

## Создание приложения в Android studio с использованием API на языке Java

*Андрюенко Иван Сергеевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### Аннотация

Целью данной статьи является создание приложения, информирующего о текущей погоде в городе Биробиджан, с помощью API технологий. Для создания мобильного приложения использовалась среда программирования Android Studio. В итоге было создано приложение, получающее погоду в городе Биробиджан в реальном времени.

**Ключевые слова:** Java, API, погода, Android studio

### Creating an application in Android studio using the API in the Java language

*Andrienko Ivan Sergeevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

The purpose of this article is to create an application that informs about the current weather in the city of Birobidzhan, using API technologies.

**Keywords:** Java, API, weather, Android studio

## 1 Введение

### 1.1 Актуальность

Сегодня рынок программного обеспечения для управления API стремительно развивается. Использование открытых API — не просто мода или веяние времени, это ответ на требования рынка. API сильно упрощает процесс программирования при создании приложений, абстрагируя базовую реализацию и предоставляя только объекты или действия, необходимые разработчику. Если графический интерфейс для почтового клиента может предоставить пользователю кнопку, которая выполнит все шаги для выборки и выделения новых писем, то API для ввода/вывода файлов может дать разработчику функцию, которая копирует файл из одного места в другое, не требуя от разработчика понимания операций файловой системы, происходящих за кулисами.

### 1.2 Обзор исследований

В своей работе Н.Б. Дворкина, Д.Е. Намиот рассмотрели практическое использование OpenCellID-проекта, поддерживающего открытую базу

данных с координатами сотовых операторов мобильной связи по всему миру. [1]. С.Э. Шмаков, Н.Н. Смекалин, Е.В. Щенникова рассмотрели способ создания Web-приложения на языке JavaScript с использованием стороннего API-интерфейса. Приводится способ эффективного создания приложения и способы взаимодействия с веб-сервером, акцент сделан на выборе архитектуры взаимодействия в зависимости от целей проекта [2]. В своей работе П.А. Сергеев использовал JSON API на основе Android-приложения по мониторингу рынка криптовалют «CoinData». Рассматриваются также общие аспекты работы с форматом JSON и сервисами API. Используемый при разработке язык программирования – Java [3]. В своей работе В.И. Васильев рассмотрел работу в среде разработки приложений Android Studio, а также работу с файлами разметки. Кроме того, в данной работе рассказывается про пошаговое создание элементарного интернет браузера [4]. А.А. Криворотов, Т.М. Марданова в своей научной статье рассмотрели пример разработки android-приложения в среде разработки Android Studio [5].

### **1.3 Цель исследования**

Цель исследования – создать приложение, получающее информацию о погоде в городе Биробиджан с помощью API технологий.

## **2 Материалы и методы**

Для получения данных о погоде использовался сервис OpenWeatherMap, предоставляющий API для доступа к данным о текущей погоде.

## **3 Результаты и обсуждения**

API (программный интерфейс приложения, интерфейс прикладного программирования) — описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой. Обычно входит в описание какого-либо интернет-протокола, программного каркаса или стандарта вызовов функций операционной системы. Часто реализуется отдельной программной библиотекой или сервисом операционной системы. Используется программистами при написании всевозможных приложений.

Для начала работы необходимо создать проект для телефона или планшета (phone and tablet), и выбрать шаблон. При выборе SDK рекомендуется выбрать «API 21: Android 5.0 (Lollipop)», так как при выборе более нового SDK будет падать процент устройств, поддерживающие создаваемое приложение.

После создания проекта и загрузки необходимых библиотек, необходимо создать примерный дизайн приложения. В Android studio это делается через специальный раздел – дизайн (рис. 1). В данном разделе для создания интерфейса надо просто перетаскивать необходимые объекты из панели слева. Создадим интерфейс для приложения (рис. 2).

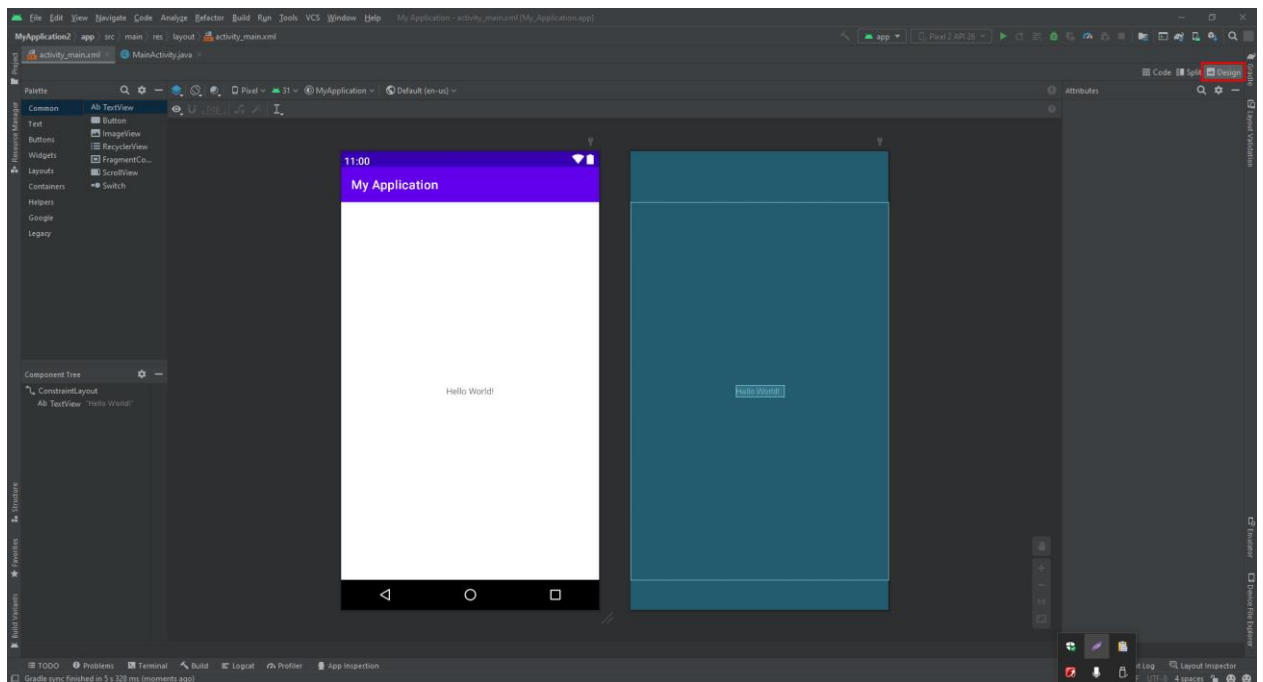


Рисунок 1 – Раздел дизайн

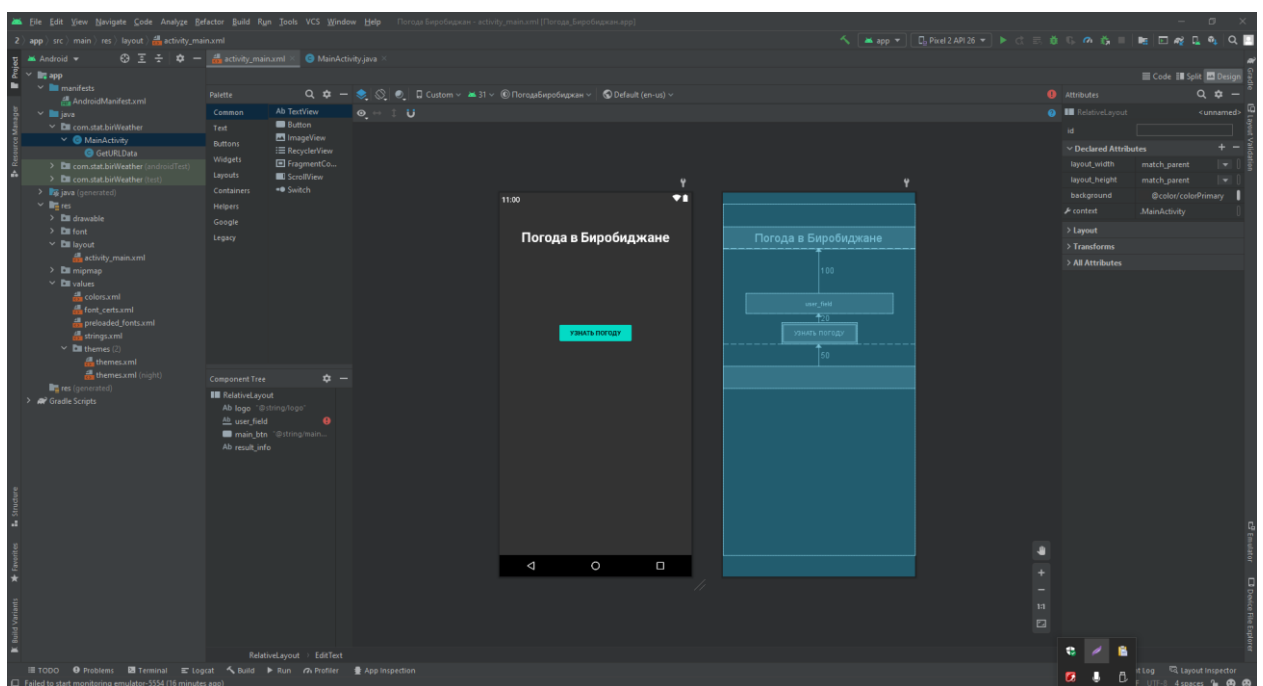


Рисунок 2 – Интерфейс будущего приложения

Теперь, после создания интерфейса, пишем код, отвечающий за саму работу приложения и получения данных с дополнительного сервиса. Для начала импортируем библиотеки, необходимые для работы (рис. 3).

```
1 package com.stat.birWeather;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.annotation.SuppressLint;
6 import android.os.AsyncTask;
7 import android.os.Bundle;
8 import android.view.View;
9 import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.TextView;
12 import android.widget.Toast;
13
14 import org.json.JSONException;
15 import org.json.JSONObject;
16
17 import java.io.BufferedReader;
18 import java.io.IOException;
19 import java.io.InputStream;
20 import java.io.InputStreamReader;
21 import java.net.HttpURLConnection;
22 import java.net.MalformedURLException;
23 import java.net.URL;
24
```

Рисунок 3 – Импорт необходимых библиотек

Теперь создаем поля, которые будут ссылаться на объекты из дизайна и устанавливаем к ним ссылки (рис.4). Это необходимо для того, чтобы в будущем уже через код задавать им какой-либо функционал.

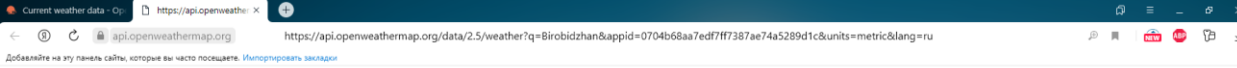
```
30 private EditText user_field;
31 private Button main_btn;
32 private TextView result_info;
33
34
35 @Override
36 protected void onCreate(Bundle savedInstanceState) {
37     super.onCreate(savedInstanceState);
38     setContentView(R.layout.activity_main);
39
40
41     user_field = findViewById(R.id.user_field);
42     main_btn = findViewById(R.id.main_btn);
43     result_info = findViewById(R.id.result_info);
44
```

Рисунок 4 – Создание ссылок на объекты из дизайн

Теперь создаем обработчик нажатия на кнопку (рис.5). Это необходимо для того, чтобы при нажатии на кнопку «Узнать погоду», приложение обращалось к сервису за информацией о погоде в Биробиджане. Для работы с сервисом, необходимо пройти по ссылке [6], и после регистрации получить API ключ в личном кабинете. Полученный ключ необходимо вставить в ссылку, полученную в личном кабинете, и при переходе по этой ссылке будет получена информация о погоде (рис.6).

```
50 main_btn.setOnClickListener(new View.OnClickListener() {
51     @Override
52     public void onClick(View view) {
53
54         if(user_field.getText().toString().trim().equals(""))
55             Toast.makeText(context MainActivity.this, R.string.no_user_input, Toast.LENGTH_LONG).show();
56         else {
57
58
59             //ключ = "0704b68aa7edf7ff7387ae74a5289d1c";
60             String url = "https://api.openweathermap.org/data/2.5/weather?q=Birbidzhan&appid=0704b68aa7edf7ff7387ae74a5289d1c&units=metric&lang=ru";
61
62
63             new GetURLData().execute(url);
64         }
65     }
66 });
67 }
```

Рисунок 5 – Создание обработчика нажатия на кнопку



```
{
  "coord": {
    "lon": 132.95,
    "lat": 48.8
  },
  "weather": [
    {
      "id": 804,
      "main": "Clouds",
      "description": "пасмурно",
      "icon": "04d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": -17.87,
    "feels_like": -23.04,
    "temp_min": -17.87,
    "temp_max": -17.87,
    "pressure": 1013,
    "humidity": 77,
    "sea_level": 1013,
    "grnd_level": 1002
  },
  "visibility": 8616,
  "wind": {
    "speed": 1.89,
    "deg": 282,
    "gust": 3.75
  },
  "clouds": {
    "all": 100
  },
  "dt": 1641187401,
  "sys": {
    "country": "RU",
    "sunrise": 1641164475,
    "sunset": 1641194599,
    "timezone": 36000,
    "id": 2026643,
    "name": "Биробиджан",
    "cod": 200
  }
}
```

Рисунок 6 – Ссылка для получения информации о погоде

Теперь пропишем код для создания URL и HTTP подключения, считывания данных и закрытия соединения (рис. 7,8).

```
66
67     @SuppressWarnings("StaticFieldLeak")
68     private class GetURLData extends AsyncTask<String, String, String> {
69
70
71     protected void onPreExecute() {
72         super.onPreExecute();
73         result_info.setText("Загрузка...");
74     }
75
76
77     @Override
78     protected String doInBackground(String... strings) {
79         HttpURLConnection connection = null;
80         BufferedReader reader = null;
81
82         try {
83
84             URL url = new URL(strings[0]);
85             connection = (HttpURLConnection) url.openConnection();
86             connection.connect();
87
88
89             InputStream stream = connection.getInputStream();
90             reader = new BufferedReader(new InputStreamReader(stream));
91
92
93             StringBuilder buffer = new StringBuilder();
94             String line = "";
95
96
97             while((line = reader.readLine()) != null)
98                 buffer.append(line).append("\n");
99
100
```

Рисунок 7 – Создание URL и HTTP подключения и считывания данных

```
101         return buffer.toString();
102
103     } catch (MalformedURLException e) {
104         e.printStackTrace();
105     } catch (IOException e) {
106         e.printStackTrace();
107     } finally {
108
109         if(connection != null)
110             connection.disconnect();
111
112         try {
113             if (reader != null)
114                 reader.close();
115         } catch (IOException e) {
116             e.printStackTrace();
117         }
118     }
119
120     return null;
121 }
```

Рисунок 8 – Закрытие соединения

Так как приложение будет работать с интернетом, необходимо предоставить ему это право. Для этого в «AndroidManifest.xml» прописываем команду «<uses-permission android:name="android.permission.INTERNET"/>».

С помощью неё у приложения будет возможность использования интернета. Теперь необходимо обработать полученные данные. Конвертируем JSON формат и выводим данные в текстовом поле (рис. 9).

```
123
124     @SuppressWarnings("SetTextI18n")
125     @Override
126     protected void onPostExecute(String result) {
127         super.onPostExecute(result);
128
129
130         try {
131             JSONObject jsonObject = new JSONObject(result);
132             result_info.setText("Температура: " + jsonObject.getJSONObject("main").getString("temp"));
133
134         } catch (JSONException e) {
135             e.printStackTrace();
136         }
137     }
138 }
139
140 }
141 }
142 }
```

Рисунок 9 – Конвертация JSON формата и вывод данных в текстовое поле

Теперь тестируем приложение. Для этого в Android Studio созданное приложение запускается на реальном устройстве (рис.9).

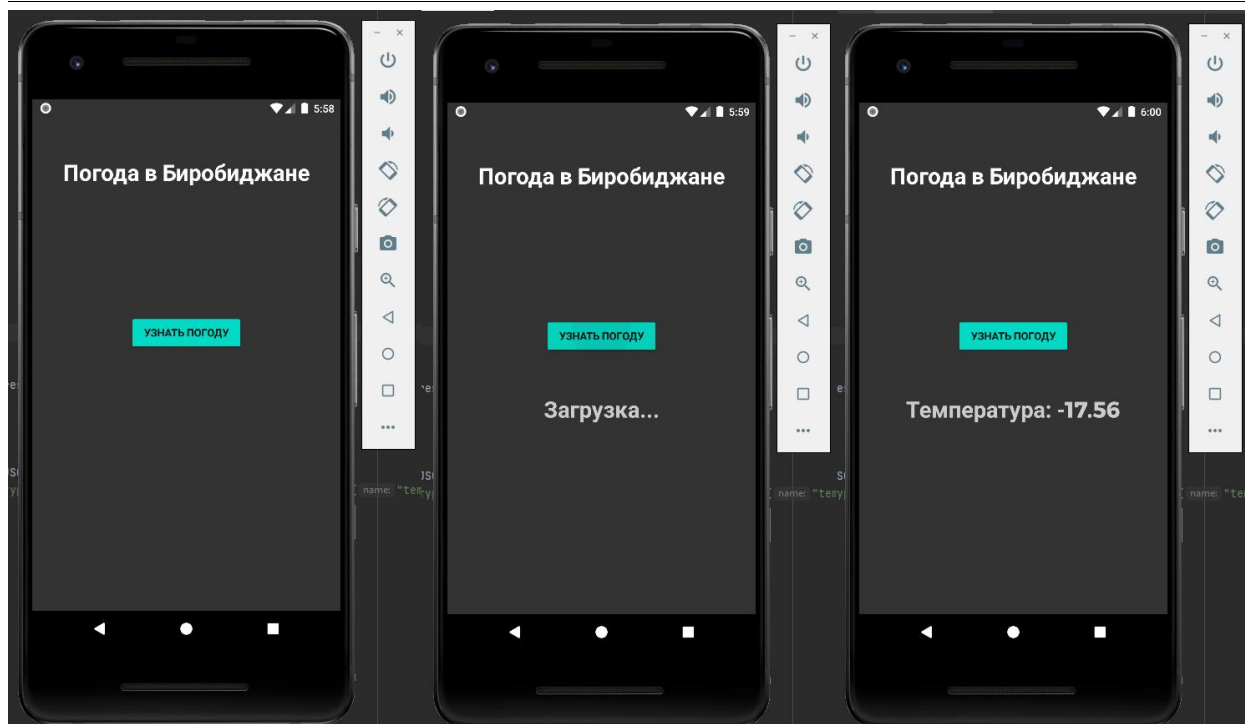


Рисунок 9 – Тестирование приложения

### Выводы

В данной работе было создано приложение, получающее данные о погоде с помощью API технологий в городе Биробиджан. В качестве источника данных о погоде использовался сервис «OpenWeatherMap».

### Библиографический список

1. Дворкина Н.Б., Намиот Д.Е. Использование OpenCellID API в мобильных сервисах. // Прикладная информатика. 2010. № 5 (29). С. 92-101.
2. Шмаков С.Э., Смекалин Н.Н., Щенникова Е.В. Создание WEB-приложения с использованием сторонних API. // XLIX Огарёвские чтения. Материалы научной конференции: в 3 частях. Саранск, 2021. С. 668-671.
3. Сергеев П.А. Использование JSON API в Android-приложении. // Актуальные направления научных исследований: перспективы развития. Сборник материалов IX Международной научно-практической конференции. Редколлегия: О.Н. Широков [и др.]. 2019. С. 50-53.
4. Васильев В.И. Создание элементарного браузера в Android studio. Форум молодых ученых. 2019. № 2 (30). С. 407-410.
5. Криворотов А.А., Марданова Т.М. Создание приложения для смартфона в среде разработки Android studio. // Наука и современность. Материалы Региональной научно-практической конференции. 2019. С. 64-66.
6. URL: <https://openweathermap.org/> (дата обращения 14.12.2021)