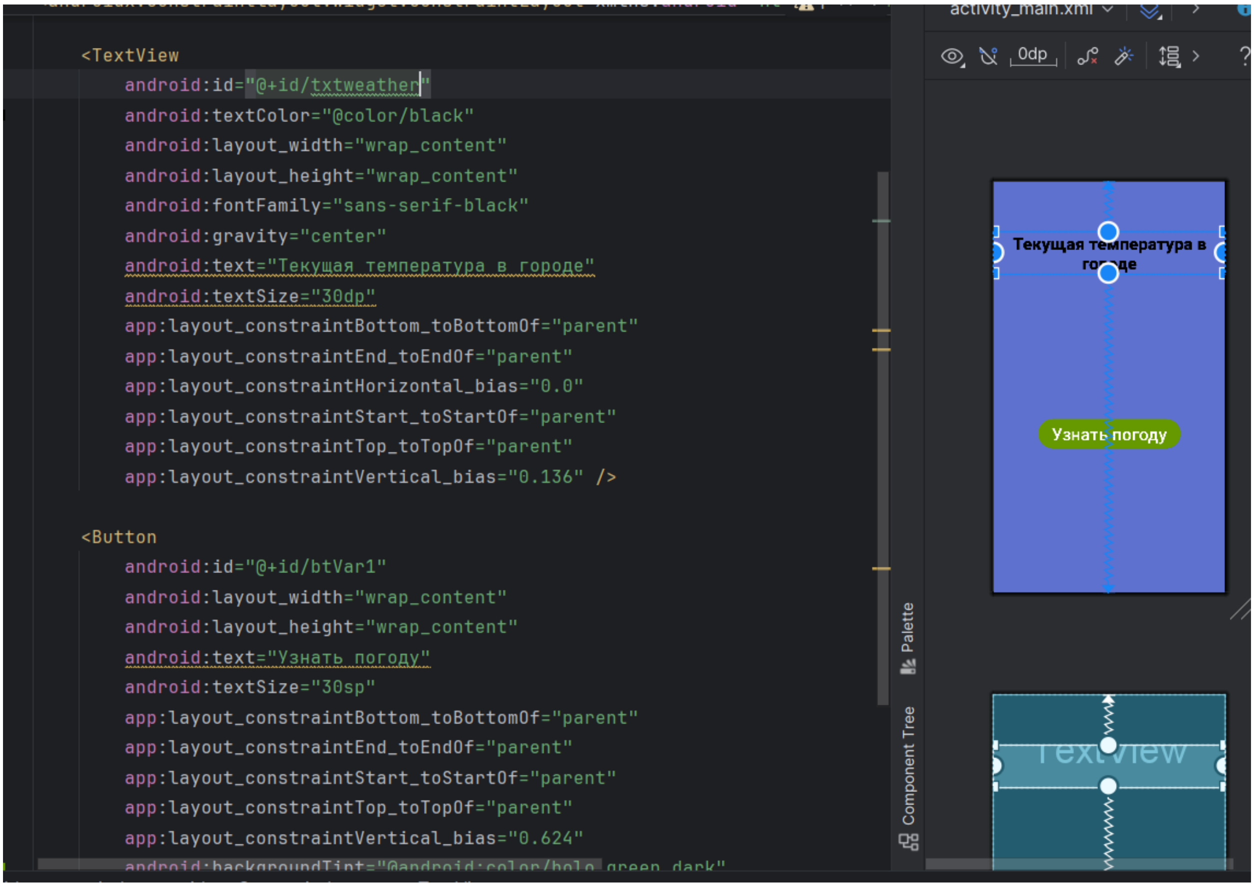
Работа с Api, на примере приложения погоды

1.В AndroidManifest дал разрешение на использования интернета и др

2.activity_main.xml

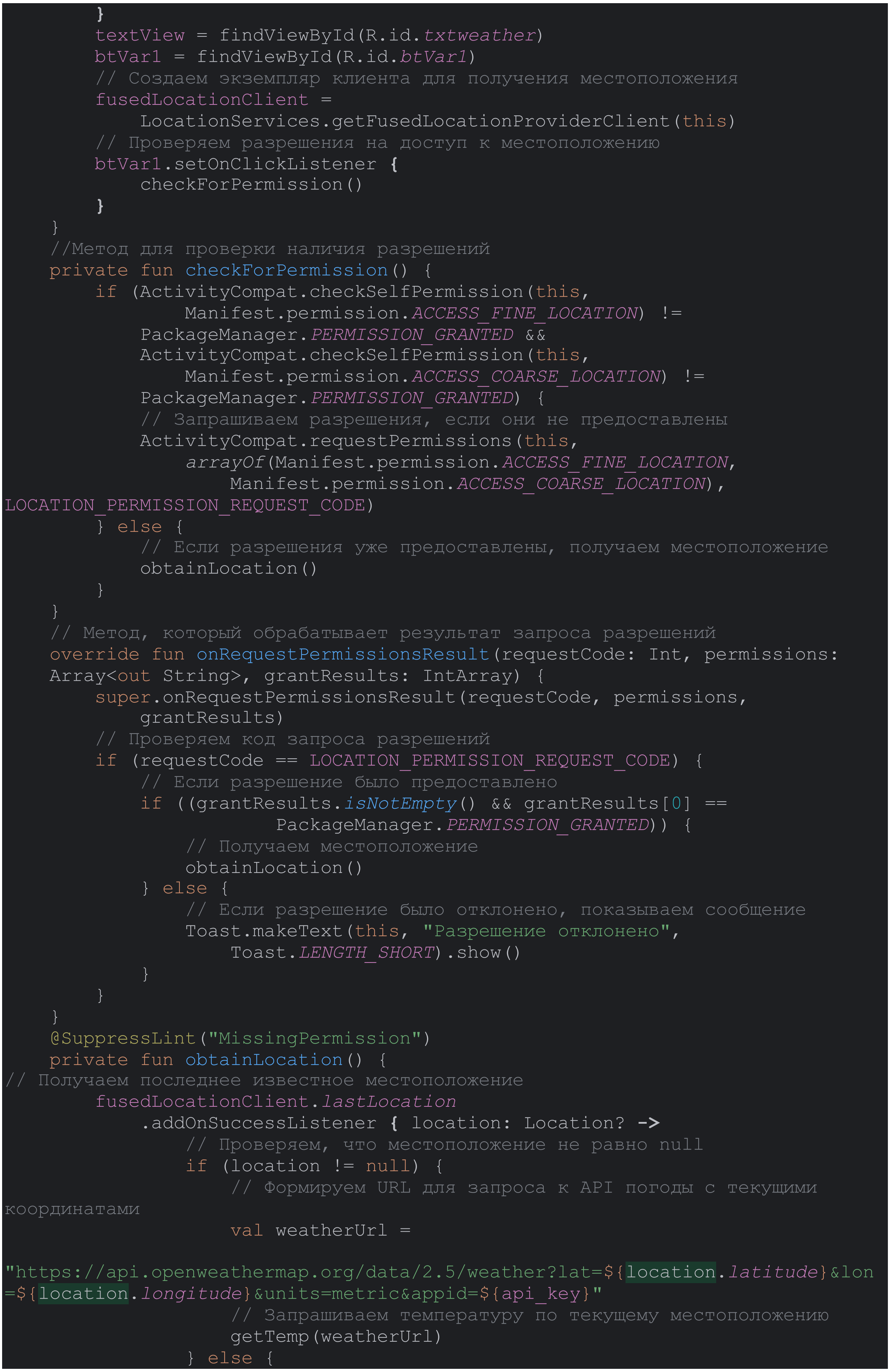


```
    <TextView
        android:id="@+id/txtweather"
        android:textColor="@color/black"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif-black"
        android:gravity="center"
        android:text="Текущая температура в городе"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.136" />

    <Button
        android:id="@+id/btVar1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Узнать погоду"
        android:textSize="30sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.624"
        android:backgroundTint="@android:color/holo_green_dark"
```

3.Main Activity

```
class MainActivity : AppCompatActivity() {
    // Ваш ключ API для доступа к OpenWeatherMap
    var api_key = "d92e83c9eeef62d2913e2b82d4fc9c6c"
    // Определяем переменные для интерфейса пользователя и клиента
местоположения
    private lateinit var btVar1: Button
    private lateinit var textView: TextView
    //FusedLocationProviderClient — это высокоуровневый API для отслеживания
местоположения, предоставляемый Google Play Services.
    private lateinit var fusedLocationClient: FusedLocationProviderClient
    // Код запроса разрешения на доступ к местоположению
    private val LOCATION_PERMISSION_REQUEST_CODE = 1
    @SuppressLint("MissingInflatedId")
    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
```



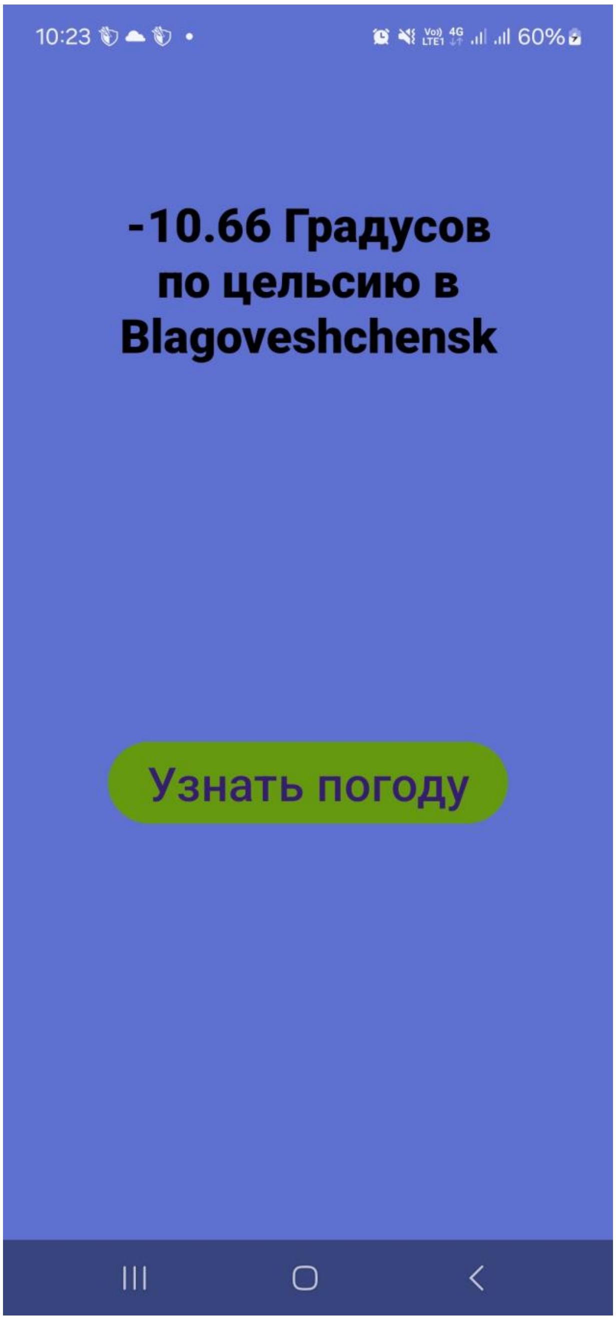
```
        }
        textView = findViewById(R.id.txtweather)
        btVar1 = findViewById(R.id.btVar1)
        // Создаем экземпляр клиента для получения местоположения
        fusedLocationClient =
            LocationServices.getFusedLocationProviderClient(this)
        // Проверяем разрешения на 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 {
                // Если разрешение было отклонено, показываем сообщение
                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) {
                    // Формируем URL для запроса к API погоды с текущими
координатами
                    val weatherUrl =

"https://api.openweathermap.org/data/2.5/weather?lat=${location.latitude}&lon
=${location.longitude}&units=metric&appid=${api_key}"
                    // Запрашиваем температуру по текущему местоположению
                    getTemp(weatherUrl)
                } else {
                    // Если не удалось получить местоположение, выводим сообщение
                    Toast.makeText(this, "Не удалось получить местоположение", Toast.LENGTH_SHORT).show()
                }
            }
            //это метод, который используется в Android для обработки неудачных результатов асинхронных операций
            .addOnFailureListener { exception ->
                Toast.makeText(this, "Location Permission not granted",
                    Toast.LENGTH_SHORT).show()
            }
    }
    private fun getTemp(url: String) {
        // Создаем очередь для запросов
        //Volley - это HTTP-библиотека, которая используется для
        // кэширования и выполнения сетевого запроса в приложениях Android.
        val queue = Volley.newRequestQueue(this)
        // Выполняем запрос к API погоды
        val stringReq = StringRequest(
            Request.Method.GET, url, { response ->
                // Получаем JSON-объект из ответа
                val obj = JSONObject(response)
                // Получаем данные о температуре из объекта
                val main: JSONObject = obj.getJSONObject("main")
                val temperature = main.getString("temp")
                println(temperature)
                // Получаем название города из объекта
                val city = obj.getString("name")
                println(city)
                val humidity = main.getString("humidity") // Получаем влажность
                println(humidity)
                // Устанавливаем текст в textView, показывая температуру и название города
                textView.text = "${temperature} Градусов по цельсию в ${city}"
                System.out.println(obj.toString())
            },
            // В случае ошибки показываем сообщение
            { textView.text = "Ошибка!" })
        // Добавляем запрос в очередь
        queue.add(stringReq)
    }
}
```

4.Также подключил библиотеки

```
implementation("com.google.android.gms:play-services-location:21.0.1")
implementation("com.android.volley:volley:1.2.1")
```

5.Результат



-10.66 Градусов по цельсию в Blagoveshchensk