**HTTP REQUEST:**

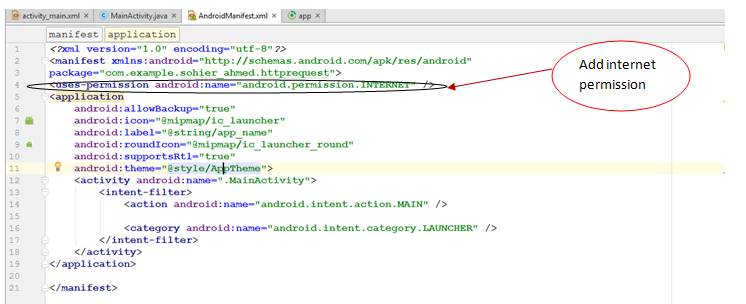
One of the ways to make http request:

**1-using Volley Library:**

Volley library is http library that makes networking for android applications easier and faster

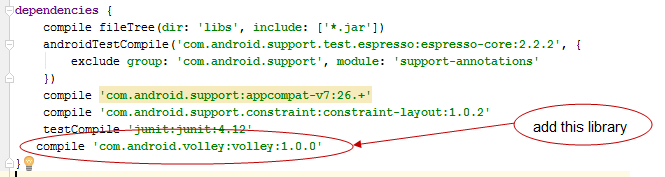
Steps to write simple program on Android that performs HTTP GET and POST request

***\*in AndroidManifest.xml :***



***In build.gradle (Module: app):\****

* -Add compile **'com.android.volley:volley:1.0.0'**



-Press  **sync now**



* Wait till sync finish successfully

***In mainActivity:\****

-create object of type Request Queue



-create String to hold URL



-Create object of type RequestString

Constructor of class RequestString takes 4 parameters: 

*1-method of request (get, post,….).*

Difference between GET and POST methods:

\*GET is used for viewing not changing.

\*POST is used for changing.

*2-url*

*3-Listener*

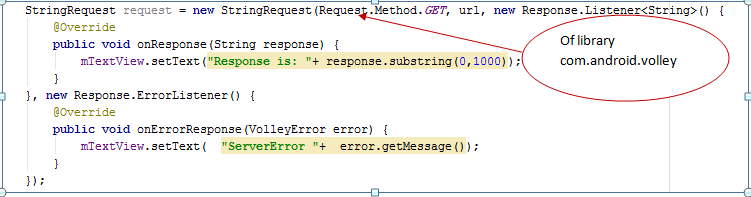
**new** Response.Listener<String>() {

@Override  
**public void** onResponse (String response) {  
 *// body of method here.*}

}

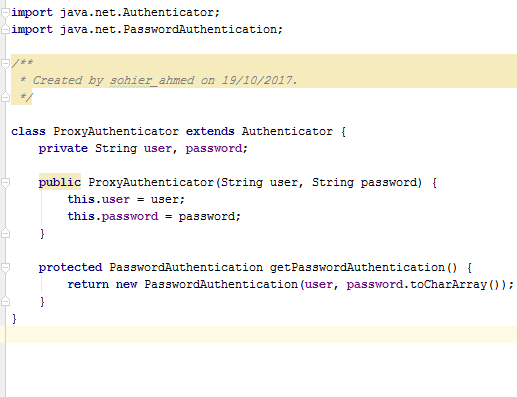
*4-ErrorListener method.*

**new** Response.ErrorListener () {  
 @Override  
 **public void** onErrorResponse(VolleyError error) {  
 mTextView.setText( **"ServerError "**+ error.getMessage());  
 }  
}



To solve network issues:

Create **class** ProxyAuthenticator **extends** Authenticator

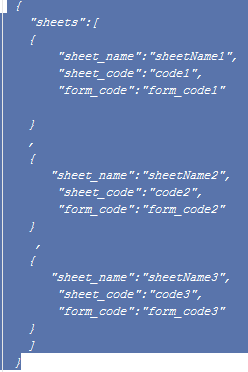


***In main activity:***



**Encoding data:**

That is the result wanted from encoding

****

Json format

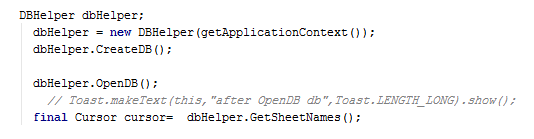
Convert results within cursor to json format

Cursor holding results

Execute query

**Step1:**

-Call method "GetSheetNames" to get available sheets.



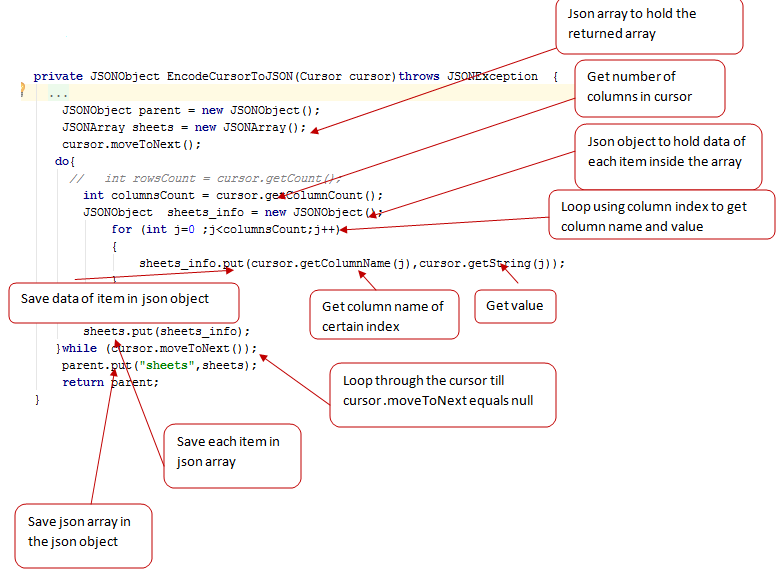
**Step2:**

-pass the cursor that holds the result of the method to be input to the method EncodeCursorToJSON

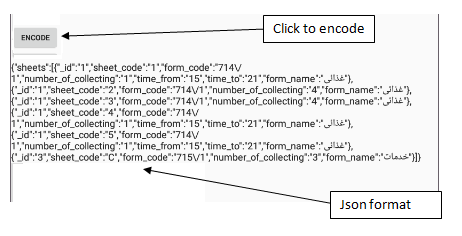




Method EncodeCursorToJSON returns JSONOBJECT which is showed in text view



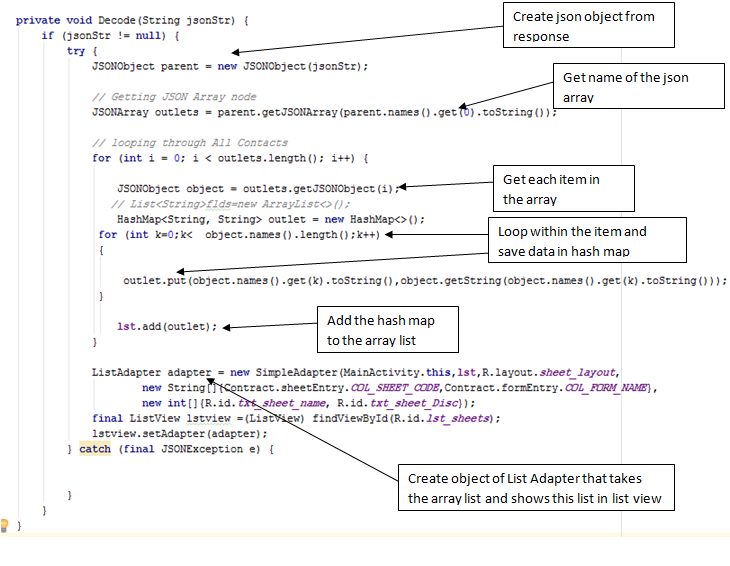
**The final output:**

* 

**Decode json object:**

**Step** **1:**

We take the jsonobject which came from the previous encoding and convert it to string to be the input of *Decode* method



This is the final output of decoding:

