Примеры получения и разбора JSON-файла на Java Android

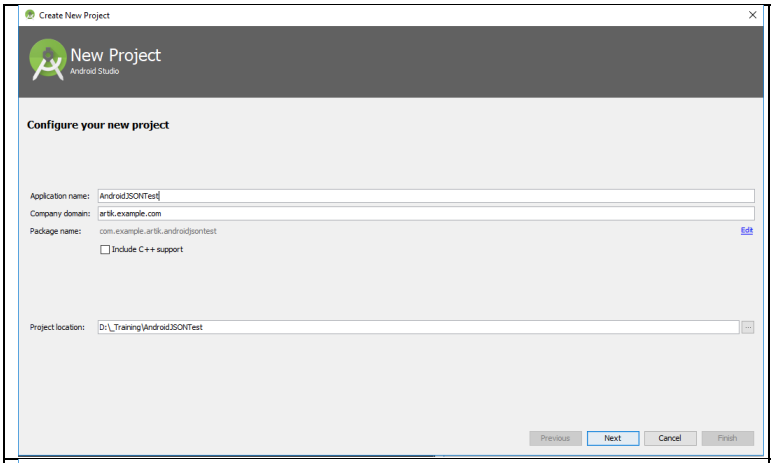
Пример доступа к файлу, который доступен в интернете.

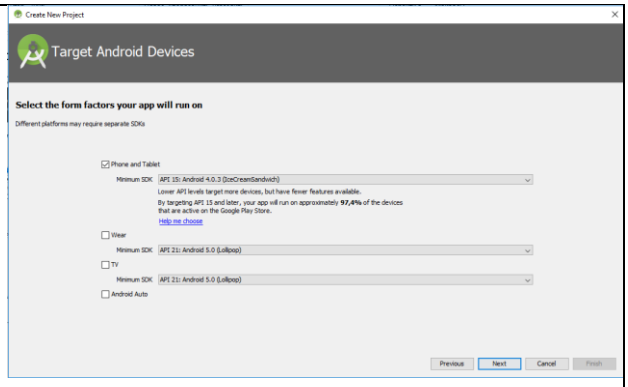
Перейдите по адресу:

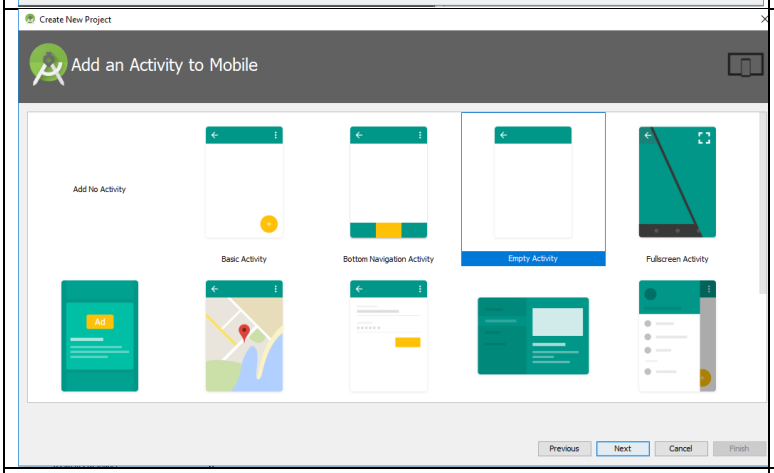
http://androiddocs.ru/api/friends.json
вы ответ получим json-файл

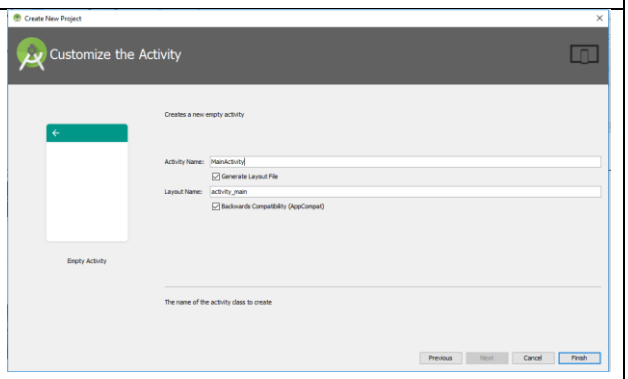

```
{"data":"dbfriends","friends":[{"id":"1","name":"Andrew","city":"Moscow","contacts":{"mobile":"+7000000","email":"andrew@androiddocs.ru","skype":"andrew"}},{"id":"2","name":"Ivan","city":"Kiev","contacts":{"mobile":"+380000000","email":"ivan@androiddocs.ru","skype":"ivan"}}]}
```


Рассмотрим, как получить его содержимое в проекте на Java Android. Для этого создадим простое Android-приложение с именем AndroidJSONTest:





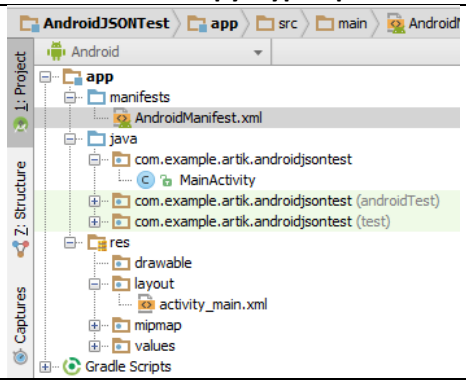




Файл манифеста выглядит так:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.artik.androidjsontest">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="AndroidJSONTest"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Структура проекта:



Нашему приложению даны права на использование интернета:

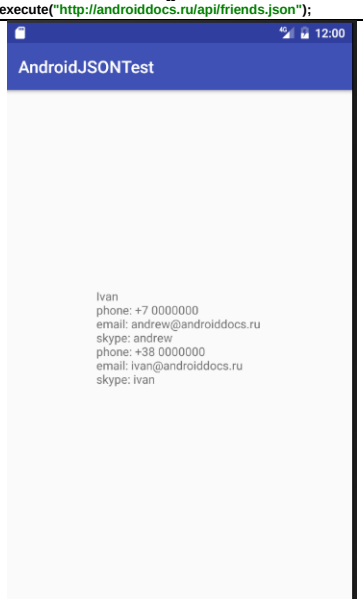
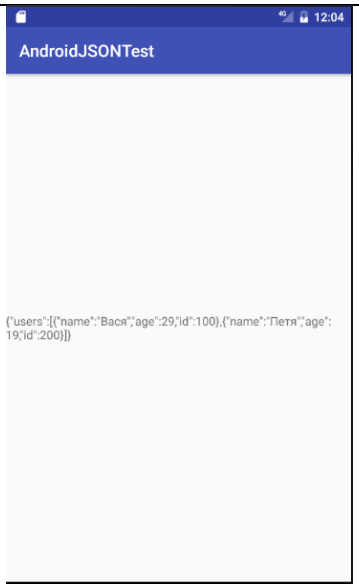
```
<uses-permission android:name="android.permission.INTERNET" />
```

Приведем код файла MainActivity.java

MainActivity

```
package com.example.artik.androidjsontest;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

import android.os.AsyncTask;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.HttpURLConnection;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

public class MainActivity extends AppCompatActivity {
    TextView tvOut;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // найдем View-элементы
        tvOut = (TextView) findViewById(R.id.tvOut);
        new JSONTask().execute("http://androiddocs.ru/api/friends.json");
        //new JSONFormLocalServiceTask().execute("http://localhost:9081/WebServiceRESTfullServer/webresources/generic/users");
        //new JSONFormLocalServiceTask().execute("http://10.0.2.2:9081/WebServiceRESTfullServer/webresources/generic/users");
        //new JSONParserTask().execute("http://10.0.2.2:9081/WebServiceRESTfullServer/webresources/generic/users");
    }
    //
    public class JSONTask extends AsyncTask<String,String,String>{...}
    //
    public class JSONFormLocalServiceTask extends AsyncTask<String,String,String>{...}
    //
    public class JSONParserTask extends AsyncTask<String,String,String>{...}
    //
}
```

Результат работы программы :

new **JSONTask().** execute("http://androiddocs.ru/api/friends.json");	new **JSONFormLocalServiceTask**() execute("http://10.0.2.2:9081/WebServiceRESTfullServer/webresources/generic/users");	new **JSONParserTask**() execute("http://10.0.2.2:9081/WebServiceRESTfullServer/webresources/generic/users");
		

Приведем тексты классов:

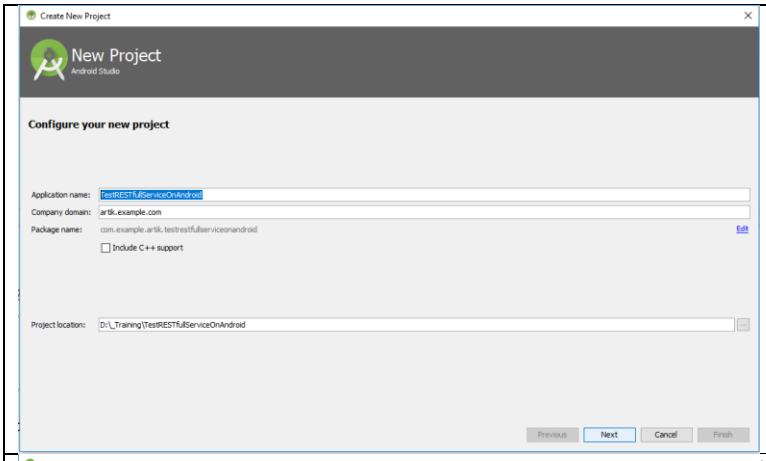
```
public class JSONTask extends AsyncTask<String,String,String>{
    @Override
    protected String doInBackground(String... params) {
        HttpURLConnection connection = null;
        BufferedReader reader = null;
        int statusCode = 0;

        try {
            URL url = new URL(params[0]);
            connection = (HttpURLConnection) url.openConnection();
            connection.setRequestProperty("Content-Type", "application/json");
            connection.setRequestMethod("GET");
            connection.connect(); //connect to server

            statusCode = connection.getResponseCode();


            if (statusCode ==  200) {
                System.out.println("Server responded with code: " + statusCode);

                InputStream stream = connection.getInputStream();

                reader = new BufferedReader(new InputStreamReader(stream));

                StringBuffer buffer = new StringBuffer();
                String line = "";

                while ((line = reader.readLine()) != null) {
                    buffer.append(line);
                }
                String finalJSON = buffer.toString();
                StringBuffer finalBufferdData = new StringBuffer();
                JSONObject dataJsonObj = new JSONObject(finalJSON);
                JSONArray friends = dataJsonObj.getJSONArray("friends");
                // 1. достаем инфо о втором друге - индекс 1
                JSONObject secondFriend = friends.getJSONObject(1);
                String secondName = secondFriend.getString("name");
                finalBufferdData.append(secondName + "\n");
                // 2. перебираем и выводим контакты каждого друга
                for (int i = 0; i < friends.length(); i++) {
                    JSONObject friend = friends.getJSONObject(i);
                    JSONObject contacts = friend.getJSONObject("contacts");
                    String phone = contacts.getString("mobile");
                    String email = contacts.getString("email");
                    String skype = contacts.getString("skype");
                    finalBufferdData.append( "phone: " + phone+ "\n");
                    finalBufferdData.append( "email: " + email+ "\n");
                    finalBufferdData.append( "skype: " + skype+ "\n");
                }
                return finalBufferdData.toString(); //pases result to POstExecute
            }
        }catch(java.net.MalformedURLException e){
            e.printStackTrace();
        }catch(IOException e){
            e.printStackTrace();
        }catch(JSONException e){
            e.printStackTrace();
        }
        finally {
            if(connection !=null) {
                connection.disconnect();
            }
            try {
                if(reader !=null) {
                    reader.close();
                }
            }catch(IOException e){
                e.printStackTrace();
            }
        }
        return null;
    }
    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
        tvOut.setText(result);
    }
}
```

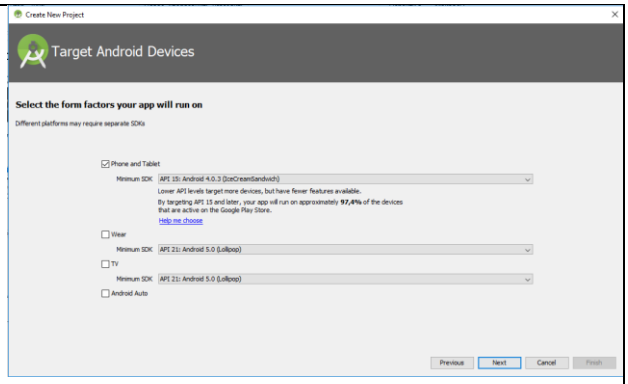
```
public class JSONFormLocalServiceTask extends AsyncTask<String,String,String>{
    @Override
    protected String doInBackground(String... params) {
        HttpURLConnection connection = null;
        BufferedReader reader = null;
        int statusCode = 0;

        try {
            URL url = new URL(params[0]);
            connection = (HttpURLConnection) url.openConnection();
            connection.setRequestProperty("Content-Type", "application/json");
            connection.setRequestMethod("GET");
            connection.connect(); //connect to server

            statusCode = connection.getResponseCode();


            if (statusCode ==  200) {
                System.out.println("Server responded with code: " + statusCode);

                InputStream stream = connection.getInputStream();

                reader = new BufferedReader(new InputStreamReader(stream));

                StringBuffer buffer = new StringBuffer();
                String line = "";

                while ((line = reader.readLine()) != null) {
                    buffer.append(line);
                }
                return buffer.toString();

            }
        }catch(java.net.MalformedURLException e){
            e.printStackTrace();
        }catch(IOException e){
            e.printStackTrace();
        }
        finally {
            if(connection !=null) {
                connection.disconnect();
            }
            try {
                if(reader !=null) {
                    reader.close();
                }
            }catch(IOException e){
                e.printStackTrace();
            }
        }
        return null;
    }

    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
        tvOut.setText(result);
    }
}
```

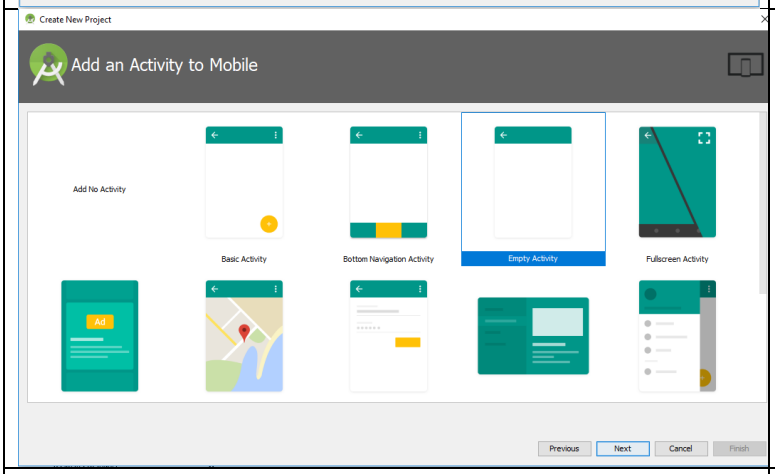
```
public class JSONParserTask extends AsyncTask<String,String,String>{
    @Override
    protected String doInBackground(String... params) {
        HttpURLConnection connection = null;
        BufferedReader reader = null;
        int statusCode = 0;

        try {
            URL url = new URL(params[0]);
            connection = (HttpURLConnection) url.openConnection();
            connection.setRequestProperty("Content-Type", "application/json");
            connection.setRequestMethod("GET");
            connection.connect(); //connect to server
            statusCode = connection.getResponseCode();

            if (statusCode ==  200) {
                System.out.println("Server responded with code: " + statusCode);

                InputStream stream = connection.getInputStream();
                reader = new BufferedReader(new InputStreamReader(stream));
                StringBuffer buffer = new StringBuffer();
                String line = "";

                while ((line = reader.readLine()) != null) {
                    buffer.append(line);
                }
                String finalJSON = buffer.toString();
                StringBuffer finalBufferdData = new StringBuffer();
                JSONObject dataJsonObj = new JSONObject(finalJSON);
                JSONArray users = dataJsonObj.getJSONArray("users");
                for(int i = 0; i < users.length();i++)
                {
                    JSONObject secondFriend = users.getJSONObject(i);
                    String name = secondFriend.getString("name");
                    String age = secondFriend.getString("age");
                    String id = secondFriend.getString("id");
                    finalBufferdData.append( "name: " + name+ "\n");
                    finalBufferdData.append( "age: " + age+ "\n");
                    finalBufferdData.append( "id: " + id+ "\n");
                }
                return finalBufferdData.toString(); //pases result to POstExecute
            }
        }catch(java.net.MalformedURLException e){
            e.printStackTrace();
        }catch(IOException e){
            e.printStackTrace();
        }
        catch(JSONException e){
            e.printStackTrace();
        }
        finally {
            if(connection !=null) {
                connection.disconnect();
            }
            try {
                if(reader !=null) {
                    reader.close();
                }
            }catch(IOException e){
                e.printStackTrace();
            }
        }
        return null;
    }
    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
        tvOut.setText(result);
    }
}
```

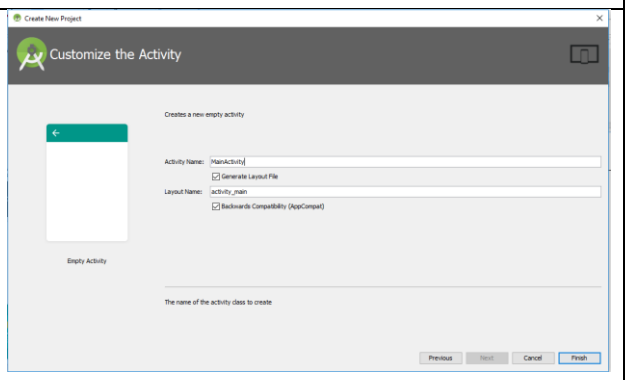
Пример 4 клиент к RESTfull сервису на Java Android.

Создадим проект TestRESTfullServiceOnAndroid в Android Studio

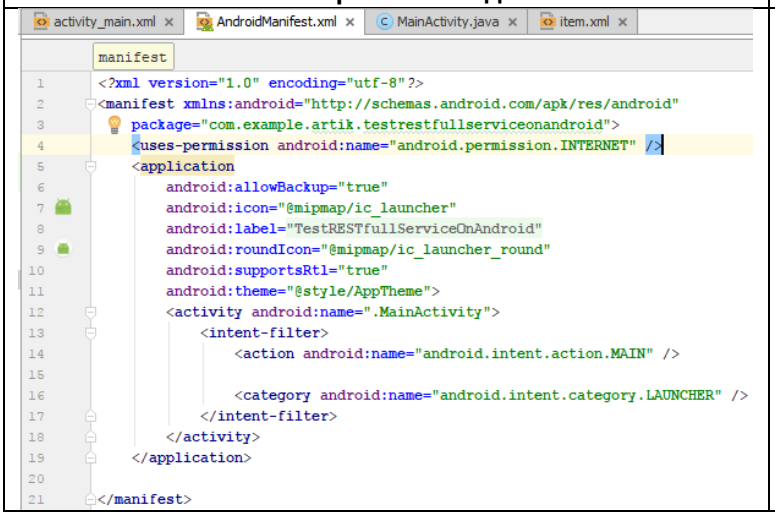




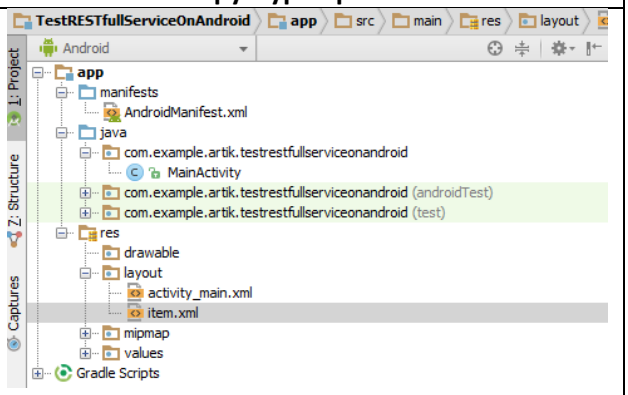




Файл манифеста выглядит так:



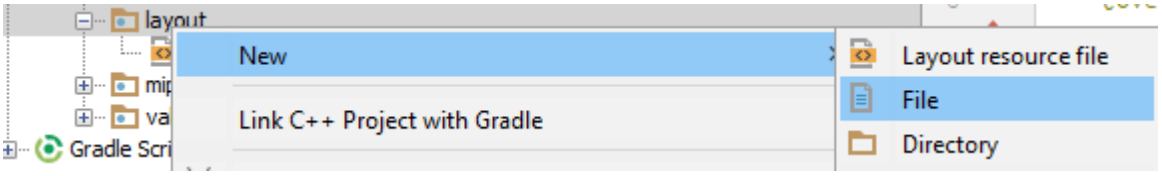
Структура проекта:



Нашему приложению даны права на использование интернета:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Создадим файл item.xml – который будет описывать внешний вид списка. Для этого вызовем контекстное меню на папке layout, выберем пункт New-File и введем имя с указанием расширения (item.xml)



Далее приведем разметки наших файлов:

item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="horizontal">
  <LinearLayout
    android:id="@+id/linearLayout1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:orientation="horizontal">
    <TextView
      android:id="@+id/tvName"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="right"
      android:layout_marginRight="20dp"
      android:text="TextView">
    </TextView>
    <TextView
      android:id="@+id/tvAge"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="right"
      android:layout_marginRight="20dp"
      android:text="TextView">
    </TextView>
    <TextView
      android:id="@+id/tvId"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="right"
      android:layout_marginRight="20dp"
      android:text="TextView">
    </TextView>
  </LinearLayout>
</LinearLayout>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
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:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context="com.example.artik.testrestfullserviceonandroid.MainActivity">

  <LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
      <ListView
       android:id="@+id/lvSimple"
       android:layout_width="match_parent"
       android:layout_height="wrap_content">
    </ListView>
  </LinearLayout>

</android.support.constraint.ConstraintLayout>
```

Содержимое файла **MainActivity.java**

```
package com.example.artik.testrestfullserviceonandroid;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
//
import android.os.AsyncTask;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.net.HttpURLConnection;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import android.widget.ListView;
import android.widget.SimpleAdapter;

//
public class MainActivity extends AppCompatActivity {
    // имена атрибутов для Map
    final String ATTRIBUTE_NAME_NAME = "name";
    final String ATTRIBUTE_NAME_AGE = "age";
    final String ATTRIBUTE_NAME_ID = "id";

    ListView lvSimple;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        new JSONParserTask(this).execute("http://10.0.2.2:9081/WebServiceRESTfullServer/webresources/generic/users");
    }
    public class JSONParserTask extends AsyncTask<String,String,String>{...}

}
```

Приведём программный код класса `JSONParserTask:`

```
public class JSONParserTask extends AsyncTask<String,String,String>{
    MainActivity mainActivity;
    public JSONParserTask( MainActivity mainActivity)
    {
        this.mainActivity = mainActivity;
    }
    @Override
    protected String doInBackground(String... params) {
        HttpURLConnection connection = null;
        BufferedReader reader = null;
        int statusCode = 0;

        try {
            URL url = new URL(params[0]);
            connection = (HttpURLConnection) url.openConnection();
            connection.setRequestProperty("Content-Type", "application/json");
            connection.setRequestMethod("GET");
            connection.connect(); //connect to server

            statusCode = connection.getResponseCode();


            if (statusCode ==  200) {
                System.out.println("Server responded with code: " + statusCode);

                InputStream stream = connection.getInputStream();

                reader = new BufferedReader(new InputStreamReader(stream));

                StringBuffer buffer = new StringBuffer();
                String line = "";

                while ((line = reader.readLine()) != null) {
                    buffer.append(line);
                }
                return buffer.toString();//pases result to POstExecut

            }

        }catch(java.net.MalformedURLException e){
            e.printStackTrace();
        }catch(IOException e){
            e.printStackTrace();
        }

        finally {
            if(connection !=null) {
                connection.disconnect();
            }
            try {
                if(reader !=null) {
                    reader.close();
                }
            }catch(IOException e){
                e.printStackTrace();
            }
        }
        return null;
    }

    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
        // массив данных
        String[] names = null;
        String[] ages = null;
        String[] ids = null;
        //
```

```
        try {
            String finalJSON = result;
            StringBuffer finalBufferdData = new StringBuffer();
            JSONObject dataJsonObj = new JSONObject(finalJSON);
            JSONArray users = dataJsonObj.getJSONArray("users");
            names = new String[users.length()];
            ages = new String[users.length()];
            ids = new String[users.length()];
            for (int i = 0; i < users.length(); i++) {
                JSONObject secondFriend = users.getJSONObject(i);
                String name = secondFriend.getString("name");
                String age = secondFriend.getString("age");
                String id = secondFriend.getString("id");
                names[i] = name;
                ages[i]=age;
                ids[i]=id;
            }
        }
        catch(JSONException e){
            e.printStackTrace();
        }
        //
        // массивы данных


        // упаковываем данные в понятную для адаптера структуру
        ArrayList<Map<String, Object>> data = new ArrayList<Map<String, Object>>(names.length);
        Map<String, Object> m;
        for (int i = 0; i < names.length; i++) {
            m = new HashMap<String, Object>();
            m.put(ATTRIBUTE_NAME_NAME, names[i]);
            m.put(ATTRIBUTE_NAME_AGE, ages[i]);
            m.put(ATTRIBUTE_NAME_ID, ids[i]);
            data.add(m);
        }
        // массив имен атрибутов, из которых будут читаться данные
        String[] from = { ATTRIBUTE_NAME_NAME, ATTRIBUTE_NAME_AGE,ATTRIBUTE_NAME_ID };
        // массив ID View-компонентов, в которые будут вставлять данные
        int[] to = { R.id.tvName, R.id.tvAge, R.id.tvId };
        // создаем адаптер
        SimpleAdapter sAdapter = new SimpleAdapter(this.mainActivity, data, R.layout.item,from,
to);

        // определяем список и присваиваем ему адаптер
        lvSimple = (ListView) findViewById(R.id.lvSimple);
        lvSimple.setAdapter(sAdapter);
        //
    }
}
```

Результат работы приложения

Команды из нашего сервиса	Приложение в эмуляторе

GenericResource.java | ResponseList.java | TestRESTfullService.java | WebS

Источник | История

```
43          return "Привет from getXml()";
44      }
45      @GET
46      @Produces(MediaType.APPLICATION_JSON)
47      //@Path("users")
48      @Path("/users")
49      public ResponseList getUsers() {
50          User customer1 = new User();
51          customer1.setId(100);
52          customer1.setName("Вася");
53          customer1.setAge(29);
54          //
55          User customer2 = new User();
56          customer2.setId(200);
57          customer2.setName("Петя");
58          customer2.setAge(19);
59          //
60          ResponseList responseList = new ResponseList();
61          responseList.getUsers().add(customer1);
62          responseList.getUsers().add(customer2);
63          return responseList;
64      }
```

WebServiceRESTfullServer > generic > users

Ресурсы: generic/users
(http://localhost:9081/WebServiceRESTfullServer/webresources/generic/users)

Выберите метод для тестирования: GET(application/json) Тестировать

Ответ:

Представление в табличной форме | Представление необработанных данных

{"users":[{"name":"Вася","age":29,"id":100},{"name":"Петя","age":19,"id":200}]}