When To Go Native Over Javascript

Harry Tormey

What’s this talk about?

What’s this talk about?

- Tooling
- Third Party Libraries
- Writing Native Code

Tooling

Expo

What is Expo?

“Tools that enable developers to build and share truly native apps that work across both iOS and Android.”

How does Native Code affect expo?

“Standard Expo projects don’t support custom native code, including third-party libraries which require custom native components. In an Expo project, you only write pure JS.”

CodePush

What is CodePush?

“A cloud service that enables React Native developers to deploy mobile app updates directly to their users’ devices.”

How does Native Code affect CodePush?

“Any product changes which touch native code (e.g. modifying your AppDelegate.m/MainActivity.java file, adding a new plugin) cannot be distributed via CodePush, and therefore, must be updated via the appropriate store(s)..”

Third Party Libraries

Navigation

Which Navigation library should I use?

- React Navigation
- React Native Navigation
- React Native Router Flux
- React Router Native
- Native Navigation

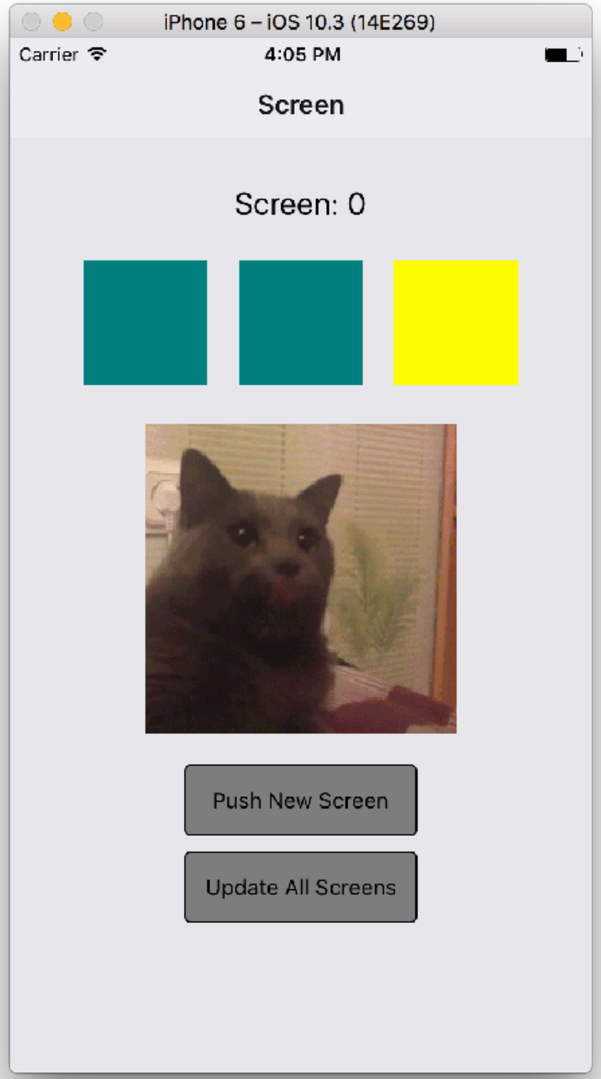
Native

- React Native Navigation
- Native Navigation

Javascript

- React Navigation
- React Native Router Flux
- React Router Native

Example App

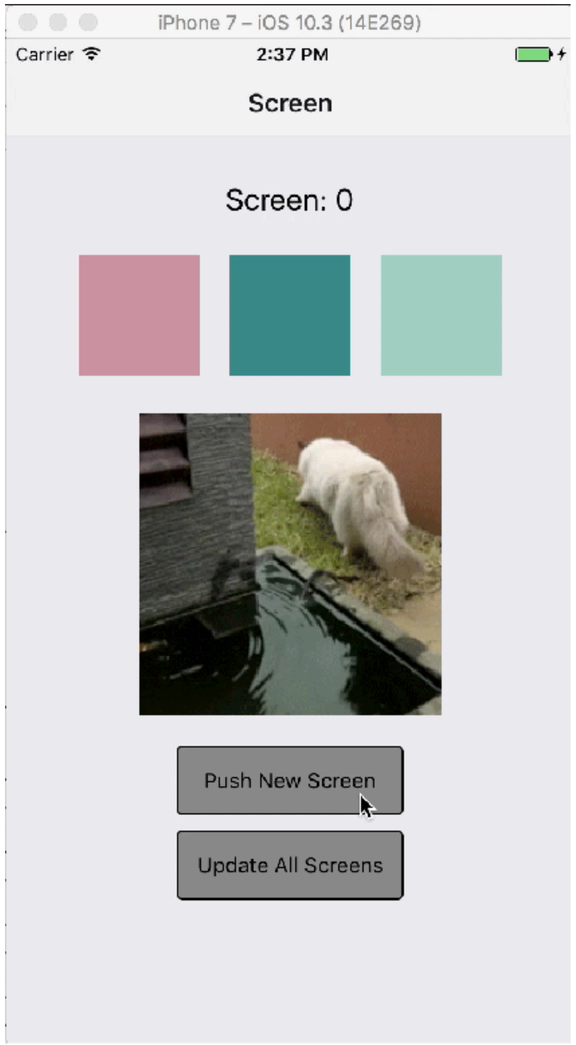

iPhone 6 – iOS 10.3 (14E269)
Carrier
4:05 PM
Screen
Screen: 0
Push New Screen
Update All Screens

Example App

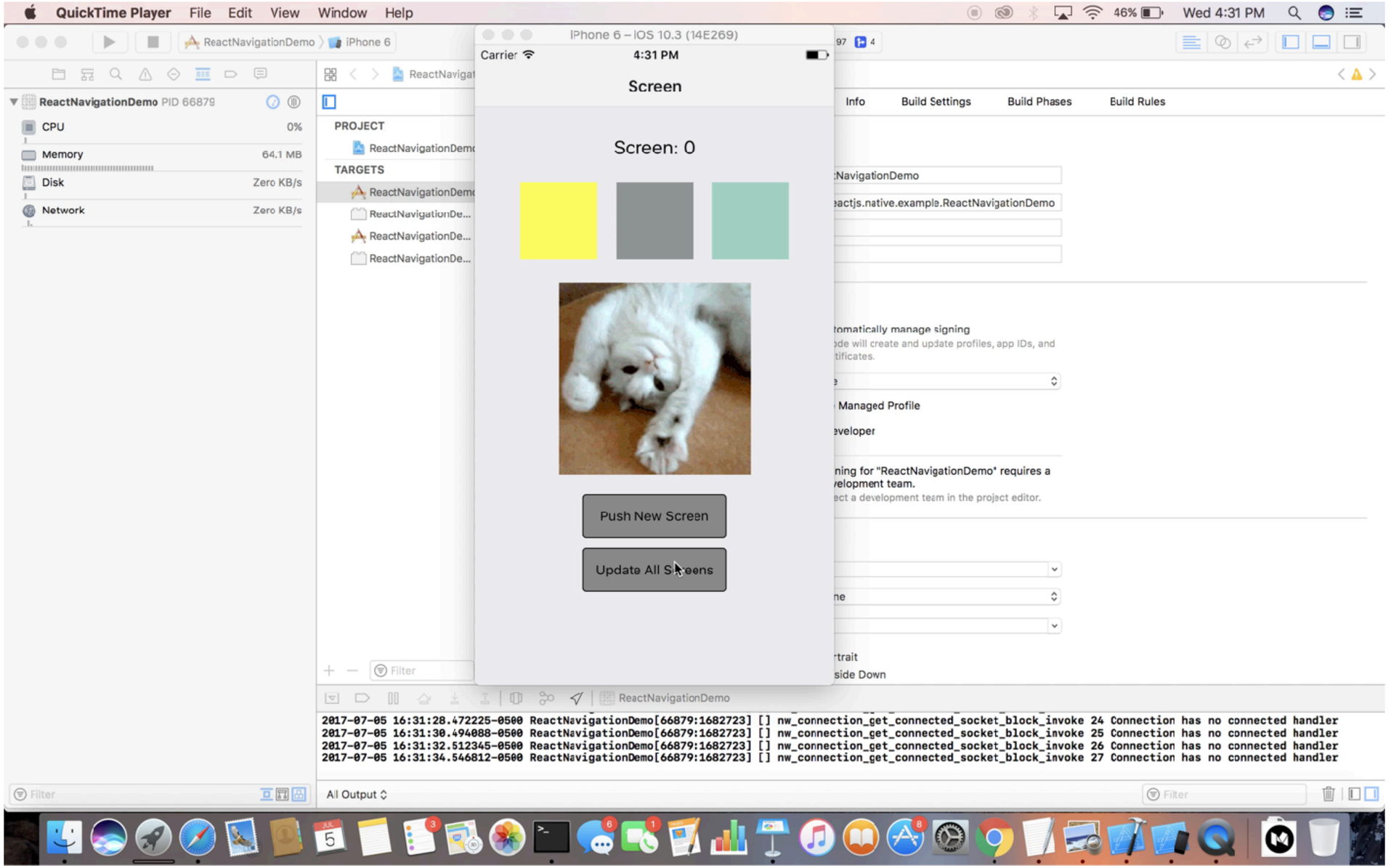
- React Navigation:
  - https://github.com/hgale/ReactNavigationDemo
- React Native Navigation:
  - https://github.com/hgale/ReactNativeNavigationDemo

What does it do?

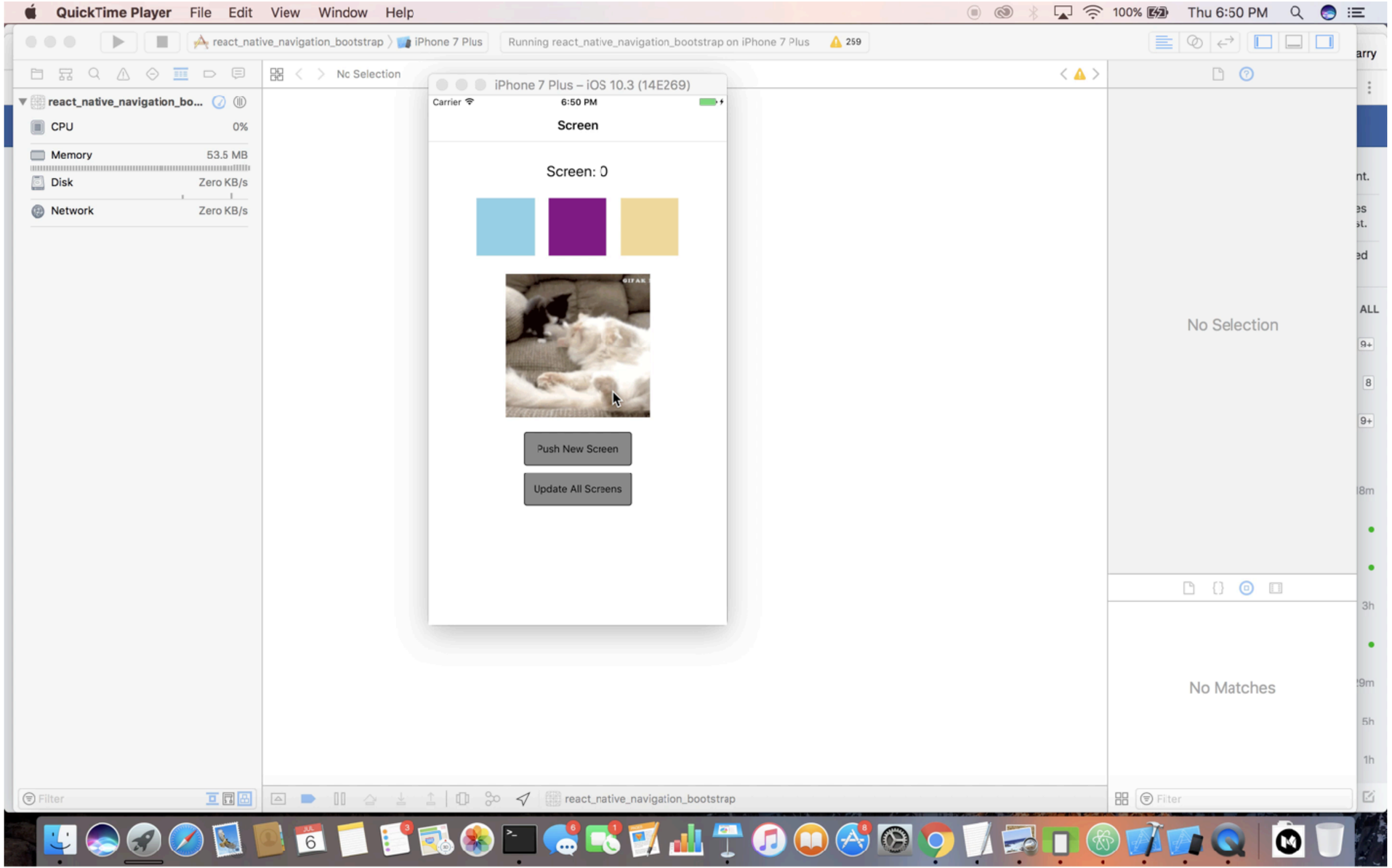
Example App


iPhone 7 – iOS 10.3 (14E269)
Carrier
2:37 PM
Screen
Screen: 0
Push New Screen
Update All Screens

React Navigation



React Native Navigation



Why does this matter?

Accessibility

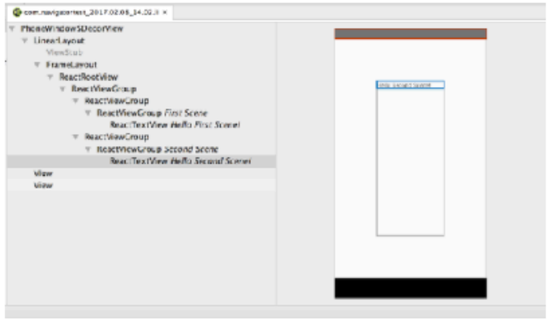
1

Navigator breaks Android screen reader [accessibility]

Erica Cooksey

When a new scene is pushed onto the Navigator, the previous scene is still present in the view tree. Thus, if a visually-impaired user enables screen reading through their device accessibility settings, views from both the previous and the current scene are read interleaved to the user. As an aside, rendering all components on the backstack creates significant GPU overdraw issues.

Attached is a screenshot of the "Additional Scenes" sample code in Android's layout inspector. The only change that I made was adding `accessibilityLabel`s to each scene ("First Scene" and "Second Scene", respectively).



February 8, 2017

Accessibility

- https://react-native.canny.io/feature-requests/p/navigator-breaks-android-screen-reader-accessibility

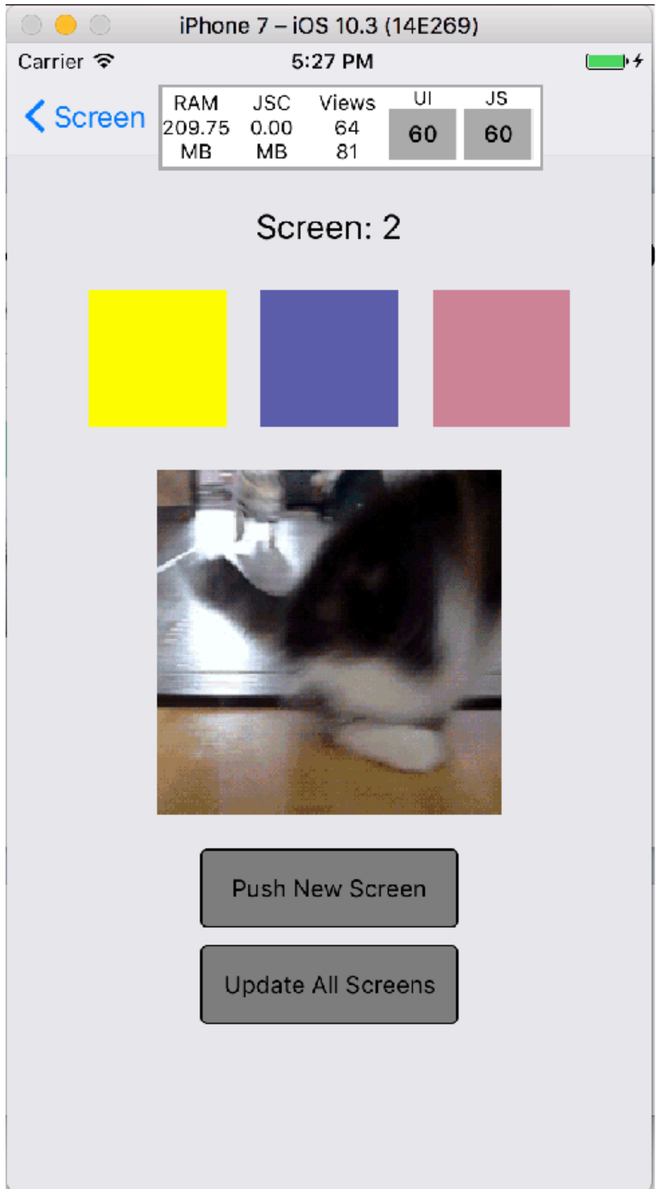
Whats the difference?

Native Feel

Performance

React Native Navigation

React Navigation



Native Libraries Pros

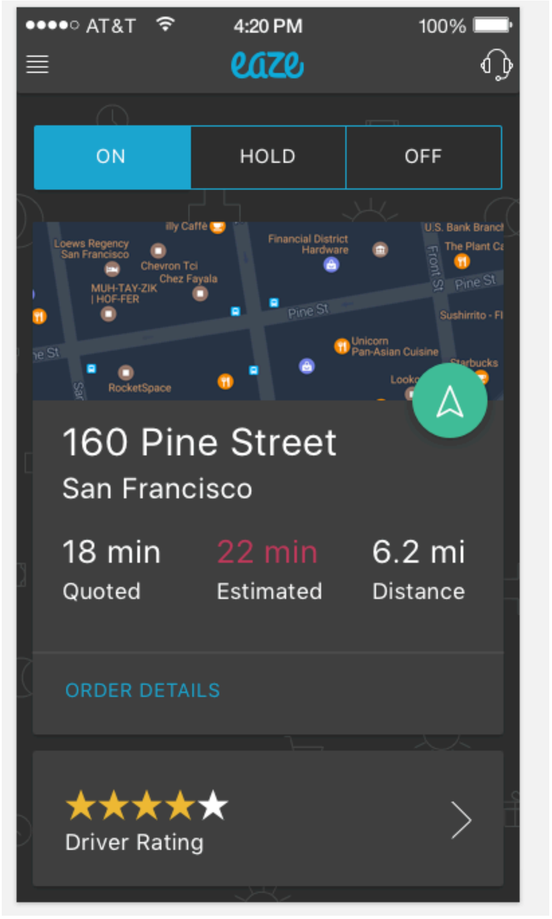
- Performance
- Feels/Behaves “Natively”

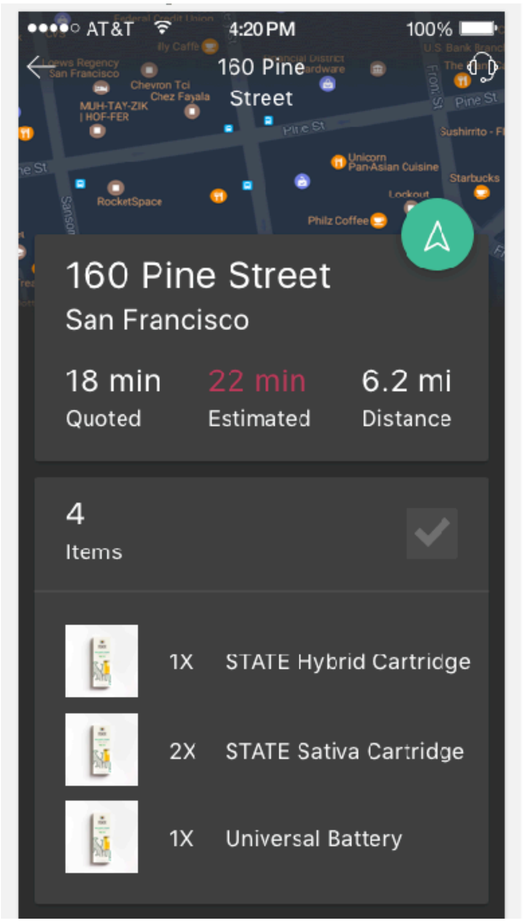
Native Libraries Cons

- Can Require Some Native Knowledge
  - Setup can be tricky
  - Native linking/build errors
- iOS/Android behave differently

Native Code

What does the Driver app do?





Chrome File Edit View History Bookmarks People Window Help Tue 10:34 AM Harry

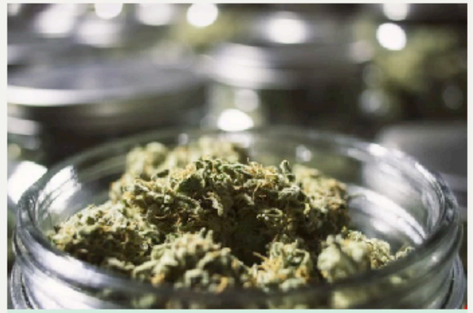
Secure https://development.eazeup.com/menu

eaze

VAPORIZERS EDIBLES PREROLLS FLOWERS CONCENTRATES ACCESSORIES

Sativa Hybrid Indica CBD

Vaporizers



Bloom Farms - Vape Pen Bat..

$22

Bloom Farms Blueberry Haze

23.3% THC | 50% CBD

$52

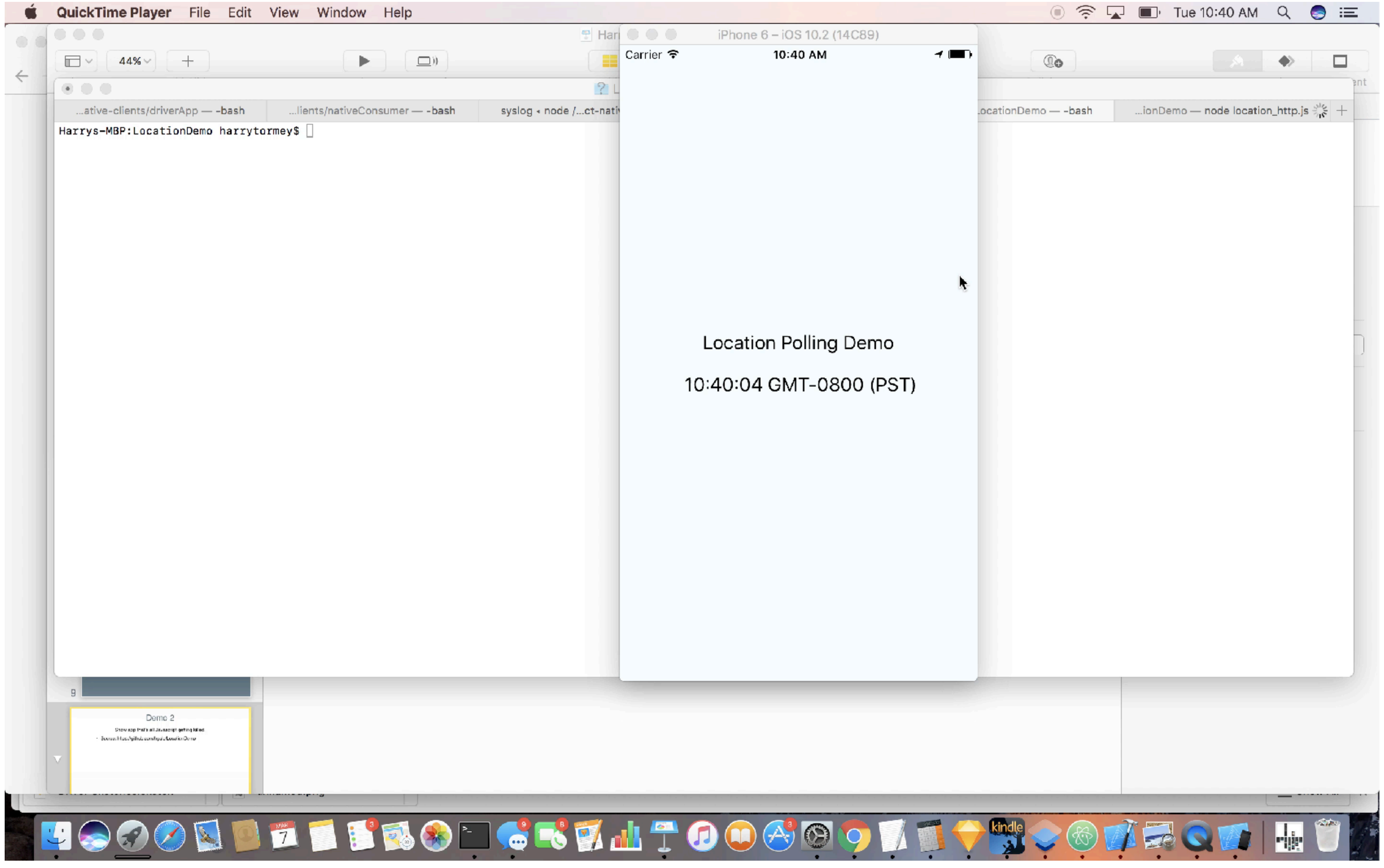
Grape Stomper Cartridge

62.2% THC | 1.8% CBD

$45

160 Pine St

Why not Javascript?


QuickTime Player File Edit View Window Help
Tue 10:40 AM
iPhone 6 – iOS 10.2 (14C89)
Carrier
10:40 AM
Location Polling Demo
10:40:04 GMT-0800 (PST)
Harrys-MBP:LocationDemo harrytormey$
...ionDemo — node location_http.js
Demo 2

Demo Source

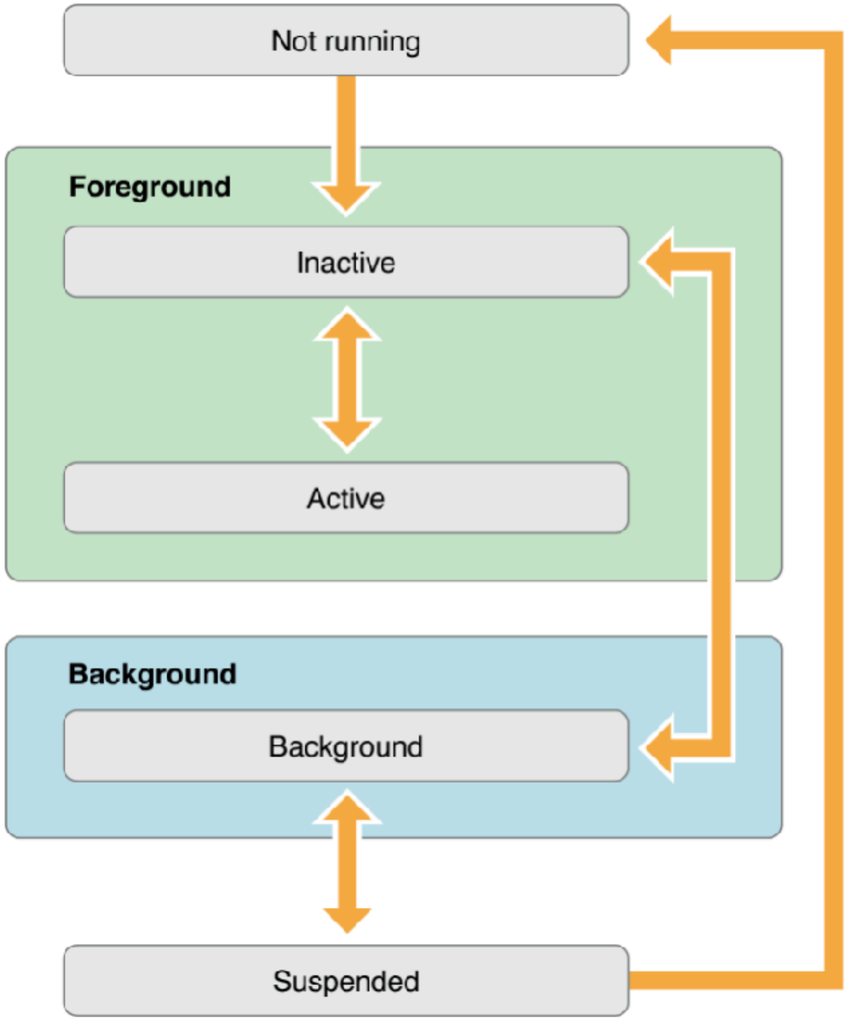
- https://github.com/hgale/LocationDemo

Why not?

```
1  import pole from 'pole'
2
3  this.setState({
4    pollLocation: pole({interval:
5                        LOCATION_POLL_FREQUENCY}, (
6                        callback) => {
7      let now =  new Date()
8      let data = JSON.stringify({'lat': 20,
9                 'long': 20, 'time': now })
10     fetch('http://localhost:8080', {
11       method: 'POST',
12       headers: {
13         'Accept': 'application/json',
14         'Content-Type': 'application/json',
15       },
16       body: data
17     })
18     this.updateTime(now)
19     callback()
20   })
21 })
```

iOS Background Modes

Figure 2-3 State changes in an iOS app



iOS Background Modes

State	Description
Not running	The app has not been launched or was running but was terminated by the system.
Inactive	The app is running in the foreground but is currently not receiving events. (It may be executing other code though.) An app usually stays in this state only briefly as it transitions to a different state.
Active	The app is running in the foreground and is receiving events. This is the normal mode for foreground apps.
Background	The app is in the background and executing code. Most apps enter this state briefly on their way to being suspended. However, an app that requests extra execution time may remain in this state for a period of time. In addition, an app being launched directly into the background enters this state instead of the inactive state. For information about how to execute code while in the background, see Background Execution.
Suspended	The app is in the background but is not executing code. The system moves apps to this state automatically and does not notify them before doing so. While suspended, an app remains in memory but does not execute any code. When a low-memory condition occurs, the system may purge suspended apps without notice to make more space for the foreground app.

Testing

- Test on a Device
- Test with app backgrounded and screen locked
- Test with app backgrounded for ~8 minutes on iOS
- Test 10-20 minutes on Android
- Test with device unplugged and not on wifi

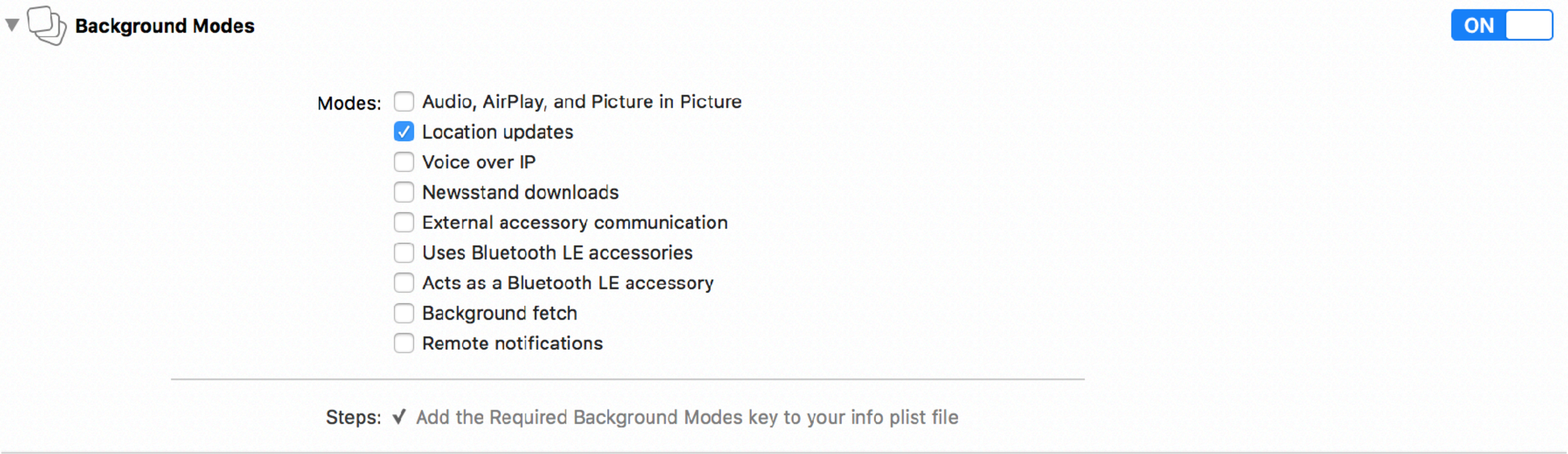
What about react-native-background-timer?

```
1
2   import BackgroundTimer from 'react-native-background-timer'
3
4   const intervalId = BackgroundTimer.setInterval(() => {
5     let now =  new Date()
6     let data = JSON.stringify({'lat': 20, 'long': 20, 'time': now })
7     fetch('http://192.168.107.24:8080', {
8       method: 'POST',
9       headers: {
10        'Accept': 'application/json',
11        'Content-Type': 'application/json',
12      },
13      body: data
14    })
15    this.updateTime(now)
16  }, LOCATION_POLL_FREQUENCY)
```

No

- On iOS Based on Executing Finite-Length Tasks.
- See: https://github.com/ocetnik/react-native-background-timer/blob/master/ios/RNBackgroundTimer.m
- How much time you get after your app gets backgrounded is determined is not guaranteed. App will be killed after roughly ~3-5 minutes.

iOS Background Modes



iOS Background Modes

NSLocationAlwaysUsageDescription

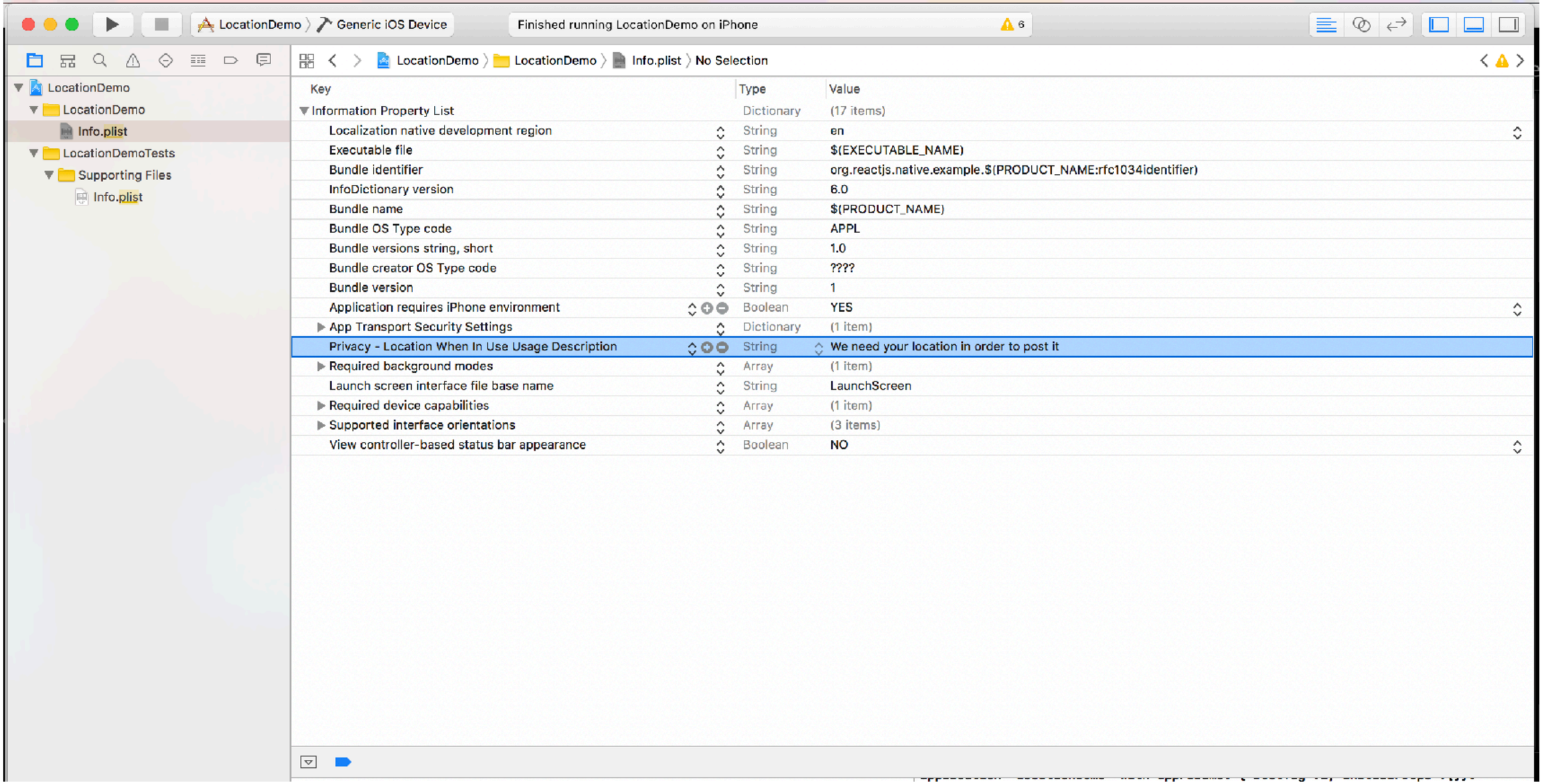
`NSLocationAlwaysUsageDescription` (`String` – iOS) This key lets you describe the reason your app accesses the user's location information at all times. Include this key when your app uses location services in a potentially nonobvious way while running in the foreground or the background. For example, a social app might include this key when it uses location information to track the user's location and display other users that are nearby. In this case, the fact that the app is tracking the user's location might not be readily apparent. The system includes the value of this key in the alert panel displayed to the user when requesting permission to use location services.

Important: To protect user privacy, an iOS app linked on or after iOS 10.0, and which accesses the user's location information, must statically declare the intent to do so. Include the NSLocationAlwaysUsageDescription key in your app's `Info.plist` file and provide a purpose string for this key. If your app attempts to access the user's location information without a corresponding purpose string, your app exits.

This key is required when you use the `requestAlwaysAuthorization` method of the `CLLocationManager` class to request authorization for location services. If this key is not present and you call the `requestAlwaysAuthorization` method, the system ignores your request and prevents your app from using location services.

This key is supported in iOS 8.0 and later.

iOS Background Modes



```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)
   launchOptions {
    self.locationManager = [[CLLocationManager alloc] init];
    self.locationManager.pausesLocationUpdatesAutomatically = NO;
    self.locationManager.allowsBackgroundLocationUpdates = YES;
    self.locationManager.delegate=self;
    self.locationManager.desiredAccuracy=kCLLocationAccuracyBest;
    [self.locationManager requestAlwaysAuthorization];
    return YES;
}

#pragma - CLLocationManagerDelegate

- (void)locationManager:(CLLocationManager *)manager didChangeAuthorizationStatus:(
   CLAuthorizationStatus)status {
    if(status==kCLAuthorizationStatusAuthorizedAlways)
        [self.locationManager startUpdatingLocation];
}

- (void)locationManager:(CLLocationManager *)manager didUpdateLocations:(NSArray *)locations {
    // Emit event
}

```

Does this work now?

```
1  import pole from 'pole'
2
3  this.setState({
4    pollLocation: pole({interval:
5                        LOCATION_POLL_FREQUENCY}, (
6                        callback) => {
7      let now =  new Date()
8      let data = JSON.stringify({'lat': 20,
9                  'long': 20, 'time': now })
10     fetch('http://localhost:8080', {
11       method: 'POST',
12       headers: {
13         'Accept': 'application/json',
14         'Content-Type': 'application/json',
15       },
16       body: data
17     })
18     this.updateTime(now)
19     callback()
20   })
21 })
```

No

React Native Timers don’t work in the background

- See Brent Vatne’s feature request: https://react-native.canny.io/feature-requests/p/background-timer-execution

```
RCT_EXPORT_METHOD(startPostingLocationTo:(NSString *)url everyInterval:(NSInteger)interval) {
    if (!url || url.length == 0 || interval <= 0) {
        return;
    }

    self.url = [NSURL URLWithString:url];

    double seconds = interval / 1000.0;

    if (self.pollingTimer) {
        [self stopPostingLocation];
    }

    self.pollingTimer = [NSTimer scheduledTimerWithTimeInterval:seconds target:self selector:@selector(postLocation:) userInfo:nil
                         repeats:YES];
}

- (NSArray<NSString *> *)supportedEvents {
    return @[@"LocationUpdated"];
}

RCT_EXPORT_METHOD(stopPostingLocation) {
    [self.pollingTimer invalidate];
    self.pollingTimer = nil;
}

- (void)postLocation:(NSTimer *)timer {
    if (!self.url && !self.currentLocation) {
        return;
    }

    NSMutableURLRequest *request = [[NSMutableURLRequest alloc] initWithURL:self.url];
    [request addValue:@"application/json" forHTTPHeaderField:@"Content-Type"];
    [request addValue:@"application/json" forHTTPHeaderField:@"Accept"];
    request.HTTPMethod = @"POST";

    NSDateFormatter *dateFormat = [[NSDateFormatter alloc] init];
    [dateFormat setDateFormat:@"yyyy-MM-dd'T'HH:mm:ss.SSS"];
    NSString *dateString = [dateFormat stringFromDate:[NSDate date]];

    NSDictionary *params =  @{@"lat":@(self.currentLocation.coordinate.latitude), @"long":@(self.currentLocation.coordinate.longitude),
                              @"time":dateString};
    NSError *error = nil;
    NSData *data = [NSJSONSerialization dataWithJSONObject:params
                                                   options:kNilOptions error:&error];
    if (!data || error) {
        return;
    }

    [request setHTTPBody:data];
    NSURLSessionDataTask *postDataTask = [self.session dataTaskWithRequest:request
                                                          completionHandler:^(NSData *data, NSURLResponse *response, NSError *requestError)
                                                                            {
                                                              NSHTTPURLResponse *httpResponse = (NSHTTPURLResponse *) response;
                                                              if (requestError || [httpResponse statusCode] != 200) {
                                                                  return;
                                                              }
                                                              [self sendEventWithName:@"LocationUpdated" body:params];
                                                          }];
    [postDataTask resume];
}
```

Things To Note

- 90% of the DriverApp is Javascript
- Code Push doesn’t support Native code
  - https://github.com/Microsoft/react-native-code-push

Note: Any product changes which touch native code (e.g. modifying your `AppDelegate.m` / `MainActivity.java` file, adding a new plugin) cannot be distributed via CodePush, and therefore, must be updated via the appropriate store(s).


eaze

Contact Me:

Email: harry.n.gale@gmail.com

Medium: http://launchdrawer.com/

Twitter: @htormey