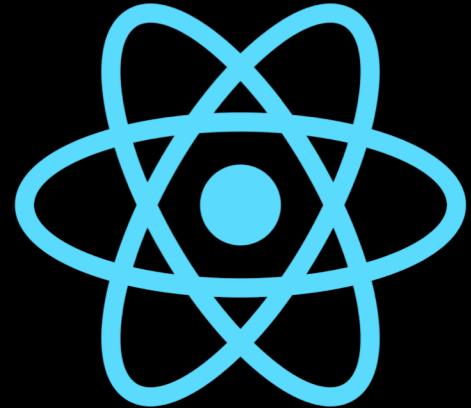




SwiftFest

RAY-20-OFF



for
developers

@ray_deck

react-native-swift

react-native-swift-cli

react-native-coreml

react-native-pods

react-native-ios-setfamily

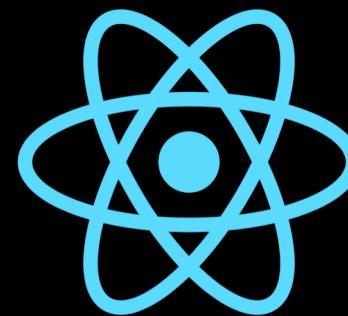
react-native-appicons

react-native-launchimages

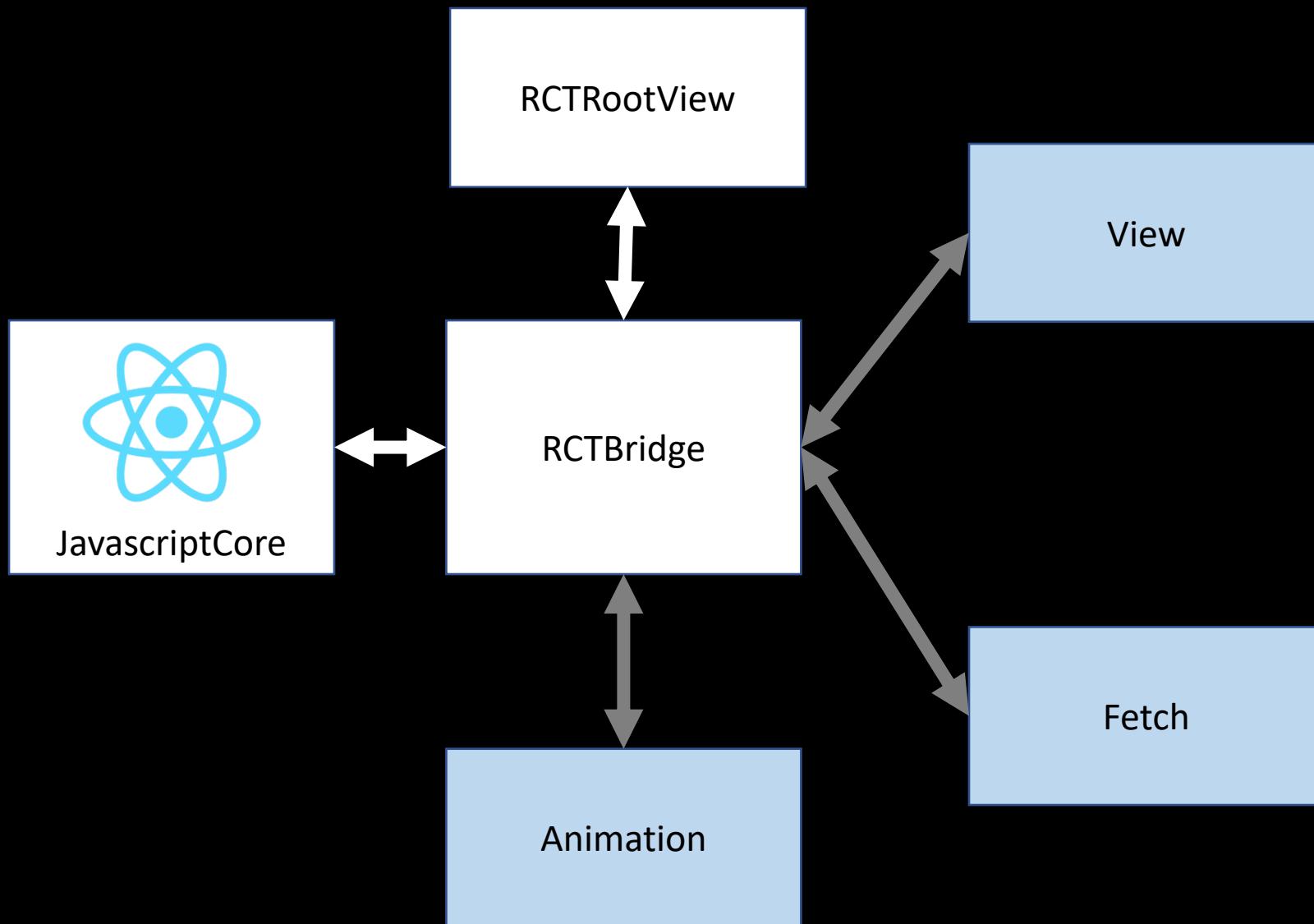
react-native-setdevteam

react-native-xcode

facebook
2015



JavaScript As Orchestrator



```
> yarn global add react-native-cli  
> react-native init <myprojectname>  
> cd <myprojectname>  
> code .  
> react-native run-ios
```

Code File Edit Selection View Go Debug Tasks Window Help

Thu 12:08 PM

App.js — sohtest

```
JS App.js x ⓘ Workspace Settings
8 import { Platform, StyleSheet, Text, View } from "react-native";
9
10 const instructions = Platform.select({
11   ios: "Press Cmd+R to reload,\n" + "Cmd+D or shake for dev menu",
12   android:
13     "Double tap R on your keyboard to reload,\n" +
14     "Shake or press menu button for dev menu"
15 });
16
17 type Props = {};
18 export default class App extends Component<Props> {
19   render() {
20     return [
21       <View style={styles.container}>
22         <Text style={styles.instructions}>To get started, edit App.js</Text>
23         <Text style={styles.instructions}>[instructions]</Text>
24         <Text style={styles.welcome}>Welcome to Swift Office Hours!</Text>
25       </View>
26     ];
27 }
```

Ln 20, Col 13 Spaces: 2 UTF-8 LF JavaScript Prettier: ✓

To get started, edit App.js
Press Cmd+R to reload,
Cmd+D or shake for dev menu

Welcome to Swift Office Hours!

Developer Tools - http://localhost:8081/debugger-ui/

Elements Console Sources Network Performance Memory Application Security

top Filter Default levels Group similar

Gather Windows

Console was cleared

Module RCTImageLoader requires main queue setup since it overrides `init` but doesn't implement `requiresMainQueueSetup`. In a future release React Native will default to initializing all native modules on a background thread unless explicitly opted-out of.

Running application sohtest {
initialProps = {
};
rootTag = 161;
}

Running application "sohtest" with appParams: {rootTag:161,"initialProps":{}}. __DEV__ === true, infoLog.js:15 development-level warning are ON, performance optimizations are OFF

Remote debugger is in a background tab which may cause apps to perform slowly. Fix this by foregrounding the tab (or opening it in a separate window).

Check register analysis...017a.xlsx

DELTA [ios, dev] ./index.js 100.0% (1/1), done.
MAP [ios, dev] ./index.js 100.0% (1/1), done.
BUNDLE [ios, dev] ./index.js 100.0% (1/1), done.
DELTA [ios, dev] ./index.js 100.0% (1/1), done.
MAP [ios, dev] ./index.js 100.0% (1/1), done.
BUNDLE [ios, dev] ./index.js 100.0% (1/1), done.
DELTA [ios, dev] ./index.js 100.0% (1/1), done.
MAP [ios, dev] ./index.js 100.0% (1/1), done.

Rapid Reloading

Code as Asset

React

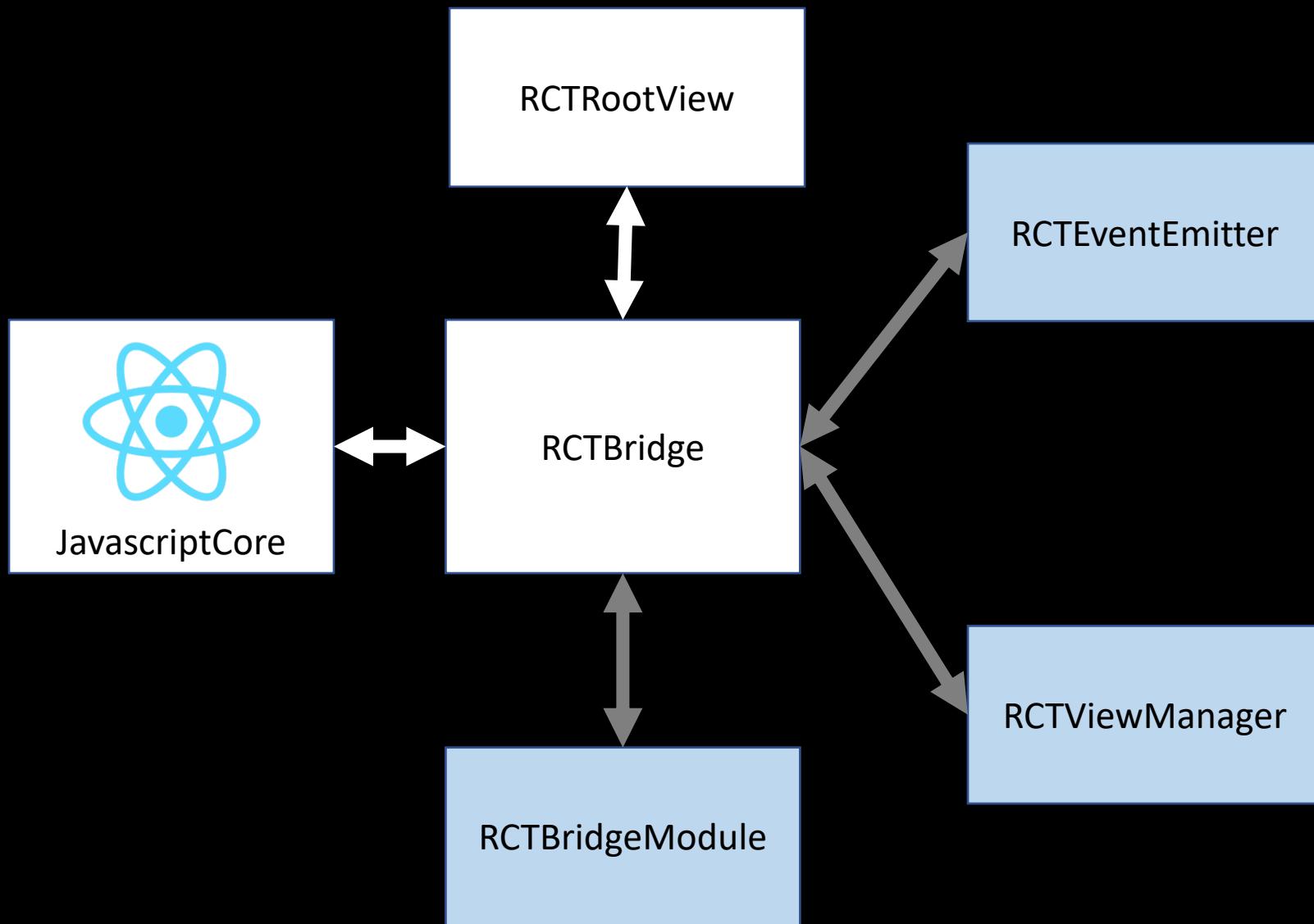
The screenshot shows the Xcode IDE interface with the following details:

- Project Navigator:** On the left, the project structure for "sohtest" is shown. It includes a main group "sohtest" containing "main.jsbundle", "AppDelegate.h", and "AppDelegate.m" (which is selected). Other files like "Images.xcassets", "Info.plist", and "LaunchScreen.xib" are also listed. Below "sohtest" are "Libraries" containing various React Native framework projects (RCTAnimation, React, RCTActionSheet, RCTBlob, RCTGeolocation, RCTImage, RCTLinking, RCTNetwork, RCTSettings, RCTText, RCTVibration, RCTWebSocket) and "sohtestTests" containing "sohtestTests.m".
- Editor:** The main editor area displays the content of "AppDelegate.m". The code is written in Objective-C and handles the application's launch process. It imports "React/RCTBundleURLProvider.h" and "React/RCTRootView.h", implements the "UIApplicationDelegate" protocol, and initializes the root view controller with a bundle URL for "index" and a fallback resource "nil". It also sets the root view's background color to white and makes it key and visible.
- Build Status:** At the top right, there are build statistics: 13 warnings and 1 error.

```
1 /**
2  * Copyright (c) 2015-present, Facebook, Inc.
3  *
4  * This source code is licensed under the MIT license found in the
5  * LICENSE file in the root directory of this source tree.
6 */
7
8 #import "AppDelegate.h"
9
10#import <React/RCTBundleURLProvider.h>
11#import <React/RCTRootView.h>
12
13 @implementation AppDelegate
14
15 - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
16 {
17     NSURL *jsCodeLocation;
18
19     jsCodeLocation = [[RCTBundleURLProvider sharedSettings] jsBundleURLForBundleRoot:@"index" fallbackResource:nil];
20
21     RCTRootView *rootView = [[RCTRootView alloc] initWithBundleURL:jsCodeLocation
22                                                       moduleName:@"sohtest"
23                                                       initialProperties:nil
24                                                       launchOptions:launchOptions];
25
26     rootView.backgroundColor = [[UIColor alloc] initWithRed:1.0f green:1.0f blue:1.0f alpha:1];
27
28     self.window = [[UIWindow alloc] initWithFrame:[UIScreen mainScreen].bounds];
29     UIViewController *rootViewController = [UIViewController new];
30     rootViewController.view = rootView;
31     self.window.rootViewController = rootViewController;
32     [self.window makeKeyAndVisible];
33     return YES;
34 }
35 @end
36
```

react-native link
2016

```
> react-native init <myproject> ; \  
yarn add react-native-camera;  
react-native link;  
react-native run-ios;  
code .
```



```
#import <React/RCTViewManager.h>
#import <React/RCTEventEmitter.h>
#import <React/RCTBridgeModule.h>

@interface RCT_EXTERN_MODULE(RNSBoston, RCTEventEmitter)
RCT_EXTERN_METHOD(demo:(NSString *)message
success:(RCTPromiseResolveBlock)success
reject:(RCTPromiseRejectBlock)reject);
RCT_EXTERN_METHOD(delayedSend:(NSString *)message
ms:(NSInteger)ms);

@end

@interface RCT_EXTERN_MODULE(RNSBostonBasicViewManager,
RCTViewManager)

@end

@interface RCT_EXTERN_MODULE(RNSBostonViewManager,
RCTViewManager)
```

Xcode 9
2017



react-native-swift

2017

react-native-swift
react-native-swift-bridge

2018

react-native-swift
react-native-swift-bridge
react-native-swift-cli

2018

```
> yarn global add react-native-swift-cli  
> react-native-swift init <mymodule>  
> rns makeapp <appname> <mymodule>  
> cd <mymodule>  
> yarn watch  
> code .  
> cd ../../<appname>  
> react-native xcode
```

The screenshot shows the Xcode interface with two editors open and a detailed project view on the right.

Left Editor: Displays the file `RNSwiftBridge.js`. The code implements a native module named `rnsmod` which contains three methods: `demo`, `delayedSend`, and `takePicture`. It uses the `NativeModules` API to interact with the native environment.

```
1 import { NativeModules } from "react-native";
2 //region Code for object rnsmod
3 const Nativernsmod = NativeModules.rnsmod;
4 const demo = async message => {
5   return await Nativernsmod.demo(message);
6 };
7 const delayedSend = (message, ms) => {
8   return Nativernsmod.delayedSend(message, ms);
9 };
10 //endregion
11 //##region Code for object rnsmodViewManager
12 const NativernsmodViewManager = NativeModules.rnsmodViewManager;
13 const takePicture = async () => {
14   return await NativernsmodViewManager.takePicture();
15 };
16 //##endregion
17 //##region Exports
18 export { demo, delayedSend, takePicture };
19 //##endregion
20
```

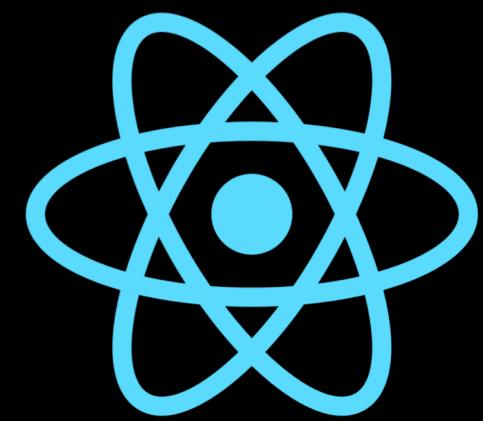
Right Sidebar: Shows the Xcode project structure for the target `rnsapp`.

- rnssapp**:
 - rnsapp**:
 - main.jsbundle
 - AppDelegate.h
 - AppDelegate.m
 - Images.xcassets
 - Info.plist
 - LaunchScreen.xib
 - main.m
 - Libraries**:
 - RCTAnimation.xcodeproj
 - React.xcodeproj
 - RCTActionSheet.xcodeproj
 - RCTBlob.xcodeproj
 - RCTGeolocation.xcodeproj
 - RCTImage.xcodeproj
 - RCTLinking.xcodeproj
 - RCTNetwork.xcodeproj
 - RCTSettings.xcodeproj
 - RCTText.xcodeproj
 - RCTVibration.xcodeproj
 - RCTWebSocket.xcodeproj
 - rnsmod.xcodeproj**:
 - rnsmod**:
 - rnsmod.swift
 - rnsmodViewManager.swift
 - rnsmodView.swift
 - rnsmodBasicViewManager.swift
 - Products**:
 - librnsmod.a
 - rn-Bridging-Header.h
 - Recovered References**:
 - rn-swift-bridge.m
 - rnsappTests**:
 - rnsappTests.m
 - Supporting Files
 - Products**
 - Frameworks**
 - Recovered References**:
 - RNPlaceholder.swift
 - librnsmod.a

Promise-Based Communication



```
//MARK: RN Methods
@objc func compileModel(_ source: String, success:@escaping RCTPromiseResolveBlock, reject:@escaping RCTPromiseRejectBlock) ->Void {
    DispatchQueue(label: "RNCoreML").async() {
        let url = URL(fileURLWithPath: source)
        do {
            let tempURL:URL = try
MLModel.compileModel(at:url)
            success(tempURL.path);
        } catch {
            reject(nil, nil, error);
        }
    }
}
@objc func classifyImageWithModel(_ source: String,
modelPath: String, success:@escaping RCTPromiseResolveBlock,
reject:@escaping RCTPromiseRejectBlock) ->Void {
```



```
import { NativeModules } from "react-native";

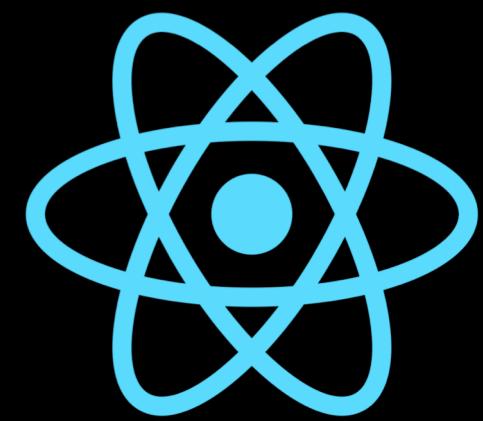
const RNCoreML: RNCoreMLClass = NativeModules.RNCoreML;
const compileModel = async sourcepath => {
  return await RNCoreML.compileModel(sourcepath);
};

const classifyImage = async (imagePath, modelPath) => {
  return await RNCoreML.classifyImageWithModel(imagePath,
modelPath);
};
```

UI Element Integration



```
import Foundation
import AVKit // This is for the camera questions asked in
getCameraDirection
@objc(rnsmodViewManager)
class rnsmodViewManager: RCTViewManager,
AVCapturePhotoCaptureDelegate {
    var currentView: rnsmodView?
    //MARK: RCTViewManager key methods
    override func view() -> rnsmodView {
        let newView = rnsmodView()
        currentView = newView
        return newView
    }
}
```



```
import { requireNativeComponent, NativeModules } from "react-native";
import React, { Component } from "react";

const NativeView = requireNativeComponent("rnsmodView", CameraView);
class CameraView extends Component {
  render() {
    return <NativeView {...this.props} />;
  }
}
CameraView.takePicture = async () => {
  try {
    const x = await NativeModules.rnsmodViewManager.takePicture();
    return x;
  } catch (e) {return null;}
};
export default CameraView;
```

```
<TouchableOpacity
  onPress={async () => {
    const result = await CameraView.takePicture();
    console.log("I got a result!!!", result);
    const newText = result
      ? "Took a picture!"
      : "Error taking picture";
    this.setState({ cameraText: newText, imageURL:
      result.url });
  }}
>
<CameraView
  style={{ width: "100%", height: 300 }}
  cameraFront={this.state.cameraFront}
/>
</TouchableOpacity>
```

Business Logic
Extended Functionality
Low-level Affordance

plist
pbxproj
FastLane

```
> react-native init <myproject> ; \
yarn add
  react-native-coreml \
  react-native-swift \
  react-native-camera \
  react-native-setdevteam \
  react-native-fix-ios-version;
react-native setdevteam ;
react-native link;
react-native run-ios --device;
code .
```

Let Designers Tweak
Let Coders Create

Swift that Matters

Thank You

<https://github.com/rhdeck/rn-for-swift-devs>

