null
            if (location != null) {
                val weatherUrl =
"https://api.openweathermap.org/data/2.5/weather?lat=${location.latitude}&lon=${location.longitude}&units=metric&appid=${api_key}"
                // Запрашиваем температуру по текущему местоположению
                getTemp(weatherUrl)
            } else {
                Toast.makeText(this, "Не удалось получить
местоположение", Toast.LENGTH_SHORT).show()
            }
        }
        .addOnFailureListener { exception ->
            Toast.makeText(this, "Location Permission not granted",
Toast.LENGTH_SHORT).show()
        }
    }

    private fun getTemp(url: String) {
        // Создаем очередь для запросов
        val queue = Volley.newRequestQueue(this)

        // Выполняем запрос к API погоды
        val stringReq = StringRequest(
            Request.Method.GET, url, { response ->
                try {
                    // Получаем JSON-объект из ответа
                    val obj = JSONObject(response)

                    // Получаем данные о температуре и влажности из объекта
                    val main: JSONObject = obj.getJSONObject("main")
                    val temperature = main.getString("temp")
                    val humidity = main.getString("humidity") // Получаем
влажность

                    // Получаем название города из объекта
                    val city = obj.getString("name")

                    // Обновляем текстовое поле с температурой и влажностью
                    textView.text = "${temperature} °C в ${city}, Влажность:
${humidity}%"
                } catch (e: JSONException) {
                    textView.text = "Ошибка парсинга данных!"
                }
            },
            // В случае ошибки показываем сообщение
            { textView.text = "Ошибка запроса!" }
        )
    }
}

```

```
// Добавляем запрос в очередь
queue.add(stringReq)
}
```

Result

