



# Connectivity

*Detecting Internet connectivity on iOS*

# Reachability

Used by iOS developers as the de facto means of checking whether an Internet connection is available.

*"A remote host is considered reachable when a data packet, sent by an application into the network stack, can leave the local device. Reachability does not guarantee that the data packet will actually be received by the host."*

- Apple Developer Documentation

# Captive Portals

Consider an iOS device connected to a router disconnected from the Internet.

*"A captive portal is a web page which is displayed to newly connected user before they are granted broader access to network resources."*

- Wikipedia

# iOS

Contacts a number of known URLs hosting a small webpage:

e.g. <http://www.apple.com/library/test/success.html>

```
<HTML><HEAD><TITLE>Success</TITLE></  
HEAD><BODY>Success</BODY></HTML>
```

If the content cannot be retrieved then a captive portal is blocking the connection. iOS can present the web page for user interaction to access the network.

However, iOS does not provide an API for this functionality.

# Connectivity

- Provides a familiar API similar to Reachability.
- Emulates the behaviour of iOS by contacting a set of URLs and checking the response when WiFi available.
- Connectivity URLs & percentage of successful connectivity checks may be specified.
- Polling option available for situations where a persistent connection is required.

# API

```
let connectivity: Connectivity = Connectivity()

connectivity.connectivityURLs = [URL(string: "https://www.apple.com/library/
test/success.html")!]

let connectivityChanged: (Connectivity) -> Void = { [weak self] connectivity in
    self?.updateConnectionStatus(connectivity.status)
}

connectivity.whenConnected = connectivityChanged
connectivity.whenDisconnected = connectivityChanged
connectivity.startNotifier()
connectivity.stopNotifier()
```

# API

```
func updateConnectionStatus(_ status:
Connectivity.ConnectivityStatus) {

    switch status {
    case .connectedViaWiFi:
    case .connectedViaWiFiWithoutInternet:
    case .connectedViaWWAN:
    case .connectedViaWWANWithoutInternet:
    case .notConnected:

    }

}
```

# Connectivity

<https://github.com/rwbutler/Connectivity>

- Cocoapods

```
pod "Connectivity"
```

- Carthage

```
github "rwbutler/Connectivity"
```



# References

- Beyond Reachability - Detecting true Internet Connectivity on iOS
  - <https://medium.com/@rwbutler/beyond-reachability-detecting-true-internet-connectivity-on-ios-928da1b60122>
- Captive Portal - Wikipedia
  - [https://en.wikipedia.org/wiki/Captive\\_portal](https://en.wikipedia.org/wiki/Captive_portal)
- SCNetworkReachability - Apple Developer Documentation
  - <https://developer.apple.com/documentation/systemconfiguration/scnetworkreachability-g7d>