Android Networking API

In **Android**, Network programming includes a request to the server and retrieve the data returned. Basically you have two APIs for working with the network:

Apache HttpClient:

- This is an open source library provided by the **Apache**.

HttpURLConnection

- This is an official API of Android, it began to be included in the version of Android 2.3, in the previous, Android using **Apache HttpClient** to work with the network.

You need to grant some permissions to the application if you want to work with the network.

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

android.permission.INTERNET:

- Add this permission, allowing your application will have the ability to connect to the network.

android.permission.ACCESS_NETWORK_STATE:

- Allows the app to check the status of your network connection.

The following code checks the network connection status:

```
private boolean checkInternetConnection() {

    ConnectivityManager connManager =
            (ConnectivityManager)
this.getSystemService(Context.CONNECTIVITY_SERVICE);


    NetworkInfo networkInfo = connManager.getActiveNetworkInfo();

    if (networkInfo == null) {
        Toast.makeText(this, "No default network is currently active",
Toast.LENGTH_LONG).show();
        return false;
    }

    if (!networkInfo.isConnected()) {
        Toast.makeText(this, "Network is not connected",
Toast.LENGTH_LONG).show();
        return false;
    }

    if (!networkInfo.isAvailable()) {
        Toast.makeText(this, "Network not available",
Toast.LENGTH_LONG).show();
        return false;
    }
    Toast.makeText(this, "Network OK", Toast.LENGTH_LONG).show();
    return true;
 }
```

➢ FOLLOWING ARE FEW LIB FOR NETWORKING TASK

1> Volley (Google)

2> OkHTTP (Square)

3> Retrofit (Square)

4> Robospice (Octo)

5> Picasso (Square)

6> ION (Clockworkmod)

7> Spring (Pivotal)

❖ Sample Code To Check Connectivity

```
ConnectivityManager cm = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo netInfo = cm.getActiveNetworkInfo();
if (netInfo != null && netInfo.isConnectedOrConnecting())
{        return true;
}
```

❖ Accessing the Internet (HTTP) :-

The most common way to transfer data to and from the network is to use HTTP. You can use HTTP to encapsulate almost any type of data and to secure the data with Secure Sockets Layer (SSL), which can be important when you transmit data that falls under privacy requirements. Also, most common ports used by HTTP are typically open from the phone networks.

❖ Parsing XML from the Network :-

A large portion of data transmitted between network resources is stored in a structured fashion in Extensible Markup Language (XML).

In particular, RSS feeds are provided in a standardized XML format, and many web services provide data using these feeds.

Parsing XML from the network is similar to parsing an XML resource file or a raw file on the file system.Android provides a fast and efficient XML Pull Parser, which is a parser of choice for networked applications.