**Naturally the next step in this process is to foster solid safe communication between our android application and our webpage.**

Below I have found a few promising links that I shall further explore respectfully:

<https://developer.android.com/training/volley/index.html>

**Volly:**

I was able to download the Android Libraries known as “Volly”; but using them in other Android Studio applications proved to be a problem while trying to create a .AAR package (needed for the introduction of the methods and functionality. This method seems promising but I can not figure a work-around here.

*Project Files Available on Request.*

<http://www.androidhive.info/2012/05/how-to-connect-android-with-php-mysql/>

**Connect Android Application with PHP MySQL:**

I created the PHP files “needed” for the connection between the ANdroid Application and a MySQL server. I keep running into errors in my Android Studio application. I am not sure if this method is promising or would solve our issue since there may be a large security risk of SQL injection attacks using this method.

*Project Files Available on Request.*

<https://www.youtube.com/watch?time_continue=3&v=nJeWmsuMZ6I&feature=emb_logo>

**How to get data from website in android: Using HttpClient**

The above video looked promising with a user establishing a connection where-in the android application could gather information off of a website link stored in a variable.

I ran with his steps until completion, it seems he uses some sort of unnamed repository that allows him to use the particular methods he does. With further research it seems that these particular methods are no longer available with the latest version of Android Studio, however a new version of the method exists “HttpURLConnection class”

*Project Files Available on Request.*

**HttpURLConnection class:**

*I made an application using the HttpURLConnection class:* using the below video as a close guide: [13 - JSON Parsing, NetworkOnMainThreadException, AsyncTask - Android Studio](https://www.youtube.com/watch?v=_7r_vdwmW0o)

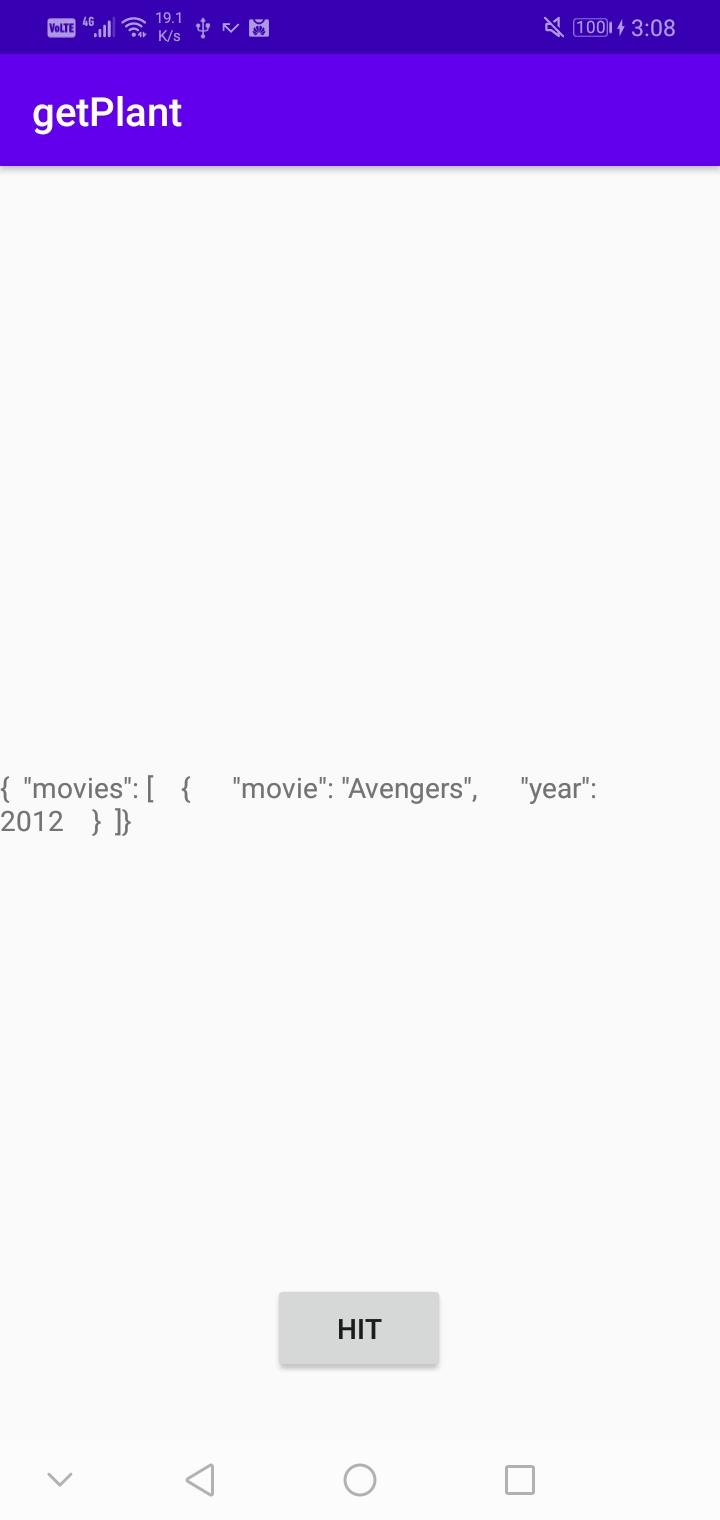
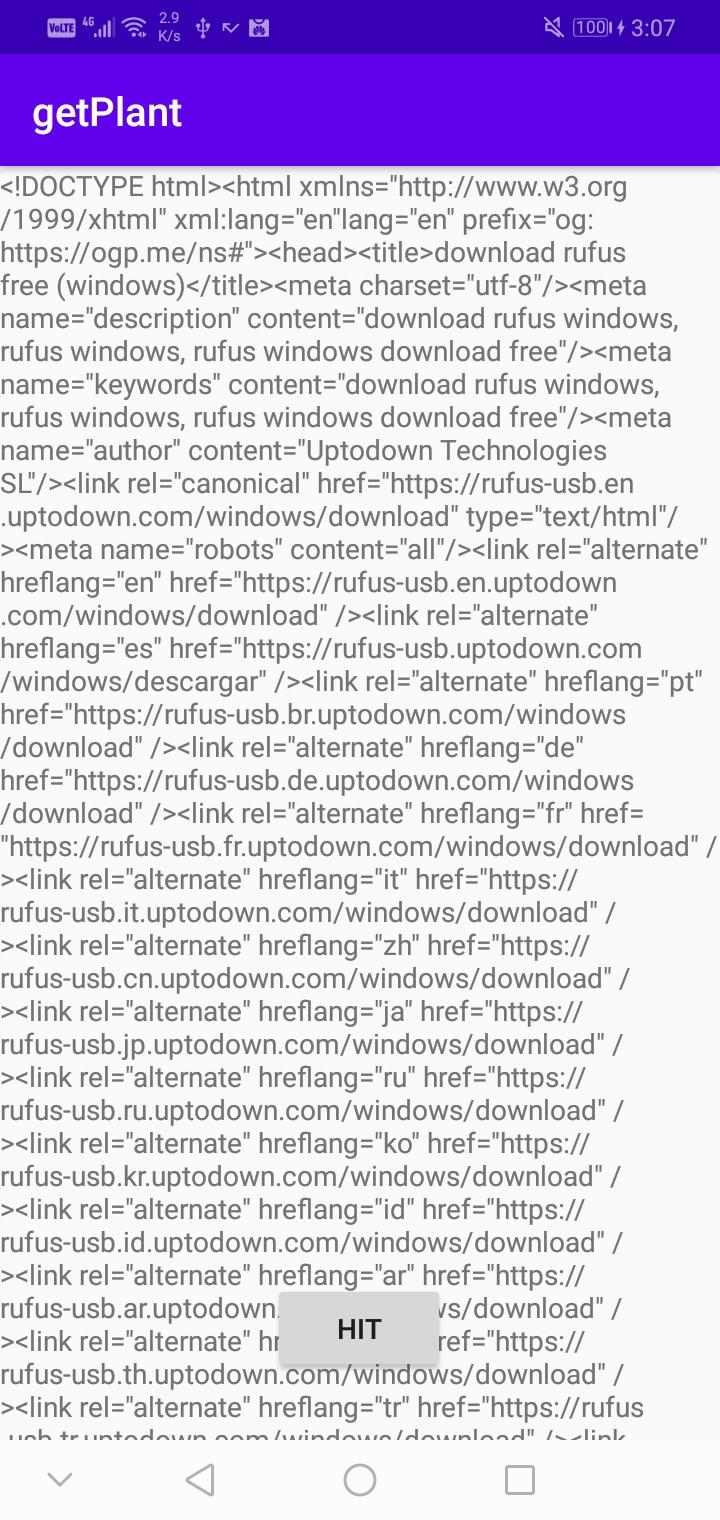
The **goal** was to ,upon request, **secure a connection between the Android Application** and a specified hard-coded website.

The primary goal was to pull JSON data from the website and display it in a text field for the time being.

**Pulling from the following websites:**

**https://jsonparsingdemo-cec5b.firebaseapp.com/jsonData/moviesDemoItem.txt**

**https://rufus-usb.en.uptodown.com/windows/download**

**Respectfully yield the following results:**

The java application that fosters this connection is located on the shared drive, the file name (regrettably) is: ***getPlant***

*Now that a viable link is being established in a trial application, with having information received from a website, the forwarding of information is the next critical aspect of our application*

Instances of the URLConnection\_class can be used both to read from and to write to the resource referenced by the URL.

How the Connection works can be explored more in detail here:

[URLConnection](https://developer.android.com/reference/java/net/URLConnection)

[Reading from and Writing to a URLConnection (The Java™ Tutorials > Custom Networking > Working with URLs)](https://docs.oracle.com/javase/tutorial/networking/urls/readingWriting.html)

[HttpURLConnection: Android Tutorial](https://blog.codavel.com/how-to-integrate-httpurlconnection)

Writing to a URL is often called ***posting*** *to a URL*. The server recognizes the POST request and reads the data sent from the client.

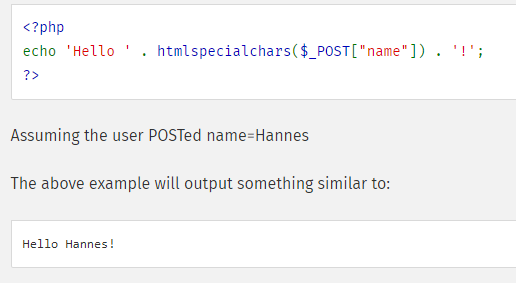
For a Java program to interact with a server-side process it simply must be able to write to a URL, thus providing data to the server.

Our Netbeans Java Project URLconnectionTesting further explores the transfer of information between a java application and a web-server.

Upon writing to various website and receiving multiple error codes such as: **405 and 500**

***500***is a very general HTTP status code that means **something unspecified has gone wrong** on the web site's server. And ***405*** is **Method Not Allowed**:

It seems (upon initial reaction) that the web servers are not designed to handle a request of the type being sent. For further testing, connection needs to be established with a local PHP file so that a post response can be received and tailored.



PHP Testing file:

<https://www.php.net/manual/en/reserved.variables.post.php>

<https://stackoverflow.com/questions/30440349/accessing-a-local-file-java-url>

+