Android Developers

Determining and Monitoring the Connectivity Status

Some of the most common uses for repeating alarms and background services is to schedule regular updates of application data from Internet resources, cache data, or execute long running downloads. But if you aren't connected to the Internet, or the connection is too slow to complete your download, why bother waking the device to schedule the update at all?

You can use the `ConnectivityManager` *(https://developer.android.com /reference/android/net/ConnectivityManager.html)* to check that you're actually connected to the Internet, and if so, what type of connection is in place.

This lesson teaches you to

Determine if you Have an Internet Connection

Determine the Type of your Internet Connection

Monitor for Changes in Connectivity

You should also read

Intents and Intent Filters

Determine if You Have an Internet Connection

There's no need to schedule an update based on an Internet resource if you aren't connected to the Internet. The following snippet shows how to use the `ConnectivityManager` *(https://developer.android.com/reference/android /net/ConnectivityManager.html)* to query the active network and determine if it has Internet connectivity.

```java
ConnectivityManager cm =
        (ConnectivityManager)context.getSystemService(Context.CONNECTIVITY_SERVICE);

NetworkInfo activeNetwork = cm.getActiveNetworkInfo();
boolean isConnected = activeNetwork != null &&
                        activeNetwork.isConnectedOrConnecting();
```

Determine the Type of your Internet Connection

It's also possible to determine the type of Internet connection currently available.

Device connectivity can be provided by mobile data, WiMAX, Wi-Fi, and ethernet connections. By querying the type of the active network, as shown below, you can alter your refresh rate based on the bandwidth available.

```java
boolean isWiFi = activeNetwork.getType() == ConnectivityManager.TYPE_WIFI;
```

Mobile data costs tend to be significantly higher than Wi-Fi, so in most cases, your app's update rate should be lower when on mobile connections. Similarly, downloads of significant size should be suspended until you have a Wi-Fi connection.

Having disabled your updates, it's important that you listen for changes in connectivity in order to resume them once an Internet connection has been established.

Monitor for Changes in Connectivity

Apps targeting Android 7.0 (API level 24) and higher do not receive `CONNECTIVITY_ACTION` *(https://developer.android.com/reference/android/net/ConnectivityManager.html#CONNECTIVITY_ACTION)* broadcasts if they declare the broadcast receiver in their manifest. Apps will still receive `CONNECTIVITY_ACTION` *(https://developer.android.com/reference/android/net/ConnectivityManager.html#CONNECTIVITY_ACTION)* broadcasts if they register their `BroadcastReceiver` *(https://developer.android.com/reference/android/content/BroadcastReceiver.html)*

with `Context.registerReceiver()` (https://developer.android.com/reference/android/content/Context.html#registerReceiver(android.content.BroadcastReceiver, android.content.IntentFilter)) and that context is still valid.