

**Đề bài:** - Viết chương trình xem dự báo thời tiết.

**Giao diện:**

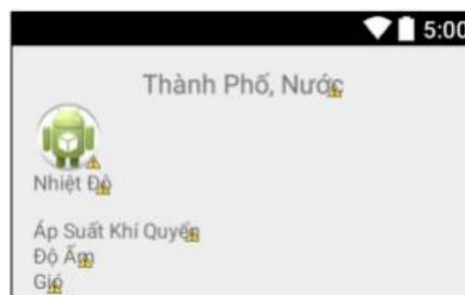


**Gợi ý thực hiện:**

- Lấy dữ liệu thông qua đường link: <http://api.openweathermap.org/data/2.5/weather?q=> (cộng thêm chuỗi "city,country" để lấy dữ liệu của thành phố thuộc nước nào)
- Dùng "JSONObject" và "JSONArray" đọc dữ liệu JSON.
- Lấy dữ liệu tải về hiển thị lên.
- Tải hình ảnh thông qua đường link: <http://openweathermap.org/img/w/> (cộng thêm tên hình .png).

**Hướng dẫn chi tiết:**

**Bước 1:** Thiết kế giao diện trong file "activity\_main.xml":



Bước 2: Tạo đối tượng vị trí:

```
public class ViTri implements Serializable {
    private static final long serialVersionUID = 1L;
    private String tenNuoc;
    private String thanhPho;
    ...
}
```

Bước 3: Tạo đối tượng thời tiết:

```
public class ThoiTiet {
    public ViTri viTri;
    public TinhTrangHienTai tinhTrangHienTai = new TinhTrangHienTai();
    public NhietDo nhietDo = new NhietDo();
    public Gio gio = new Gio();
    public byte[] hìnhAnh;
    public class TinhTrangHienTai {
        private String tenHinh;
        private float apSuat;
        private float doAm;
        ...
    }
    public class NhietDo {
        private float nd;
        ...
    }
}
```

```
public class Gio {
    private float tocDo;
    ...
}
```

Bước 4: Tạo lớp lấy dữ liệu ra gán vào đối tượng thời tiết từ chuỗi dữ liệu trả về:

```
public class JSONWeatherParser {
    public static ThoiTiet getWeather(String data) throws JSONException {
        ThoiTiet thoiTiet = new ThoiTiet();
        JSONObject jsonObj = new JSONObject(data);
        ViTri loc = new ViTri();
        JSONObject sysObj = getObject("sys", jsonObj);
        loc.setTenNuoc(getString("country", sysObj));
        loc.setThanhPho(getString("name", jsonObj));
        thoiTiet.viTri = loc;
        JSONObject mainObj = getObject("main", jsonObj);
        thoiTiet.tinhTrangHienTai.setDoAm(getInt("humidity", mainObj));
        thoiTiet.tinhTrangHienTai.setApSuat(getInt("pressure", mainObj));
        thoiTiet.nhietDo.setND(getFloat("temp", mainObj));
        JSONArray jArr = jsonObj.getJSONArray("weather");
        JSONObject JSONWeather = jArr.getJSONObject(0);
        thoiTiet.tinhTrangHienTai.setTenHinh(getString("icon",
            JSONWeather));
        return thoiTiet;
    }
    private static JSONObject getObject(String tagName, JSONObject jsonObj)
        throws JSONException {
        JSONObject subObj = jsonObj.getJSONObject(tagName);
        return subObj;
    }
}
```

```

private static JSONObject getObject(String tagName, JSONObject jsonObj)
throws JSONException {
    JSONObject subObj = jsonObj.getJSONObject(tagName);
    return subObj;
}
private static String getString(String tagName, JSONObject jsonObj)
throws JSONException {
    return jsonObj.getString(tagName);
}
private static float getFloat(String tagName, JSONObject jsonObj)
throws JSONException {
    return (float) jsonObj.getDouble(tagName);
}
private static int getInt(String tagName, JSONObject jsonObj)
throws JSONException {
    return jsonObj.getInt(tagName);
}
}

```

*Chú thích:* Ở đây có lớp cần chú ý đó là JSONObject (là một đối tượng của dữ liệu JSON nó cung cấp cho các phương thức để lấy dữ liệu theo key) và JSONArray (là danh sách các đối tượng JSONObject).

*Bước 5:* Tạo lớp thực hiện lấy dữ liệu trả về và tải hình:

```

private static String BASE_URL =
"http://api.openweathermap.org/data/2.5/weather?q=";
private static String IMG_URL = "http://openweathermap.org/img/w/";
public String getWeatherData(String location) {
    HttpURLConnection con = null;
    InputStream is = null;
    try {
        con = (HttpURLConnection) (new URL(BASE_URL + location))
        .openConnection();
        con.setRequestMethod("GET");
        con.setDoInput(true);
        con.setDoOutput(true);
        con.connect();
        StringBuffer buffer = new StringBuffer();
        is = con.getInputStream();
        BufferedReader br = new BufferedReader(new
        InputStreamReader(is));
        String line = null;
        while ((line = br.readLine()) != null)
            buffer.append(line + "\r\n");
        is.close();
        con.disconnect();
        return buffer.toString();
    } catch (Throwable t) {
        t.printStackTrace();
    } finally {
        try {
            is.close();
        } catch (Throwable t) {
        }
        try {
            con.disconnect();
        } catch (Throwable t) {
        }
    }
    return null;
}
}

```

```

public byte[] getImage(String code) {
    HttpURLConnection con = null;
    InputStream is = null;
    try {
        con = (HttpURLConnection) (new URL(IMG_URL + code + ".png"))
            .openConnection();
        con.setRequestMethod("GET");
        con.setDoInput(true);
        con.setDoOutput(true);
        con.connect();
        is = con.getInputStream();
        byte[] buffer = new byte[1024];
        ByteArrayOutputStream baos = new ByteArrayOutputStream();

```

```

        while (is.read(buffer) != -1)
            baos.write(buffer);
        return baos.toByteArray();
    } catch (Throwable t) {
        t.printStackTrace();
    } finally {
        try {
            is.close();
        } catch (Throwable t) {
        }
        try {
            con.disconnect();
        } catch (Throwable t) {
        }
    }
    return null;
}

```

Bước 6: Viết xử lý trong MainActivity:

```

private TextView cityText;
private TextView temp;
private TextView press;
private TextView windSpeed;
private TextView hum;
private ImageView imgView;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    String city = "HaNoi,vn";
    cityText = (TextView) findViewById(R.id.cityText);
    temp = (TextView) findViewById(R.id.temp);
    hum = (TextView) findViewById(R.id.hum);
    press = (TextView) findViewById(R.id.press);
    windSpeed = (TextView) findViewById(R.id.windSpeed);
    imgView = (ImageView) findViewById(R.id.condIcon);
    JSONWeatherTask task = new JSONWeatherTask();
    task.execute(new String[]{city});
}
private class JSONWeatherTask extends AsyncTask<String,
    Void, ThoiTiet> {
    @Override
    protected ThoiTiet doInBackground(String... params) {
        ThoiTiet thoiTiet = new ThoiTiet();
        String data = ((new WeatherHttpClient())
            .getWeatherData(params[0]));
        try {
            thoiTiet = JSONWeatherParser.getWeather(data);
            thoiTiet.hinhAnh = (new WeatherHttpClient())
                .getImage(thoiTiet.tinhTrangHienTai

```

```

        .getTenHinh()));
    } catch (JSONException e) {
        e.printStackTrace();
    }
    return thoiTiet;
}

@Override
protected void onPostExecute(ThoiTiet thoiTiet) {
    super.onPostExecute(thoiTiet);
    try {
        if (thoiTiet.hinhAnh != null && thoiTiet.hinhAnh.length>0){
            Bitmap img = BitmapFactory.decodeByteArray(
                thoiTiet.hinhAnh, 0,
                thoiTiet.hinhAnh.length);
            imageView.setImageBitmap(img);
        }
        cityText.setText(thoiTiet.viTri.getThanhPho()
            + "," + thoiTiet.viTri.getTenNuoc());
        temp.setText(Math.round((thoiTiet.nhietDo
            .getND() - 273.15))+ "C");
        hum.setText("" + thoiTiet.tinhTrangHienTai
            .getDoAm() + "%");
        press.setText("" + thoiTiet.tinhTrangHienTai
            .getApSuat() + " hPa");
        windSpeed.setText("" + thoiTiet.gio.getTocDo() + " mps");
    } catch (Exception e) {
    }
}
}
}

```