* **Using Android Networking APIs**

Android lets your application connect to the internet or any other local network and allows you to perform network operations.

A device can have various types of network connections. This chapter focuses on using either a Wi-Fi or a mobile network connection.

**Checking Network Connection**

Before you perform any network operations, you must first check that are you connected to that network or internet e.t.c. For this android provides **ConnectivityManager** class. You need to instantiate an object of this class by calling **getSystemService()** method. Its syntax is given below −

ConnectivityManager check = (ConnectivityManager)

this.context.getSystemService(Context.CONNECTIVITY\_SERVICE);

Once you instantiate the object of ConnectivityManager class, you can use **getAllNetworkInfo** method to get the information of all the networks. This method returns an array of **NetworkInfo**. So you have to receive it like this.

NetworkInfo[] info = check.getAllNetworkInfo();

The last thing you need to do is to check **Connected State** of the network. Its syntax is given below –

for (int i = 0; i<info.length; i++){

if (info[i].getState() == NetworkInfo.State.CONNECTED){

Toast.makeText(context, "Internet is connected

Toast.LENGTH\_SHORT).show();

}

}

Apart from this connected states, there are other states a network can achieve. They are listed below –

|  |  |
| --- | --- |
| **Sr.No** | **State** |
| 1 | Connecting |
| 2 | Disconnected |
| 3 | Disconnecting |
| 4 | Suspended |
| 5 | Unknown |

Performing Network Operations

After checking that you are connected to the internet, you can perform any network operation. Here we are fetching the html of a website from a url.AD

Android provides **HttpURLConnection** and **URL** class to handle these operations. You need to instantiate an object of URL class by providing the link of website. Its syntax is as follows −

String link = "http://www.google.com";

URL url = new URL(link);

After that you need to call **openConnection** method of url class and receive it in a HttpURLConnection object. After that you need to call the **connect** method of HttpURLConnection class.

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

conn.connect();

And the last thing you need to do is to fetch the HTML from the website. For this you will use **InputStream** and **BufferedReader** class. Its syntax is given below −

InputStream is = conn.getInputStream();

BufferedReader reader = new BufferedReader(new InputStreamReader(is, "UTF-8"));

String webPage = "",data="";

while ((data = reader.readLine()) != null){

webPage += data + "\n";

}

Apart from this connect method, there are other methods available in HttpURLConnection class. They are listed below −

|  |  |
| --- | --- |
| **Sr.No** | **Method & description** |
| 1 | **disconnect()**  This method releases this connection so that its resources may be either reused or closed |
| 2 | **getRequestMethod()**  This method returns the request method which will be used to make the request to the remote HTTP server |
| 3 | **getResponseCode()**  This method returns response code returned by the remote HTTP server |
| 4 | **setRequestMethod(String method)**  This method Sets the request command which will be sent to the remote HTTP server |
| 5 | **usingProxy()**  This method returns whether this connection uses a proxy server or not |