

BÀI TẬP BUỔI 12 (CHƯƠNG 6)

Học phần: Phát triển ứng dụng trên thiết bị di động_TN216

Sinh viên thực hiện: **Họ và tên – MSSV**

Lưu ý

- Sử dụng file mẫu trên Hệ thống học trực tuyến của Trường để làm bài.
- Hiểu rõ mục tiêu, ý nghĩa, và mã lệnh của các bài tập.
- Sử dụng Snipping Tool để chụp hình kết quả thực hiện các yêu cầu.

Mục tiêu

- Thiết kế giao diện với các Layout và điều khiển thông dụng. Sử dụng danh sách tùy biến.
- Lập trình xử lý sự kiện.
- Trao đổi dữ liệu với Web Server thông qua Web Services.

Kết quả thực hiện

Tạo ứng dụng TraoDoiDuLieuVoiWebServer cho phép trao đổi dữ liệu với WebServer

- **Bước 1: tạo dự án mới và các đối tượng hỗ trợ**
 - + Tạo dự án mới, chọn mẫu Empty Views Activity, đặt tên cho dự án là TraoDoiDuLieuVoiWebServer, Package Name là thud.traodoidulieuvoiwebserver.
 - + Trong res/values, thêm các Styles cần thiết: TextStyle và ButtonStyle.
 - + Trong AndroidManifest, khai báo quyền sử dụng Internet và quyền truy xuất trạng thái kết nối mạng.

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

- **Bước 2: thêm lớp Publics khai báo các biến và phương thức sử dụng chung**
 - + Khai báo biến chuỗi lưu địa chỉ Web Server gọi và nhận dữ liệu,
 - + Thêm phương thức kiểm tra trạng thái kết nối Internet,
 - + Thêm phương thức đổi từ InputStream (dữ liệu nhận từ Server) thành String.

```
public class Publics {

    public static String URLNHANDULIEU =
        "https://api.imgflip.com/get_memes";
    public static String URLGOIDULIEU =
        "http://hmkcode.appspot.com/jsonservlet";

    public static boolean HasInternet(Context context) {
        ConnectivityManager conn = (ConnectivityManager)
            context.getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo netInfo = conn.getActiveNetworkInfo();
        return ((netInfo != null) && (netInfo.isConnected()));
    }

    public static String StreamToString(InputStream tream){
        StringBuilder builder = new StringBuilder();
        BufferedReader reader = new BufferedReader(
            new InputStreamReader(tream));
        String temp;
        try {
            while ((temp = reader.readLine()) != null) {
                builder.append(temp).append('\n');
            }
            tream.close();
        } catch (IOException e) {
        }
        return builder.toString();
    }
}
```

- **Bước 3: tạo Activity thực hiện nhận dữ liệu từ Web Server, thiết kế giao diện hiển thị dữ liệu nhận được trong ListView**
 - + Tạo Activity mới với tên NhanDuLieuTuServer, tập tin giao diện nhandulieu
 - + Thiết kế giao diện: tập tin giao diện res/layout/nhandulieu.xml như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".NhanDuLieuTuServer">
```

```
<ListView
    android:id="@+id/lst_dulieu"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1" />

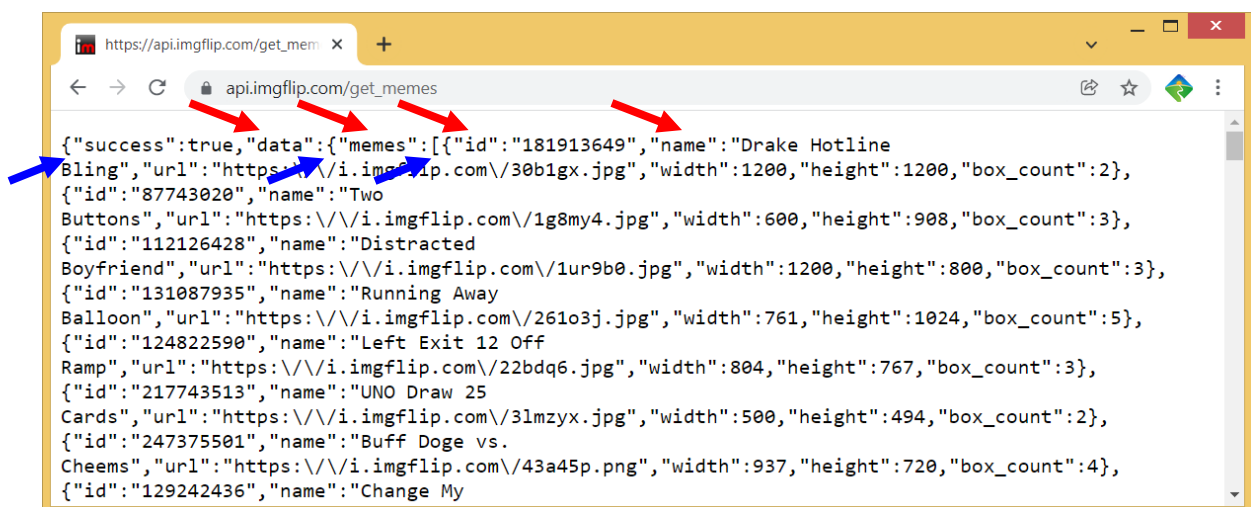
<LinearLayout
    android:gravity="center"
    android:layout_height="wrap_content"
    android:layout_marginTop="1dp"
    android:layout_width="match_parent"
    android:orientation="horizontal">

    <Button
        style="@style/ButtonStyle"
        android:id="@+id/btn_nhan"
        android:onClick="ThucHienNhanDuLieu"
        android:text="Nhận dữ liệu" />

    <Button
        style="@style/ButtonStyle"
        android:id="@+id/btn_xoa"
        android:onClick="XoaDuLieu"
        android:text="Xóa dữ liệu" />

</LinearLayout>
</LinearLayout>
```

– Bước 2: phân tích dữ liệu nhận được từ WebServer (Hình 1)



Hình 1. Dữ liệu nhận được từ WebServer (https://api.imgflip.com/get_memes)

- + Dữ liệu nhận được là 1 đối tượng JSON (JSONObject): có dấu { ở đầu
- + Tiếp theo nhận đối tượng JSON có tên data: **“data”:{**
- + Tiếp theo nhận mảng đối tượng JSON (JSONArray) có tên memes: **“memes”:[**
- + Tiếp theo nhận lần lượt từng đối tượng JSON trong mảng, mỗi đối tượng truy xuất 2 nội dung là id và name:

```
{ "id":"181913649", "name":"Drake Hotline Bling", ..., "box_count":2 },  
{ "id":"87743020", "name":"Two Buttons", ..., "box_count":3}, ...
```

– **Bước 4: điều chỉnh mã lệnh Java cho NhanDuLieuTuServer**

- + Khai báo các biến hỗ trợ,
- + Trong phương thức onCreate: nhận và khai báo danh sách hiển thị dữ liệu,
- + Thêm lớp cục bộ NhanDuLieu kế thừa AsyncTask, thực hiện:
 - ✓ Nhận dữ liệu từ Server,
 - ✓ Thêm dữ liệu vào danh sách gồm id và name,
 - ✓ Thông báo kết quả, cập nhật Adapter cho phép hiển thị dữ liệu trong danh sách.
- + Thêm phương thức ThucHienNhanDuLieu xử lý sự kiện cho nút lệnh “Nhận dữ liệu”: kiểm tra trạng thái Internet và thực thi lớp NhanDuLieu.
- + Thêm phương thức XoaDuLieu xử lý sự kiện cho nút lệnh “Xóa dữ liệu”: thực hiện xóa dữ liệu trong danh sách.
- + Điều chỉnh trong AndroidManifest: gán NhanDuLieuTuServer là Activity chính,
- + Thực thi kiểm tra kết quả.

```
public class NhanDuLieuTuServer extends AppCompatActivity {  
  
    List<String> listDuLieu;  
    ListView listviewDuLieu;  
    ArrayAdapter<String> adapter;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.nhandulieu);  
  
        listviewDuLieu = findViewById(R.id.lst_dulieu);  
        listDuLieu = new ArrayList<String>();  
        adapter = new ArrayAdapter<String>(this,  
            android.R.layout.simple_list_item_1, listDuLieu);  
        listviewDuLieu.setAdapter(adapter);  
    }  
}
```

```
public void ThucHienNhanDuLieu(View view){
    if (Publics.HasInternet(this)) {
        NhanDuLieu nhanDuLieu = new NhanDuLieu();
        nhanDuLieu.execute();
    } else {
        Toast.makeText(this, "Lỗi kết nối Internet!",
            Toast.LENGTH_LONG).show();
    }
}

private class NhanDuLieu extends AsyncTask<Void, Void, Integer>{
    @Override
    protected Integer doInBackground(Void... voids){
        int ketqua = 0;
        try {
            URL url = new URL(Publics.URLNHANDULIEU);
            HttpURLConnection conn =
                (HttpURLConnection) url.openConnection();
            conn.setRequestMethod("GET");
            conn.setRequestProperty("Content-Type",
                "application/json");
            conn.setDoOutput(false);
            conn.setDoInput(true);
            conn.setReadTimeout(10000);
            conn.setConnectTimeout(15000);
            conn.connect();
            InputStream in = new BufferedInputStream(
                conn.getInputStream());
            String strDuLieu = Publics.StreamToString(in);
            int responseCode = conn.getResponseCode();
            if (responseCode == HttpURLConnection.HTTP_OK){
                JSONObject jsonObj = new JSONObject(strDuLieu);
                JSONObject objData = jsonObj.getJSONObject("data");
                JSONArray arrDuLieu = objData.getJSONArray("memes");
                for (int i = 0; i < arrDuLieu.length(); i++){
                    JSONObject dulieu = arrDuLieu.getJSONObject(i);
                    String id = dulieu.getString("id");
                    String hoten = dulieu.getString("name");
                    listDuLieu.add(id + " - " + hoten);
                    ketqua++;
                }
            }
            in.close();
        }
    }
}
```

```
        conn.disconnect();
    } catch (Exception e) {
    }
    return ketqua;
}

@Override
protected void onPostExecute(Integer integer) {
    super.onPostExecute(integer);
    if (integer > 0) {
        setResult(RESULT_OK);
        adapter.notifyDataSetChanged();
        Toast.makeText(NhanDuLieuTuServer.this,
            "Nhận được " + integer + " thông tin mới!",
            Toast.LENGTH_LONG).show();
    } else {
        setResult(RESULT_CANCELED);
        Toast.makeText(getApplicationContext(),
            "Không nhận được thông tin mới!",
            Toast.LENGTH_SHORT).show();
    }
}

public void XoaDuLieu(View view){
    listDuLieu.clear();
    adapter.notifyDataSetChanged();
}
}
```

Trao đổi dữ liệu với Web Server	Trao đổi dữ liệu với Web Server
	181913649 - Drake Hotline Bling
	87743020 - Two Buttons
	112126428 - Distracted Boyfriend
	131087935 - Running Away Balloon
	124822590 - Left Exit 12 Off Ramp
	217743513 - UNO Draw 25 Cards
	247375501 - Buff Doge vs. Cheems
	129242436 - Change My Mind
	131940431 - Gru's Plan
	222403160 - Bernie I Am Once Again Asking For Your Support
	438680 - Batman Slapping Robin
NHẬN DỮ LIỆU XÓA DỮ LIỆU	NHẬN DỮ LIỆU XÓA DỮ LIỆU

Hình 2. Kiểm thử Nhận dữ liệu từ Web Server

- **Bước 5: tạo Activity thực hiện gửi dữ liệu đến Web Server, thiết kế giao diện**
 - + Tạo Activity mới với tên GoiDuLieuDenServer, tập tin giao diện goidulieu
 - + Thiết kế giao diện: tập tin giao diện res/layout/goidulieu.xml như bên dưới
 - + Điều chỉnh trong AndroidManifest: gán GoiDuLieuDenServer là Activity chính,
 - + Thực thi kiểm tra kết quả.

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".GoiDuLieuDenServer">

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/layout_hoten"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="7dp">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/edt_hoten"
            style="@style/TextStyle"
            android:hint="Nhập họ tên" />
    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/layout_quocgia"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="7dp">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/edt_quocgia"
            style="@style/TextStyle"
            android:hint="Nhập quốc gia" />
    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/layout_noidung"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="7dp">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/edt_noidung"
            style="@style/TextStyle"
            android:hint="Nhập nội dung" />
    </com.google.android.material.textfield.TextInputLayout>

    <LinearLayout
        android:gravity="center"
        android:layout_height="wrap_content"
        android:layout_marginTop="1dp"
        android:layout_width="match_parent"
        android:orientation="horizontal">

        <Button
            style="@style/ButtonStyle"
            android:id="@+id/btn_goi"
            android:onClick="ThucHienGoiDuLieu"
```



```
        android:text="Gửi dữ liệu" />

        <Button
            style="@style/ButtonStyle"
            android:id="@+id/btn_xoa"
            android:onClick="XoaNoiDung"
            android:text="Xóa nội dung" />
    </LinearLayout>
</LinearLayout>
```



Hình 3. Gửi dữ liệu đến Web Server khi chưa có mã lệnh Java

- **Bước 6: điều chỉnh mã lệnh Java cho GoiDuLieuDenServer**
 - + Khai báo các biến hỗ trợ,
 - + Trong phương thức onCreate: nhận các đối tượng,
 - + Thêm lớp cục bộ GoiDuLieu kế thừa AsyncTask thực hiện:
 - ✓ Kiểm tra dữ liệu nhập trong các EditText,
 - ✓ Gửi dữ liệu gồm 3 thông tin: name, country, twitter đến Web Server,
 - ✓ Thông báo kết quả thực hiện.
 - + Thêm phương thức ThucHienGoiDuLieu xử lý sự kiện cho nút lệnh “Gửi dữ liệu”: kiểm tra trạng thái Internet và thực thi lớp GoiDuLieu.
 - + Thêm phương thức XoaNoiDung xử lý sự kiện cho nút lệnh “Xóa nội dung”: xóa nội dung trong các EditText.

```
public class GoiDuLieuDenServer extends AppCompatActivity {

    TextInputEditText edtHoTen, edtQuocGia, edtNoiDung;
    TextInputLayout layoutHoTen, layoutQuocGia, layoutNoiDung;
    String strHoTen, strQuocGia, strNoiDung;
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.goidulieu);
    edtHoTen = findViewById(R.id.edt_hoten);
    layoutHoTen = findViewById(R.id.layout_hoten);
    edtQuocGia = findViewById(R.id.edt_quocgia);
    layoutQuocGia = findViewById(R.id.layout_quocgia);
    edtNoiDung = findViewById(R.id.edt_noidung);
    layoutNoiDung = findViewById(R.id.layout_noidung);
}

public void ThucHienGoiDuLieu(View view) {
    strHoTen = edtHoTen.getText().toString().trim();
    if(strHoTen.length() == 0) {
        layoutHoTen.setError("Lỗi chưa nhập họ tên!");
        edtHoTen.requestFocus();
        return;
    }
    else
        layoutHoTen.setError(null);
    strQuocGia = edtQuocGia.getText().toString().trim();
    if(strQuocGia.length() == 0) {
        layoutQuocGia.setError("Lỗi chưa nhập quốc gia!");
        edtQuocGia.requestFocus();
        return;
    }
    else
        layoutQuocGia.setError(null);
    strNoiDung = edtNoiDung.getText().toString().trim();
    if(strNoiDung.length() == 0) {
        layoutNoiDung.setError("Lỗi chưa nhập nội dung!");
        edtNoiDung.requestFocus();
        return;
    }
    else
        layoutNoiDung.setError(null);

    if (Publics.HasInternet(this)) {
        GoiDuLieu goiDuLieu = new GoiDuLieu();
        goiDuLieu.execute();
    } else {
        Toast.makeText(this, "Lỗi kết nối Internet!",
            Toast.LENGTH_LONG).show();
    }
}
}

```

```

private class GoiDuLieu extends
    AsyncTask<Void, Void, String> {
    @Override
    protected String doInBackground(Void... arg0) {
        String ketqua = "";
        try {
            URL url = new URL(Publics.URLGOIDULIEU);
            HttpURLConnection conn = (HttpURLConnection)
                url.openConnection();
            conn.setRequestMethod("POST");
            conn.setRequestProperty("Content-Type",
                "application/json");
            conn.setDoOutput(true);
            conn.setDoInput(false);
            conn.setReadTimeout(10000);
            conn.setConnectTimeout(15000);
            conn.connect();

            JSONObject dulieu = new JSONObject();
            dulieu.put("name", strHoTen);
            dulieu.put("country", strQuocGia);
            dulieu.put("twitter", strNoiDung);

            OutputStream os = conn.getOutputStream();
            OutputStreamWriter out = new OutputStreamWriter(os,
                "UTF-8");
            BufferedWriter writer = new BufferedWriter(out);
            writer.write(dulieu.toString());
            writer.flush();
            ketqua = conn.getResponseMessage();
            writer.close();
            out.close();
            os.close();
            conn.disconnect();
        } catch (Exception e) {
        }
        return ketqua;
    }

    @Override
    protected void onPostExecute(String s) {
        super.onPostExecute(s);
        if (s.equalsIgnoreCase("OK")) {
            setResult(RESULT_OK);
        }
    }
}

```

```
        Toast.makeText(GoiDuLieuDenServer.this,
                        "Gửi dữ liệu thành công!",
                        Toast.LENGTH_LONG).show();
    } else {
        setResult(RESULT_CANCELED);
        Toast.makeText(GoiDuLieuDenServer.this,
                        "Gửi dữ liệu không thành công!",
                        Toast.LENGTH_SHORT).show();
    }
}

}

}

public void XoaNoiDung(View view){
    edtHoTen.setText("");
    edtQuocGia.setText("");
    edtNoiDung.setText("");
    edtHoTen.requestFocus();
}
}
```

– **Bước 7: điều chỉnh cấu hình hỗ trợ thực hiện gửi dữ liệu**

- + Tạo thêm tập tin xml cấu hình bảo mật mạng: tại thư mục res, tạo tập tin tài nguyên network-security-config.xml lưu trong thư mục xml như sau:
 - ✓ R_Click lên thư mục res, chọn New, chọn Android resource file,
 - ✓ Đặt tên tập tin tài nguyên (File name) là network-security-config,
 - ✓ Chọn kiểu tài nguyên (Resource type) là XML,
 - ✓ Đặt tên thư mục lưu tập tin tài nguyên (Directory name) là xml.
- + Khai báo trong AndroidManifest.xml: sử dụng file cấu hình bảo mật mạng trong thẻ Application,
- + Thực thi kiểm tra kết quả.

Tập tin network-security-config:

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <base-config cleartextTrafficPermitted="true" />
    <domain-config cleartextTrafficPermitted="true">
        <domain includeSubdomains = "true">
            http://hmkcode.appspot.com</domain>
        </domain>
    </domain-config>
</network-security-config>
```

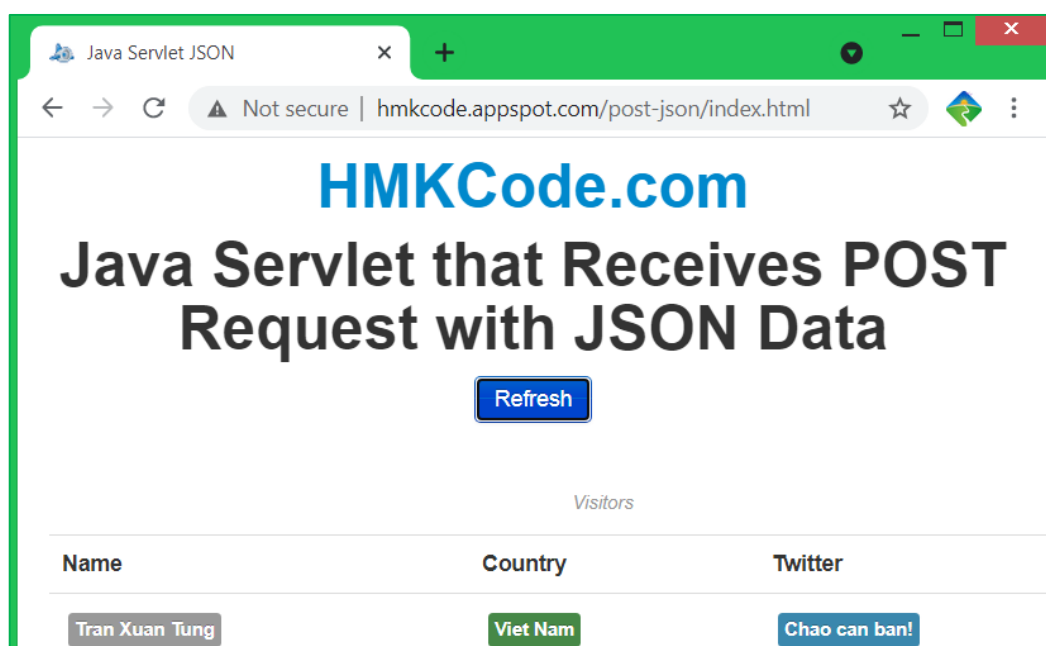
Khai báo trong AndroidManifest.xml:

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/traodoidulieu"
    android:label="Trao đổi dữ liệu với Web Server"
    android:roundIcon="@mipmap/traodoidulieu"
    android:supportsRtl="true"
    android:theme="@style/Theme.TraoDoiDuLieuVoiWebServer"
    android:networkSecurityConfig="@xml/network_security_config" >
    ...
    <activity android:name=".GoiDuLieuDenServer"
        android:label="Gửi dữ liệu đến Web Server">
        <intent-filter>
            <action android:name = "android.intent.action.MAIN"/>

            <category android:name =
                "android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
```

– **Bước 8: thực thi và kiểm tra kết quả**

- + Thực thi gửi dữ liệu đến Web Server,
- + Kiểm tra kết quả: URL là <http://hmckode.appspot.com/post-json/index.html>, dữ liệu đã gửi hiển thị như Hình 4.



Hình 4. Dữ liệu đã gửi

Lưu ý: hiện tại Web Server đang lỗi nên chức năng gửi dữ liệu đến Web Server thực hiện không thành công (hình 4 là kết quả thực thi năm 2021).

– **Bước 9: điều chỉnh giao diện MainActivity làm thực đơn cho ứng dụng**

- + Thiết kế giao diện: tập tin giao diện res/layout/activity_main.xml như bên dưới,
- + Khai báo trong AndroidManifest.xml: gán MainActivity là Activity chính,
- + Thực thi kiểm tra kết quả.

```
<?xml version="1.0" encoding="utf-8"?>
<ListView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/lst_menu"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    tools:context=".MainActivity" />
```

– **Bước 10: điều chỉnh mã lệnh Java cho MainActivity**

- + Sử dụng ListView làm thực đơn cho ứng dụng,
- + Thực thi kiểm tra kết quả.

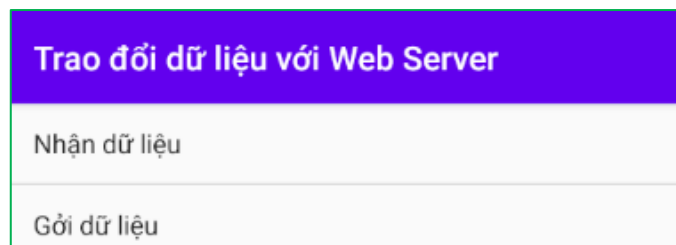
```
public class MainActivity extends AppCompatActivity {

    String[] arrMenu = {"Nhận dữ liệu", "Gửi dữ liệu"};
    ListView listviewDuLieu;
    ArrayAdapter<String> adapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

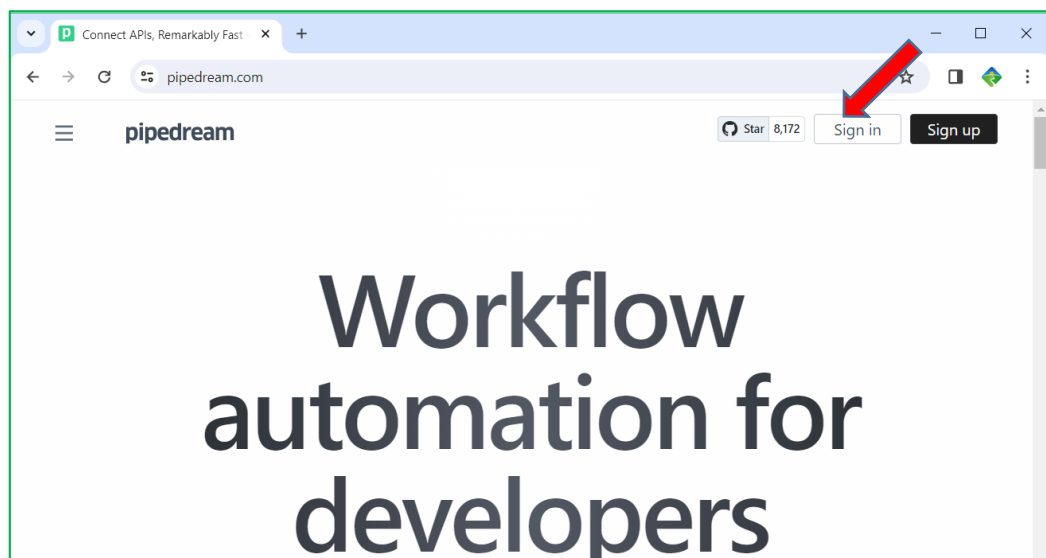
        listviewDuLieu = findViewById(R.id.lst_menu);
        adapter = new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, arrMenu);
        listviewDuLieu.setAdapter(adapter);
        listviewDuLieu.setOnItemClickListener(new ChonCongViec());
    }
}
```

```
private class ChonCongViec implements
    android.widget.AdapterView.OnItemClickListener {
    @Override
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        Intent intent = new Intent(MainActivity.this,
            NhanDuLieuTuServer.class);
        if (position == 1)
            intent = new Intent(MainActivity.this,
                GoiDuLieuDenServer.class);
        startActivity(intent);
    }
}
```



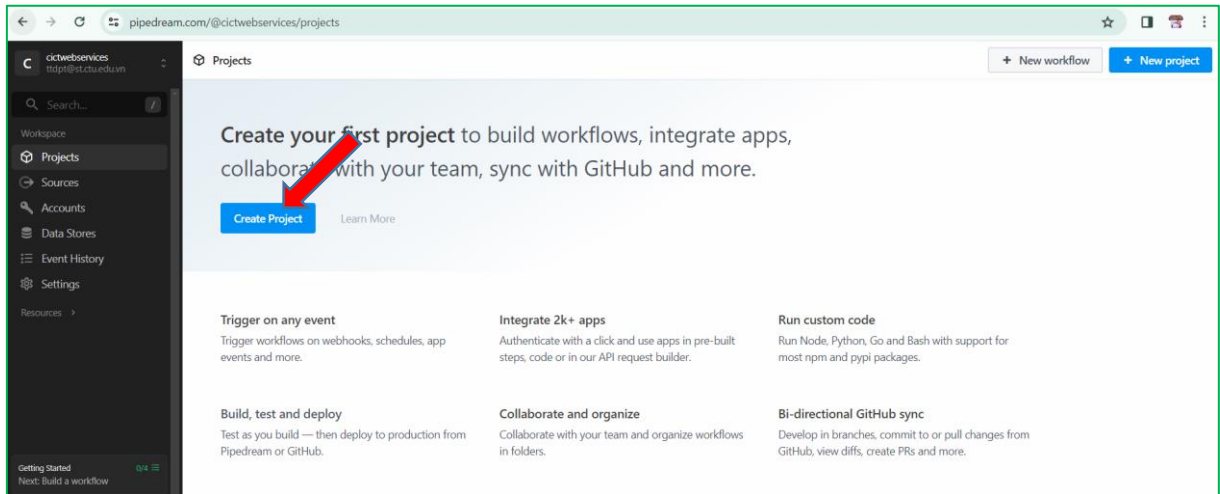
Hình 5. ListView làm thực đơn cho ứng dụng

- Bước 11 (**Nội dung làm thêm, không bắt buộc**): sử dụng Pipedream kiểm thử mã lệnh gửi dữ liệu đến Web Server
- + Vào Pipedream (<https://pipedream.com>)

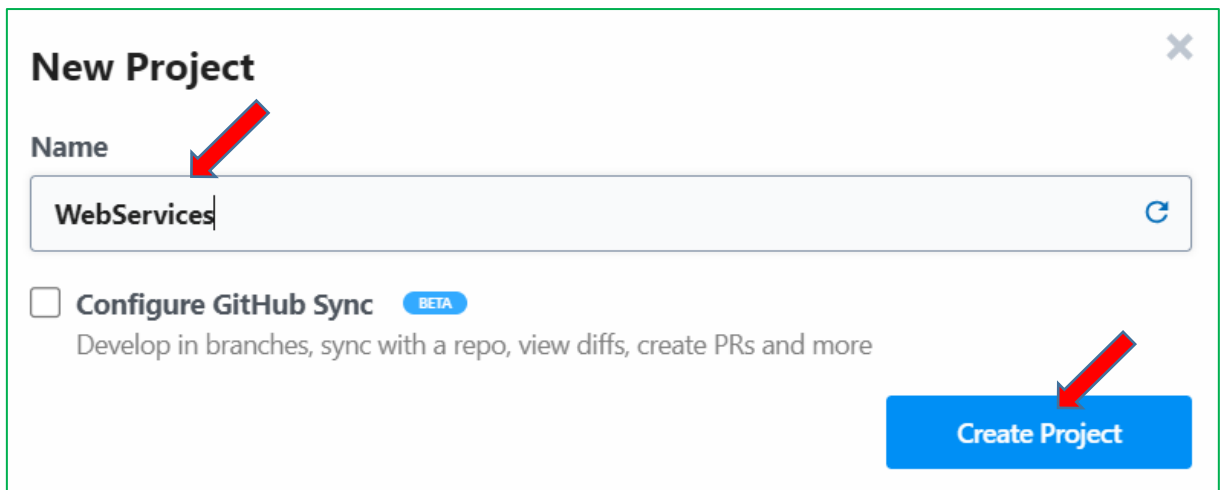


Hình 6. Sử dụng Pipedream

- + Có thể sử dụng tài khoản Email (Sign in with Google) để đăng nhập vào Pipedream, nhập các thông tin theo quy định
- + Tạo **dự án** mới như Hình 7, đặt tên dự án như Hình 8

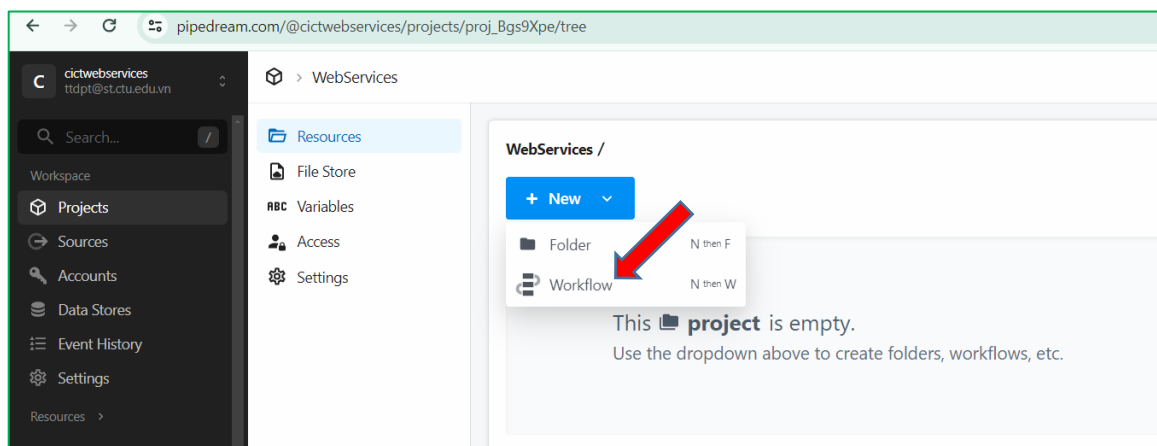


Hình 7. Tạo dự án mới



Hình 8. Đặt tên dự án

- + Tạo **Workflow** mới như Hình 9, đặt tên Workflow như Hình 10



Hình 9. Tạo Workflow mới

Create new workflow

Workflow Name
SendJsonData

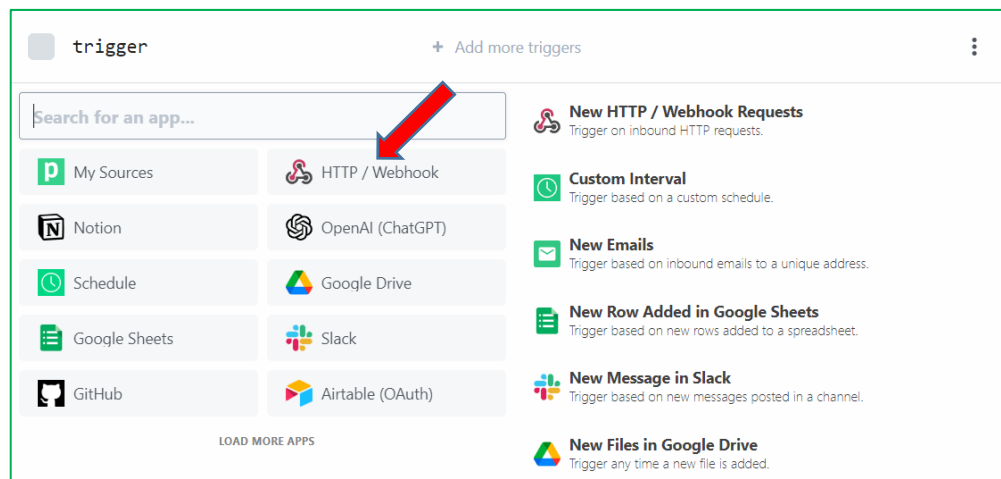
Execution Controls
Execution time at 256 MB memory costs 1 credit per 30-second interval. [Learn more.](#)
Timeout
1s 2m 4m 6m 8m 10m 12m 30 seconds
Memory
256MB 2048MB 4096MB 6144MB 8192MB 10240MB 256 MB
1 credit per workflow execution

☒ **Send error notifications**
☐ **Automatically retry on errors** Advanced
☐ **Disable data retention**
☐ **Limit concurrency** Advanced
☐ **Limit execution rate** Advanced
☐ **Eliminate cold starts** Advanced
☐ **Run in VPC** Business

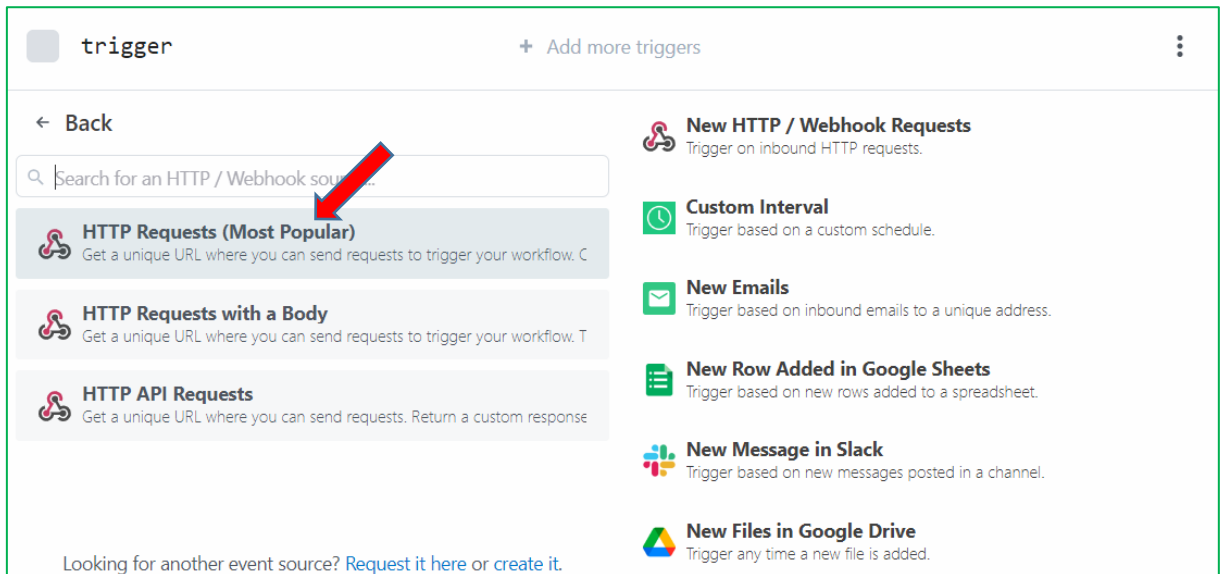
Create Workflow

Hình 10. Đặt tên Workflow

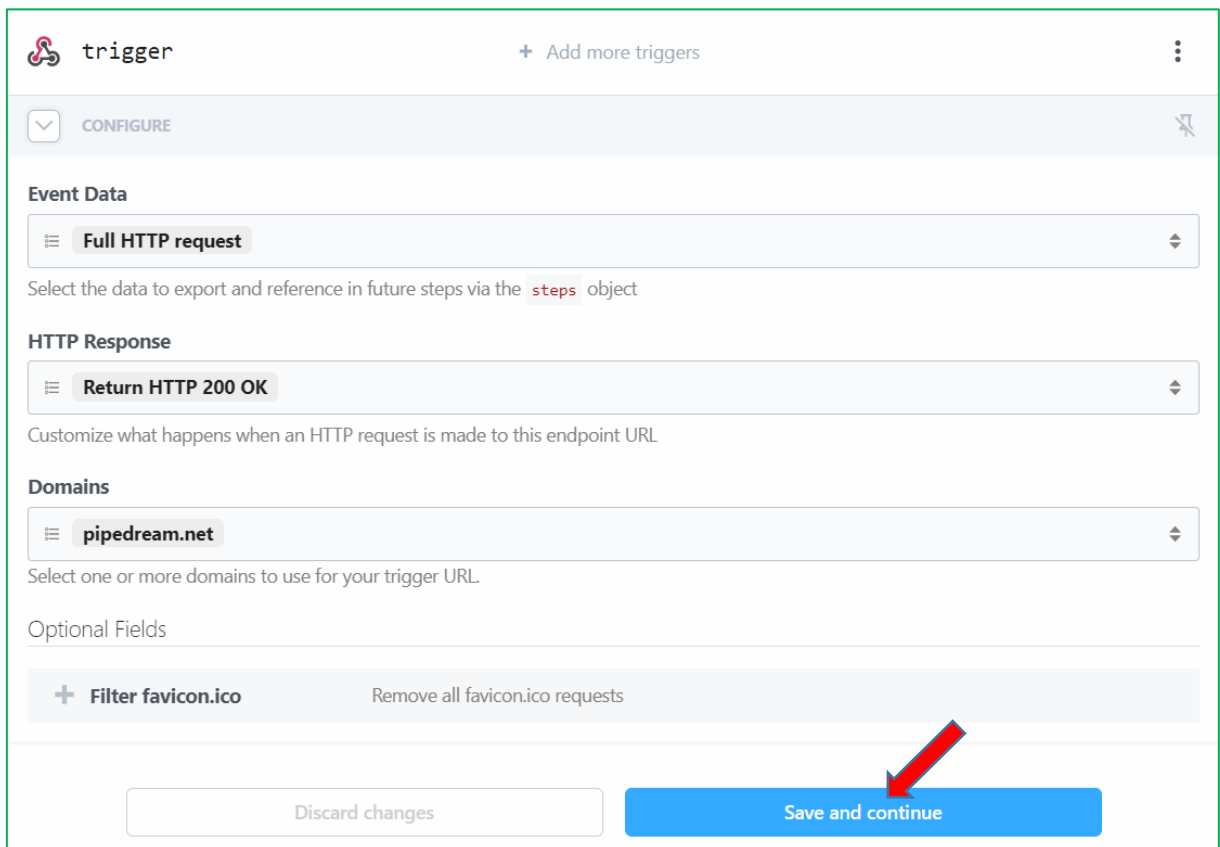
- + Thêm **trigger** mới như Hình 11, Hình 12, và Hình 13



Hình 11. Tạo trigger mới

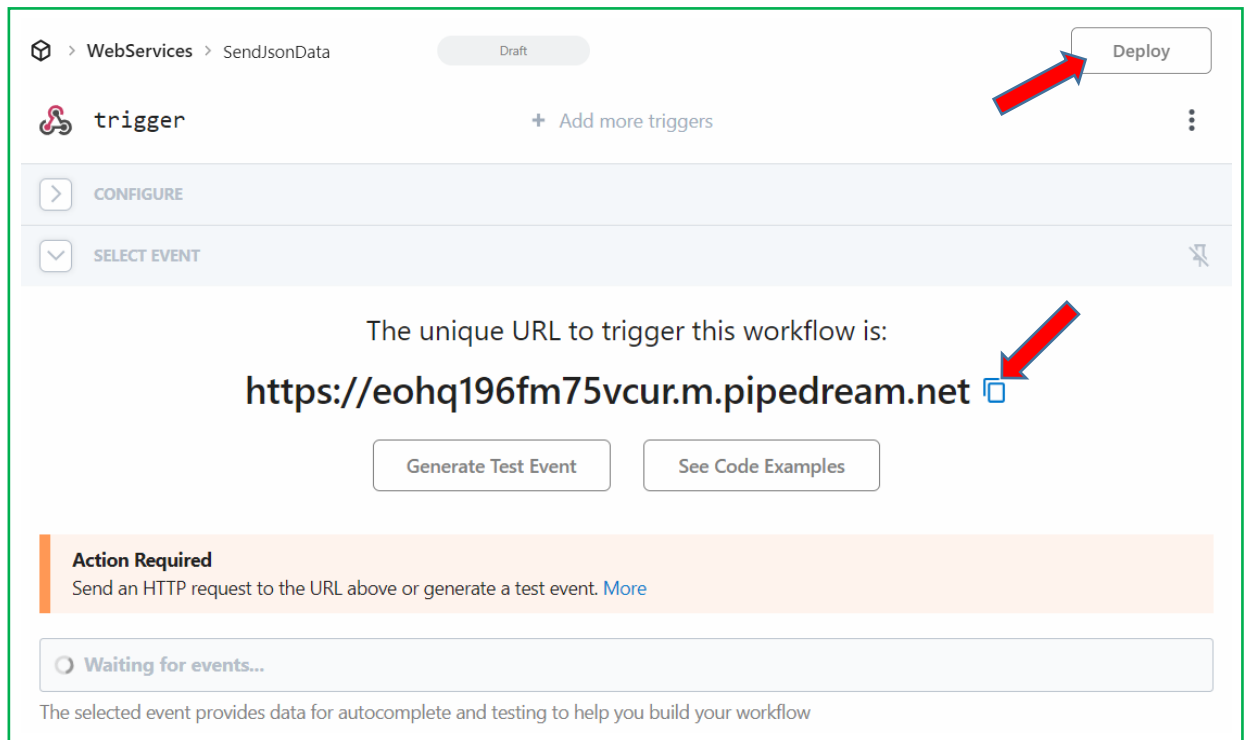


Hình 12. Chọn loại trigger mới



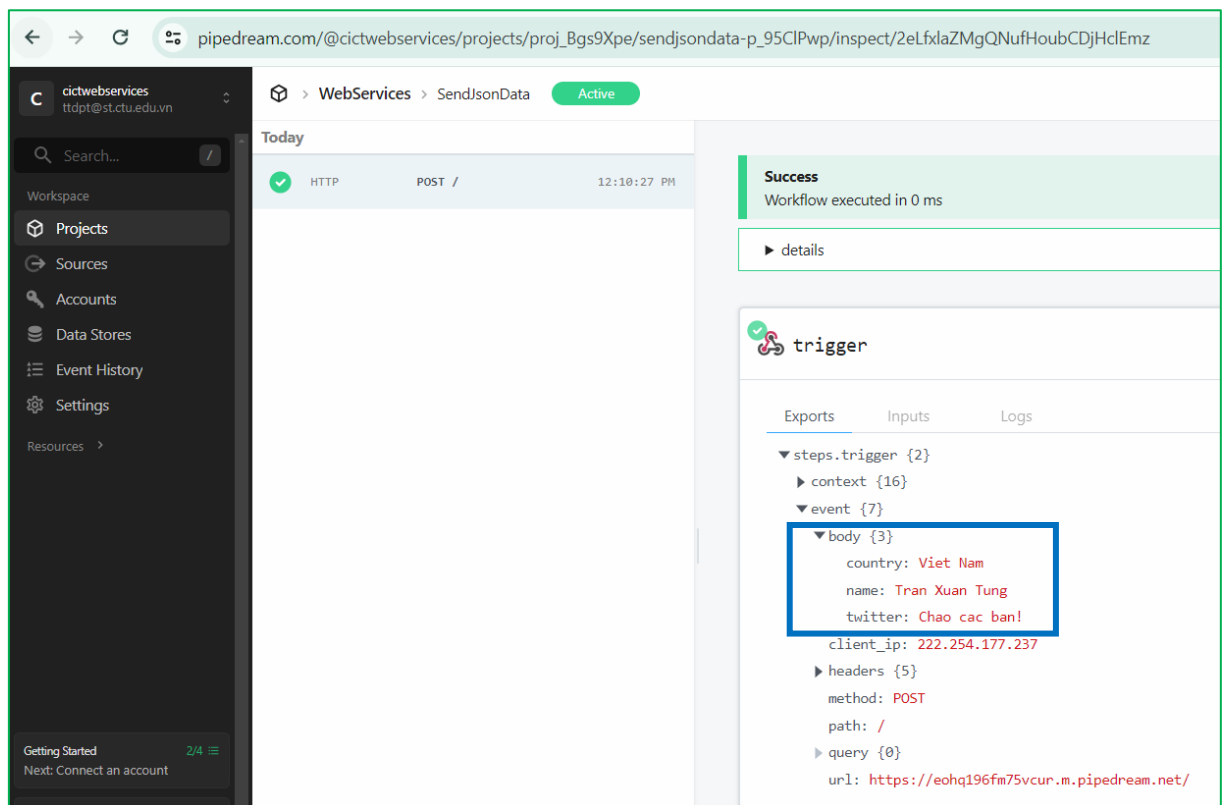
Hình 13. Lưu trigger mới

- + Lưu địa chỉ URL và triển khai Workflow (Deploy) → chọn OK hoàn thành



Hình 14. Lưu địa chỉ URL và triển khai Workflow (Deploy)

- + Sử dụng địa chỉ URL ở trên làm địa chỉ gửi dữ liệu, thực thi ứng dụng và kiểm tra kết quả.



Hình 1. Kiểm tra kết quả