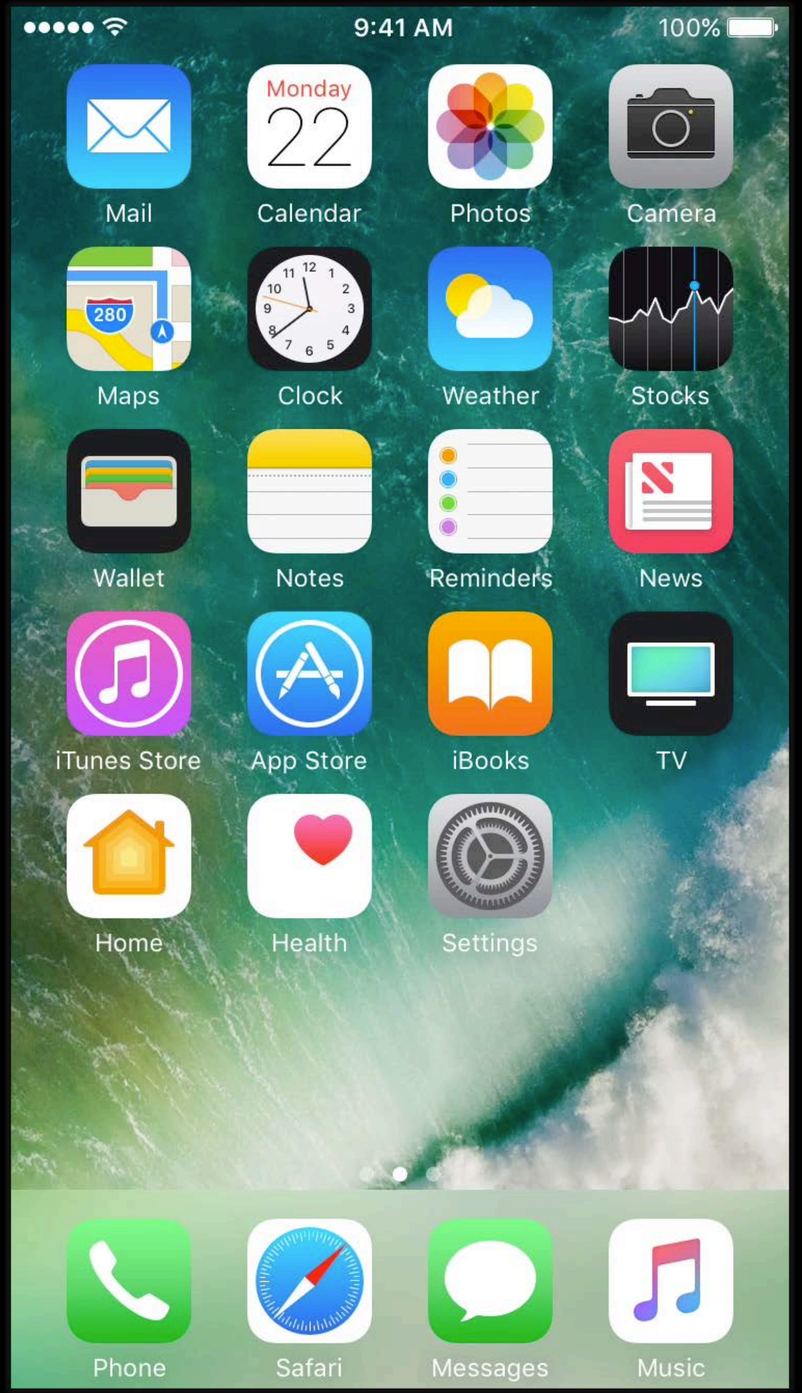
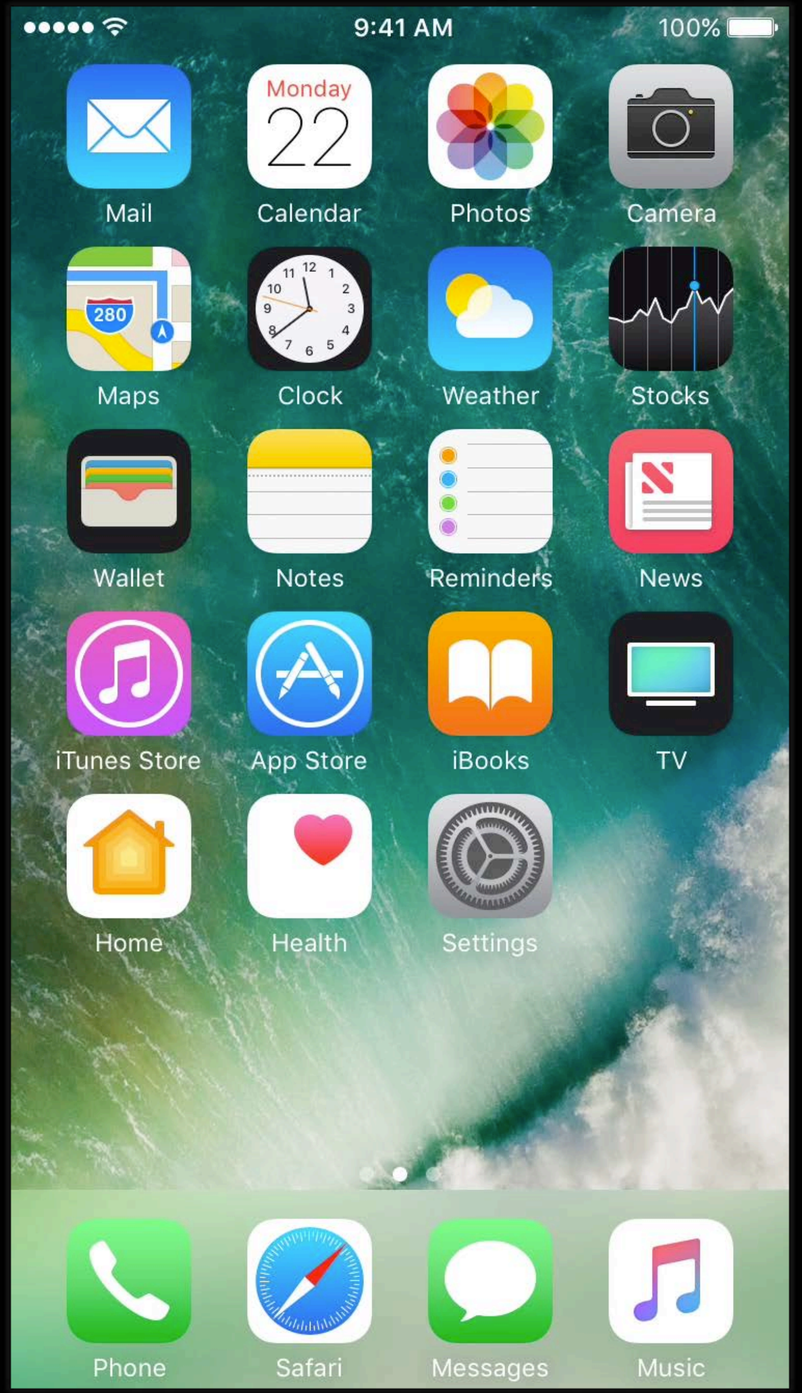
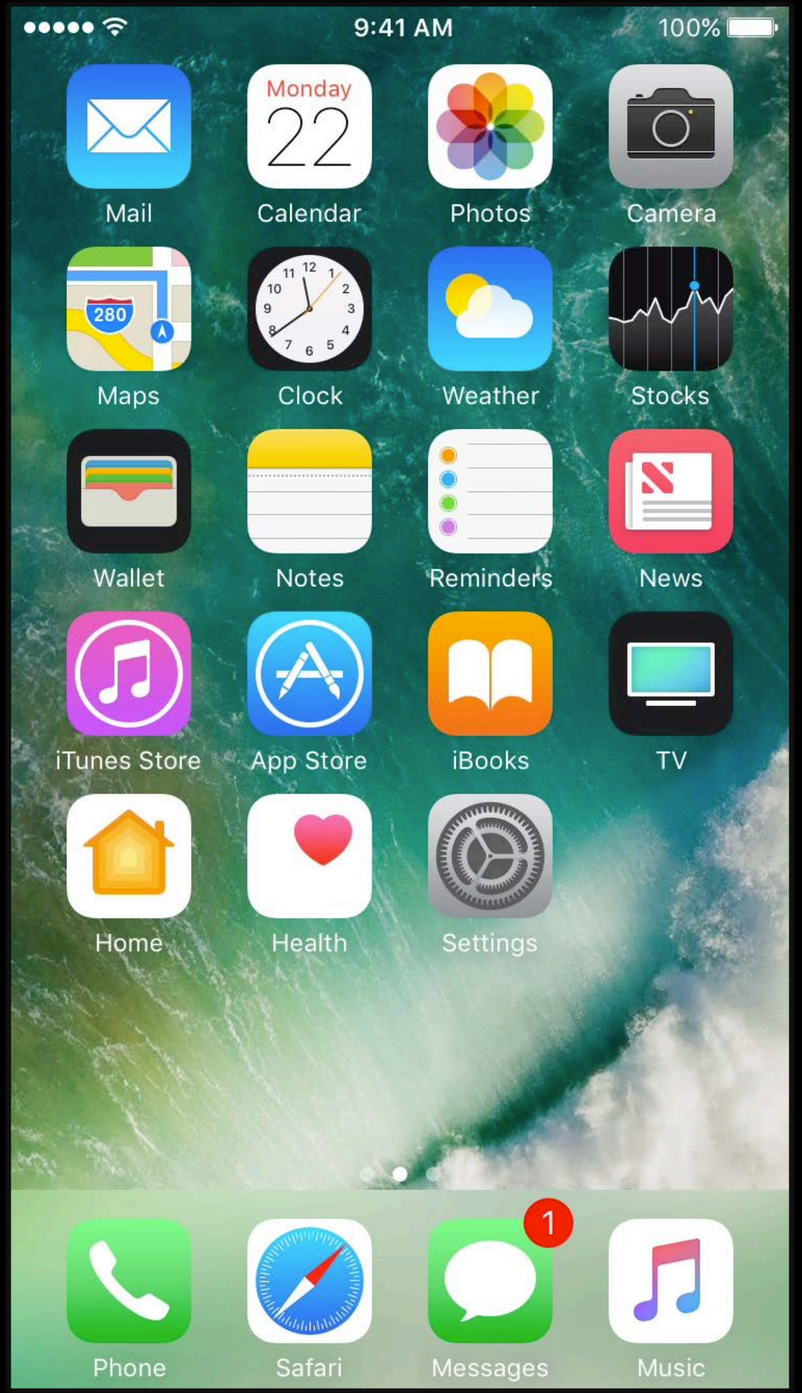
Filtering Unwanted Messages with Identity Lookup

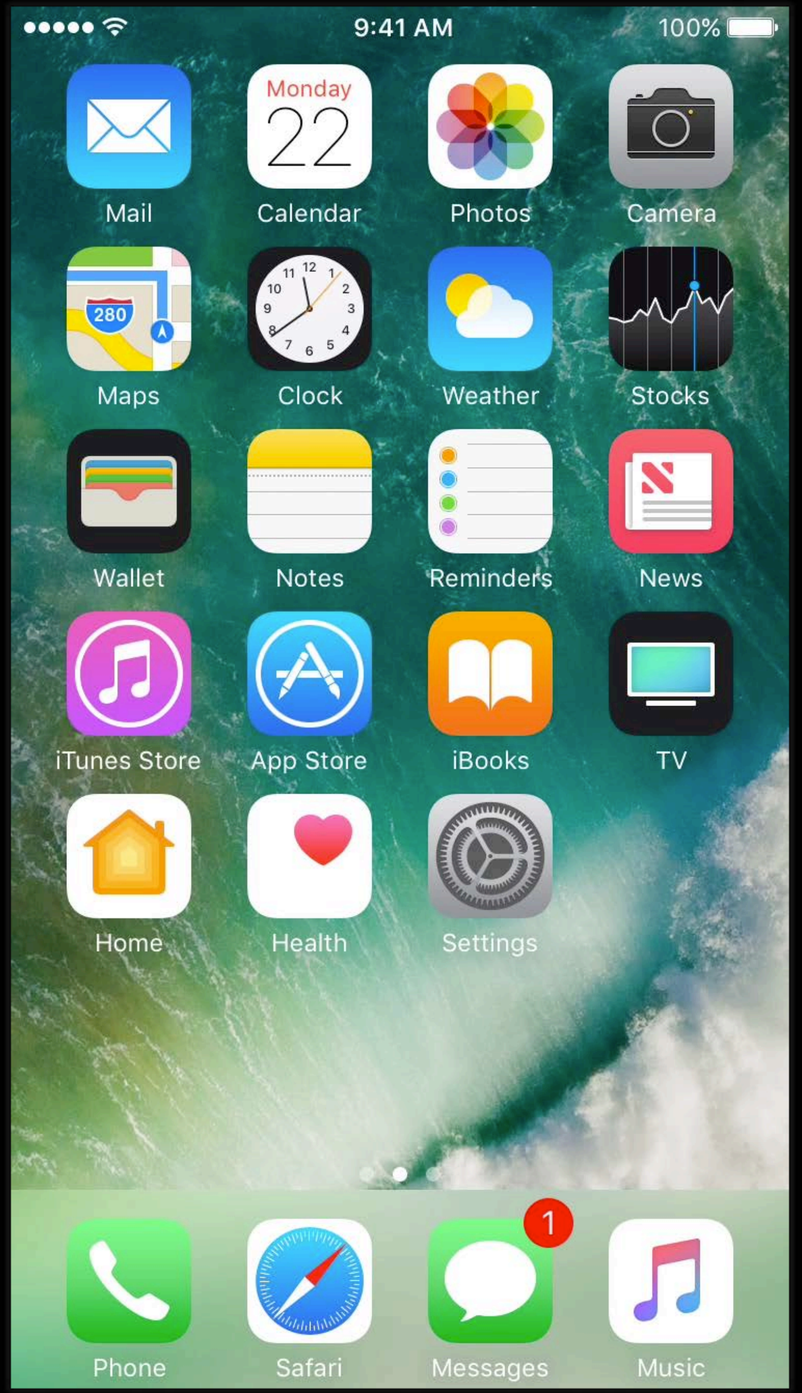
Session 249

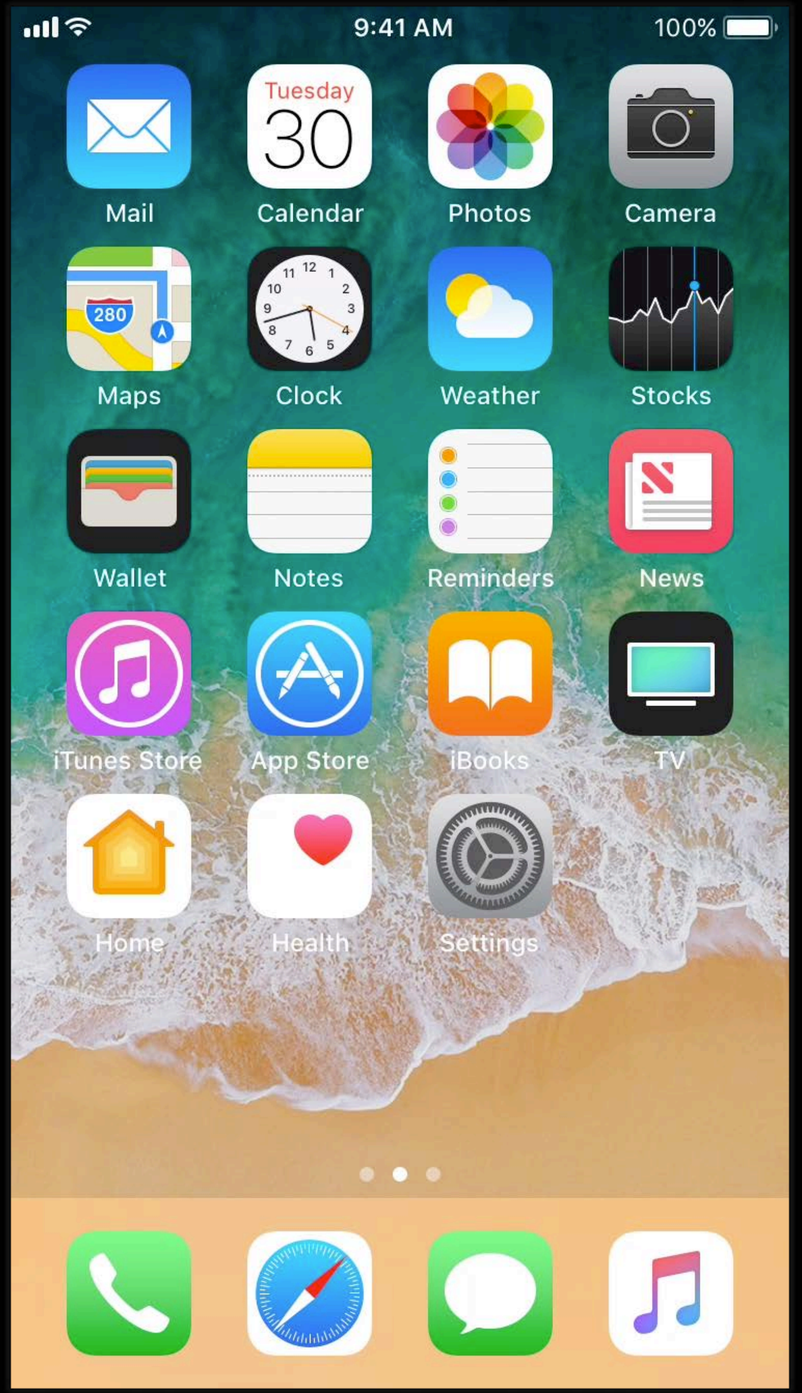
Stuart Montgomery, iOS Engineer

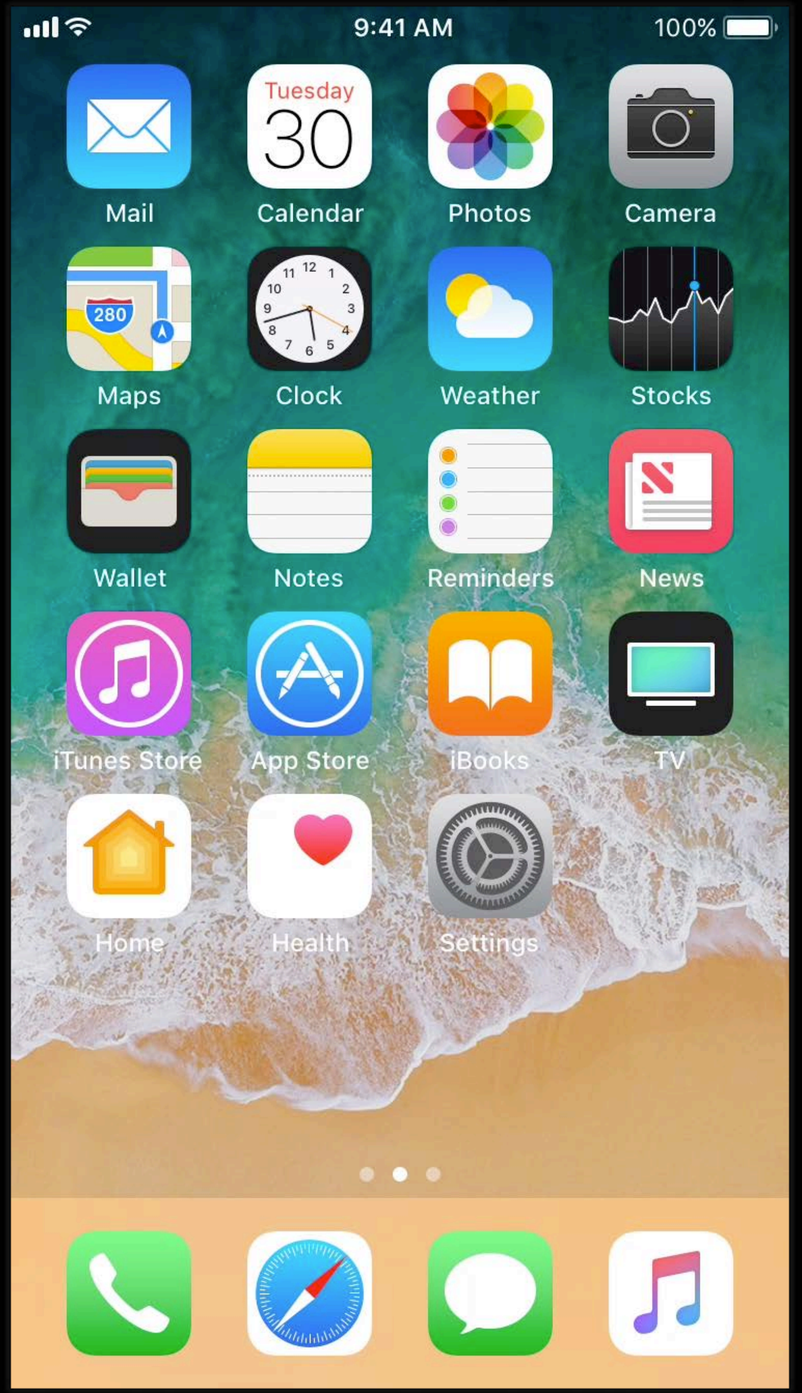

9:41 AM
100%
Mail
Monday
22
Calendar
Photos
Camera
280
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
Phone
Safari
Messages
Music

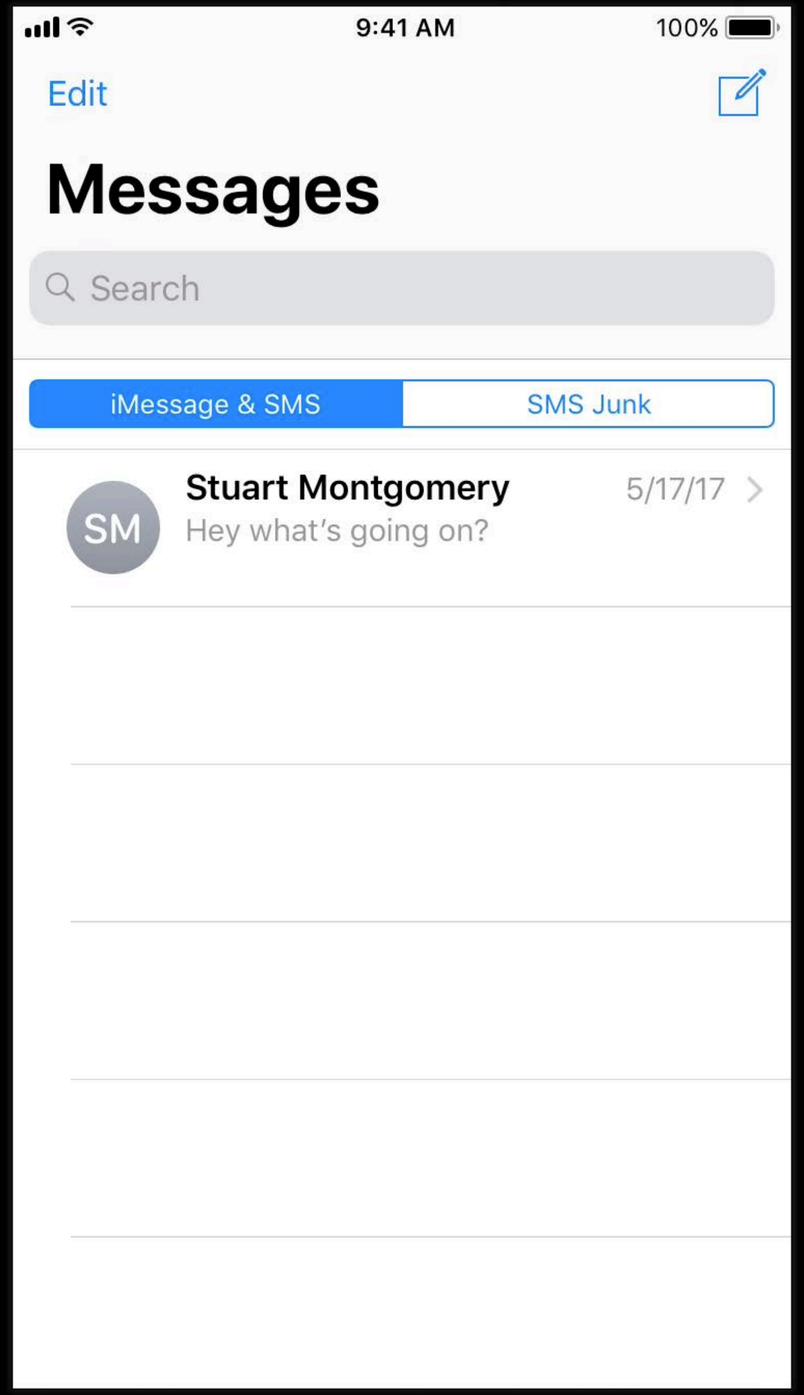

9:41 AM
100%
Mail
Monday
22
Calendar
Photos
Camera
280
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
Phone
Safari
Messages
Music

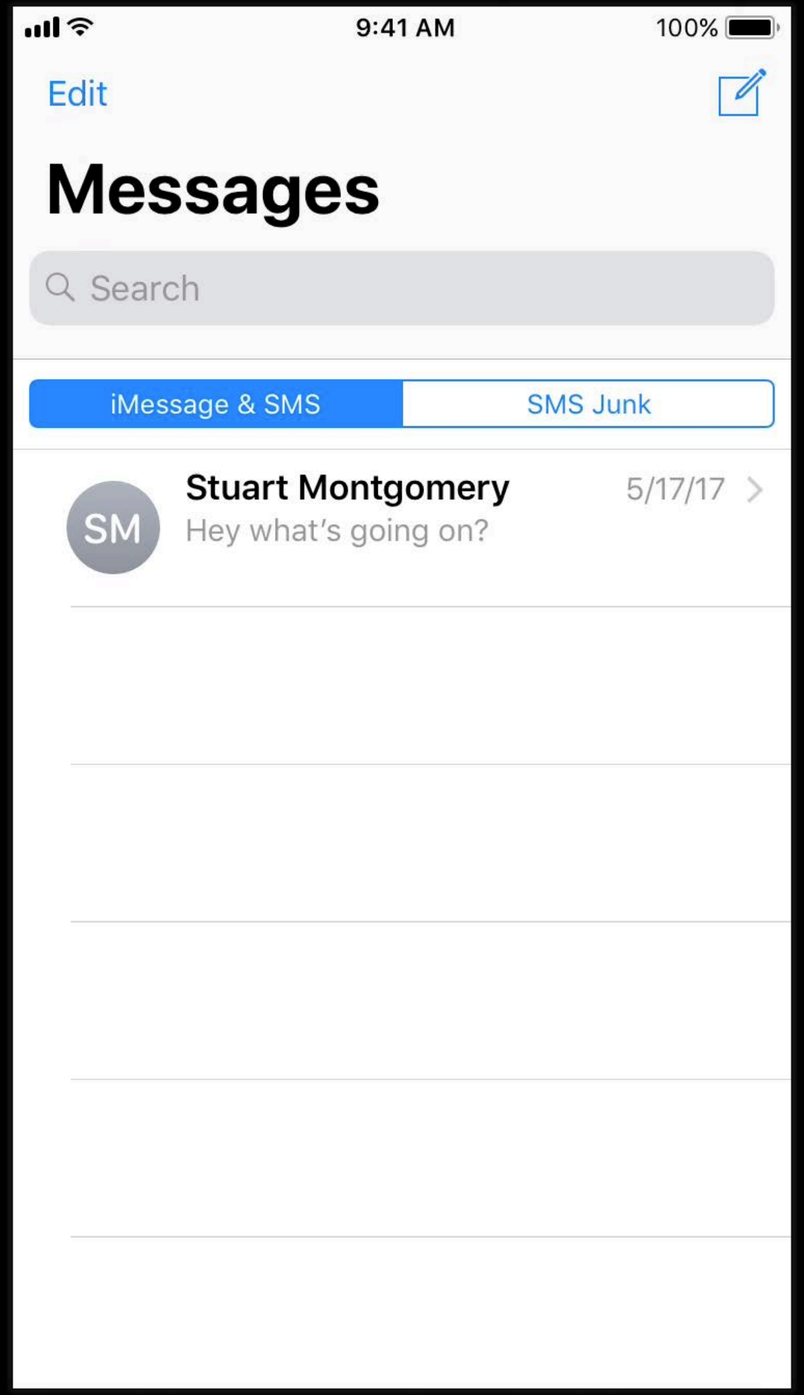

9:41 AM
100%
Mail
Monday
22
Calendar
Photos
Camera
280
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
1
Phone
Safari
Messages
Music

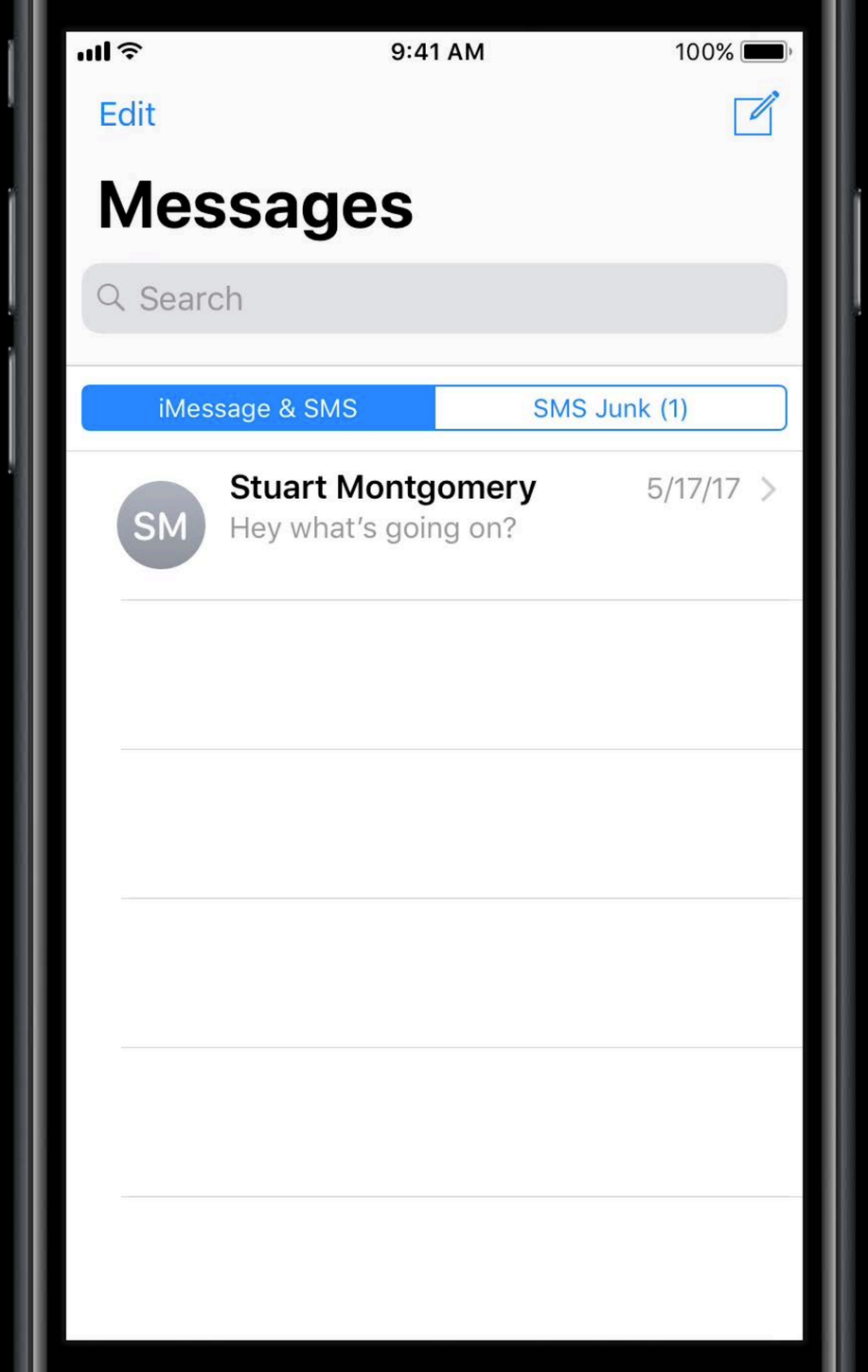

9:41 AM
100%
Mail
Monday
22
Calendar
Photos
Camera
280
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings
1
Phone
Safari
Messages
Music

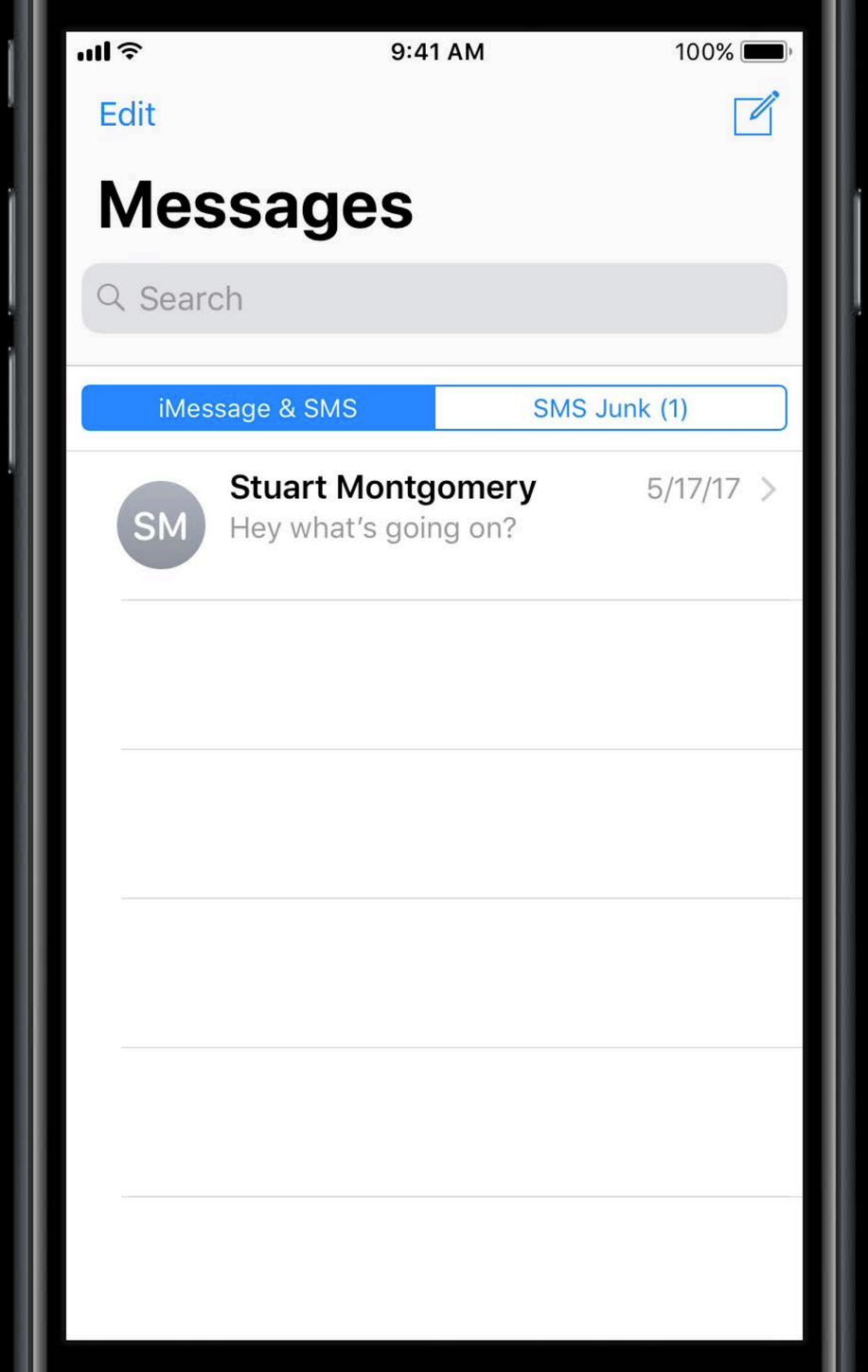

9:41 AM
100%
Mail
Tuesday
30
Calendar
Photos
Camera
280
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings

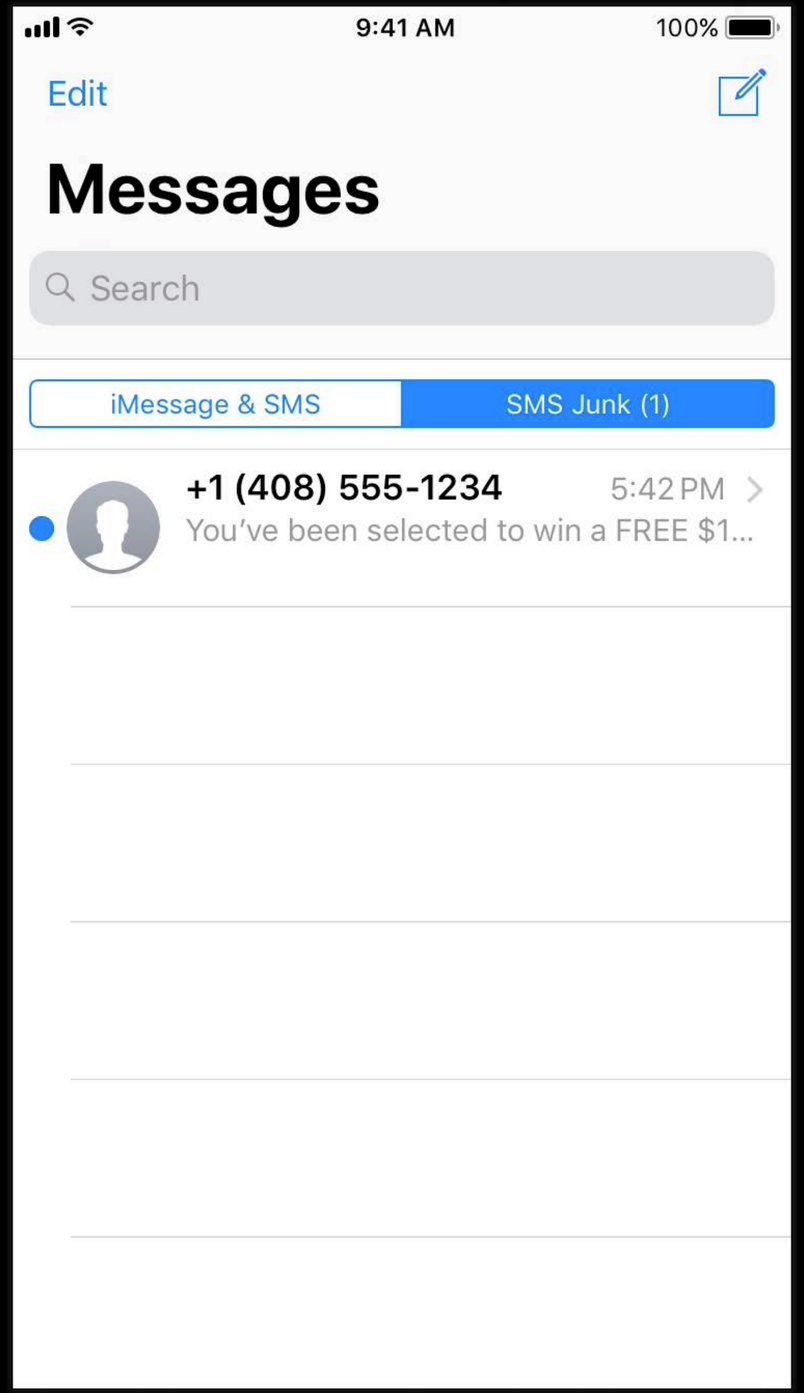

9:41 AM
100%
Mail
Tuesday
30
Calendar
Photos
Camera
280
Maps
Clock
Weather
Stocks
Wallet
Notes
Reminders
News
iTunes Store
App Store
iBooks
TV
Home
Health
Settings

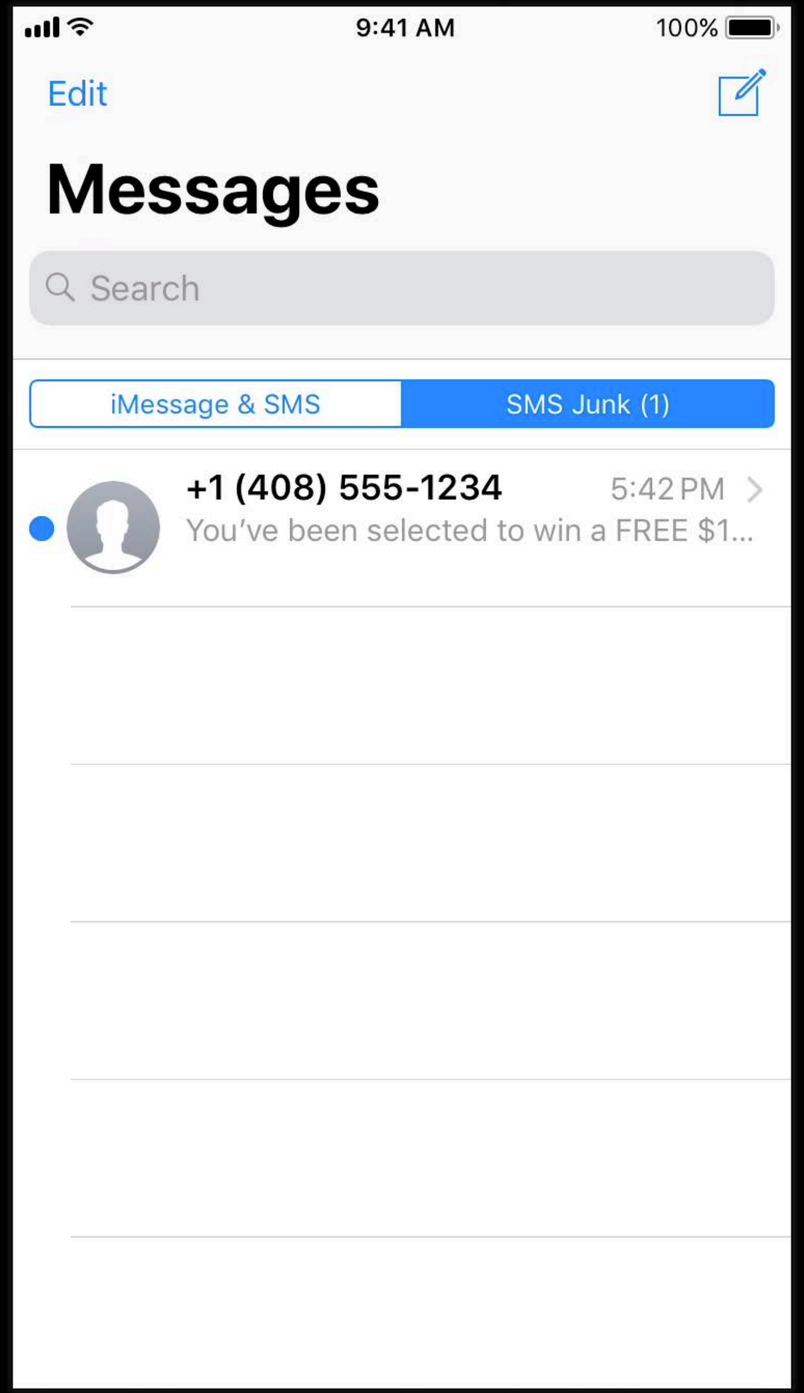

9:41 AM
100%
Edit
Messages
Search
iMessage & SMS
SMS Junk
SM
Stuart Montgomery
5/17/17
Hey what's going on?


9:41 AM
100%
Edit
Messages
Search
iMessage & SMS
SMS Junk
SM
Stuart Montgomery
5/17/17
Hey what's going on?


9:41 AM
100%
Edit
Messages
Search
iMessage & SMS
SMS Junk (1)
SM
Stuart Montgomery
5/17/17
Hey what's going on?


9:41 AM
100%
Edit
Messages
Search
iMessage & SMS
SMS Junk (1)
SM
Stuart Montgomery
5/17/17
Hey what's going on?


9:41 AM
100%
Edit
Messages
Search
iMessage & SMS
SMS Junk (1)
+1 (408) 555-1234
5:42 PM
You've been selected to win a FREE $1...


9:41 AM
100%
Edit
Messages
Search
iMessage & SMS
SMS Junk (1)
+1 (408) 555-1234
5:42 PM
You've been selected to win a FREE $1...

Motivation

Motivation

Increasing nuisance for users

Motivation

Increasing nuisance for users

Sometimes phishing attempts

Motivation

Increasing nuisance for users

Sometimes phishing attempts

Delivered via carrier, unlike iMessage

Motivation

Increasing nuisance for users

Sometimes phishing attempts

Delivered via carrier, unlike iMessage

Leverage expertise in detecting unwanted messages

Message Filter extensions

Privacy considerations

Network deferral

Demo

Message Filter extensions

Privacy considerations

Network deferral

Demo

Message Filter extensions

Privacy considerations

Network deferral

Demo

Message Filter extensions

Privacy considerations

Network deferral

Demo

Message Filter extensions

Privacy considerations

Network deferral

Demo

Message Filter Extensions

Message Filter Extensions

NEW

Message Filter Extensions

NEW

New app extension point

Message Filter Extensions

NEW

New app extension point

Part of Identity Lookup framework, new in iOS 11

Message Filter Extensions

NEW

New app extension point

Part of Identity Lookup framework, new in iOS 11

Must be enabled by user in Messages settings

Message Filter Extensions

NEW

New app extension point

Part of Identity Lookup framework, new in iOS 11

Must be enabled by user in Messages settings

One extension enabled at a time

Message Filter Extensions

NEW

New app extension point

Part of Identity Lookup framework, new in iOS 11

Must be enabled by user in Messages settings

One extension enabled at a time

Invoked for each SMS from an unknown sender

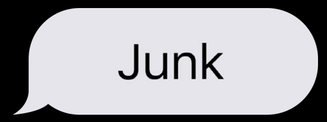
Message Filter Extensions

Offline-only

Messages

Message Filter Extensions

Offline-only



Messages

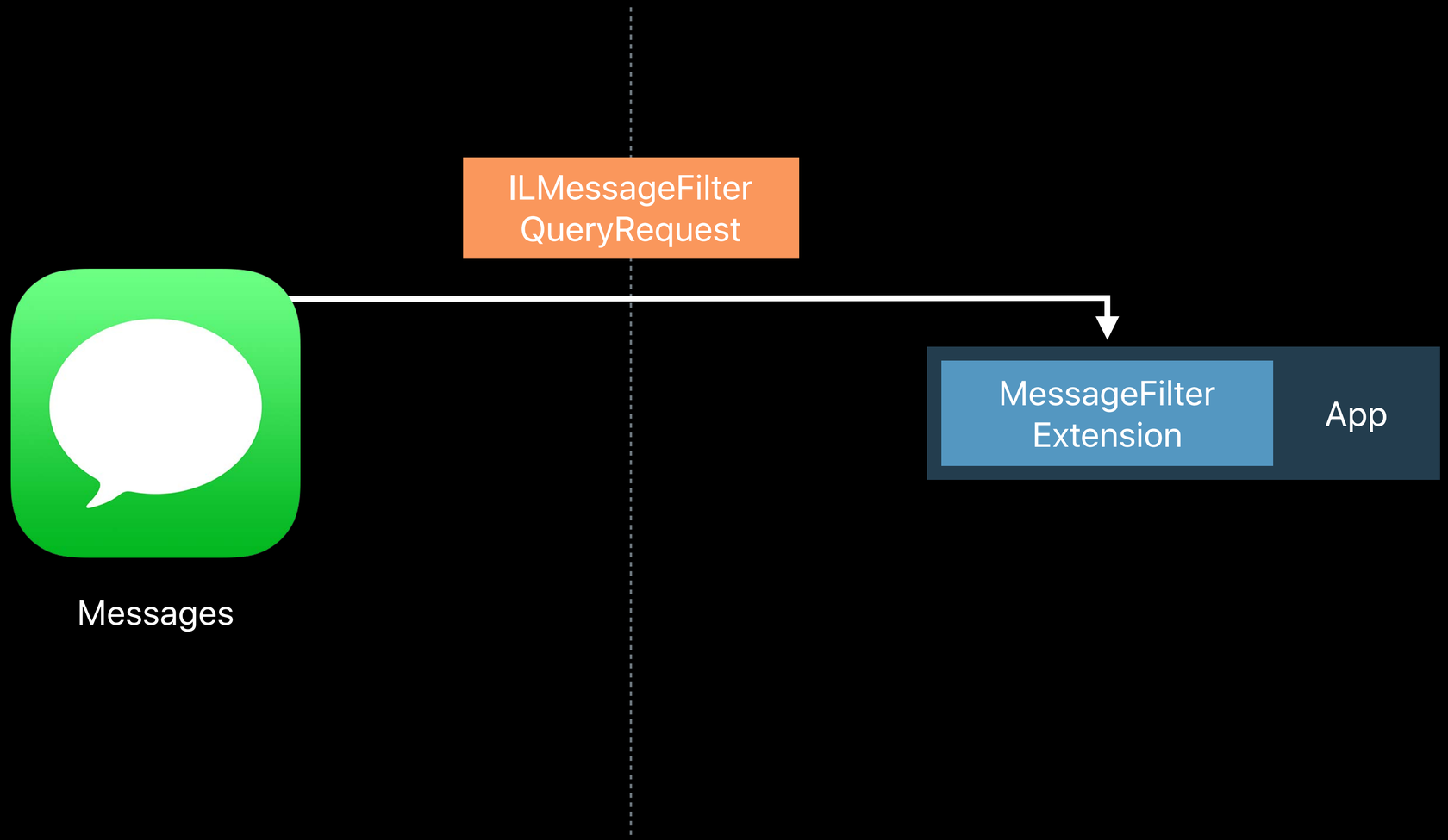
Message Filter Extensions

Offline-only

Messages

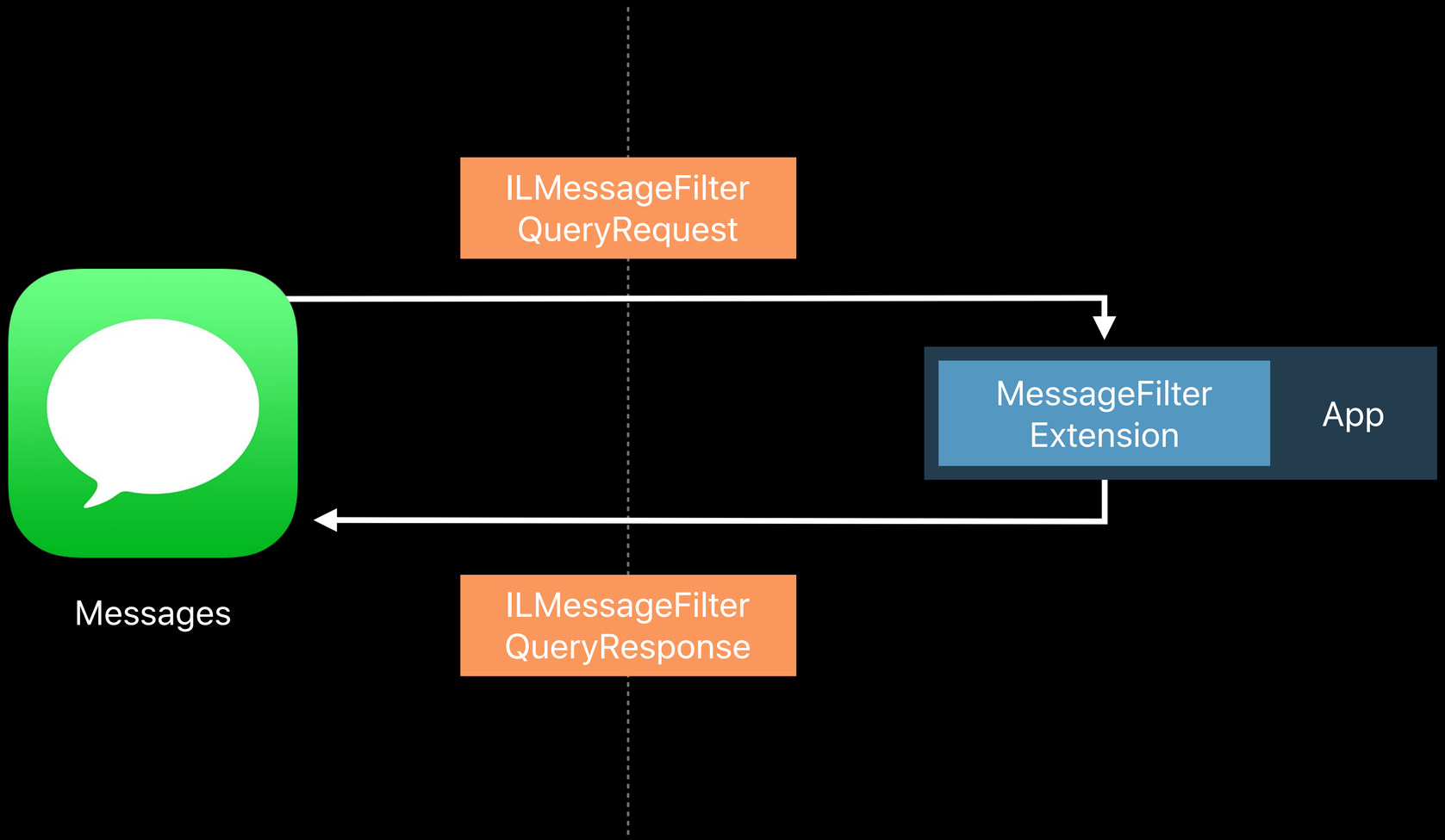
Message Filter Extensions

Offline-only



Message Filter Extensions

Offline-only



Privacy Considerations

Privacy Considerations

Privacy Considerations

Recipient’s phone number not sent to extension, only sender’s

Privacy Considerations

Recipient’s phone number not sent to extension, only sender’s

Restrictions on extension

Privacy Considerations

Recipient’s phone number not sent to extension, only sender’s

Restrictions on extension

- Cannot write to files shared with containing app

Privacy Considerations

Recipient’s phone number not sent to extension, only sender’s

Restrictions on extension

- Cannot write to files shared with containing app
- Cannot perform networking

Privacy Considerations

Recipient’s phone number not sent to extension, only sender’s

Restrictions on extension

- Cannot write to files shared with containing app
- Cannot perform networking

Do not export messages outside extension’s container

Message Criteria

Requirements for messages to be sent to extension

Message Criteria

Requirements for messages to be sent to extension

SMS and MMS only

Message Criteria

Requirements for messages to be sent to extension

SMS and MMS only

Never iMessage

Message Criteria

Requirements for messages to be sent to extension

SMS and MMS only

Never iMessage

Only unknown senders (not in recipient’s Contacts)

Message Criteria

Requirements for messages to be sent to extension

SMS and MMS only

Never iMessage

Only unknown senders (not in recipient’s Contacts)

After multiple responses

Message Criteria

Requirements for messages to be sent to extension

SMS and MMS only

Never iMessage

Only unknown senders (not in recipient’s Contacts)

After multiple responses

- Stops sending to extension

Message Criteria

Requirements for messages to be sent to extension

SMS and MMS only

Never iMessage

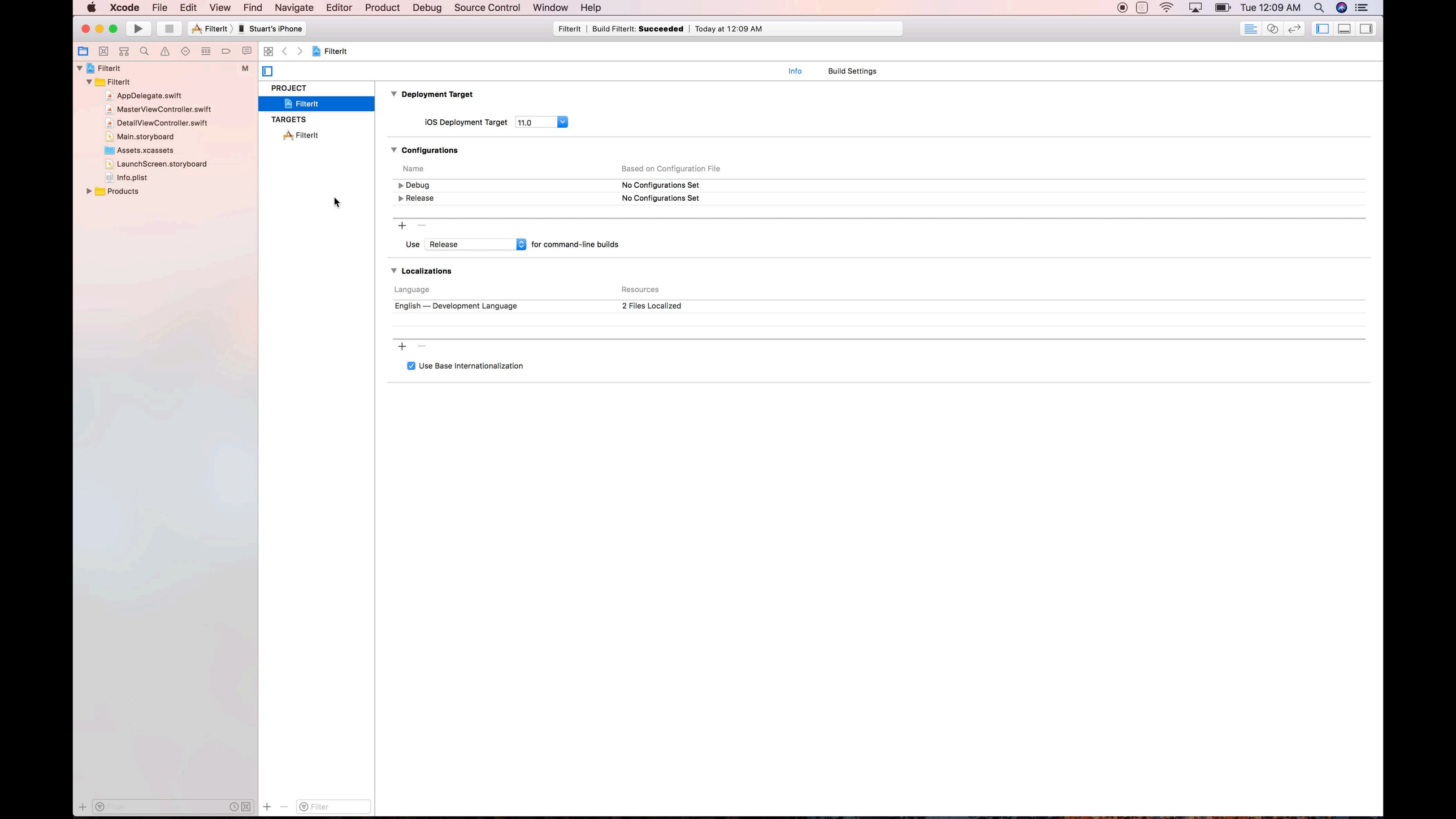
Only unknown senders (not in recipient’s Contacts)

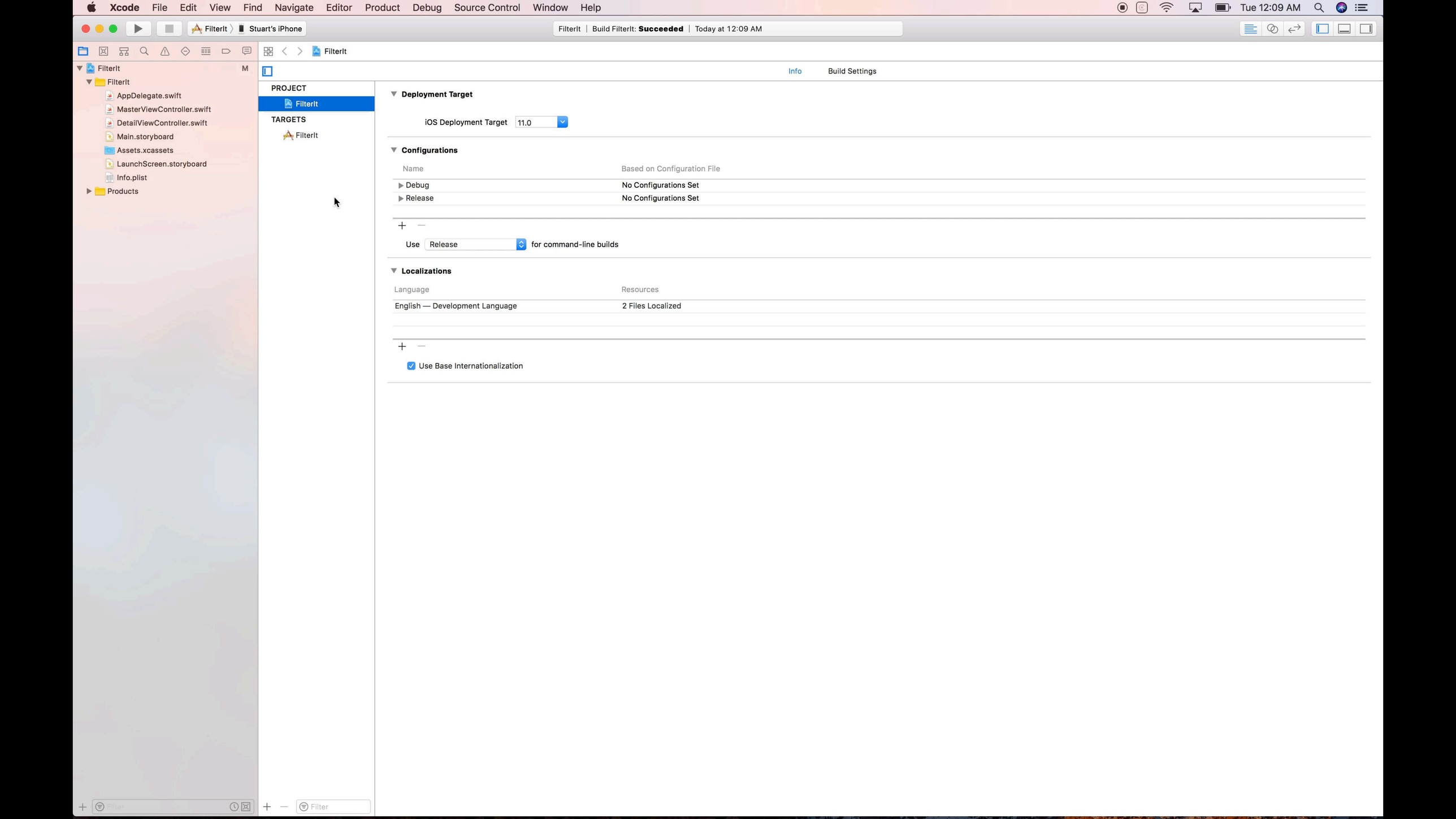
After multiple responses

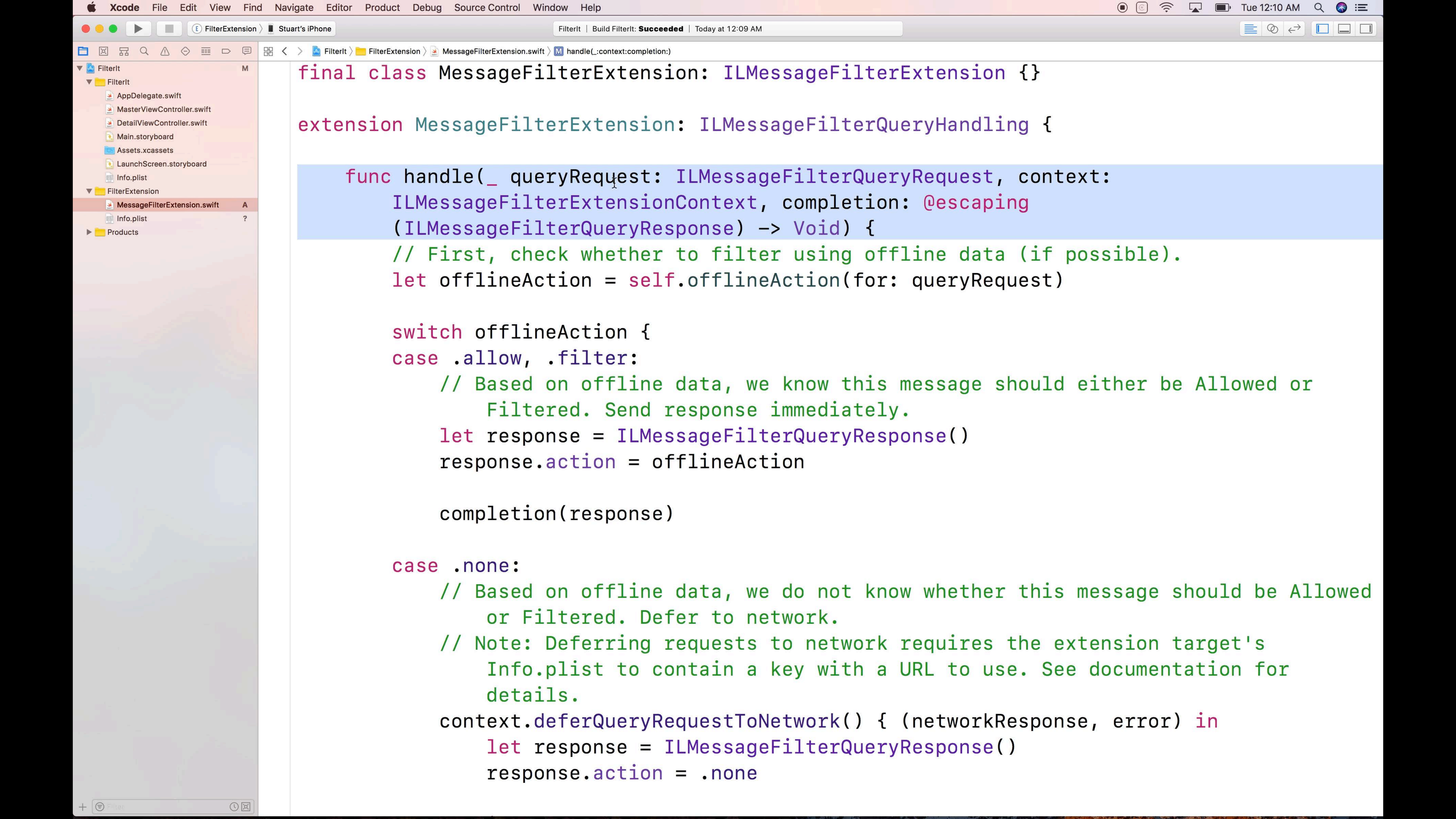
- Stops sending to extension
- Junk threads promoted to non-junk

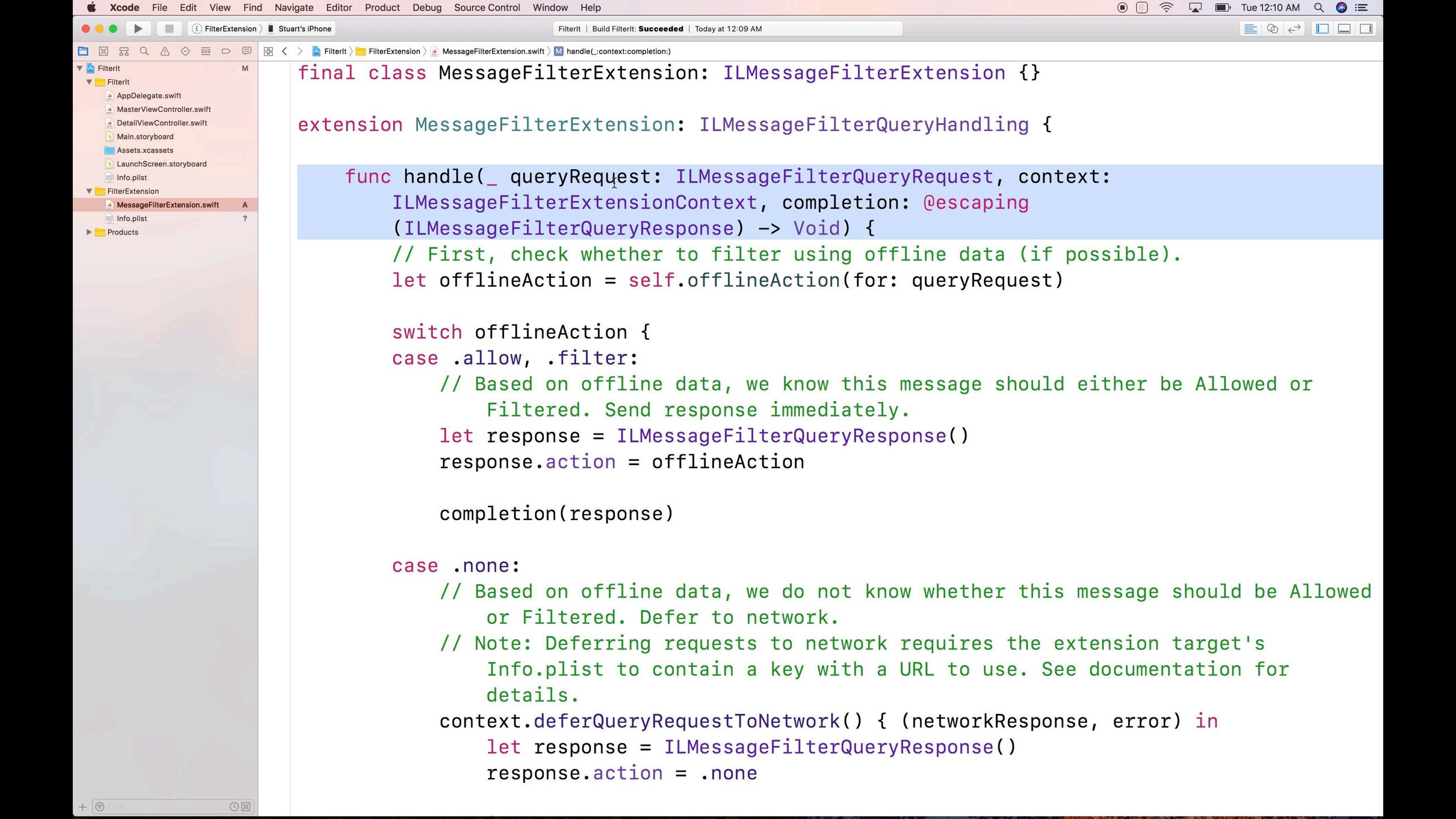
Demo

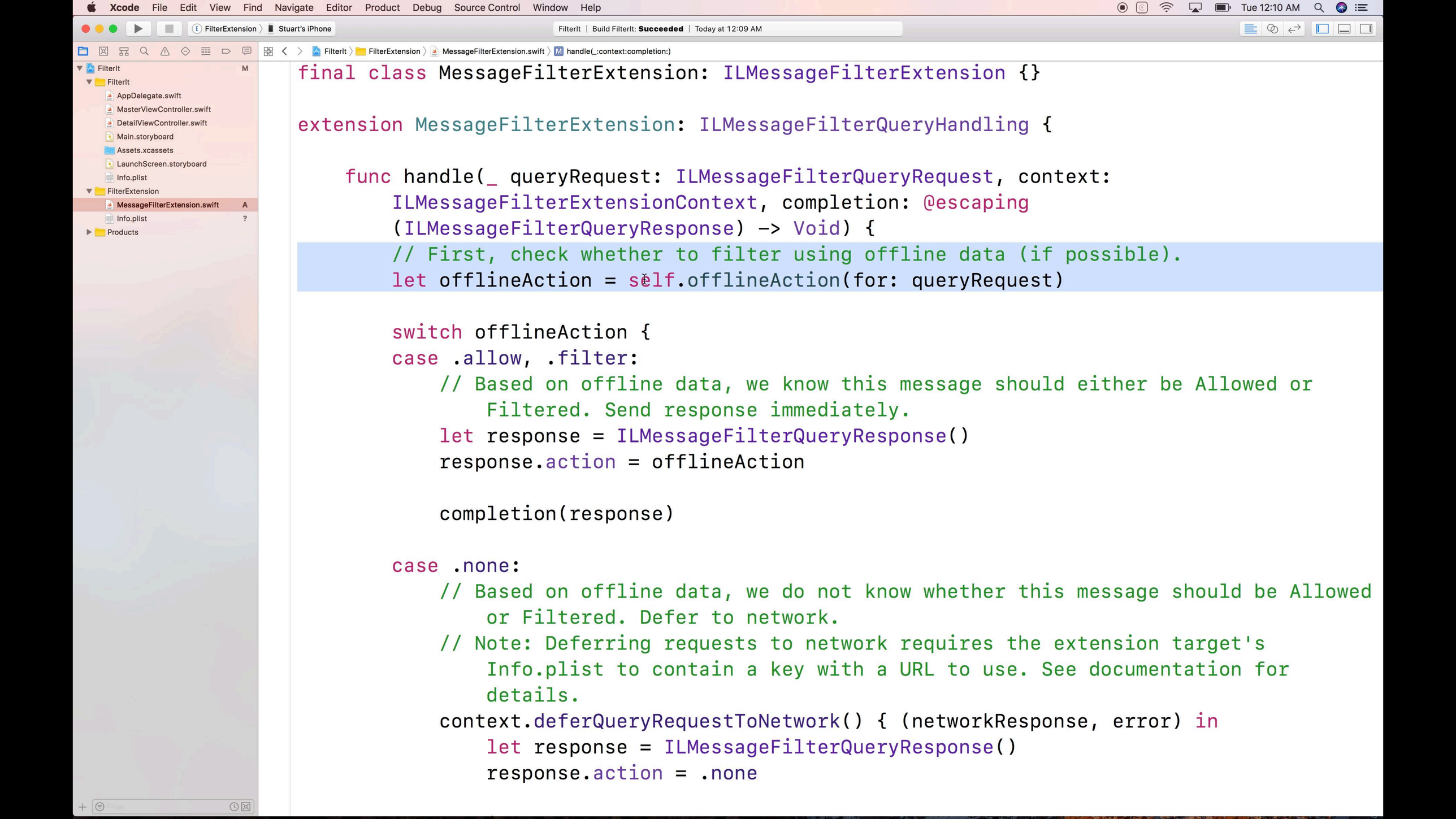
Offline-only extension

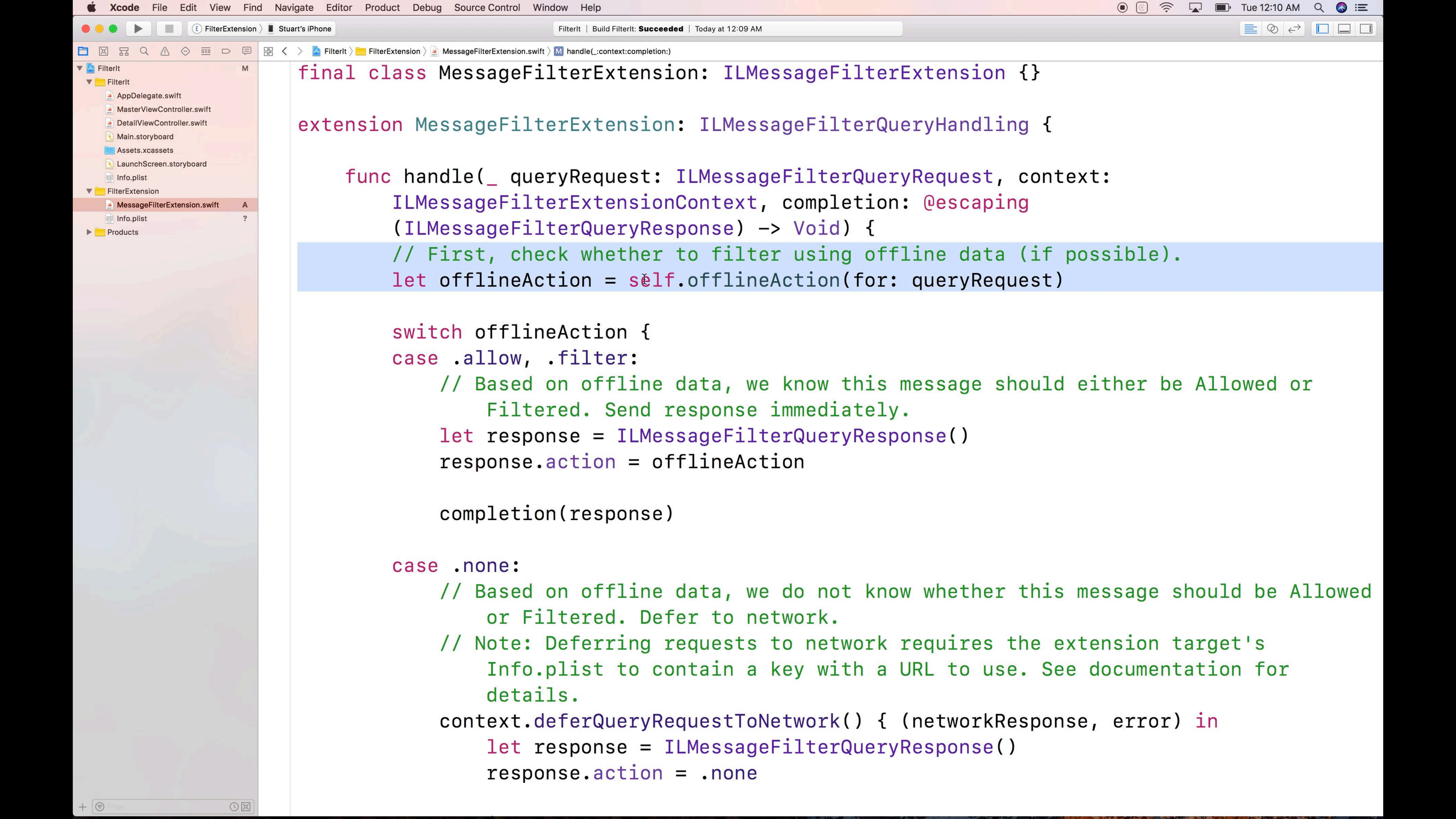

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 12:09 AM
FilterIt 〉 Stuart's iPhone
FilterIt | Build FilterIt: Succeeded | Today at 12:09 AM
FilterIt
FilterIt
FilterIt
AppDelegate.swift
MasterViewController.swift
DetailViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
Info
Build Settings
PROJECT
FilterIt
TARGETS
FilterIt
Deployment Target
iOS Deployment Target
11.0
Configurations
Name
Based on Configuration File
Debug
No Configurations Set
Release
No Configurations Set
Use
Release
for command-line builds
Localizations
Language
Resources
English — Development Language
2 Files Localized
Use Base Internationalization
Filter
Filter

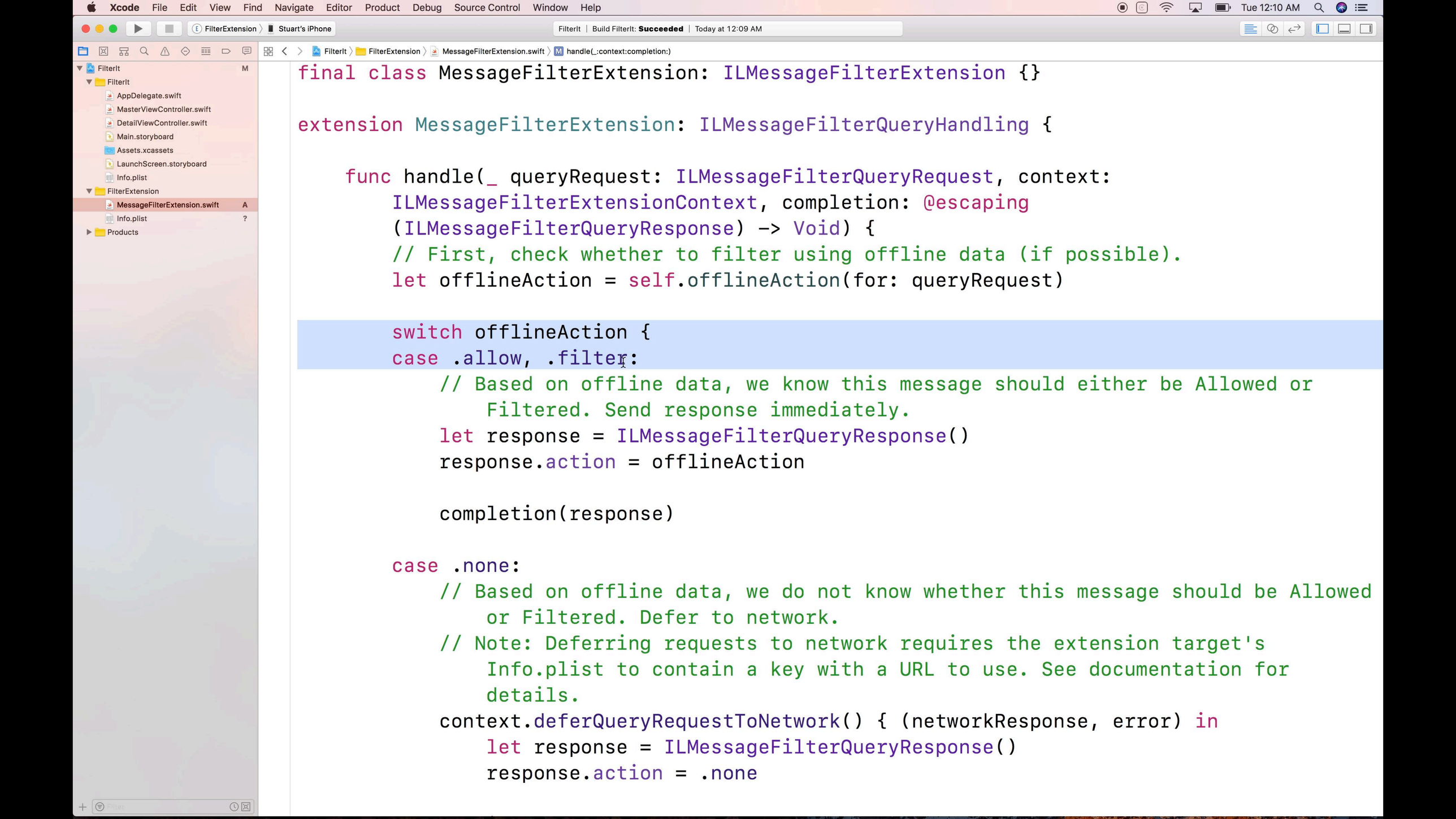

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 12:09 AM
FilterIt 〉 Stuart's iPhone
FilterIt | Build FilterIt: Succeeded | Today at 12:09 AM
FilterIt
FilterIt
FilterIt
AppDelegate.swift
MasterViewController.swift
DetailViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
Info
Build Settings
PROJECT
FilterIt
TARGETS
FilterIt
Deployment Target
iOS Deployment Target
11.0
Configurations
Name
Based on Configuration File
Debug
No Configurations Set
Release
No Configurations Set
Use
Release
for command-line builds
Localizations
Language
Resources
English — Development Language
2 Files Localized
Use Base Internationalization
Filter
Filter

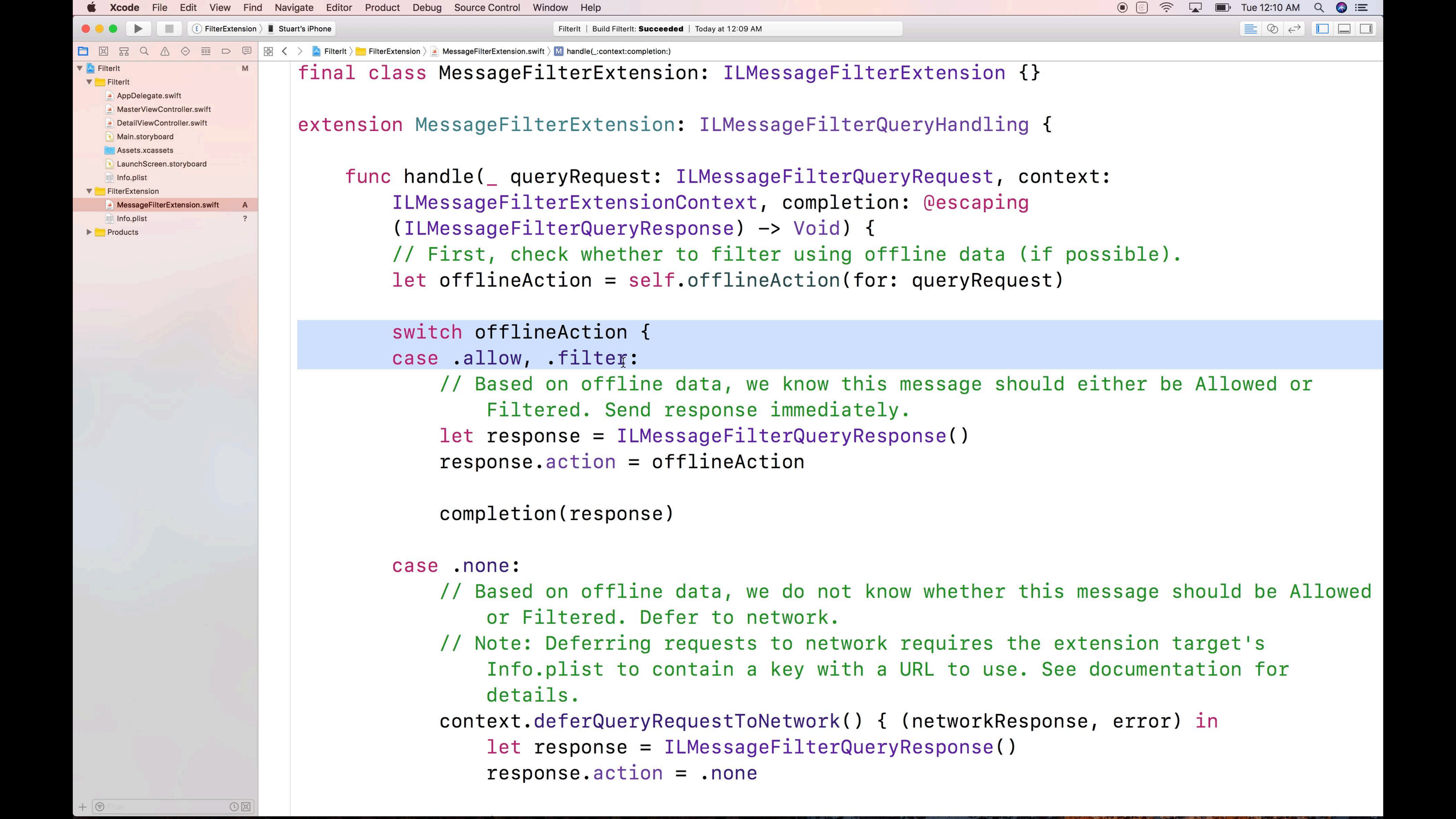

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 12:10 AM
FilterExtension Stuart's iPhone
FilterIt | Build FilterIt: Succeeded | Today at 12:09 AM
FilterIt FilterExtension MessageFilterExtension.swift handle(_:context:completion:)
FilterIt
FilterIt
AppDelegate.swift
MasterViewController.swift
DetailViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
FilterExtension
MessageFilterExtension.swift
Info.plist
Products
final class MessageFilterExtension: ILMessageFilterExtension {}
extension MessageFilterExtension: ILMessageFilterQueryHandling {
func handle(_ queryRequest: ILMessageFilterQueryRequest, context:
ILMessageFilterExtensionContext, completion: @escaping
(ILMessageFilterQueryResponse) -> Void) {
// First, check whether to filter using offline data (if possible).
let offlineAction = self.offlineAction(for: queryRequest)
switch offlineAction {
case .allow, .filter:
// Based on offline data, we know this message should either be Allowed or
Filtered. Send response immediately.
let response = ILMessageFilterQueryResponse()
response.action = offlineAction
completion(response)
case .none:
// Based on offline data, we do not know whether this message should be Allowed
or Filtered. Defer to network.
// Note: Deferring requests to network requires the extension target's
Info.plist to contain a key with a URL to use. See documentation for
details.
context.deferQueryRequestToNetwork() { (networkResponse, error) in
let response = ILMessageFilterQueryResponse()
response.action = .none

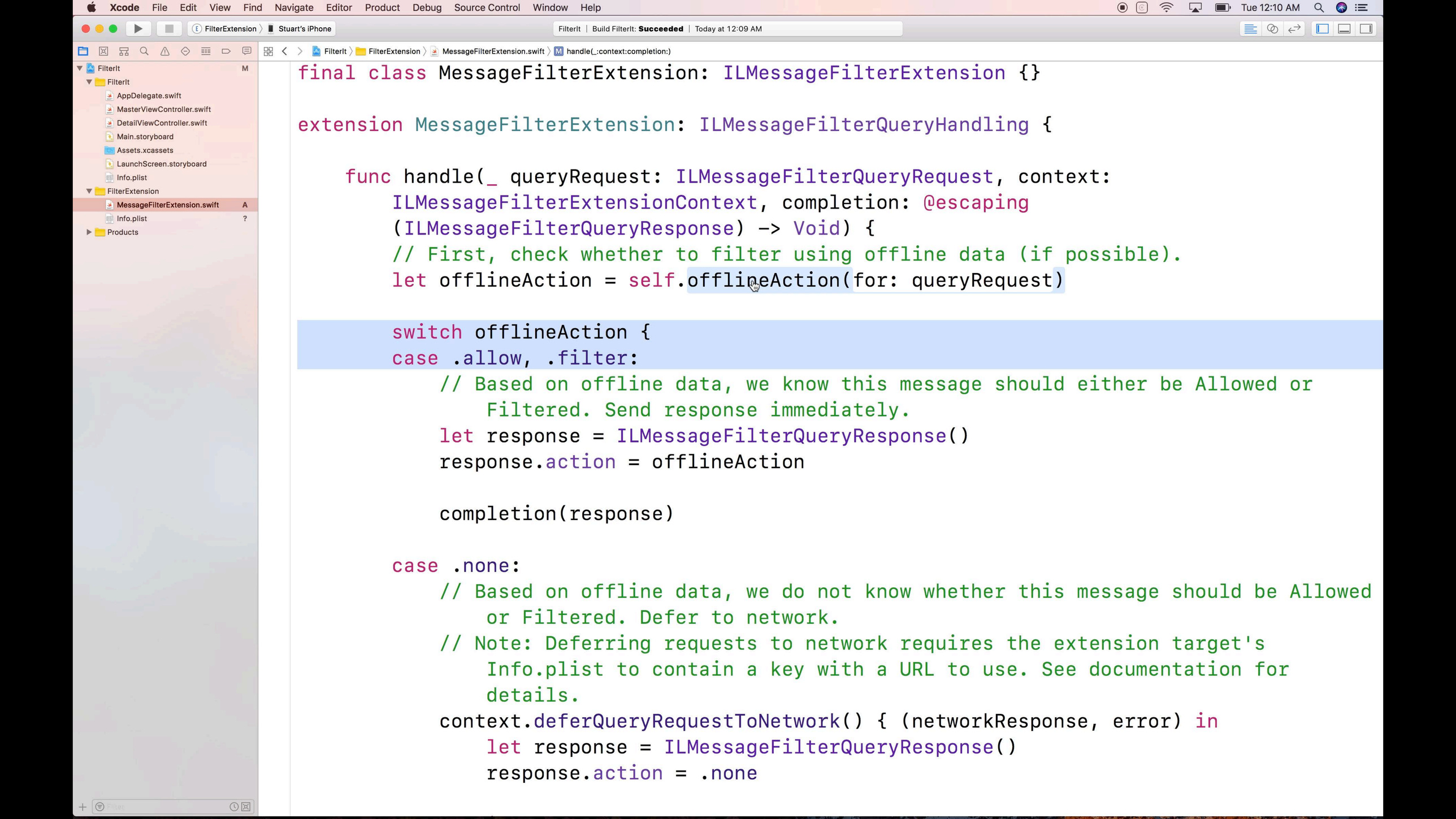

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 12:10 AM
FilterExtension Stuart's iPhone
FilterIt | Build FilterIt: Succeeded | Today at 12:09 AM
FilterIt FilterExtension MessageFilterExtension.swift handle(_:context:completion:)
FilterIt
FilterIt
AppDelegate.swift
MasterViewController.swift
DetailViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
FilterExtension
MessageFilterExtension.swift
Info.plist
Products
final class MessageFilterExtension: ILMessageFilterExtension {}
extension MessageFilterExtension: ILMessageFilterQueryHandling {
func handle(_ queryRequest: ILMessageFilterQueryRequest, context:
ILMessageFilterExtensionContext, completion: @escaping
(ILMessageFilterQueryResponse) -> Void) {
// First, check whether to filter using offline data (if possible).
let offlineAction = self.offlineAction(for: queryRequest)
switch offlineAction {
case .allow, .filter:
// Based on offline data, we know this message should either be Allowed or
Filtered. Send response immediately.
let response = ILMessageFilterQueryResponse()
response.action = offlineAction
completion(response)
case .none:
// Based on offline data, we do not know whether this message should be Allowed
or Filtered. Defer to network.
// Note: Deferring requests to network requires the extension target's
Info.plist to contain a key with a URL to use. See documentation for
details.
context.deferQueryRequestToNetwork() { (networkResponse, error) in
let response = ILMessageFilterQueryResponse()
response.action = .none

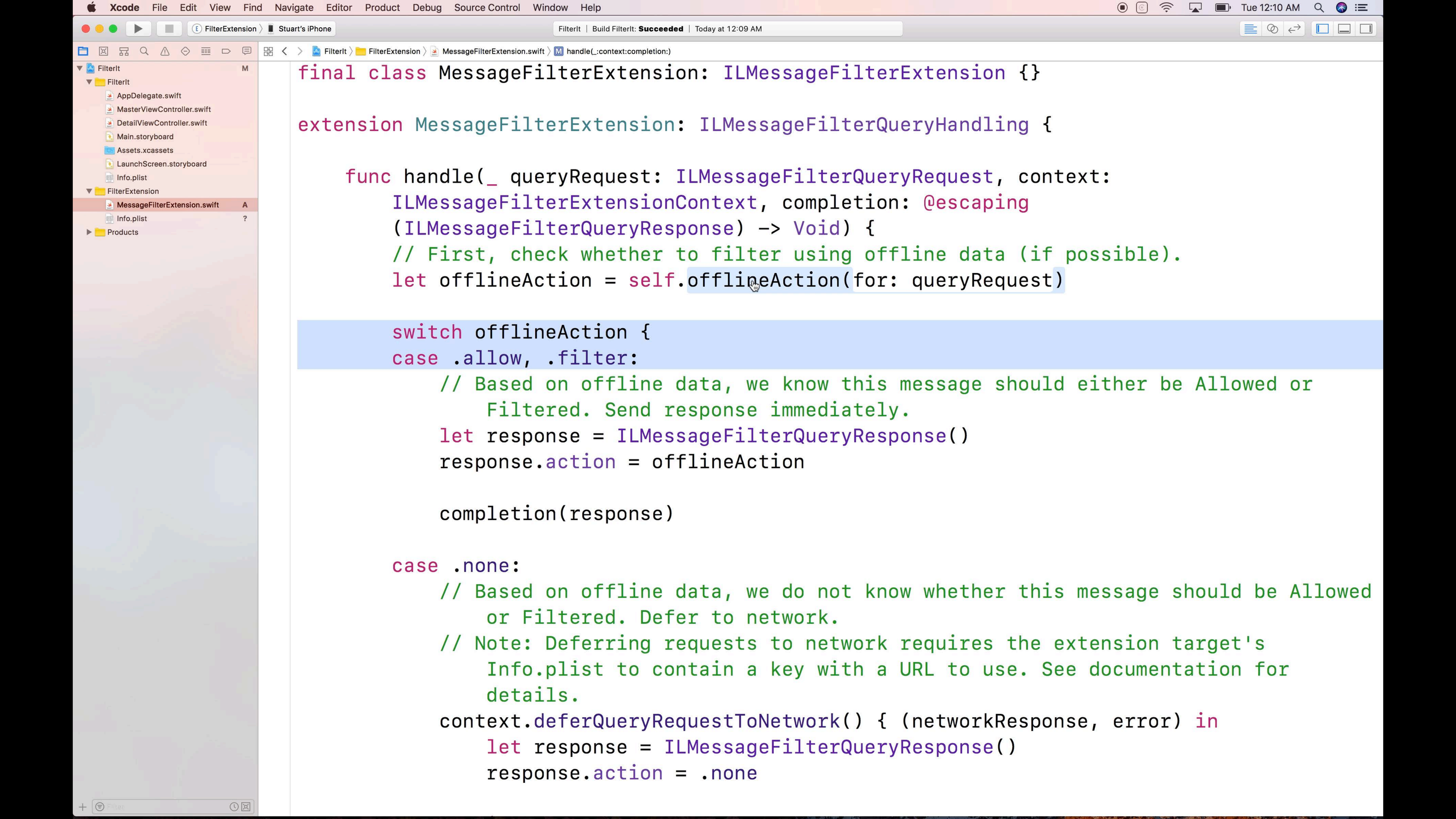

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 12:10 AM
FilterExtension Stuart's iPhone
FilterIt | Build FilterIt: Succeeded | Today at 12:09 AM
FilterIt FilterExtension MessageFilterExtension.swift handle(_:context:completion:)
FilterIt
FilterIt
AppDelegate.swift
MasterViewController.swift
DetailViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
FilterExtension
MessageFilterExtension.swift
Info.plist
Products
final class MessageFilterExtension: ILMessageFilterExtension {}
extension MessageFilterExtension: ILMessageFilterQueryHandling {
func handle(_ queryRequest: ILMessageFilterQueryRequest, context:
ILMessageFilterExtensionContext, completion: @escaping
(ILMessageFilterQueryResponse) -> Void) {
// First, check whether to filter using offline data (if possible).
let offlineAction = self.offlineAction(for: queryRequest)
switch offlineAction {
case .allow, .filter:
// Based on offline data, we know this message should either be Allowed or
Filtered. Send response immediately.
let response = ILMessageFilterQueryResponse()
response.action = offlineAction
completion(response)
case .none:
// Based on offline data, we do not know whether this message should be Allowed
or Filtered. Defer to network.
// Note: Deferring requests to network requires the extension target's
Info.plist to contain a key with a URL to use. See documentation for
details.
context.deferQueryRequestToNetwork() { (networkResponse, error) in
let response = ILMessageFilterQueryResponse()
response.action = .none

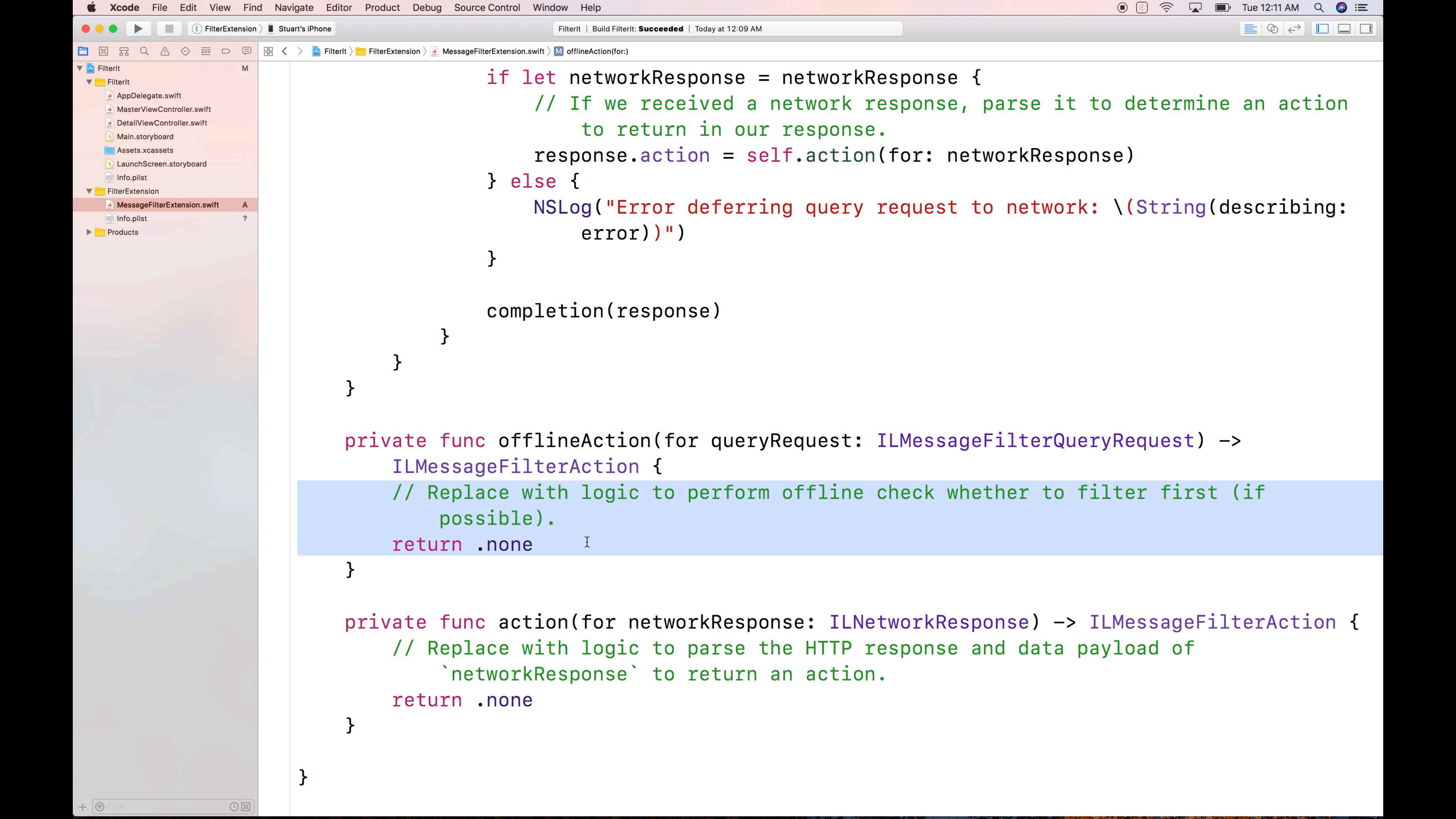

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 12:10 AM
FilterExtension Stuart's iPhone
FilterIt | Build FilterIt: Succeeded | Today at 12:09 AM
FilterIt FilterExtension MessageFilterExtension.swift handle(_:context:completion:)
FilterIt
FilterIt
AppDelegate.swift
MasterViewController.swift
DetailViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
FilterExtension
MessageFilterExtension.swift
Info.plist
Products
final class MessageFilterExtension: ILMessageFilterExtension {}
extension MessageFilterExtension: ILMessageFilterQueryHandling {
func handle(_ queryRequest: ILMessageFilterQueryRequest, context:
ILMessageFilterExtensionContext, completion: @escaping
(ILMessageFilterQueryResponse) -> Void) {
// First, check whether to filter using offline data (if possible).
let offlineAction = self.offlineAction(for: queryRequest)
switch offlineAction {
case .allow, .filter:
// Based on offline data, we know this message should either be Allowed or
Filtered. Send response immediately.
let response = ILMessageFilterQueryResponse()
response.action = offlineAction
completion(response)
case .none:
// Based on offline data, we do not know whether this message should be Allowed
or Filtered. Defer to network.
// Note: Deferring requests to network requires the extension target's
Info.plist to contain a key with a URL to use. See documentation for
details.
context.deferQueryRequestToNetwork() { (networkResponse, error) in
let response = ILMessageFilterQueryResponse()
response.action = .none

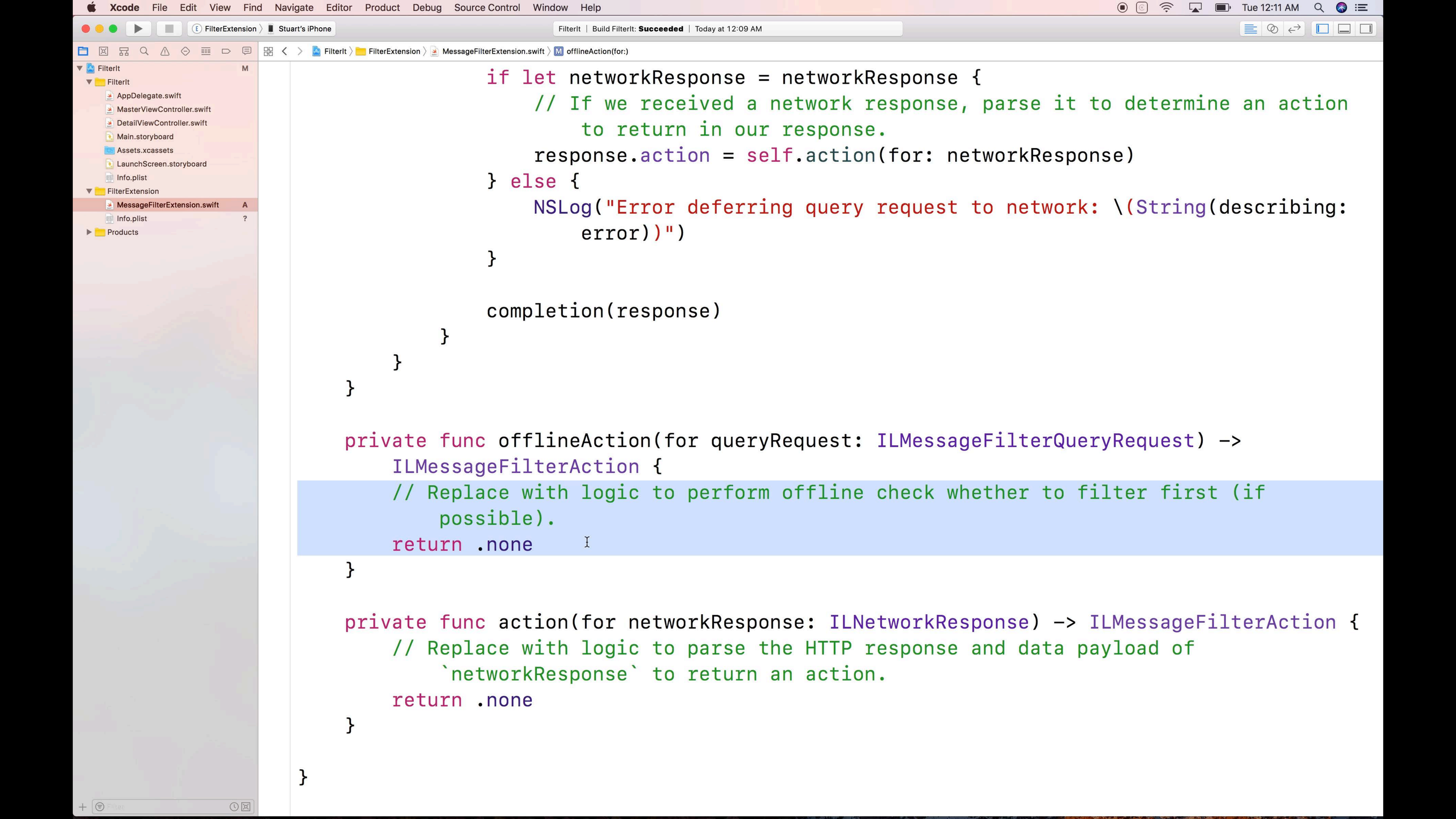

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 12:10 AM
FilterExtension Stuart's iPhone
FilterIt | Build FilterIt: Succeeded | Today at 12:09 AM
FilterIt FilterExtension MessageFilterExtension.swift handle(_:context:completion:)
FilterIt
FilterIt
AppDelegate.swift
MasterViewController.swift
DetailViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
FilterExtension
MessageFilterExtension.swift
Info.plist
Products
final class MessageFilterExtension: ILMessageFilterExtension {}
extension MessageFilterExtension: ILMessageFilterQueryHandling {
func handle(_ queryRequest: ILMessageFilterQueryRequest, context:
ILMessageFilterExtensionContext, completion: @escaping
(ILMessageFilterQueryResponse) -> Void) {
// First, check whether to filter using offline data (if possible).
let offlineAction = self.offlineAction(for: queryRequest)
switch offlineAction {
case .allow, .filter:
// Based on offline data, we know this message should either be Allowed or
Filtered. Send response immediately.
let response = ILMessageFilterQueryResponse()
response.action = offlineAction
completion(response)
case .none:
// Based on offline data, we do not know whether this message should be Allowed
or Filtered. Defer to network.
// Note: Deferring requests to network requires the extension target's
Info.plist to contain a key with a URL to use. See documentation for
details.
context.deferQueryRequestToNetwork() { (networkResponse, error) in
let response = ILMessageFilterQueryResponse()
response.action = .none


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 12:10 AM
FilterExtension Stuart's iPhone
FilterIt | Build FilterIt: Succeeded | Today at 12:09 AM
FilterIt FilterExtension MessageFilterExtension.swift handle(_:context:completion:)
FilterIt
FilterIt
AppDelegate.swift
MasterViewController.swift
DetailViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
FilterExtension
MessageFilterExtension.swift
Info.plist
Products
final class MessageFilterExtension: ILMessageFilterExtension {}
extension MessageFilterExtension: ILMessageFilterQueryHandling {
func handle(_ queryRequest: ILMessageFilterQueryRequest, context:
ILMessageFilterExtensionContext, completion: @escaping
(ILMessageFilterQueryResponse) -> Void) {
// First, check whether to filter using offline data (if possible).
let offlineAction = self.offlineAction(for: queryRequest)
switch offlineAction {
case .allow, .filter:
// Based on offline data, we know this message should either be Allowed or
Filtered. Send response immediately.
let response = ILMessageFilterQueryResponse()
response.action = offlineAction
completion(response)
case .none:
// Based on offline data, we do not know whether this message should be Allowed
or Filtered. Defer to network.
// Note: Deferring requests to network requires the extension target's
Info.plist to contain a key with a URL to use. See documentation for
details.
context.deferQueryRequestToNetwork() { (networkResponse, error) in
let response = ILMessageFilterQueryResponse()
response.action = .none


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 12:10 AM
FilterExtension Stuart's iPhone
FilterIt | Build FilterIt: Succeeded | Today at 12:09 AM
FilterIt FilterExtension MessageFilterExtension.swift handle(_:context:completion:)
FilterIt
FilterIt
AppDelegate.swift
MasterViewController.swift
DetailViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
FilterExtension
MessageFilterExtension.swift
Info.plist
Products
final class MessageFilterExtension: ILMessageFilterExtension {}
extension MessageFilterExtension: ILMessageFilterQueryHandling {
func handle(_ queryRequest: ILMessageFilterQueryRequest, context:
ILMessageFilterExtensionContext, completion: @escaping
(ILMessageFilterQueryResponse) -> Void) {
// First, check whether to filter using offline data (if possible).
let offlineAction = self.offlineAction(for: queryRequest)
switch offlineAction {
case .allow, .filter:
// Based on offline data, we know this message should either be Allowed or
Filtered. Send response immediately.
let response = ILMessageFilterQueryResponse()
response.action = offlineAction
completion(response)
case .none:
// Based on offline data, we do not know whether this message should be Allowed
or Filtered. Defer to network.
// Note: Deferring requests to network requires the extension target's
Info.plist to contain a key with a URL to use. See documentation for
details.
context.deferQueryRequestToNetwork() { (networkResponse, error) in
let response = ILMessageFilterQueryResponse()
response.action = .none


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 12:10 AM
FilterExtension Stuart's iPhone
FilterIt | Build FilterIt: Succeeded | Today at 12:09 AM
FilterIt FilterExtension MessageFilterExtension.swift handle(_:context:completion:)
FilterIt
FilterIt
AppDelegate.swift
MasterViewController.swift
DetailViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
FilterExtension
MessageFilterExtension.swift
Info.plist
Products
final class MessageFilterExtension: ILMessageFilterExtension {}
extension MessageFilterExtension: ILMessageFilterQueryHandling {
func handle(_ queryRequest: ILMessageFilterQueryRequest, context:
ILMessageFilterExtensionContext, completion: @escaping
(ILMessageFilterQueryResponse) -> Void) {
// First, check whether to filter using offline data (if possible).
let offlineAction = self.offlineAction(for: queryRequest)
switch offlineAction {
case .allow, .filter:
// Based on offline data, we know this message should either be Allowed or
Filtered. Send response immediately.
let response = ILMessageFilterQueryResponse()
response.action = offlineAction
completion(response)
case .none:
// Based on offline data, we do not know whether this message should be Allowed
or Filtered. Defer to network.
// Note: Deferring requests to network requires the extension target's
Info.plist to contain a key with a URL to use. See documentation for
details.
context.deferQueryRequestToNetwork() { (networkResponse, error) in
let response = ILMessageFilterQueryResponse()
response.action = .none

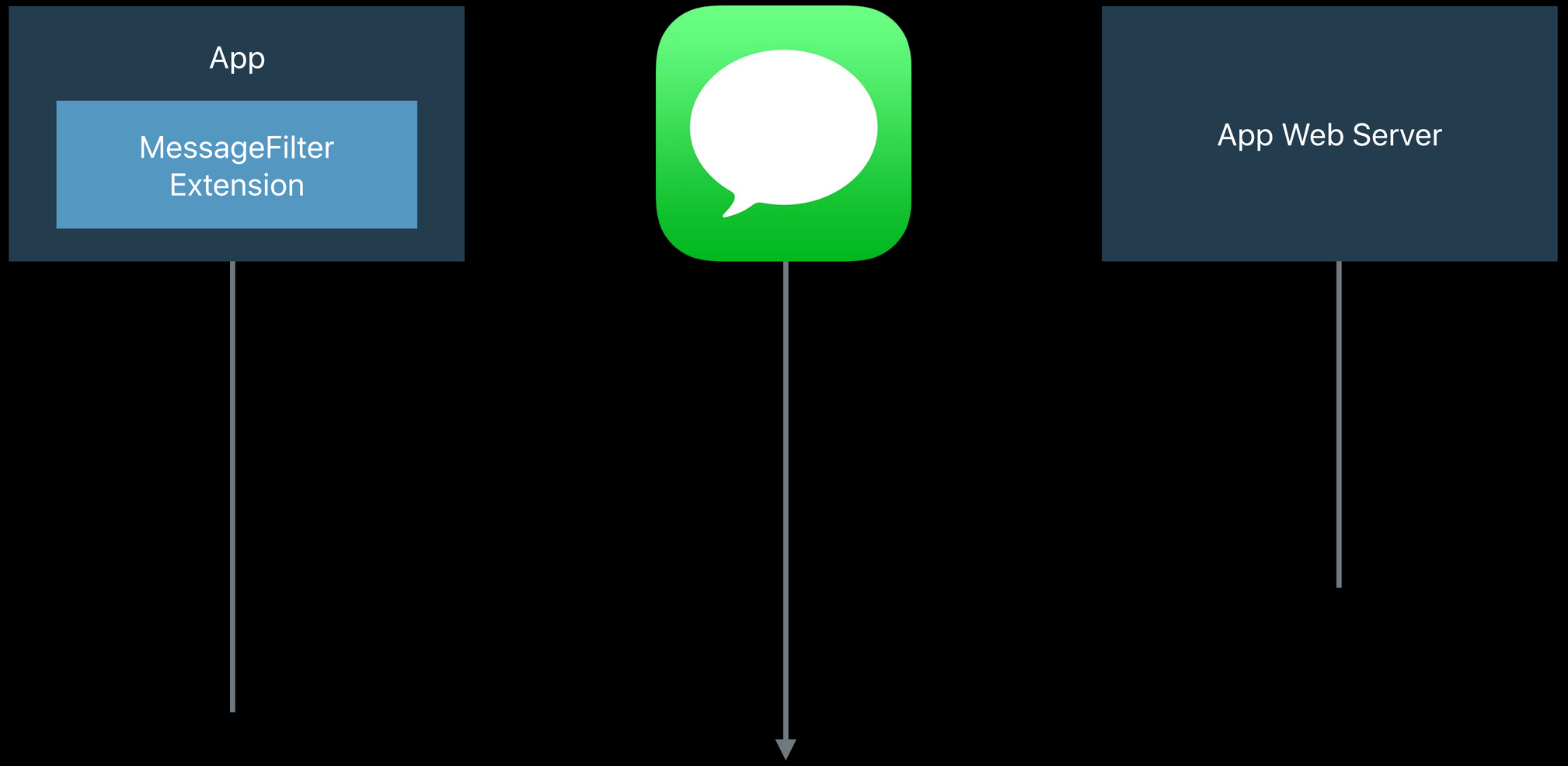

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 12:11 AM
FilterExtension Stuart's iPhone
FilterIt | Build FilterIt: Succeeded | Today at 12:09 AM
FilterIt FilterExtension MessageFilterExtension.swift offlineAction(for:)
FilterIt
FilterIt
AppDelegate.swift
MasterViewController.swift
DetailViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
FilterExtension
MessageFilterExtension.swift
Info.plist
Products
if let networkResponse = networkResponse {
// If we received a network response, parse it to determine an action to return in our response.
response.action = self.action(for: networkResponse)
} else {
NSLog("Error deferring query request to network: \(String(describing: error))")
}
completion(response)
}
}
}
private func offlineAction(for queryRequest: ILMessageFilterQueryRequest) -> ILMessageFilterAction {
// Replace with logic to perform offline check whether to filter first (if possible).
return .none
}
private func action(for networkResponse: ILNetworkResponse) -> ILMessageFilterAction {
// Replace with logic to parse the HTTP response and data payload of `networkResponse` to return an action.
return .none
}
}


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 12:11 AM
FilterExtension Stuart's iPhone
FilterIt | Build FilterIt: Succeeded | Today at 12:09 AM
FilterIt FilterExtension MessageFilterExtension.swift offlineAction(for:)
FilterIt
FilterIt
AppDelegate.swift
MasterViewController.swift
DetailViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
FilterExtension
MessageFilterExtension.swift
Info.plist
Products
if let networkResponse = networkResponse {
// If we received a network response, parse it to determine an action to return in our response.
response.action = self.action(for: networkResponse)
} else {
NSLog("Error deferring query request to network: \(String(describing: error))")
}
completion(response)
}
}
}
private func offlineAction(for queryRequest: ILMessageFilterQueryRequest) -> ILMessageFilterAction {
// Replace with logic to perform offline check whether to filter first (if possible).
return .none
}
private func action(for networkResponse: ILNetworkResponse) -> ILMessageFilterAction {
// Replace with logic to parse the HTTP response and data payload of `networkResponse` to return an action.
return .none
}
}

Network Deferral

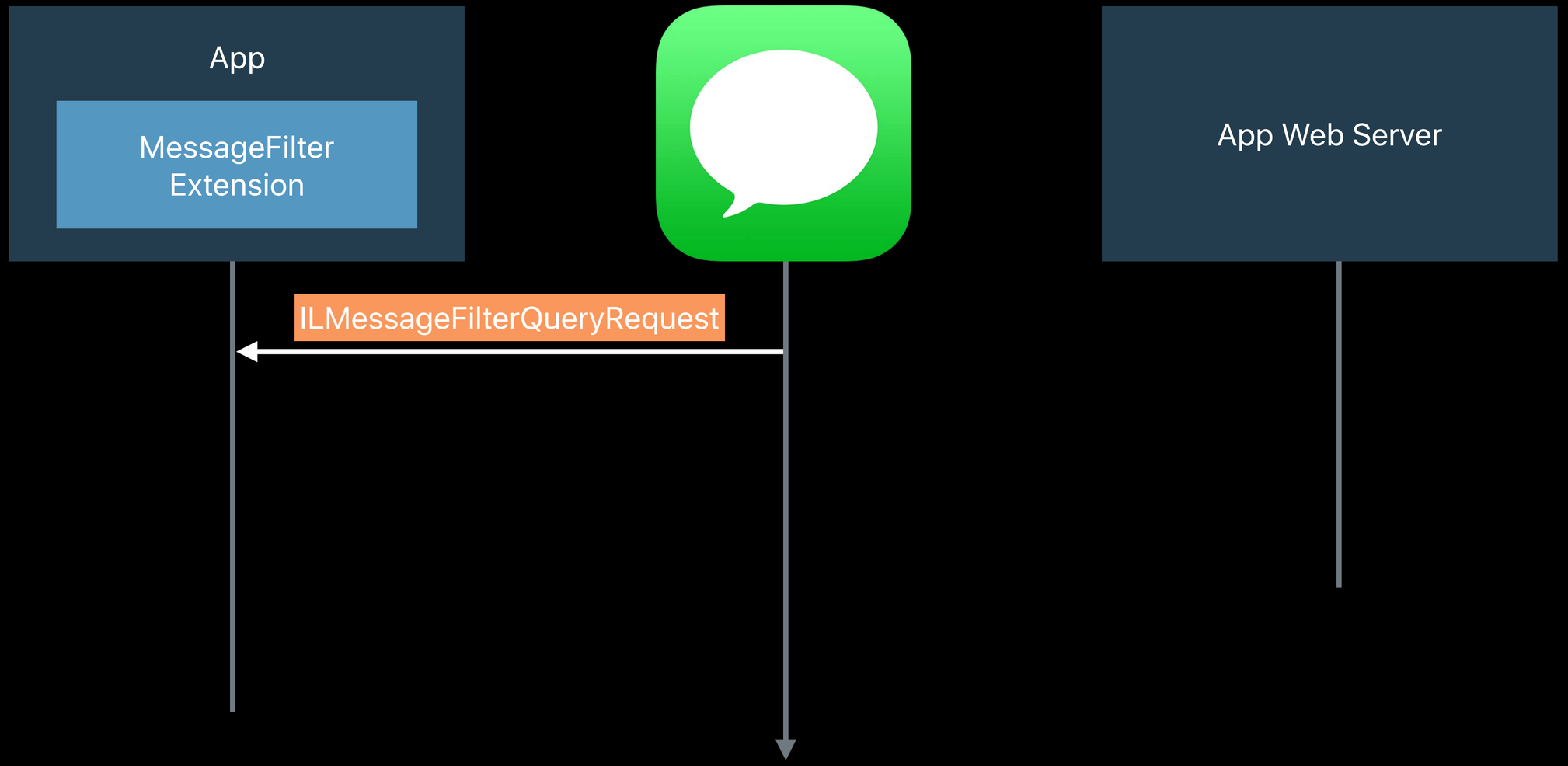
Message Filter Extensions

With network deferral



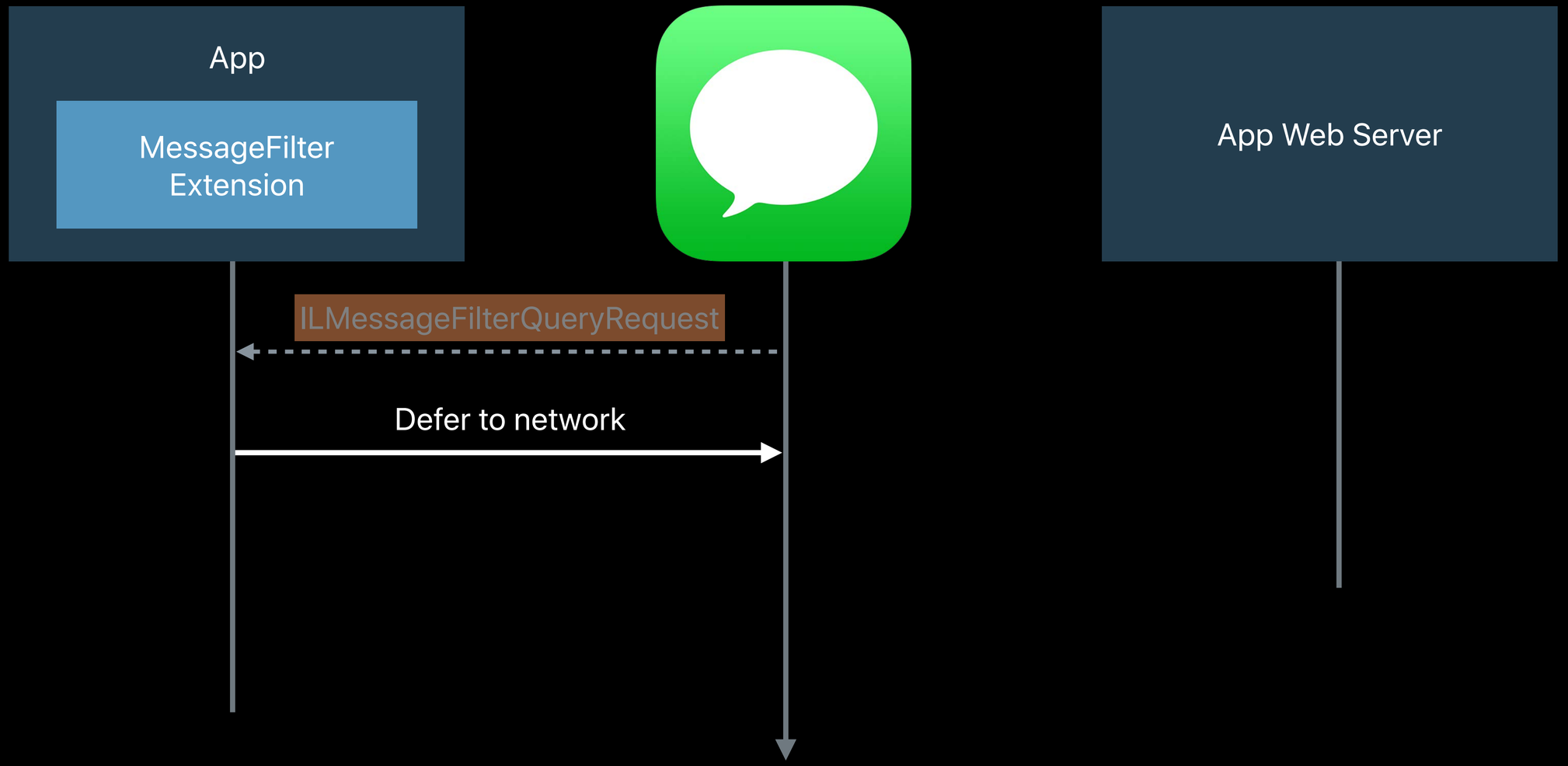
Message Filter Extensions

With network deferral



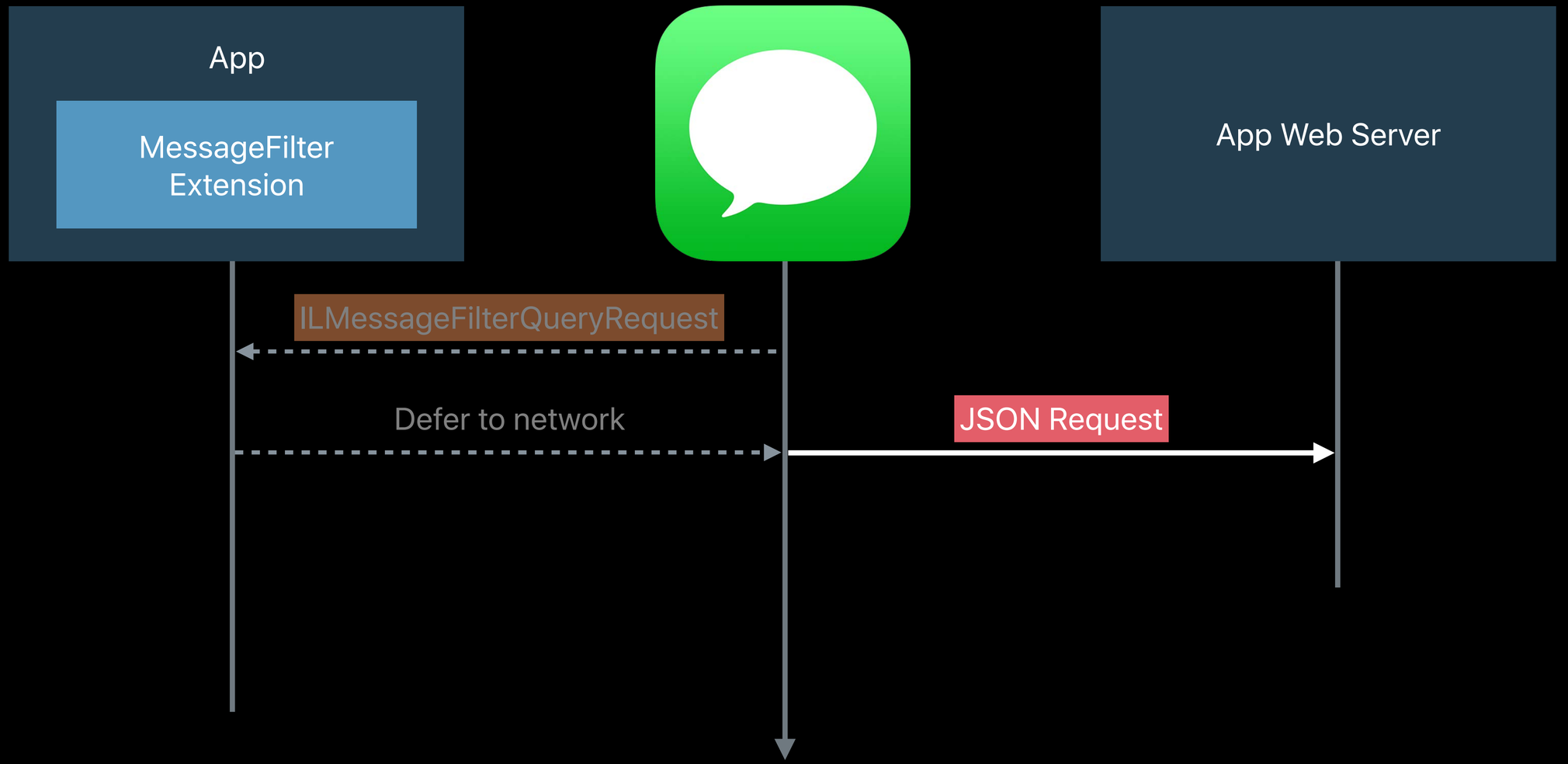
Message Filter Extensions

With network deferral



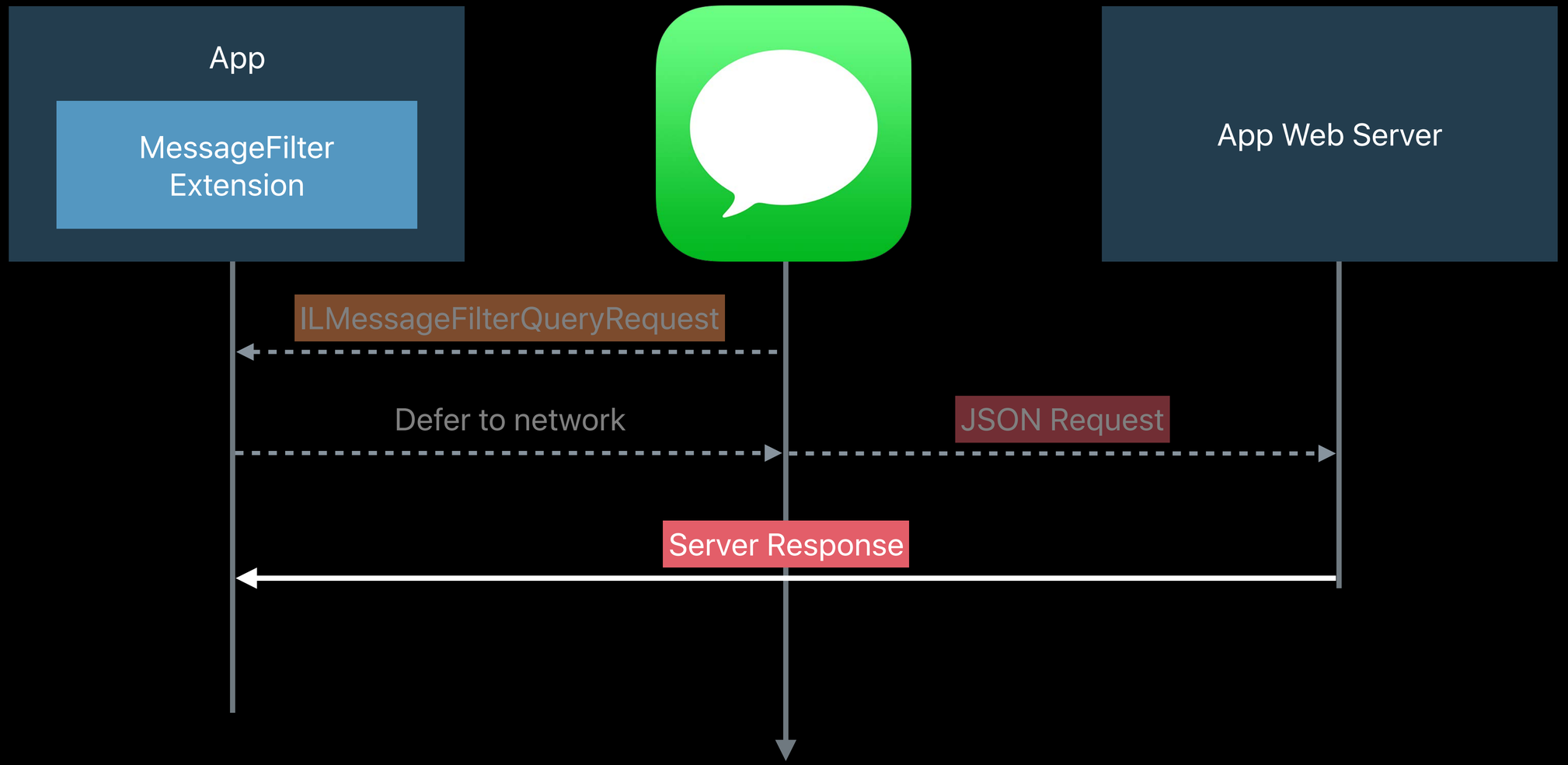
Message Filter Extensions

With network deferral



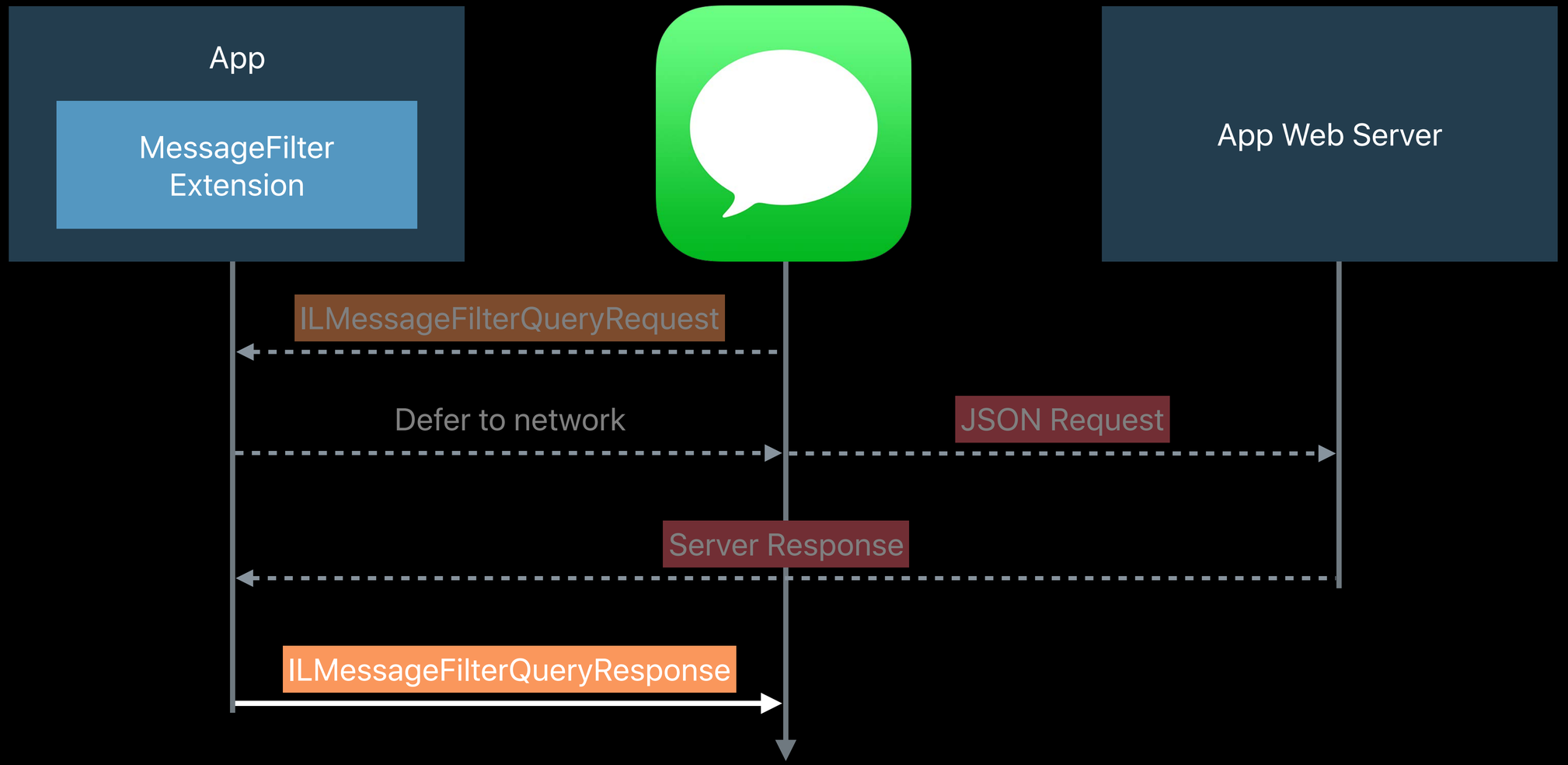
Message Filter Extensions

With network deferral



Message Filter Extensions

With network deferral



Network Deferral Restrictions

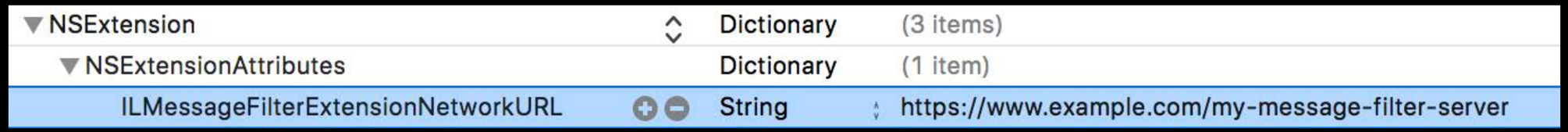
Network Deferral Restrictions

Network requests contain no personally identifiable information

Network Deferral Restrictions

Network requests contain no personally identifiable information

URL specified in extension’s Info.plist



Network Deferral Restrictions

Network requests contain no personally identifiable information

URL specified in extension’s Info.plist

Network Deferral Restrictions

Network requests contain no personally identifiable information

URL specified in extension’s Info.plist

HTTPS required, no App Transport Security (ATS) overrides

Network Deferral Restrictions

Network requests contain no personally identifiable information

URL specified in extension’s Info.plist

HTTPS required, no App Transport Security (ATS) overrides

Extension’s app must use Associated Domains (apple-app-site-association)

Network Deferral Restrictions

Network requests contain no personally identifiable information

URL specified in extension’s Info.plist

HTTPS required, no App Transport Security (ATS) overrides

Extension’s app must use Associated Domains (apple-app-site-association)

Network Deferral Restrictions

Network requests contain no personally identifiable information

URL specified in extension’s Info.plist

HTTPS required, no App Transport Security (ATS) overrides

Extension’s app must use Associated Domains (apple-app-site-association)

Cookies are not preserved

Network Request and Response Formats

Network Request and Response Formats

Network request uses JSON and includes

Network Request and Response Formats

Network request uses JSON and includes

- Message sender (phone number or email)

Network Request and Response Formats

Network request uses JSON and includes

- Message sender (phone number or email)
- Message body

Network Request and Response Formats

Network request uses JSON and includes

- Message sender (phone number or email)
- Message body
- Version of app (CFBundleVersion)

Network Request and Response Formats

Network request uses JSON and includes

- Message sender (phone number or email)
- Message body
- Version of app (CFBundleVersion)
- Version of the JSON request format (currently 1)

Network Request and Response Formats

Network request uses JSON and includes

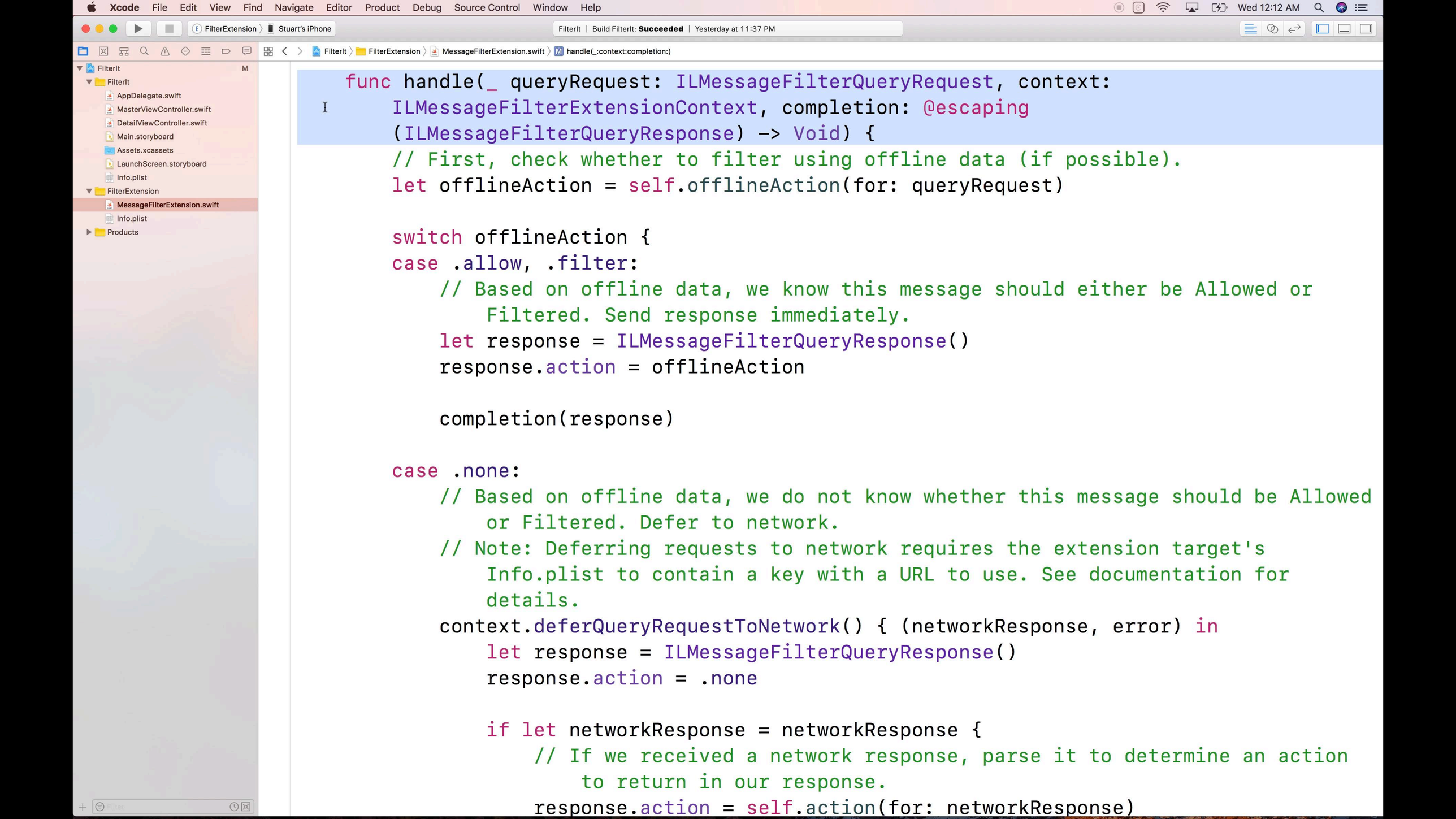
- Message sender (phone number or email)
- Message body
- Version of app (CFBundleVersion)
- Version of the JSON request format (currently 1)

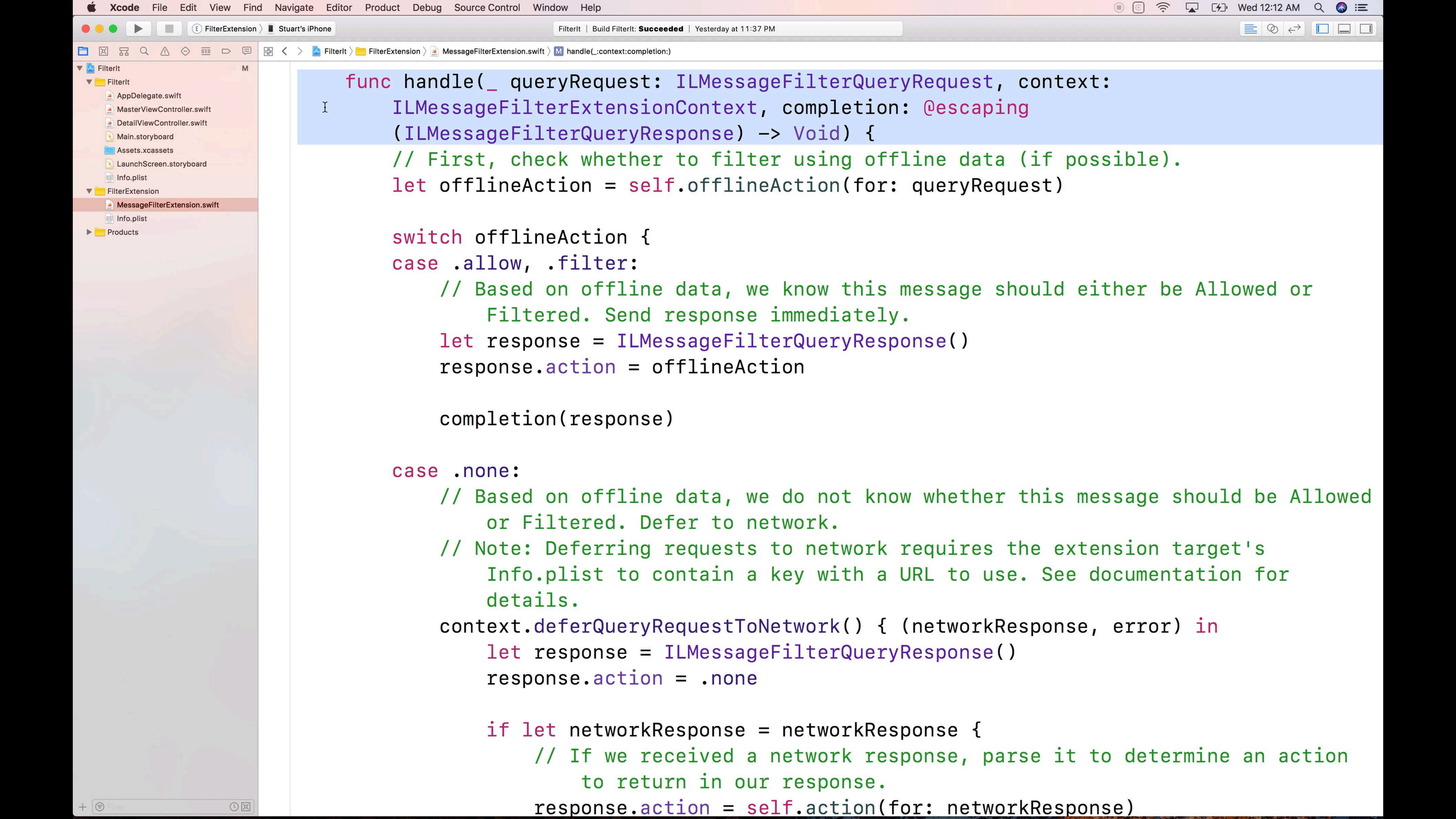
Response format is up to your app to define

```
// Example of Network HTTP Request Sent to Server

POST /server-url-path HTTP/1.1
Accept: */*
Content-Type: application/json; charset=utf-8
Content-Length: 148

{   "_version": 1,
    "query": {
        "sender": "14085550001",
        "message": {
            "text": "This is a message"
        }
    },
    "app": {
        "version": "1.1"
    }
}
```

Demo

Network-deferring extension


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Wed 12:12 AM
FilterExtension Stuart's iPhone
FilterIt | Build FilterIt: Succeeded | Yesterday at 11:37 PM
FilterIt FilterExtension MessageFilterExtension.swift handle(_:context:completion:)
FilterIt
FilterIt
AppDelegate.swift
MasterViewController.swift
DetailViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
FilterExtension
MessageFilterExtension.swift
Info.plist
Products
func handle(_ queryRequest: ILMessageFilterQueryRequest, context:
ILMessageFilterExtensionContext, completion: @escaping
(ILMessageFilterQueryResponse) -> Void) {
// First, check whether to filter using offline data (if possible).
let offlineAction = self.offlineAction(for: queryRequest)
switch offlineAction {
case .allow, .filter:
// Based on offline data, we know this message should either be Allowed or
Filtered. Send response immediately.
let response = ILMessageFilterQueryResponse()
response.action = offlineAction
completion(response)
case .none:
// Based on offline data, we do not know whether this message should be Allowed
or Filtered. Defer to network.
// Note: Deferring requests to network requires the extension target's
Info.plist to contain a key with a URL to use. See documentation for
details.
context.deferQueryRequestToNetwork() { (networkResponse, error) in
let response = ILMessageFilterQueryResponse()
response.action = .none
if let networkResponse = networkResponse {
// If we received a network response, parse it to determine an action
to return in our response.
response.action = self.action(for: networkResponse)


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Wed 12:12 AM
FilterExtension Stuart's iPhone
FilterIt | Build FilterIt: Succeeded | Yesterday at 11:37 PM
FilterIt FilterExtension MessageFilterExtension.swift handle(_:context:completion:)
FilterIt
FilterIt
AppDelegate.swift
MasterViewController.swift
DetailViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
FilterExtension
MessageFilterExtension.swift
Info.plist
Products
func handle(_ queryRequest: ILMessageFilterQueryRequest, context:
ILMessageFilterExtensionContext, completion: @escaping
(ILMessageFilterQueryResponse) -> Void) {
// First, check whether to filter using offline data (if possible).
let offlineAction = self.offlineAction(for: queryRequest)
switch offlineAction {
case .allow, .filter:
// Based on offline data, we know this message should either be Allowed or
Filtered. Send response immediately.
let response = ILMessageFilterQueryResponse()
response.action = offlineAction
completion(response)
case .none:
// Based on offline data, we do not know whether this message should be Allowed
or Filtered. Defer to network.
// Note: Deferring requests to network requires the extension target's
Info.plist to contain a key with a URL to use. See documentation for
details.
context.deferQueryRequestToNetwork() { (networkResponse, error) in
let response = ILMessageFilterQueryResponse()
response.action = .none
if let networkResponse = networkResponse {
// If we received a network response, parse it to determine an action
to return in our response.
response.action = self.action(for: networkResponse)

Summary

Message Filter extensions

Powerful but subject to privacy restrictions

Try making one!

More Information

https://developer.apple.com/wwdc17/249

Related Sessions

Privacy and Your Apps	Executive Ballroom	Tuesday 11:20AM
What's New in Foundation	Hall 2	Wednesday 11:00AM
Seamless Linking to Your App		WWDC15

Labs

CallKit and Identity Lookup Lab	Technology Lab H	Tuesday 12:40PM–2:50PM


WWDC17