

# COMS\_510 Android Fundamental Part3

Chen-Yeou (Charles) Yu

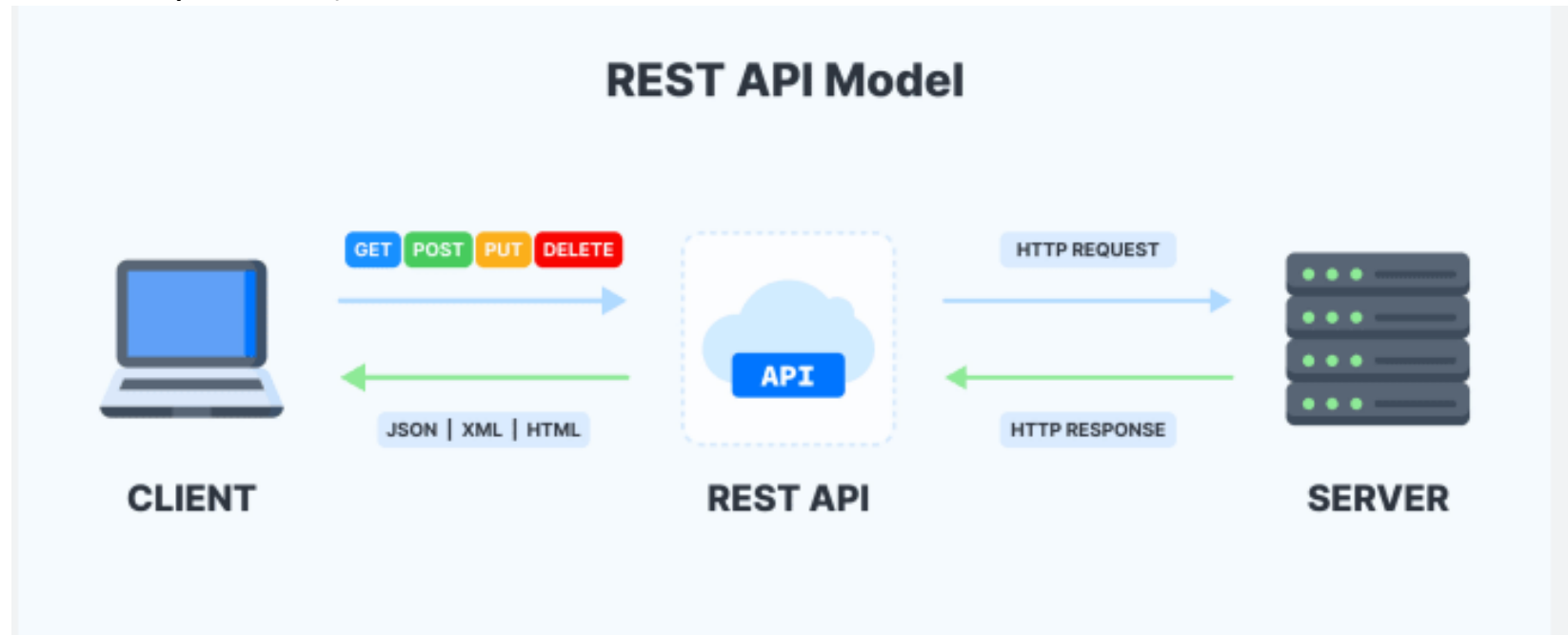
11/6/2017

# Outlines

- [Demo1] NumVerify

# A Restful API: NumVerify

- Sometimes, a RESTful API can help us in doing queries, **programmatically**.
  - Phone numbers
  - News (if the website provides)
  - Local Info. (if the website provides)
  - and more...!



# A Restful API: NumVerify

- You can query a phone number
- You can use this service for free for 1 month
  - numverify.com
- Registration is a must, and you will get a an API key



# A Restful API: NumVerify

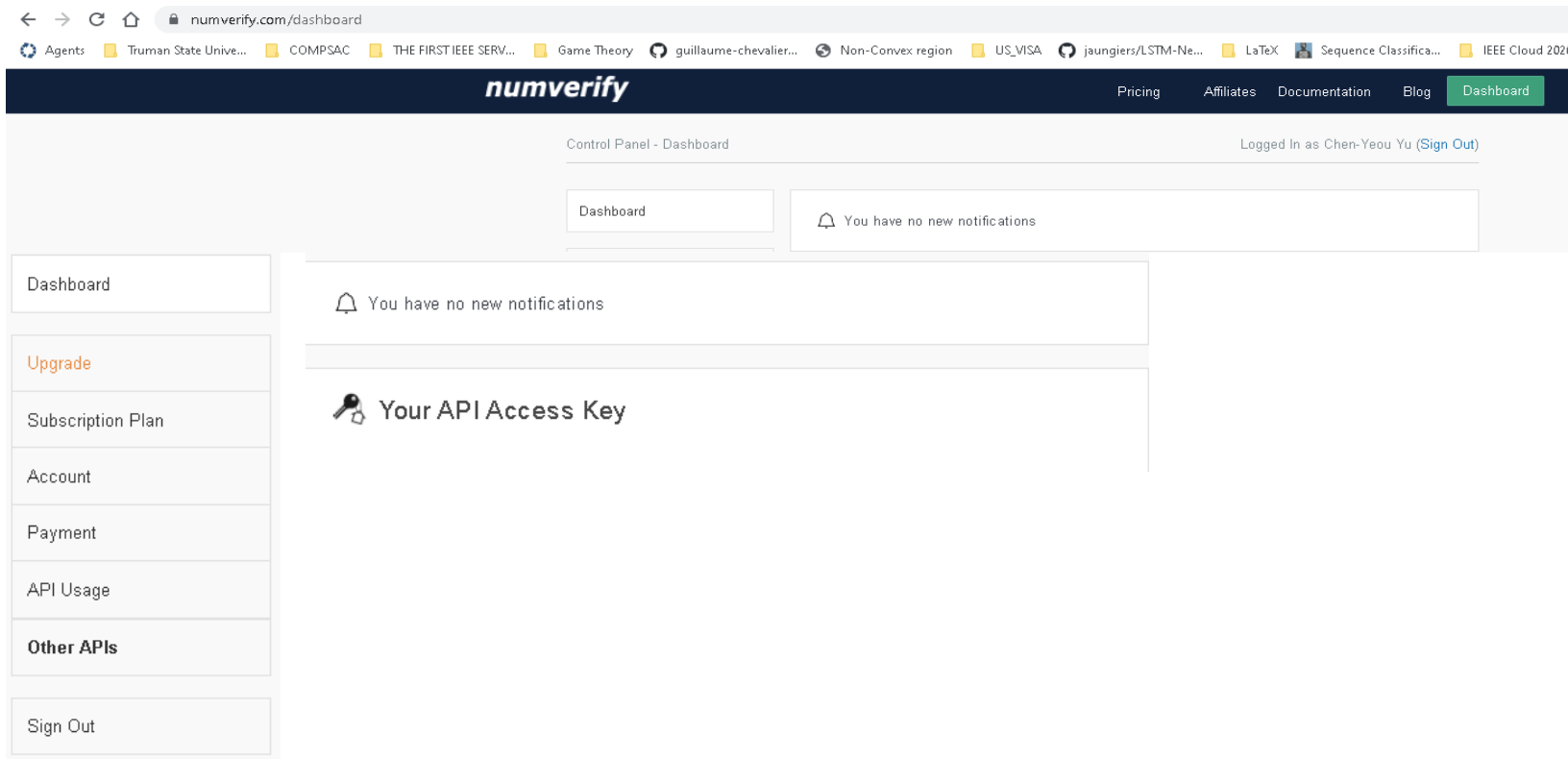
- The result of the query can be well-formatted or JSON

FORMATTED	
Valid	✓
Local Format	4158586273
Intl. Format	+14158586273
Country	United States of America
Location	Novato
Carrier	AT&T Mobility LLC
Line Type	Mobile

JSON	
<pre>{   "valid": true,   "local_format": "4158586273",   "intl_format": "+14158586273",   "country_code": "US",   "country_name": "United States of America",   "location": "Novato",   "carrier": "AT&amp;T Mobility LLC",   "line_type": "mobile" }</pre>	

# A Restful API: NumVerify

- After a login to the website, click the “Dashboard” and you will see your API Access Key.
- No Worries! It is free to use for 1 month!



# A Restful API: Num Verify

Click the  
“Documentation”  
tab

Pricing Affiliates Documentation Blog Dashboard

## API response objects:

Each API response consists of 9 individual JSON response objects:

Object	Description
"valid"	Returns <code>true</code> if the specified phone number is valid.
"number"	Returns the phone number you specified in a clean format. (stripped of any special characters)
"local_format"	Returns the local (national) format of the specified phone number.
"international_format"	Returns the international format of the specified phone number.
"country_prefix"	Returns the international country dial prefix for the specified phone number.
"country_code"	Returns the 2-letter country code assigned to the specified phone number.
"country_name"	Returns the full country name assigned to the specified phone number.
"location"	If available, returns the location (city, state, or county) assigned to the specified phone number.
"carrier"	Returns the name of the carrier which the specified phone number is registered with.
"line_type"	Returns the line type of the specified phone number (See: <a href="#">Line Type Detection</a> )

## Make an API Request

Since all existing validation data is returned by the same main API endpoint, making a phone number verification request to the numverify API is simple.

### Most basic API request:

Chen-Yeou, call the API yourself simply by clicking on the request URL below: (You should also have a look at the Browser URL)

```
http://apilayer.net/api/validate
? access_key = YOUR_ACCESS_KEY
& number = 14158586273
```

As you can see, in addition to the `access_key` parameter, there is only one required parameter (`number`) to start validating phone numbers.

### Optional parameters:

```
country_code    specify a country code you intend to
                use a national phone number in your request
format          use to order to request prettified
                JSON result set (use only for debugging!)
callback        append your preferred JSONP callback
                name. (See "JSONP Callbacks" section)
```

## API Response

All numverify validation data is returned in universal and lightweight JSON format. Find below a standard API result set:

```
{
  "valid": true,
  "number": "14158586273",
  "local_format": "4158586273",
  "international_format": "+14158586273",
  "country_prefix": "+1",
  "country_code": "US",
  "country_name": "United States of America",
  "location": "Novato",
  "carrier": "AT&T Mobility LLC",
  "line_type": "mobile"
}
```

# A Restful API: NumVerify

- You need to put your access key (API KEY) here, as a Java String
- Put a phone number you want to check. I put Truman's number as an example
- A TextView is used to show the returned info from our query (against the phone number)

```
public class MainActivity extends AppCompatActivity {  
    // Set endpoint and your access key, you need to get access key for API first.  
    // Please go to https://numverify.com/  
    1 usage  
    public String access_key = "YOUR_ACCESS_KEY_HERE";  
    // Input the number you want to query  
    1 usage  
    public String phone_number = "16607854000";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        final TextView mStatusTextView = (TextView) findViewById(R.id.status_text);  
  
        // Instantiate the RequestQueue.  
        RequestQueue queue = Volley.newRequestQueue(context: this);  
        String url = "http://apilayer.net/api/validate?access_key=" + access_key + "&number=" + phone_number;
```



# A Restful API: NumVerify

- Here is the query string provided in the documentation

```
// Instantiate the RequestQueue.  
RequestQueue queue = Volley.newRequestQueue( context: this);  
String url = "http://apilayer.net/api/validate?access_key=" + access_key + "&number=" + phone_number;
```

- Volley!? Seriously?
  - If you used to be in my Database class, you know it is a kind of library used by Android to query the databases
  - We can make a quick guess, the backend of NumVerify is possibly a MySQL (very likely) or a PostgreSQL (a sibling of MySQL)
  - Yes, we setup a RequestQueue object here

# A Restful API: NumVerify

- Everything is put in the onCreate()
- Our crucial job is to build our StringRequest
- You can also use JsonRequest
- Finally, we use the RequestQueue object (queue), to add() the StringRequest
- Easy!

```
// Request a string response from the provided URL.
StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
    new Response.Listener<String>() {
        3 usages
        @Override
        public void onResponse(String response) {
            // Display the first 500 characters of the response string.
            if (response != null) {
                // mStatusTextView.setText("Response is: " + response.substring(0, 500));
                mStatusTextView.setText("Response is: " + response);
            } else {
                mStatusTextView.setText("There is nothing in the response");
            }
        }
    }, new Response.ErrorListener() {
        2 usages
        @Override
        public void onErrorResponse(VolleyError error) {
            // mStatusTextView.setText("That didn't work!");
            mStatusTextView.setText(error.toString());
        }
    });

// Add the request to the RequestQueue.
queue.add(stringRequest);
}
```

# A Restful API: NumVerify

- Here is the documentation, you can easily find StringRequest vs. JsonRequest
  - <https://javadoc.io/static/com.android.volley/volley/1.1.1/com/android/volley/toolbox/package-summary.html>

# A Restful API: NumVerify

- All you need to do is to override the callback function: onResponse()
- You can also limit how many characters of response string
- Attach the response String onto the TextView

```
// Request a string response from the provided URL.
StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
    new Response.Listener<String>() {
        3 usages
        @Override
        public void onResponse(String response) {
            // Display the first 500 characters of the response string.
            if (response != null) {
                // mStatusTextView.setText("Response is: " + response.substring(0, 500));
                mStatusTextView.setText("Response is: " + response);
            } else {
                mStatusTextView.setText("There is nothing in the response");
            }
        }
    }, new Response.ErrorListener() {
        2 usages
        @Override
        public void onErrorResponse(VolleyError error) {
            // mStatusTextView.setText("That didn't work!");
            mStatusTextView.setText(error.toString());
        }
    });

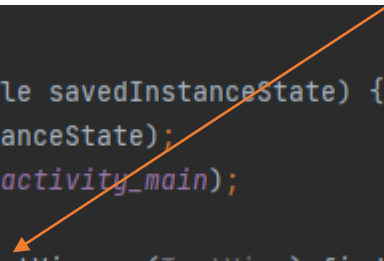
// Add the request to the RequestQueue.
queue.add(stringRequest);
}
```

# A Restful API: NumVerify

- In slide #8, I had a mStatusTextView to show the returned string

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

    final TextView mStatusTextView = (TextView) findViewById(R.id.status_text);
```



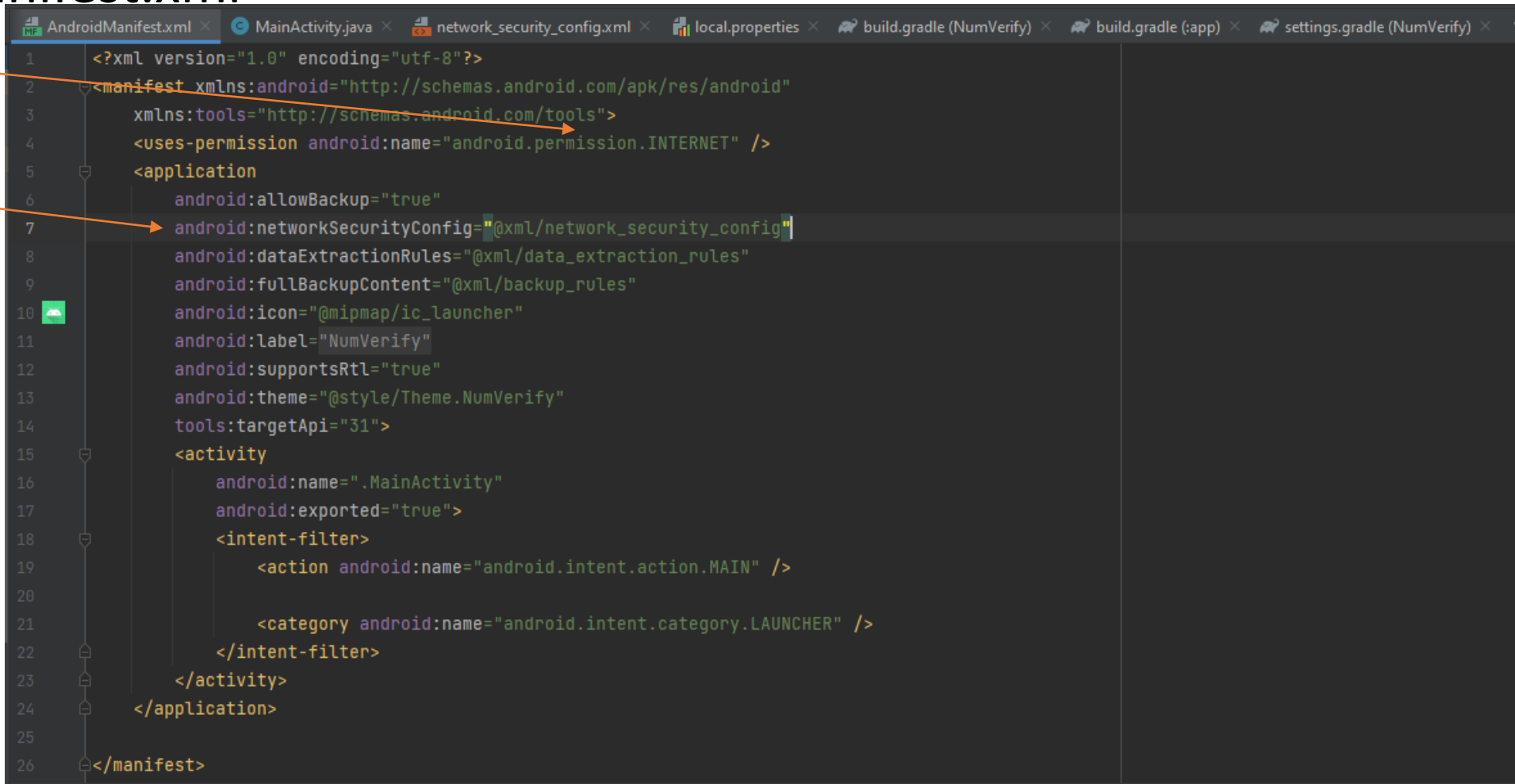
- A couple of other things, I do the following setup.
- If we don't do that, we will have run-time errors. The reason is that, in the later versions of Android, they have more security related mechanism. (check the next page)

# A Restful API: NumVerify

- In AndroidManifest.xml

- Access right

- Network security config.

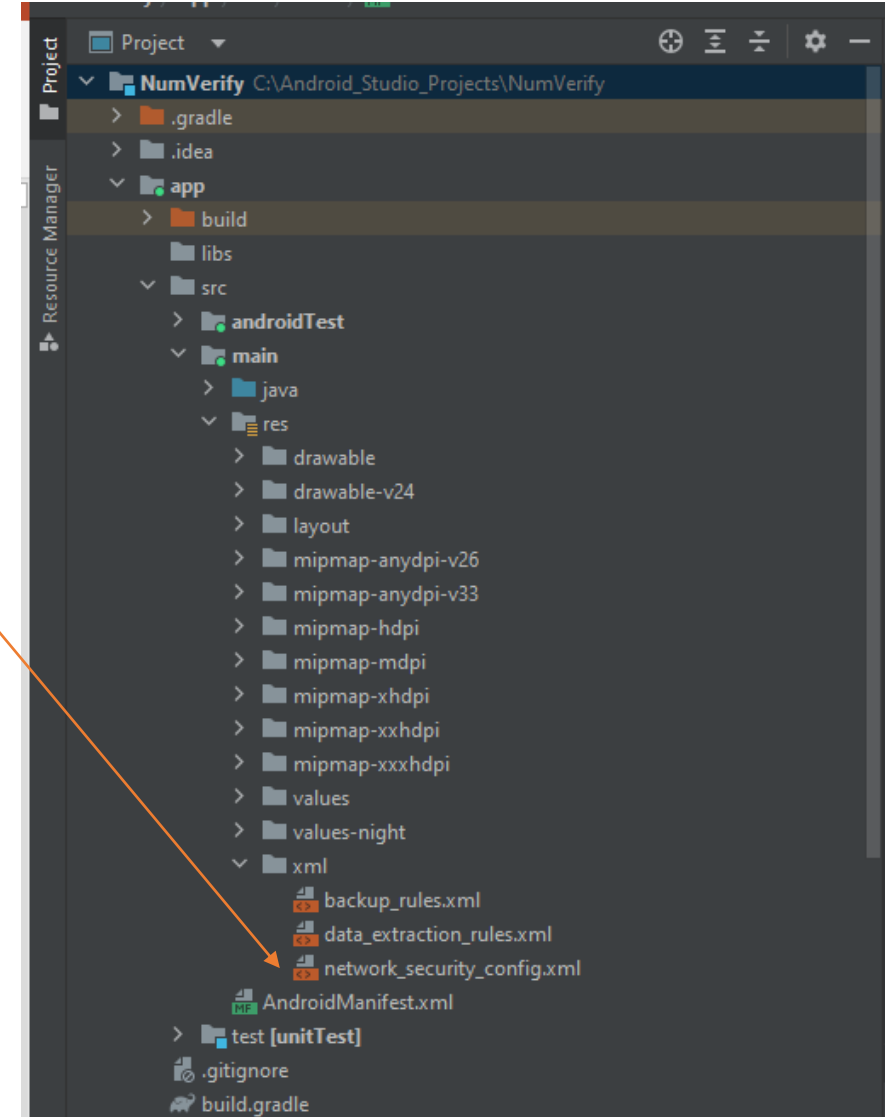


```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3  xmlns:tools="http://schemas.android.com/tools">
4  <uses-permission android:name="android.permission.INTERNET" />
5  <application
6  android:allowBackup="true"
7  android:networkSecurityConfig="@xml/network_security_config"
8  android:dataExtractionRules="@xml/data_extraction_rules"
9  android:fullBackupContent="@xml/backup_rules"
10 android:icon="@mipmap/ic_launcher"
11 android:label="NumVerify"
12 android:supportsRtl="true"
13 android:theme="@style/Theme.NumVerify"
14 tools:targetApi="31">
15   <activity
16     android:name=".MainActivity"
17     android:exported="true">
18     <intent-filter>
19       <action android:name="android.intent.action.MAIN" />
20
21       <category android:name="android.intent.category.LAUNCHER" />
22     </intent-filter>
23   </activity>
24 </application>
25
26 </manifest>
```

# A Restful API: NumVerify

- I created a network\_security\_config.xml
- I put it into the folder:
  - /app/src/main/res/xml
- After Android v8.0, the clear text transmission over the internet is not allowed, by default
- I use this to make it allowed

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <network-security-config xmlns:android="http://schemas.android.com/apk/res/android">
3   <domain-config cleartextTrafficPermitted="true">
4     <domain includeSubdomains="true">apilayer.net</domain>
5   </domain-config>
6 </network-security-config>
```



# A Restful API: NumVerify

- In the very beginning, I was thinking it is my query string is wrong:

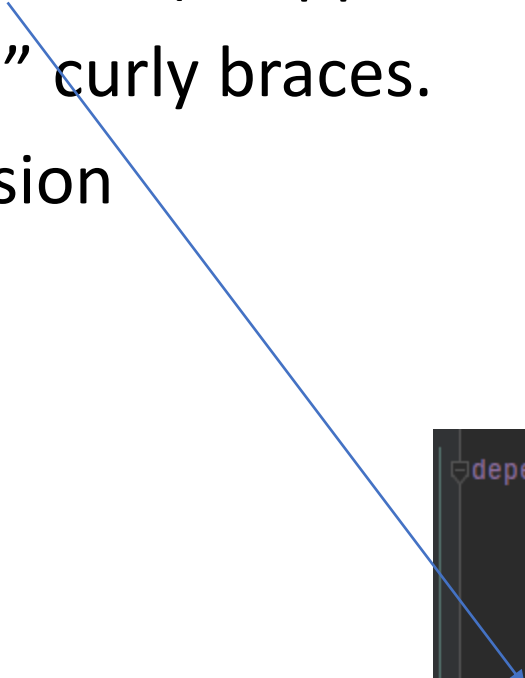
```
String url ="http://apilayer.net/api/validate?access_key=" + access_key + "&number=" + phone_number;
```

- If I changed to “https”, the returned info will tell you: base on your plan (free of \$), you cannot access such a kind of service.
- So, “http” is correct



# A Restful API: NumVerify

- Finally, in order to use Volley, I changed the configuration for this library, in the **build.gradle** (in app's level)
- In the “dependencies” curly braces.
- 1.2.1 is the latest version



```
dependencies {  
  
    implementation 'androidx.appcompat:appcompat:1.6.1'  
    implementation 'com.google.android.material:material:1.8.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
    implementation 'com.android.volley:volley:1.2.1'  
    testImplementation 'junit:junit:4.13.2'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'  
}
```

# A Restful API: NumVerify

- See? Our Truman phone number is a landline
- Location is Kirksville
- You might be able to get more info. if you pay \$\$

^ \_ ^

