Developer Forums   Search by keywords or tags   ?   Ask

NEHotspotHelper pseudo API

👁 907

I've been playing around with NEHotspotHelper recently and one of the things I've found is that, because the API is effectively parameter-block based, it's not always clear which commands take which parameters. To get this sorted in my head I created a pseudo API that shows all of the commands with their parameters and function results tightly constrained. The result is pasted in below.

For example, from the NEHotspotHelper documentation and headers, it's not obvious that the

```
.Evaluate
```

command is allowed to do networking but the

```
.FilterScanList
```

command is not. In my interface, however, that is obvious because

```
filter(_:)
```

callback does not get a

```
context
```

parameter.

IMPORTANT This is not meant to be a usable API (for example, in a real app all of the callbacks are asynchronous), but rather a compact representation of the interface between the hotspot subsystem and your helper.

Share and Enjoy — Quinn "The Eskimo!" Apple Developer Relations, Developer Technical Support, Core OS/Hardware

```
let myEmail = "eskimo" + "1" + "@apple.com"
```

```swift
protocol HotspotDispatcherDelegate {
    func filter(networkList: [NEHotspotNetwork]) -> [FilterResult]
    func evaluate(context context: Context) -> NEHotspotHelperConfidence
        // .High or .Low yields the "Captive" transition in the authentication state machine
        // .None yields the "Not Captive" transition
    func authenticate(context context: Context) -> AuthenticateResult
    func presentUI(context context: Context) -> PresentUIResult
    func maintain(context context: Context) -> MaintainResult
    func logoff(context context: Context)
}

enum FilterConfidence {
    case low
    case high
}

struct FilterResult {
    let network: NEHotspotNetwork
    let confidence: FilterConfidence
    let password: String?
}

enum AuthenticateResult {
    case success
    case failure
    case temporaryFailure
    case uiRequired
    case unsupportedNetwork
}

enum PresentUIResult {
    case success
    case failure
    case temporaryFailure
    case unsupportedNetwork
}

enum MaintainResult {
    case success
    case failure
    case authenticationRequired
}

struct Context {
    func createTCPConnection(endpoint: NWEndpoint) -> NWTCPConnection
    func createUDPSession(endpoint: NWEndpoint) -> NWUDPSession
    func bindRequest(request: NSMutableURLRequest)
}
```

Network   Asked 4 years ago by eskimo 🍎 ↗

Reply to this question

Replies

This is great. I have been creating a spreadsheet of inputs and outputs on a per-command basis.

I guess it makes sense to not allow networking during FilterResult, as no network is bound. I would hope there would be more information on how to use the network in the evaluate phase. If create***Connection and bindRequest are the only APIs permitted on an interface that isn't published globally as "available", then I wish that was called out explicitly in the documentation with an example.

Posted 3 years ago by clamont ↗

 Developer   Apple Developer Forums

Discover	Design	Develop	Distribute	Support
macOS	Human Interface Guidelines	Xcode	Developer Program	Articles
iOS	Resources	Swift	App Store	Developer Forums
watchOS	Videos	Swift Playgrounds	App Review	Feedback & Bug Reporting
tvOS	Apple Design Awards	TestFlight	Mac Software	System Status
Safari and Web	Fonts	Documentation	Apps for Business	Contact Us
Games	Accessibility	Videos	Safari Extensions	
Business	Internationalization	Downloads	Marketing Resources	**Account**
Education	Accessories		Trademark Licensing	Certificates, Identifiers & Profiles
WWDC				App Store Connect

To view the latest developer news, visit News and Updates.