REFACTORING YOUR APP IN RX

PRELUDE

SHOW OF HANDS 🖐

1. How many people here’ve heard of Functional Reactive Programming?
2. How many people here’ve heard of Reactive Swift/RxSwift?
3. How many people here’ve used Reactive Swift/RxSwift?

What this talk is *NOT* about

Introduction to RxSwift

- ~~Introduction to RxSwift~~

RxSwift by @codeOfRobin at Swift Delhi meetup Chapter 4 @swiftindiagroup

#swift #swiftlang #rx #swiftdelhi

11:58 PM - 2 Jun 2017

- RxSwift with MVVM

- ~~RxSwift with MVVM~~

Stories
Examples
Experiences

I DON'T KNOW RxSwift 😰

Conferences aren't about learning things, they're about learning what to learn, about networking, and meeting great people.

– Someone on Twitter

So chill ❄

2.0.0 

NETWORKING

Rx ➡ streams of values over time.

Sockets ➡ streams of data packets over time on a wire

➡ Rx ❤️ Sockets

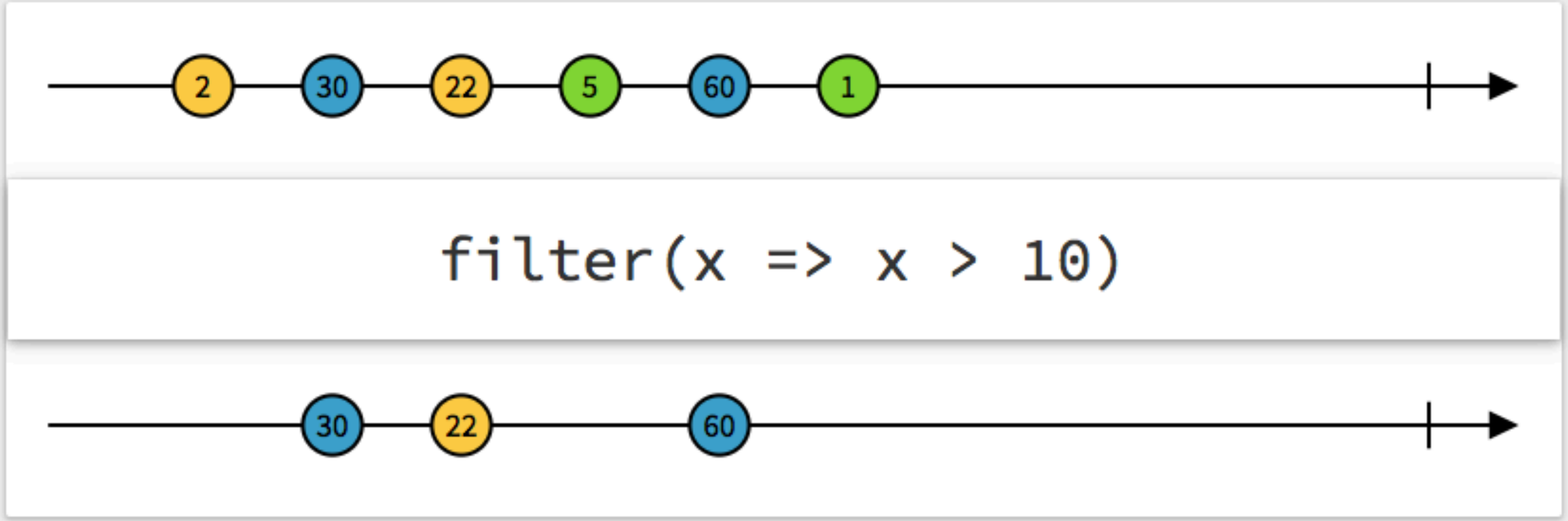
WHAT ABOUT HTTP REQUESTS?

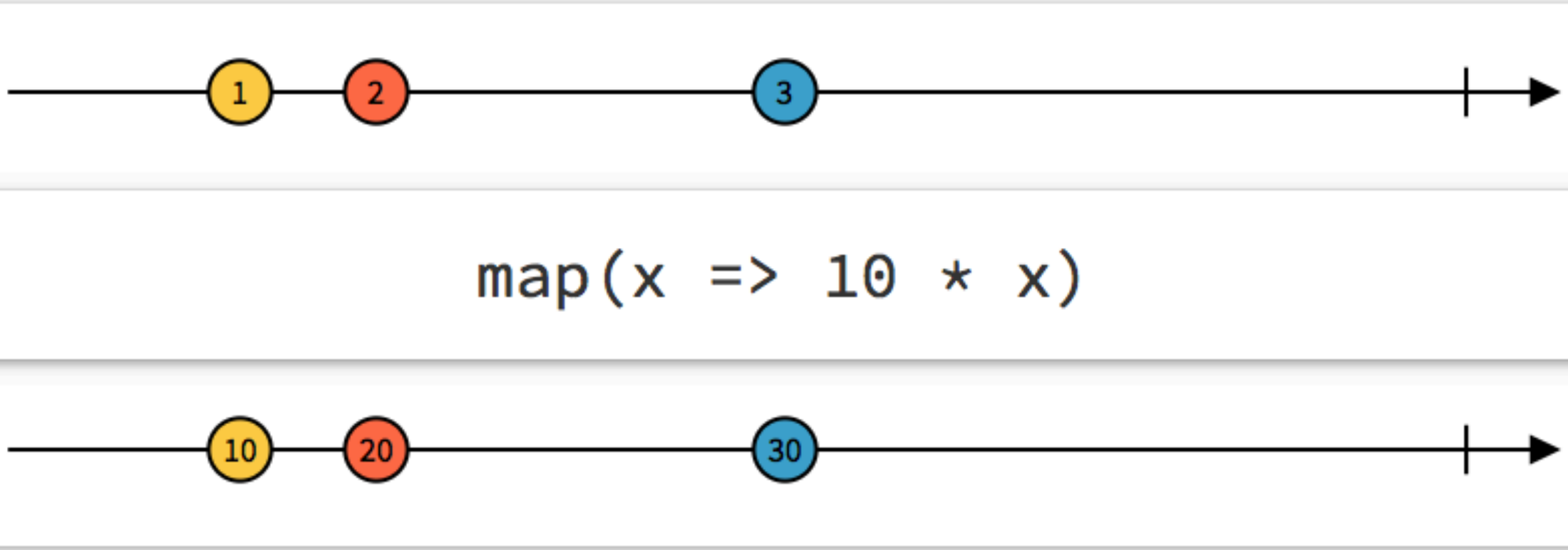
Is Rx really a good fit for HTTP calls? IMO, Promises/Futures are a much better fit.

9:18 AM - 18 Sep 2017

flatMapLatest()

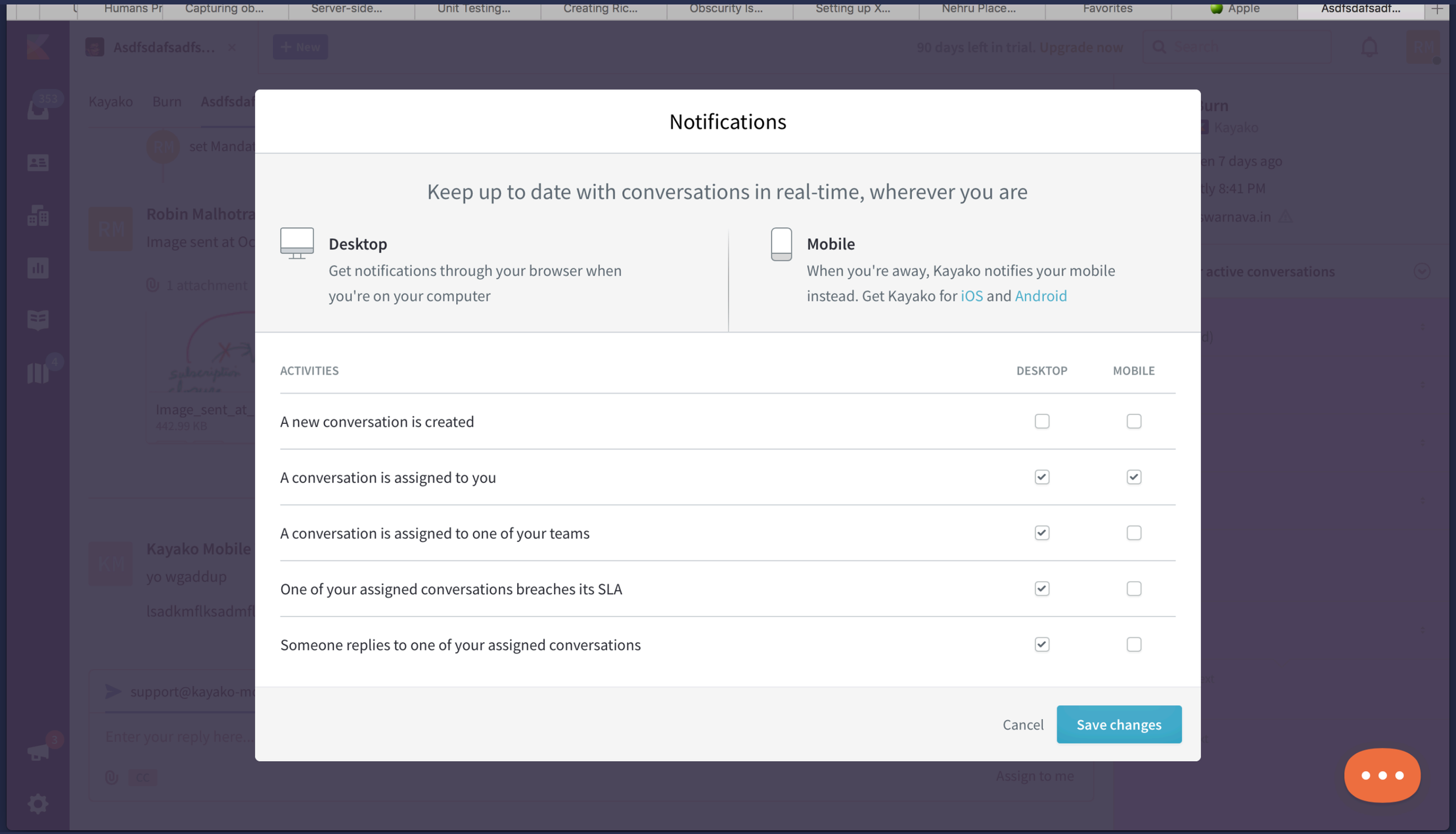
Interactive diagrams of Rx Observables

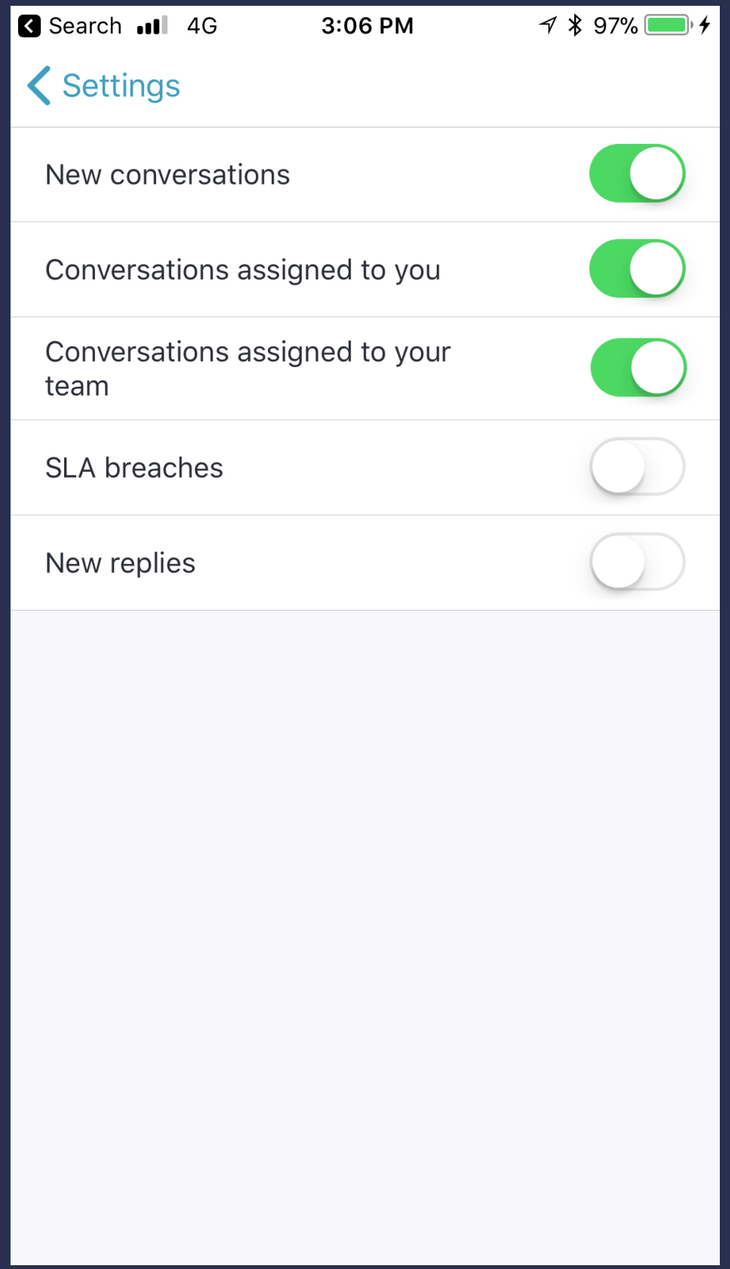


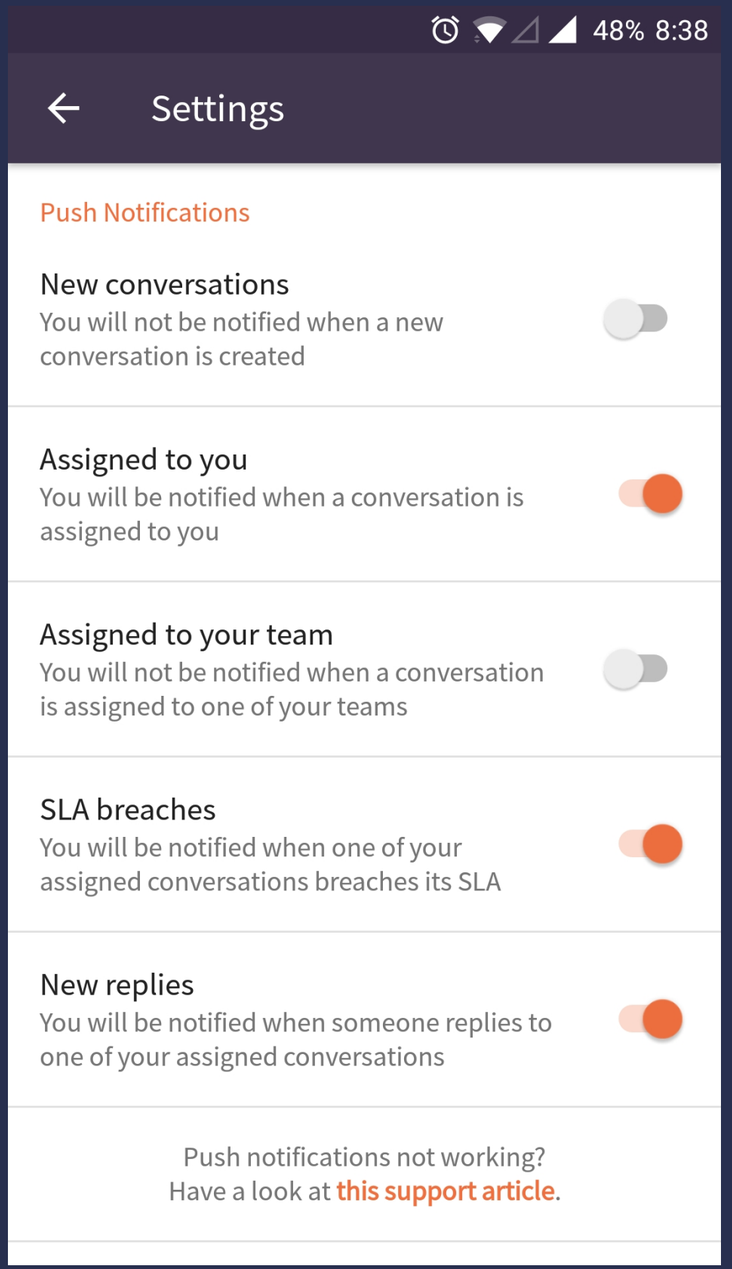

1
2
3
map(x => 10 * x)
10
20
30

LIVE DEMO

NOTIFICATION PREFERENCES


Notifications
Keep up to date with conversations in real-time, wherever you are
Desktop
Get notifications through your browser when you're on your computer
Mobile
When you're away, Kayako notifies your mobile instead. Get Kayako for iOS and Android
ACTIVITIES
DESKTOP
MOBILE
A new conversation is created
A conversation is assigned to you
A conversation is assigned to one of your teams
One of your assigned conversations breaches its SLA
Someone replies to one of your assigned conversations
Cancel
Save changes


Search 4G 3:06 PM 97%
Settings
New conversations
Conversations assigned to you
Conversations assigned to your team
SLA breaches
New replies


48% 8:38
Settings
Push Notifications
New conversations
You will not be notified when a new conversation is created
Assigned to you
You will be notified when a conversation is assigned to you
Assigned to your team
You will not be notified when a conversation is assigned to one of your teams
SLA breaches
You will be notified when one of your assigned conversations breaches its SLA
New replies
You will be notified when someone replies to one of your assigned conversations
Push notifications not working?
Have a look at this support article.

PUT https://kayako-mobile-testing.kayako.com/api/v1/notification_preferences Params Send Save

Authorization Headers (5) Body Pre-request Script Tests Cookies Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
{
  "values": [
    {
      "notification_type": "case_created",
      "channel_desktop": true,
      "channel_mobile": true
    },
    {
      "notification_type": "case_assigned_to_agent",
      "channel_desktop": true,
      "channel_mobile": true
    },
    {
      "notification_type": "case_assigned_to_team",
      "channel_desktop": false,
      "channel_mobile": true
    }
  ]
}
```

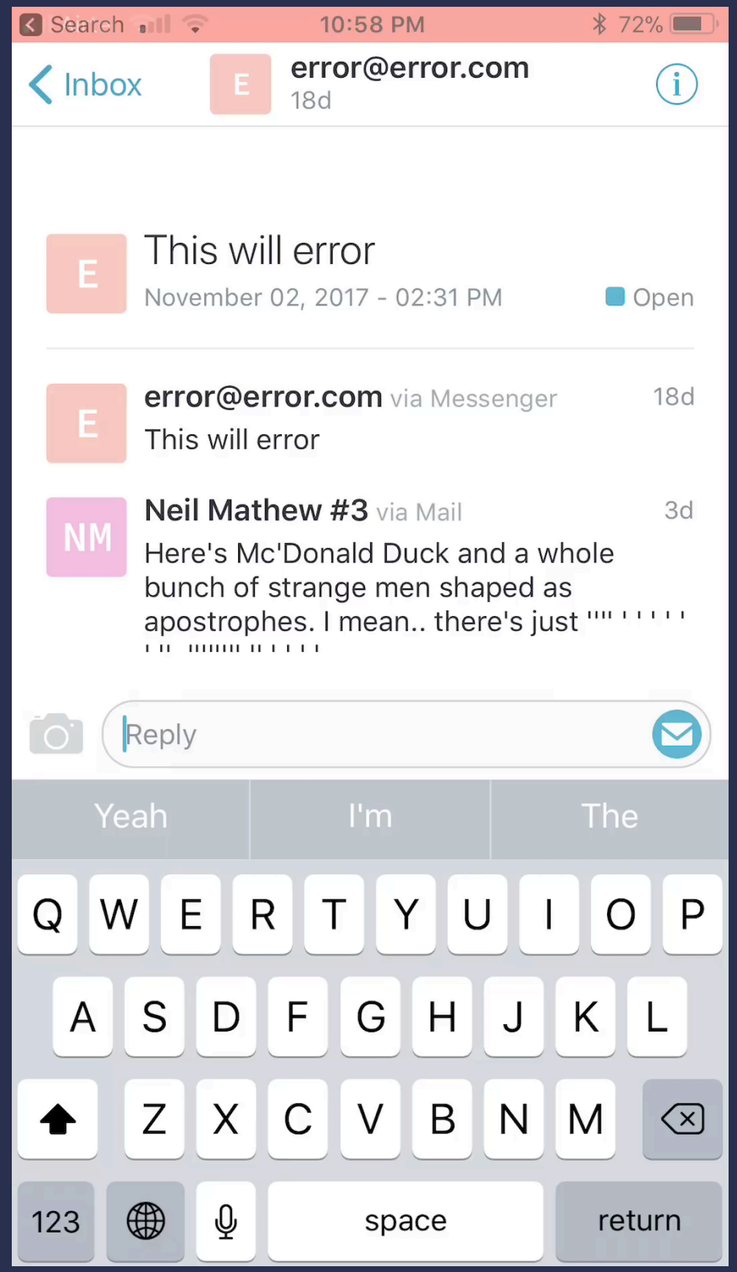
Body Cookies Headers (14) Tests Status: 200 OK Time: 553 ms Size: 603 B

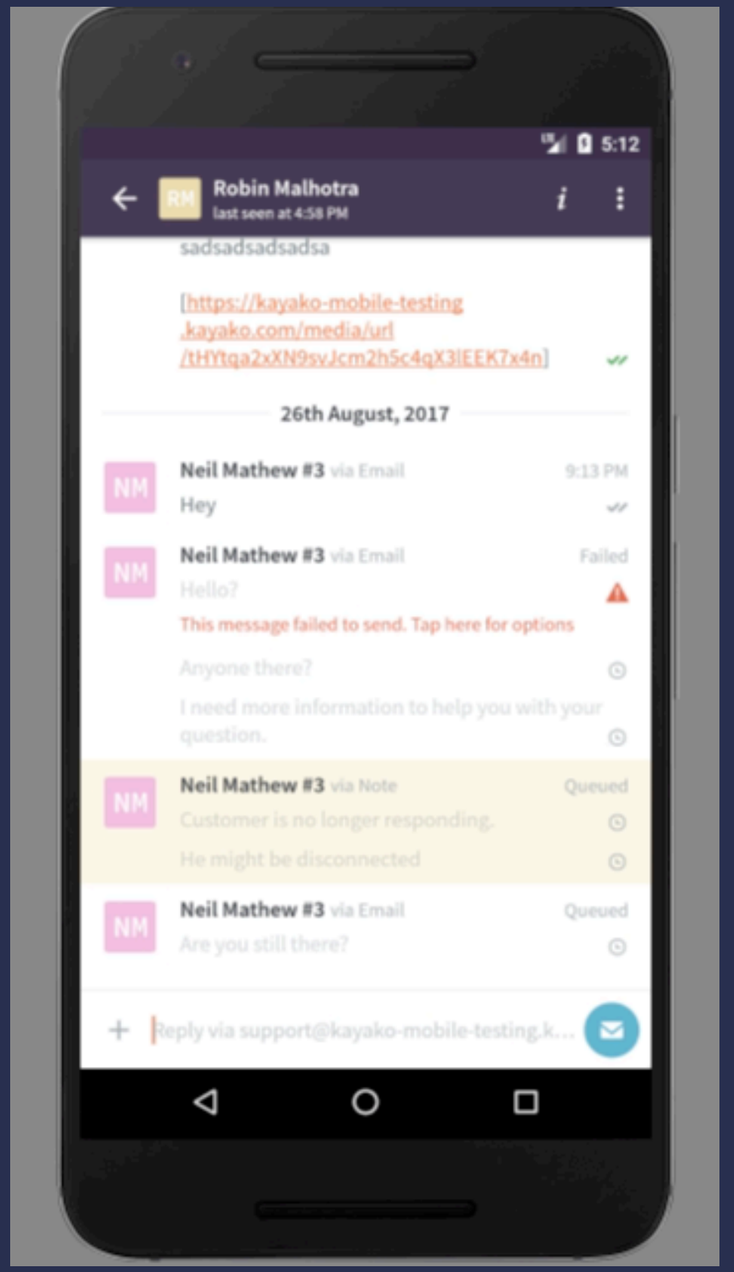
Pretty Raw Preview JSON

```
{
    "status": 200,
    "total_count": 3,
    "session_id": "jpItIUVdvq3cMMj9xaB5G1f946c083c3435114e83b9cb7a7c46ed2a3bac1amporFdZfw0PZuQymy3ChjHgY2"
}
```

MAXIMIZE RESPONSIVENESS
MINIMIZE REQUESTS

FIRST IDEA: A REQUEST QUEUE 🤔


Search
10:58 PM
72%
Inbox
E
error@error.com
18d
E
This will error
November 02, 2017 - 02:31 PM
Open
E
error@error.com via Messenger
18d
This will error
NM
Neil Mathew #3 via Mail
3d
Here's Mc'Donald Duck and a whole bunch of strange men shaped as apostrophes. I mean.. there's just
Reply
Yeah
I'm
The
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
123
space
return


5:12
Robin Malhotra
last seen at 4:58 PM
sadsadsadsadsa
[https://kayako-mobile-testing.kayako.com/media/url/tHYtqa2xXN9svJcm2h5c4qX3lEEK7x4n]
26th August, 2017
Neil Mathew #3 via Email
9:13 PM
Hey
Neil Mathew #3 via Email
Failed
Hello?
This message failed to send. Tap here for options
Anyone there?
I need more information to help you with your question.
Neil Mathew #3 via Note
Queued
Customer is no longer responding.
He might be disconnected
Neil Mathew #3 via Email
Queued
Are you still there?
Reply via support@kayako-mobile-testing.k...

CONSISTENCY 💯
LATENCY 😞

STATE

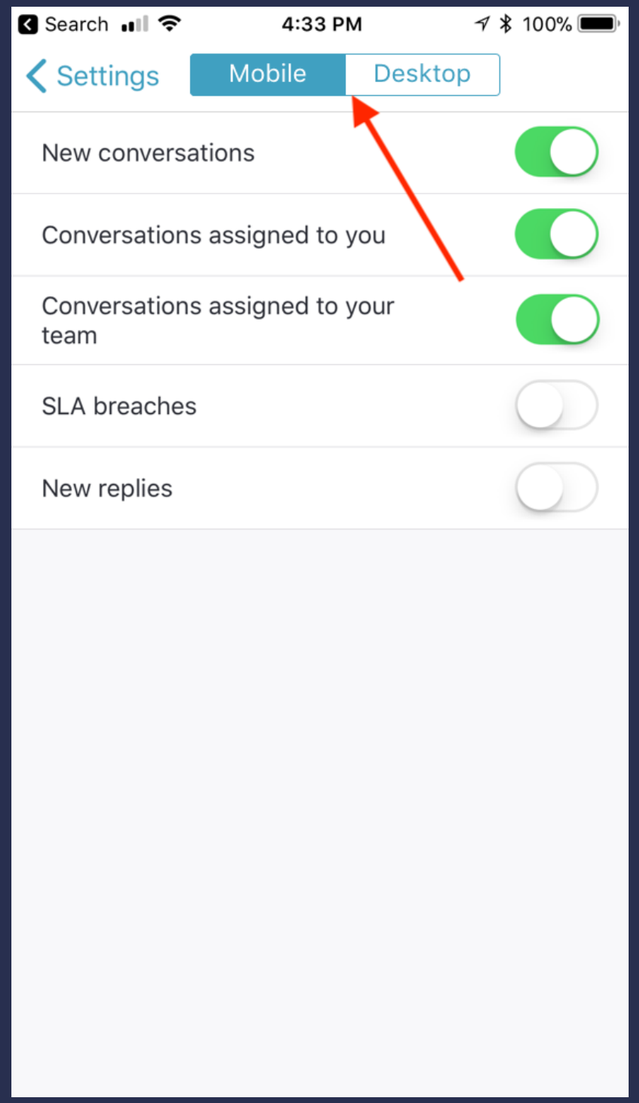
1. Dictionary of `[id: Bool]`
2. only send a dictionary if state has actually changed

IMPERATIVE APPROACH

- It. Was. Hell.

```
let timer: Timer
let request: DataRequest

// every time state actually changed
{
    timer.invalidate()
    self.request = client.createNewRequest()
    // wait for that callback to complete
}
```

Imagine if we added desktop preferences (sizes exaggerated for visibility reasons 😉)



```
let timer: Timer
let request: DataRequest
```

```
let mobileTimer: Timer
let mobileRequest: DataRequest
let desktopTimer: Timer
let desktopRequest: DataRequest

// and the checks would be worse too
```

This gets untenable very quickly

WITH RX

```
let throttledSwitch = switchSignals
    .asObservable()
    .debounce(3.0, scheduler: MainScheduler.instance)
    .distinctUntilChanged { (array1, array2) -> Bool in
    if array1.count != array2.count {
        return false
    }

    for (elem1, elem2) in zip(array1, array2) {
        if elem1.1 != elem2.1 {
            return false
        }
    }
    return true
}

throttledSwitch
    .subscribe(onNext: { ([(NotificationPreference, Bool)]) in ⚠️ R
        //send request
    })
```

and if we wanted to cancel requests automagically ✨ , we have our good old friend `flatMapLatest`

```
throttledSwitch
    .flatMapLatest(NotificationPreferencesViewController.buildRequest)
    .subscribe(onNext: { ([NotificationPreference]) in ⚠️ Result of call to 'sub
        code
    })
```

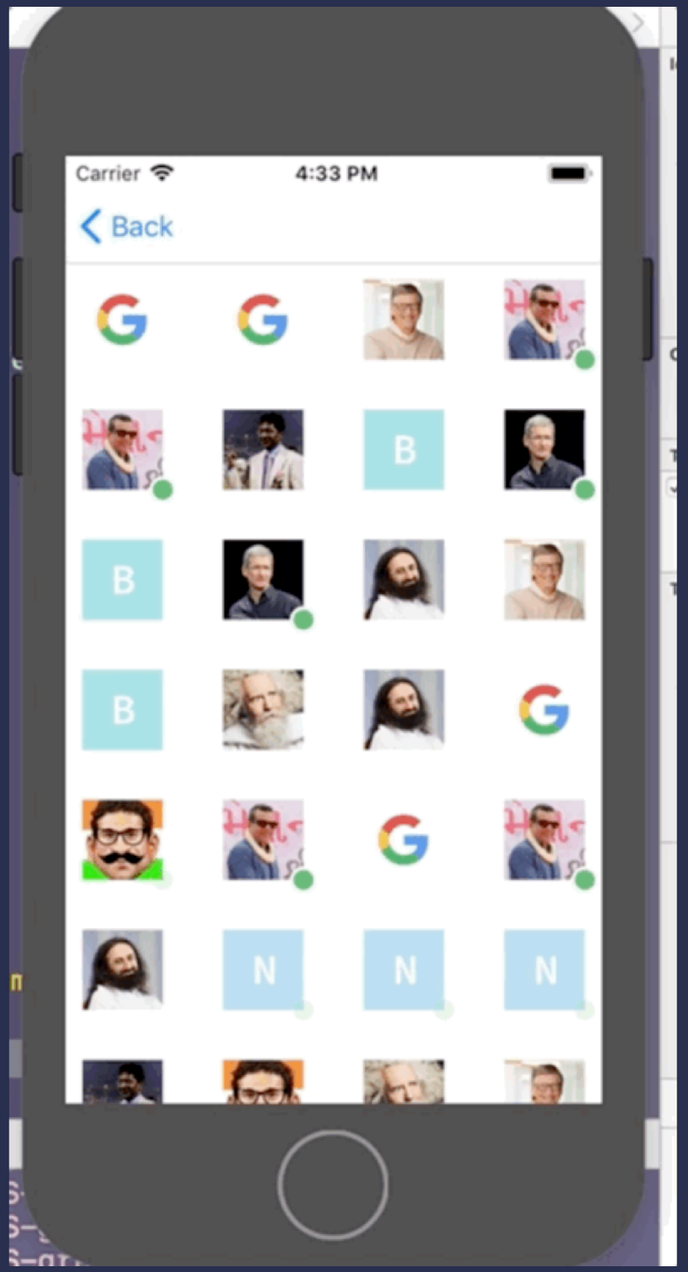
ONLINE INDICATORS

Meet Crusty

Don't call him "Jerome"

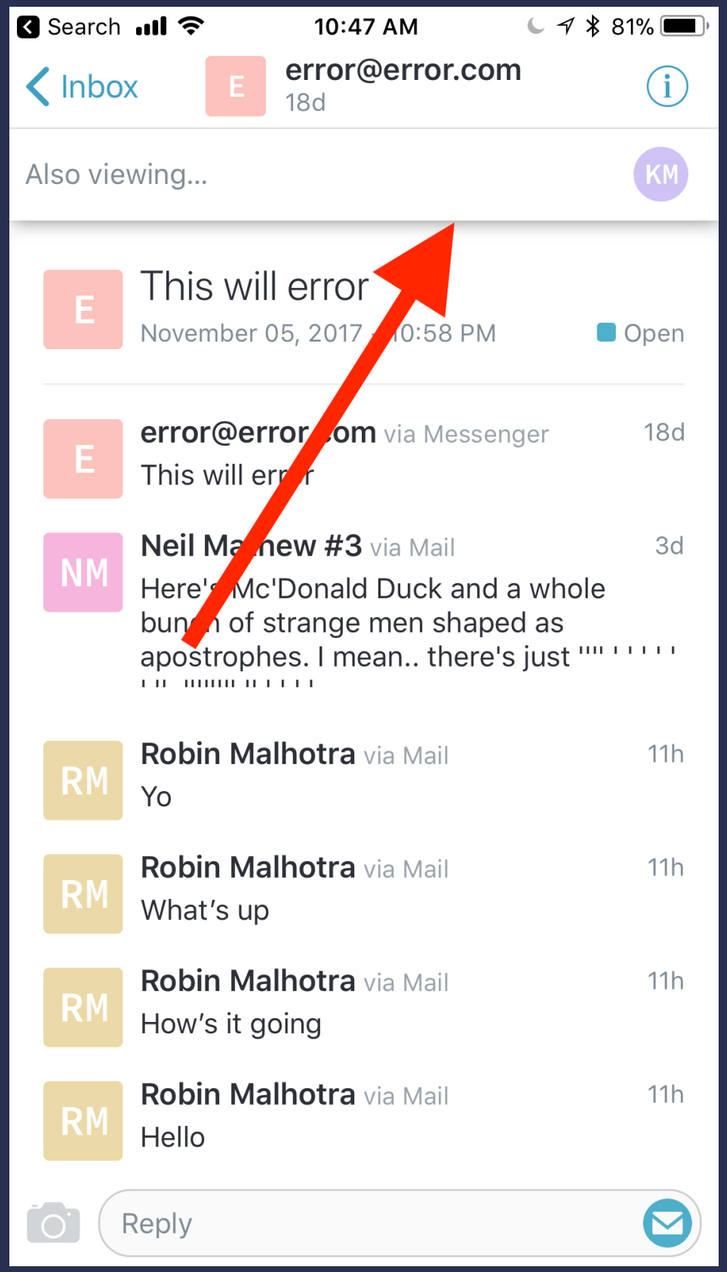
```
protocol OnlineUpdateDelegate: class {
    func subscribeToState(with callback: (OnlineState) -> Void)
}


class SomeView: UIView {

    weak var delegate: OnlineUpdateDelegate?

    override init(frame: CGRect) {
        super.init(frame: frame)
        self.delegate?.subscribeToState(with: { (state) in
            //update view
        })
    }
```

```
let disposeBag = DisposeBag()
let onlineSubscription: Observable<OnlineState>?
let avatar: Avatar

init(_ avatar: Avatar, onlineSubscription: Observable<OnlineState>? = nil) {
    self.onlineSubscription = onlineSubscription
    self.avatar = avatar
    super.init()
}
```

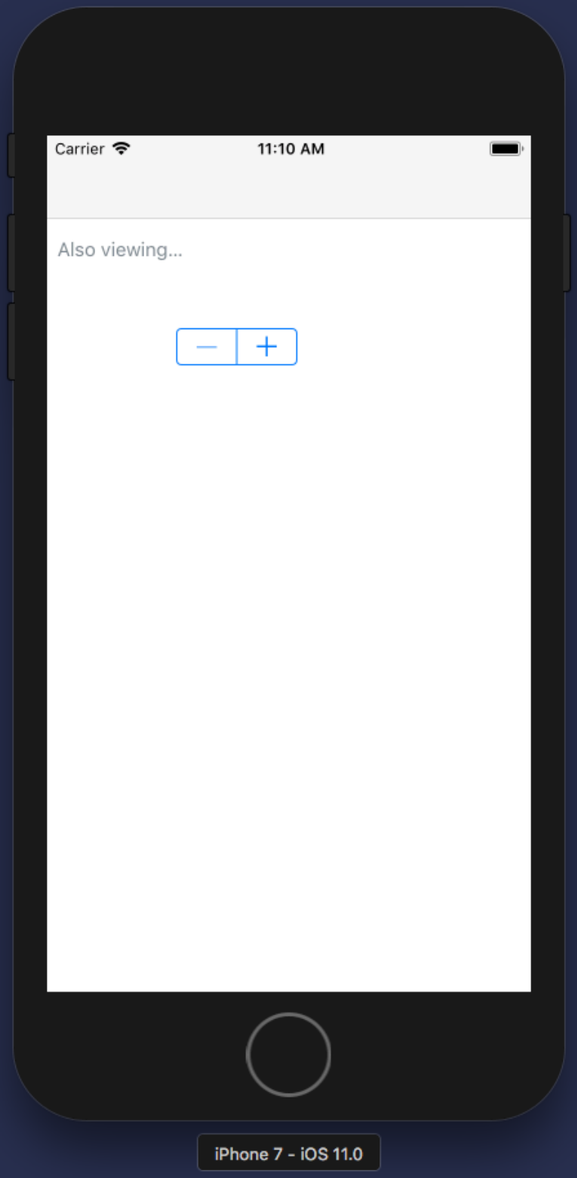

Carrier
4:33 PM
Back
B
B
B
N
N
N

TESTABILITY + PROTOTYPING

= 

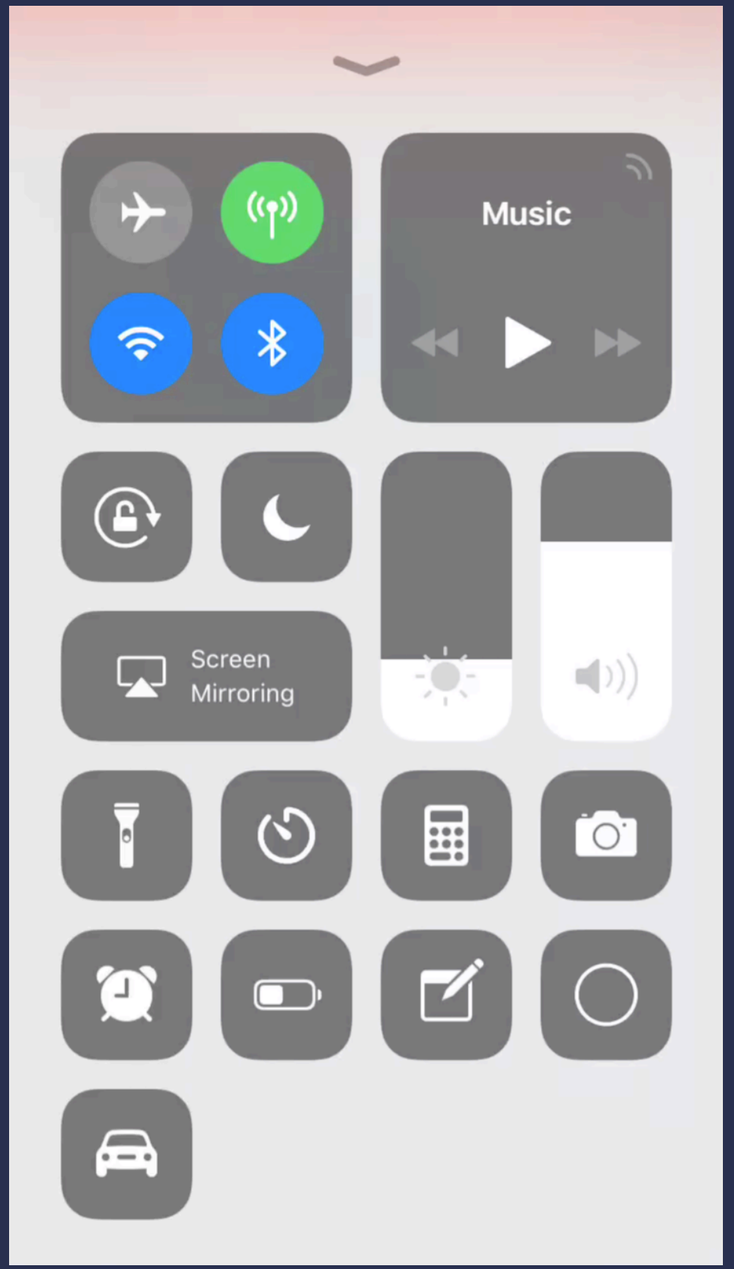

Search
10:47 AM
81%
Inbox
E
error@error.com
18d
Also viewing...
KM
E
This will error
November 05, 2017 10:58 PM
Open
E
error@error.com via Messenger
18d
This will error
NM
Neil Mathew #3 via Mail
3d
Here's Mc'Donald Duck and a whole bunch of strange men shaped as apostrophes. I mean.. there's just '''' ' ' ' ' ' ' ' ' ''''''' ' ' ' ' '
RM
Robin Malhotra via Mail
11h
Yo
RM
Robin Malhotra via Mail
11h
What's up
RM
Robin Malhotra via Mail
11h
How's it going
RM
Robin Malhotra via Mail
11h
Hello
Reply

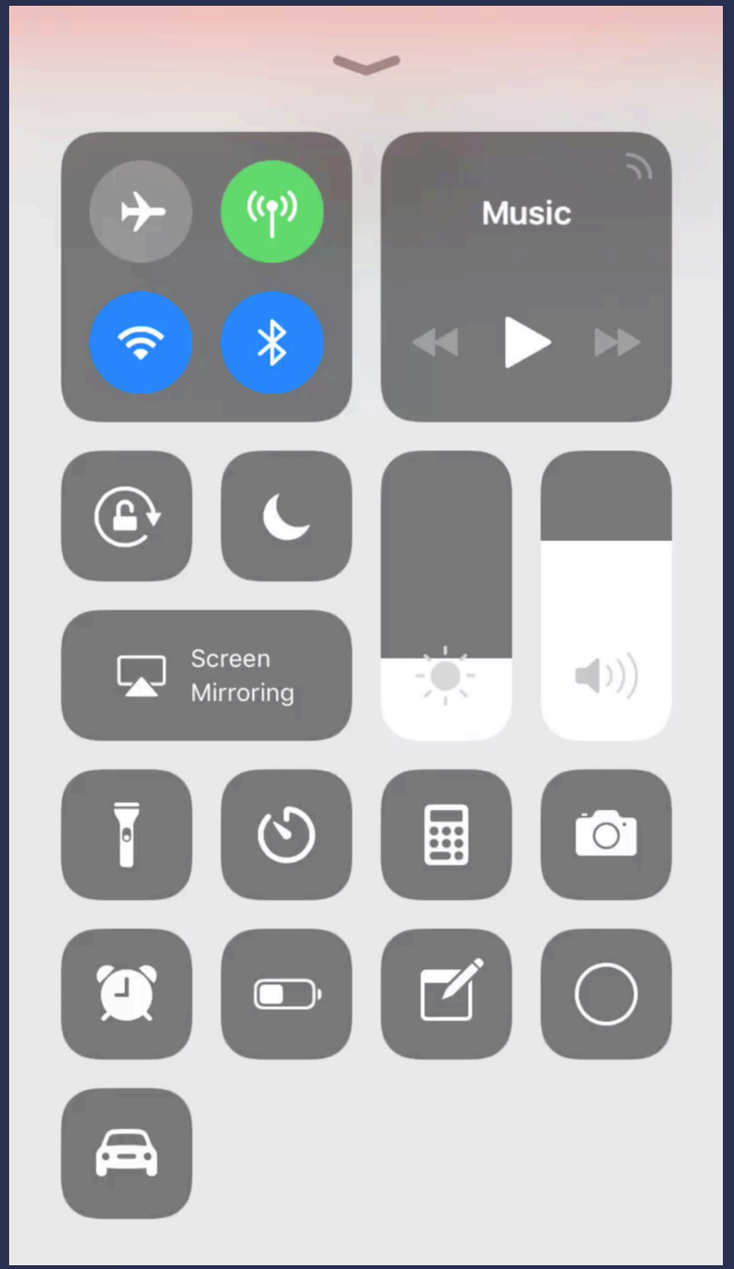
```
// Presence bar init
public init(observable: Observable<[URL]>, text: String) {
    self.avatarStack = PresenceAvatars(observable: truncatedAvatars.asObservable())
    self.numberNode = NumberNode(overFlowNumber.asObservable())
    super.init()
```


Carrier
11:10 AM
Also viewing...
iPhone 7 - iOS 11.0

SUBJECTS

```
self.node = PresenceBar(observable: subject.asObservable(), text: "Also viewing…")
```


Music
Screen
Mirroring


Music
Screen
Mirroring

NETWORKING (AGAIN!)

THINGS I WANT IN MY NETWORKING CLIENT

1. Testability
2. Should do exactly what it says. No 🐒 patching.

Average API Client (with Foundation)

```
8
9  class APIClient {
0      var accessToken = ""
1      let session: URLSession
2
3      init(session: URLSession) {
4          self.session = session
5      }
6
7      func getFoos(onGettingFoos closure: @escaping (Result<[Foo]>) -> Void)
         -> URLSessionDataTask {
8          let request = URLRequest.init(url: fooURL!)
9          let task = session.dataTask(with: request) { (data, response,
             error) in
0              //parse foos
1              //run closure
2          }
3          task.resume()
4          return task
5      }
6  }
```

```swift
func createBody(parameters: [String: String], boundary: String, file: AttachmentCreationModel?) -> Data {
        let body = NSMutableData()

        let boundaryPrefix = "--\(boundary)\r\n"

        for (key, value) in parameters {
            body.appendString(boundaryPrefix)
            body.appendString("Content-Disposition: form-data; name=\"\(key)\"\r\n\r\n")
            body.appendString("\(value)\r\n")
        }

        body.appendString(boundaryPrefix)

        if let file = file {
            body.appendString("Content-Disposition: form-data; name=\"files[\(index)]\"; filename=\"\(file.filename)\"\r\n")
            body.appendString("Content-Type: \(file.mimeType)\r\n\r\n")
            body.append(file.data)
            body.appendString("\r\n")
            body.appendString("--".appending(boundary.appending("--")))
        }

        return body as Data
    }
```

URLSession Features 😞👎

URLSession Testability 💯

Average API Client (with Alamofire)

```
10
11 class APIClient {
12
13   enum Result<T> {
14     case success(T)
15     case failure(Error)
16   }
17
18   struct Foo {
19   }
20
21   var accessToken = ""
22
23   func getFoos(onGettingFoos closure: @escaping (Result<[Foo]>) -> Void)  -> DataRequest {
24     let request = URLRequest.init(url: fooURL!)
25     return Alamofire.request(request).responseJSON(completionHandler: { (response) in
26       switch response.result {
27       case .success(_):
28         //parse foos
29         break
30       case .failure(_):
31         //error
32         break
33       }
34     })
35   }
36 }
```

RequestRetrier

The `RequestRetrier` protocol allows a `Request` that encountered an `Error` while being executed to be retried. When using both the `RequestAdapter` and `RequestRetrier` protocols together, you can create credential refresh systems for OAuth1, OAuth2, Basic Auth and even exponential backoff retry policies. The possibilities are endless. Here's an example of how you could implement a refresh flow for OAuth2 access tokens.

> **DISCLAIMER:** This is **NOT** a global `OAuth2` solution. It is merely an example demonstrating how one could use the `RequestAdapter` in conjunction with the `RequestRetrier` to create a thread-safe refresh system.

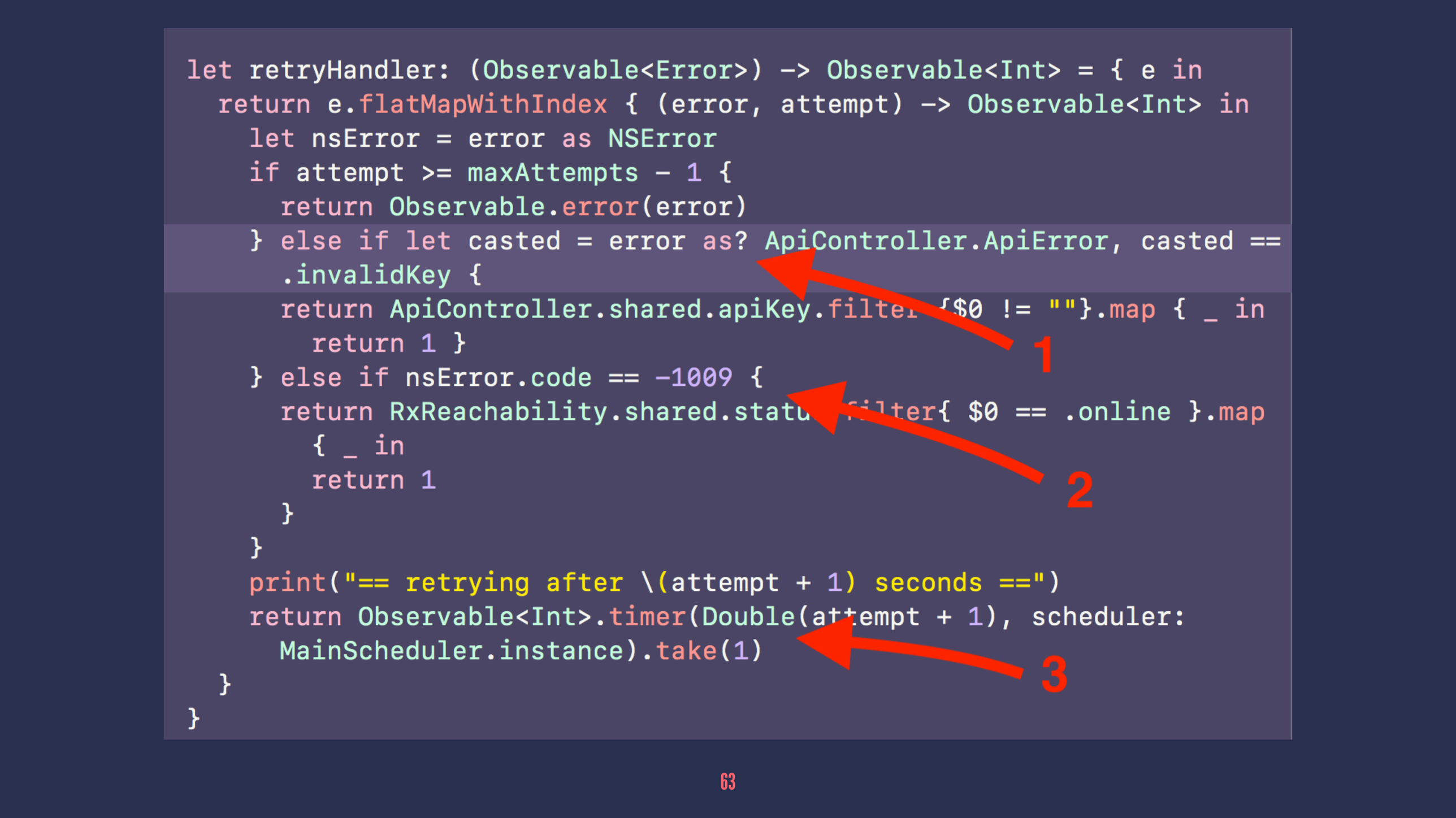
> To reiterate, **do NOT copy** this sample code and drop it into a production application. This is merely an example. Each authentication system must be tailored to a particular platform and authentication type.

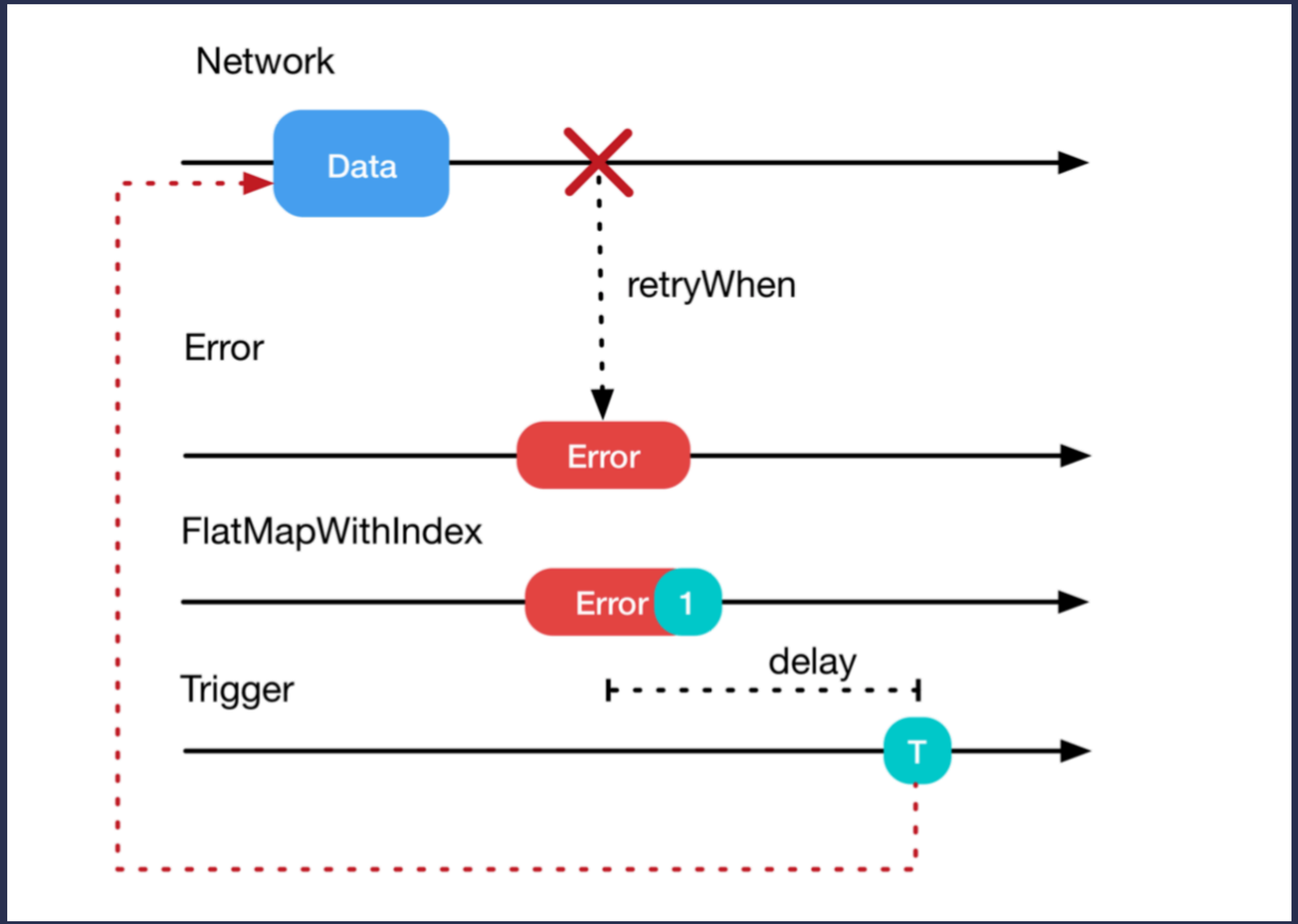
```
class OAuth2Handler: RequestAdapter, RequestRetrier {
    private typealias RefreshCompletion = (_ succeeded: Bool, _ accessToken: String?, _ refreshToken: String?) -> Void

    private let sessionManager: SessionManager = {
        let configuration = URLSessionConfiguration.default
        configuration.httpAdditionalHeaders = SessionManager.defaultHTTPHeaders

        return SessionManager(configuration: configuration)
    }()

    private let lock = NSLock()

    private var clientID: String
    private var baseURLString: String
    private var accessToken: String
    private var refreshToken: String

    private var isRefreshing = false
    private var requestsToRetry: [RequestRetryCompletion] = []

    // MARK: - Initialization

    public init(clientID: String, baseURLString: String, accessToken: String, refreshToken: String) {
        self.clientID = clientID
        self.baseURLString = baseURLString
        self.accessToken = accessToken
        self.refreshToken = refreshToken
    }

    // MARK: - RequestAdapter

    func adapt(_ urlRequest: URLRequest) throws -> URLRequest {
        if let urlString = urlRequest.url?.absoluteString, urlString.hasPrefix(baseURLString) {
            var urlRequest = urlRequest
            urlRequest.setValue("Bearer " + accessToken, forHTTPHeaderField: "Authorization")
            return urlRequest
        }

        return urlRequest
    }

    // MARK: - RequestRetrier

    func should(_ manager: SessionManager, retry request: Request, with error: Error, completion: @escaping RequestRetryCompletion) {
        lock.lock() ; defer { lock.unlock() }

        if let response = request.task?.response as? HTTPURLResponse, response.statusCode == 401 {
            requestsToRetry.append(completion)

            if !isRefreshing {
                refreshTokens { [weak self] succeeded, accessToken, refreshToken in
                    guard let strongSelf = self else { return }

                    strongSelf.lock.lock() ; defer { strongSelf.lock.unlock() }

                    if let accessToken = accessToken, let refreshToken = refreshToken {
                        strongSelf.accessToken = accessToken
                        strongSelf.refreshToken = refreshToken
                    }

                    strongSelf.requestsToRetry.forEach { $0(succeeded, 0.0) }
                    strongSelf.requestsToRetry.removeAll()
                }
            }
        } else {
            completion(false, 0.0)
        }
    }

    // MARK: - Private - Refresh Tokens

    private func refreshTokens(completion: @escaping RefreshCompletion) {
        guard !isRefreshing else { return }

        isRefreshing = true

        let urlString = "\(baseURLString)/oauth2/token"

        let parameters: [String: Any] = [
            "access_token": accessToken,
            "refresh_token": refreshToken,
            "client_id": clientID,
            "grant_type": "refresh_token"
        ]

        sessionManager.request(urlString, method: .post, parameters: parameters, encoding: JSONEncoding.default)
            .responseJSON { [weak self] response in
                guard let strongSelf = self else { return }

                if
                    let json = response.result.value as? [String: Any],
                    let accessToken = json["access_token"] as? String,
                    let refreshToken = json["refresh_token"] as? String
                {
                    completion(true, accessToken, refreshToken)
                } else {
                    completion(false, nil, nil)
                }

                strongSelf.isRefreshing = false
            }
    }
}
let baseURLString = "https://some.domain-behind-oauth2.com"

let oauthHandler = OAuth2Handler(
    clientID: "12345678",
    baseURLString: baseURLString,
    accessToken: "abcd1234",
    refreshToken: "ef56789a"
)

let sessionManager = SessionManager()
sessionManager.adapter = oauthHandler
sessionManager.retrier = oauthHandler

let urlString = "\(baseURLString)/some/endpoint"

sessionManager.request(urlString).validate().responseJSON { response in
    debugPrint(response)
}
```

Alamofire Features 💯

Alamofire Testability 😞👎

```
let retryHandler: (Observable<Error>) -> Observable<Int> = { e in
  return e.flatMapWithIndex { (error, attempt) -> Observable<Int> in
    let nsError = error as NSError
    if attempt >= maxAttempts - 1 {
      return Observable.error(error)
    } else if let casted = error as? ApiController.ApiError, casted ==
      .invalidKey {
      return ApiController.shared.apiKey.filter {$0 != ""}.map { _ in
        return 1 }
    } else if nsError.code == -1009 {
      return RxReachability.shared.status.filter{ $0 == .online }.map
        { _ in
        return 1
      }
    }
    print("== retrying after \(attempt + 1) seconds ==")
    return Observable<Int>.timer(Double(attempt + 1), scheduler:
      MainScheduler.instance).take(1)
  }
}
```



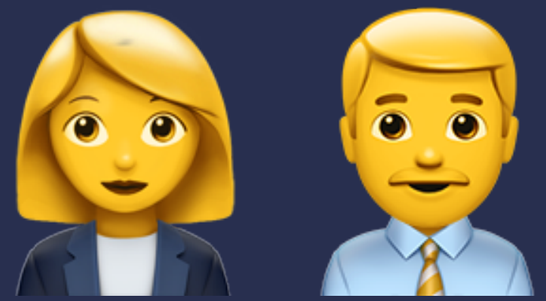

Network
Data
retryWhen
Error
Error
FlatMapWithIndex
Error
1
delay
Trigger
T


NETWORKING

But I don't wanna rewrite my code 😢

```
extension Case {
    typealias TagsResult = Result<[String], TagError>
    static func tagsSearch(_ searchQuery: String) -> Observable<[String]> {

        return Observable.create({ (observer) -> Disposable in
            guard let keychainInfo = KeychainInfo.getKeychainInfo() else {
                observer.onError(RxError.noElements)
                return Disposables.create()
            }

            let req = Client.sharedClient.searchForTags(withURL:
              keychainInfo.baseURL, sessionID: keychainInfo.sessionID, searchQuery:
              searchQuery) { (tags) in
                observer.onNext(tags.map{ $0.name })
            }

            return Disposables.create {
                req.cancel()
            }
        })
    }

}
```

WHAT ABOUT MANAGEMENT



MAKE THE CASE FOR IT!!

FINAL WORDS

RxSwift is *not* always the best way to solve a problem

BUILD GREAT APPS