

```
package com.example.covid_19alertapp.extras;
```

```
import android.content.Context;
```

```
import android.net.ConnectivityManager;
```

```
import android.net.Network;
```

```
import android.net.NetworkInfo;
```

```
import android.net.NetworkRequest;
```

```
import android.os.Build;
```

```
import androidx.annotation.RequiresApi;
```

```
public abstract class Internet {
```

```
    public static boolean isInternetAvailable(Context context){
```

```
        /*
```

```
        this method is deprecated for API 29
```

```
        use for one time network availability check
```

```
        */
```

```
        ConnectivityManager cm =
```

```
            (ConnectivityManager)context.getSystemService(Context.CONNECTIVITY_SERVICE);
```

```
        NetworkInfo activeNetwork = cm.getActiveNetworkInfo();
```

```
        return activeNetwork != null &&
```

```

        activeNetwork.isConnectedOrConnecting();
    }

    /*
    better newer method with callbacks
    works with API 24(or 29?) and above
    */

    private static Context context;

    public static boolean isNetworkConnected;

    public void CheckNetwork(Context context) {
        this.context = context;
    }

    // Network Check

    @RequiresApi(api = Build.VERSION_CODES.N)
    public void registerNetworkCallback()
    {
        try {

            ConnectivityManager connectivityManager = (ConnectivityManager)
context.getSystemService(Context.CONNECTIVITY_SERVICE);

            NetworkRequest.Builder builder = new NetworkRequest.Builder();

            connectivityManager.registerDefaultNetworkCallback(new
ConnectivityManager.NetworkCallback(){

```

```

@Override

public void onAvailable(Network network) {

    isNetworkConnected = true; // Global Static Variable

}

@Override

public void onLost(Network network) {

    isNetworkConnected = false; // Global Static Variable

}

}

);

isNetworkConnected = false;

}catch (Exception e){

    isNetworkConnected = false;

}

}

public void unRegisterNetworkCallBack(){

    // find out

}

}

```