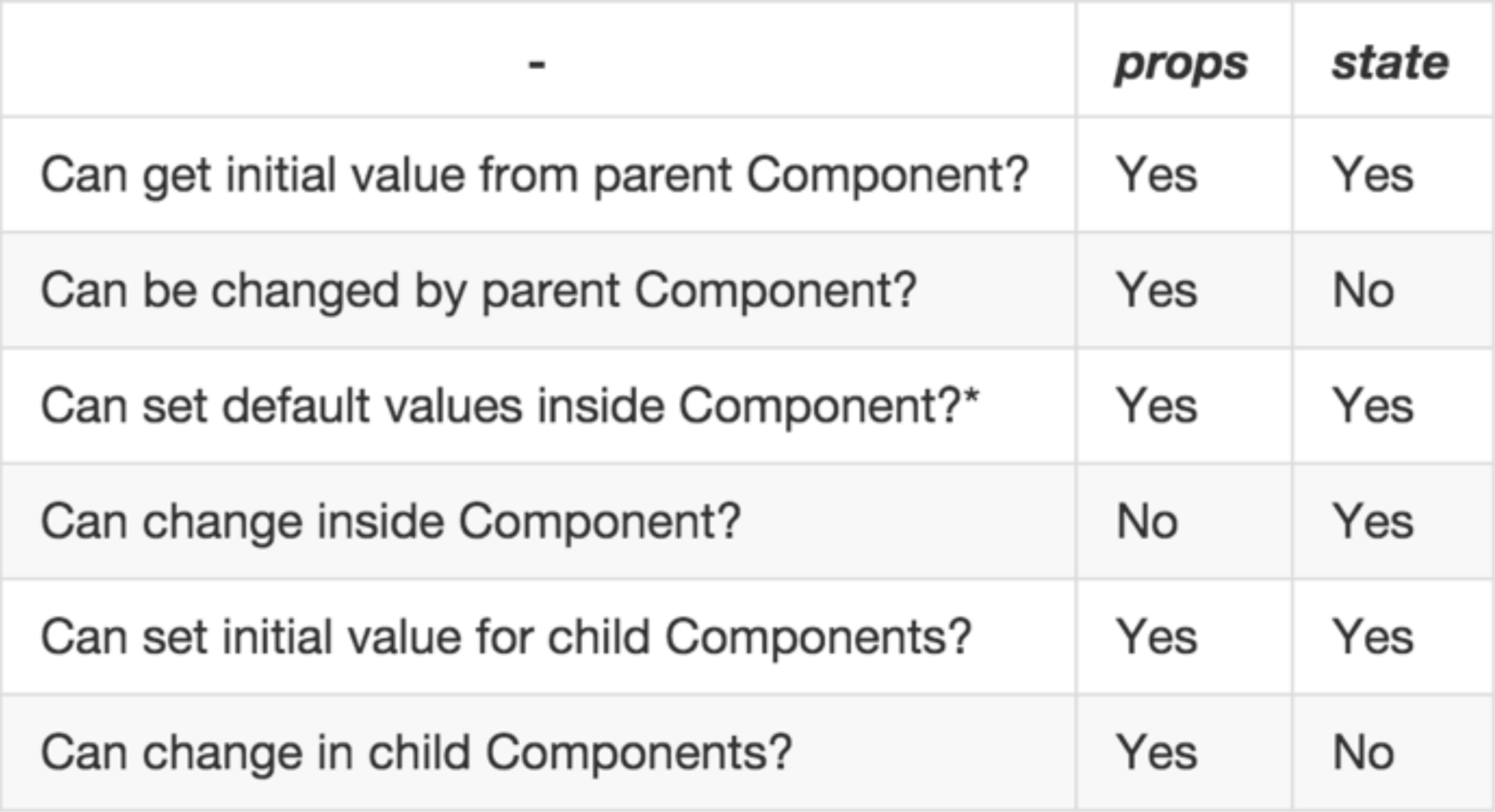
RN总结

Philsong 2015/11/27

有状态的组件

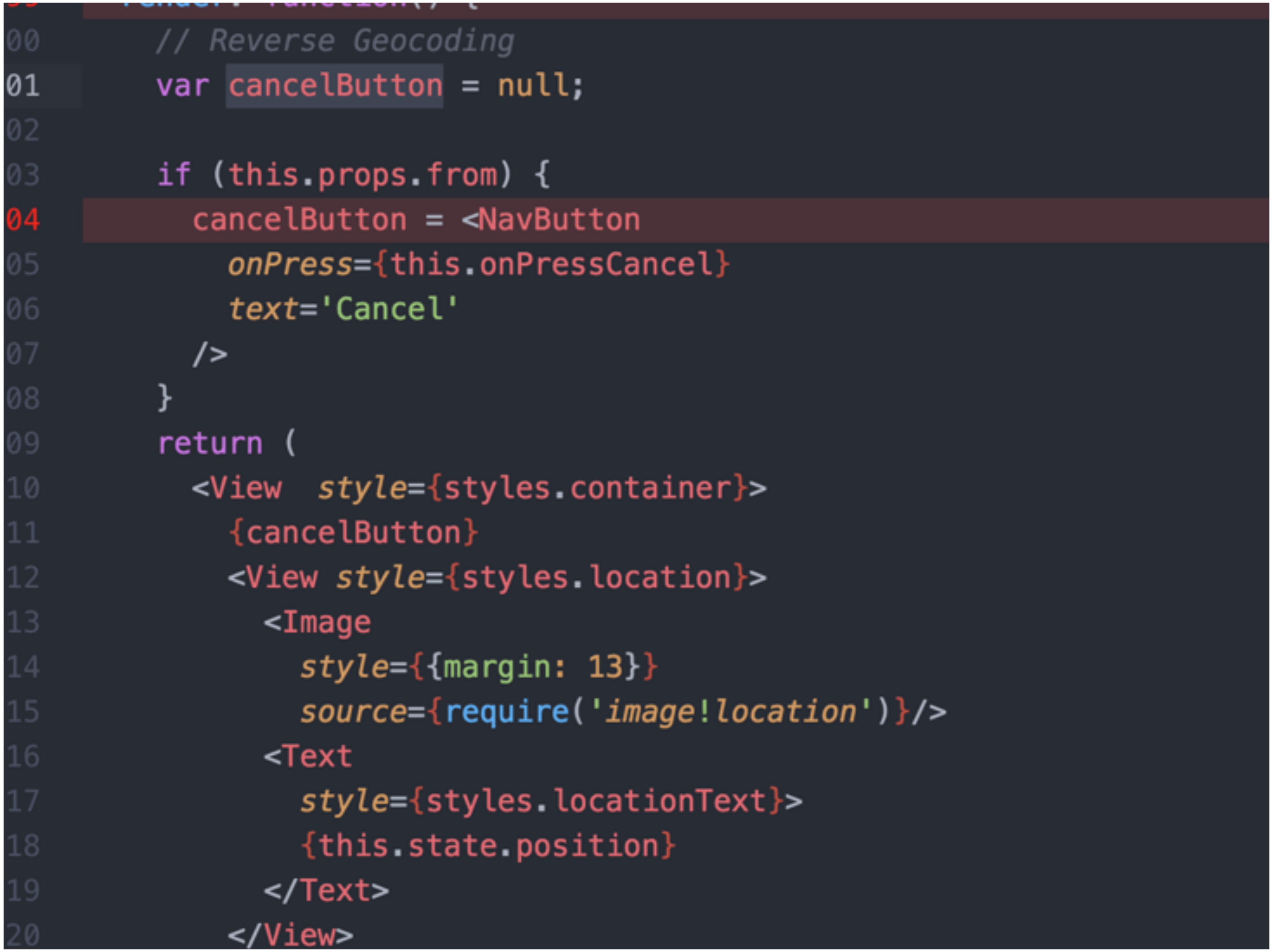
组件：（通过 this.**props**）接受**输入**数据），（通过 this.**state**）保持**内部状态**数据。当一个组件的**状态数据**的变化，展现的标记将被重新调用 **render()** 更新，并以**事件代理**的方式通知控制逻辑。

Props和State比较



-	***props***	***state***
Can get initial value from parent Component?	Yes	Yes
Can be changed by parent Component?	Yes	No
Can set default values inside Component?*	Yes	Yes
Can change inside Component?	No	Yes
Can set initial value for child Components?	Yes	Yes
Can change in child Components?	Yes	No

实例：利用props控制logic和ui



```
render: function() {
  // Reverse Geocoding
  var cancelButton = null;

  if (this.props.from) {
    cancelButton = <NavButton
      onPress={this.onPressCancel}
      text='Cancel'
    />
  }
  return (
    <View  style={styles.container}>
      {cancelButton}
      <View style={styles.location}>
        <Image
          style={{margin: 13}}
          source={require('image!location')}/>
        <Text
          style={styles.locationText}>
          {this.state.position}
        </Text>
      </View>
```

ref 属性

从 render() 中返回的内容并不是实际渲染出来的子组件实例。从 render() 返回的仅仅是子组件层级树实例在特定时间的一个描述。－－ 一个快照

所以，我们如何与真正的 input 支撑实例（ backing instance ） 交流?

做法很简单：

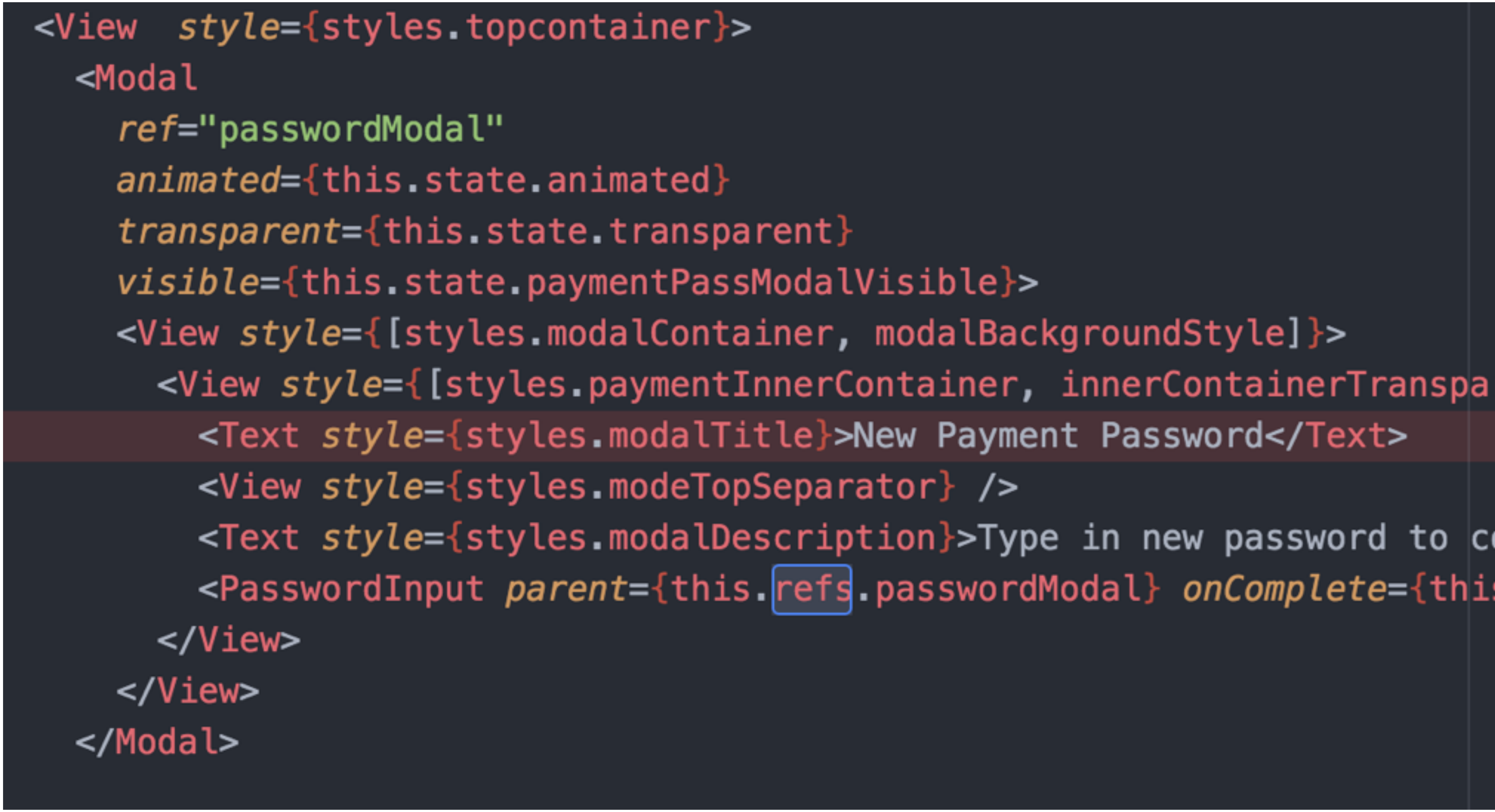
1、绑定一个 ref 属性到 render 返回的东西上面去，例如：

```
<input ref="myInput" />
```

2、在其它代码中（典型地事件处理代码），通过 this.refs 获取支撑实例（ backing instance ），就像这样：

```
this.refs.myInput
```

ref 实例

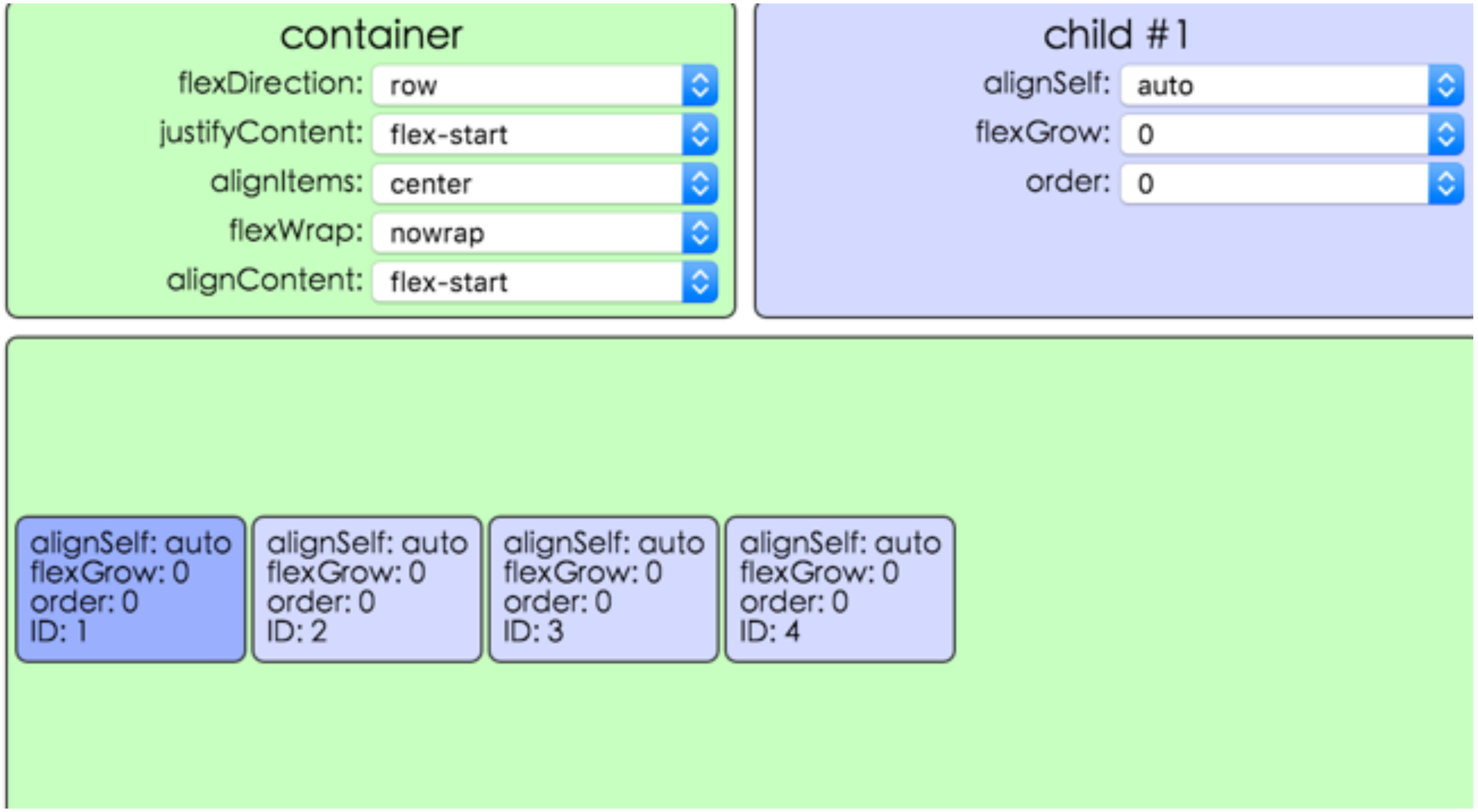


```
<View  style={styles.topcontainer}>
  <Modal
    ref="passwordModal"
    animated={this.state.animated}
    transparent={this.state.transparent}
    visible={this.state.paymentPassModalVisible}>
    <View style={[styles.modalContainer, modalBackgroundStyle]}>
      <View style={[styles.paymentInnerContainer, innerContainerTranspa
        <Text style={styles.modalTitle}>New Payment Password</Text>
        <View style={styles.modeTopSeparator} />
        <Text style={styles.modalDescription}>Type in new password to c
        <PasswordInput parent={this.refs.passwordModal} onComplete={thi
      </View>
    </View>
  </Modal>
```

布局

Flex Layout

http://blog.krawaller.se/posts/a-react-app-demonstrating-css3-flexbox/



container

flexDirection: row

justifyContent: flex-start

alignItems: center

flexWrap: nowrap

alignContent: flex-start

child #1

alignSelf: auto

flexGrow: 0

order: 0

alignSelf: auto
flexGrow: 0
order: 0
ID: 1

alignSelf: auto
flexGrow: 0
order: 0
ID: 2

alignSelf: auto
flexGrow: 0
order: 0
ID: 3

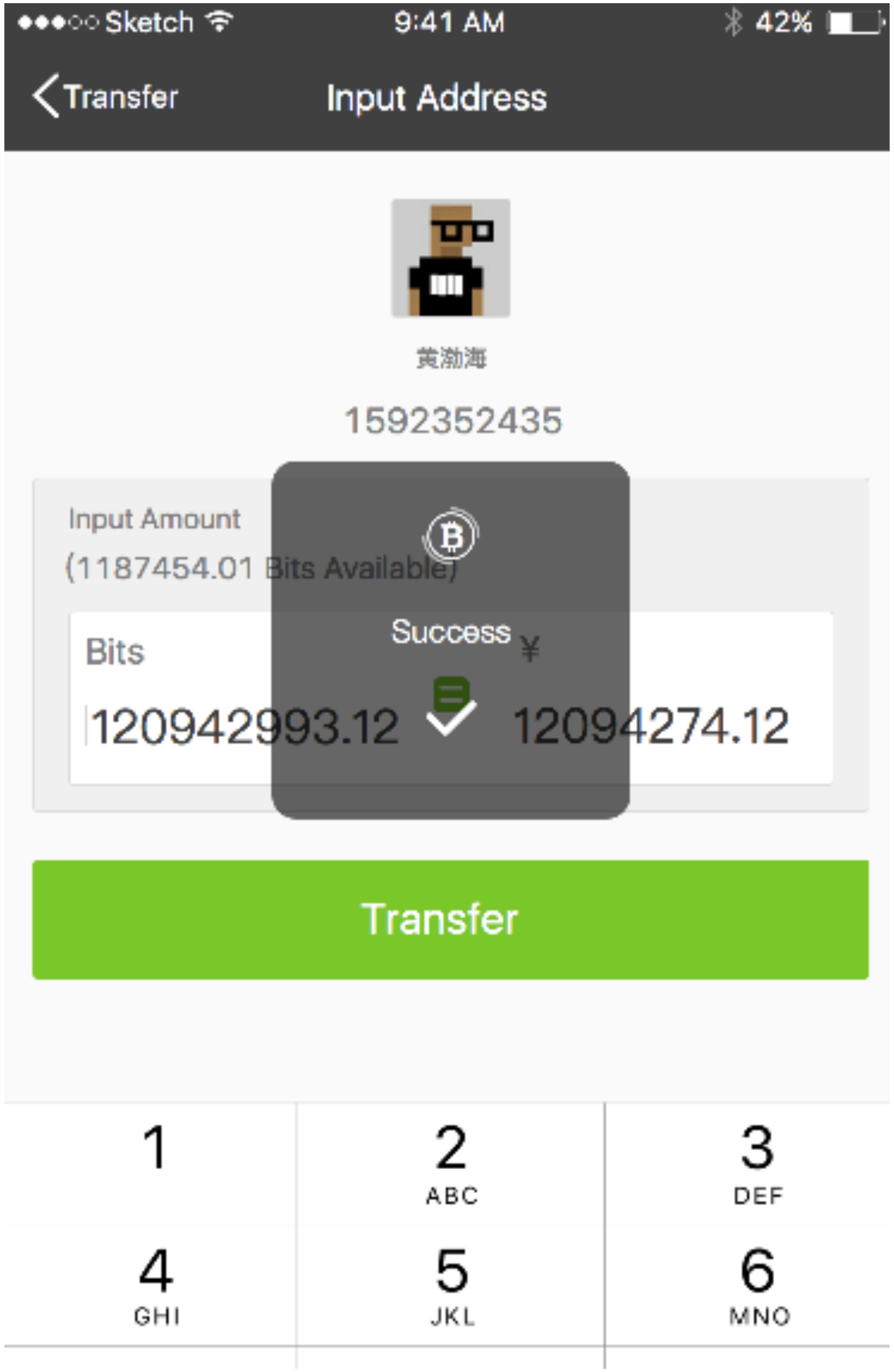
alignSelf: auto
flexGrow: 0
order: 0
ID: 4

几点

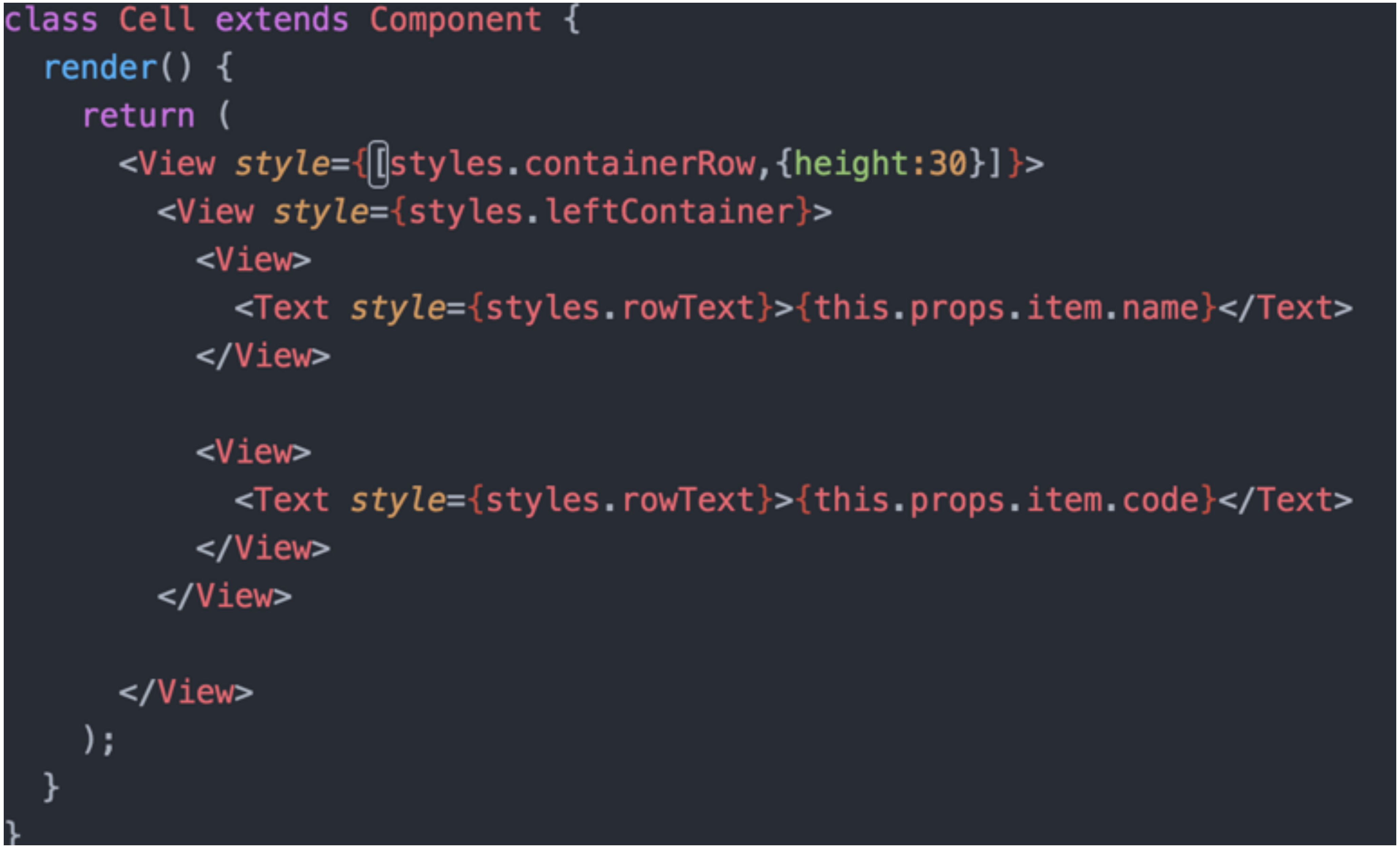
若是平均间隔和宽度的尽量用space-between或者space-around

rn flex是react flex的一个子集，不支持alignContent等属性.

flex:1可以铺满，不要用marginBottom:10000



style=[styles.container, {xx:yy,mm:nn}]

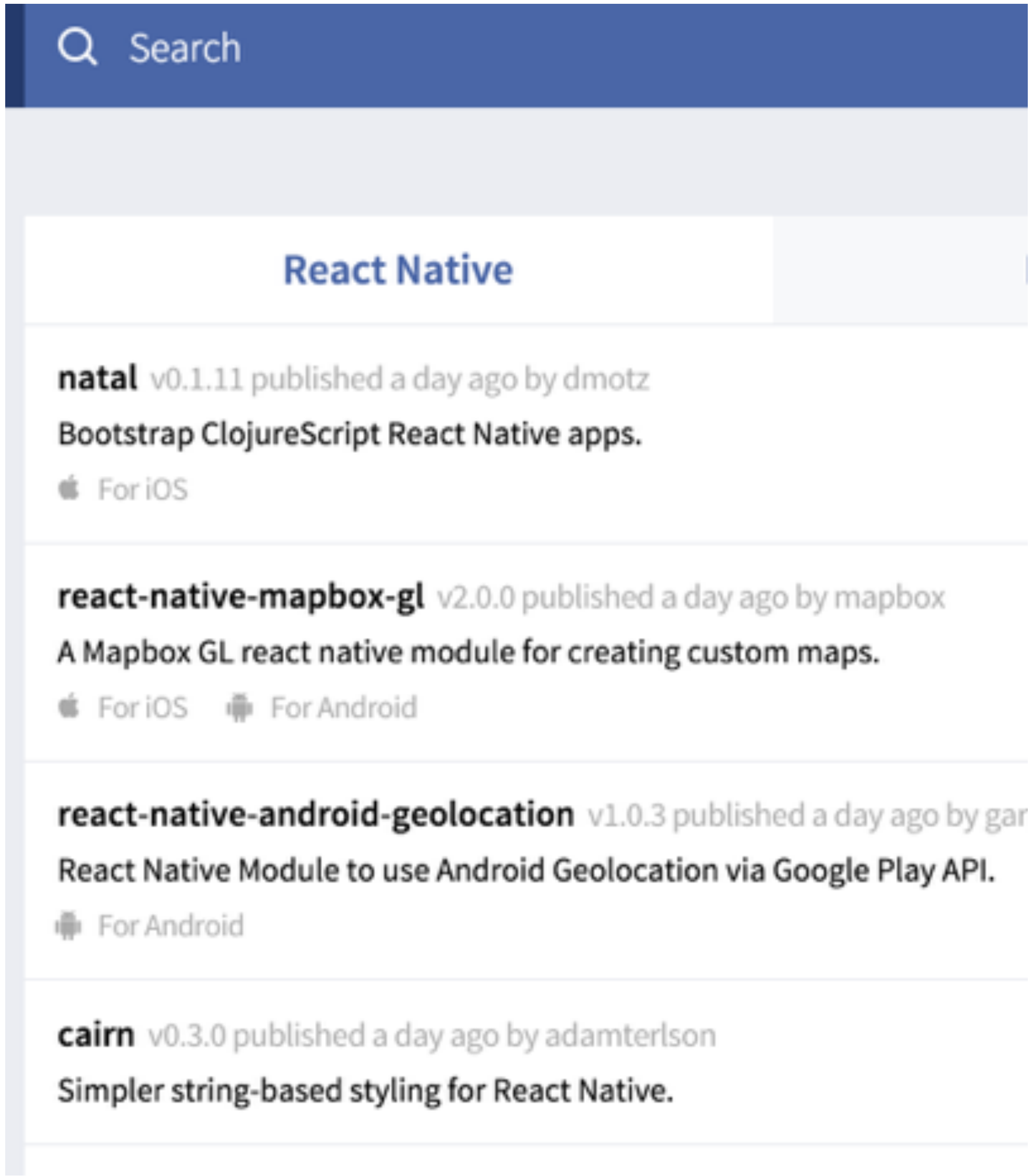


```
class Cell extends Component {
  render() {
    return (
      <View style={[styles.containerRow,{height:30}]}>
        <View style={styles.leftContainer}>
          <View>
            <Text style={styles.rowText}>{this.props.item.name}</Text>
          </View>

          <View>
            <Text style={styles.rowText}>{this.props.item.code}</Text>
          </View>
        </View>

      </View>
    );
  }
}
```

编写iOS组件

第三方组件

https://react.parts/native



React Native

natal v0.1.11 published a day ago by dmotz
Bootstrap ClojureScript React Native apps.
For iOS

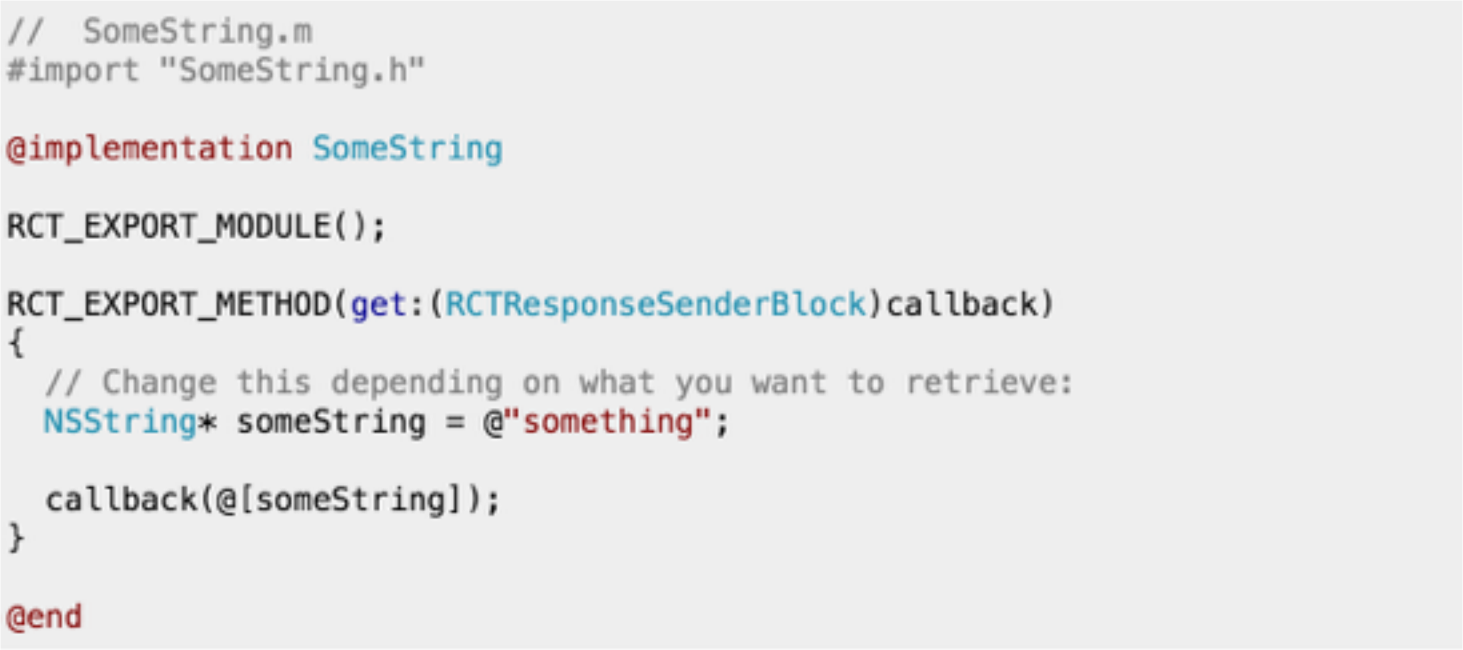
react-native-mapbox-gl v2.0.0 published a day ago by mapbox
A Mapbox GL react native module for creating custom maps.
For iOS For Android

react-native-android-geolocation v1.0.3 published a day ago by gar
React Native Module to use Android Geolocation via Google Play API.
For Android

cairn v0.3.0 published a day ago by adamterlson
Simpler string-based styling for React Native.

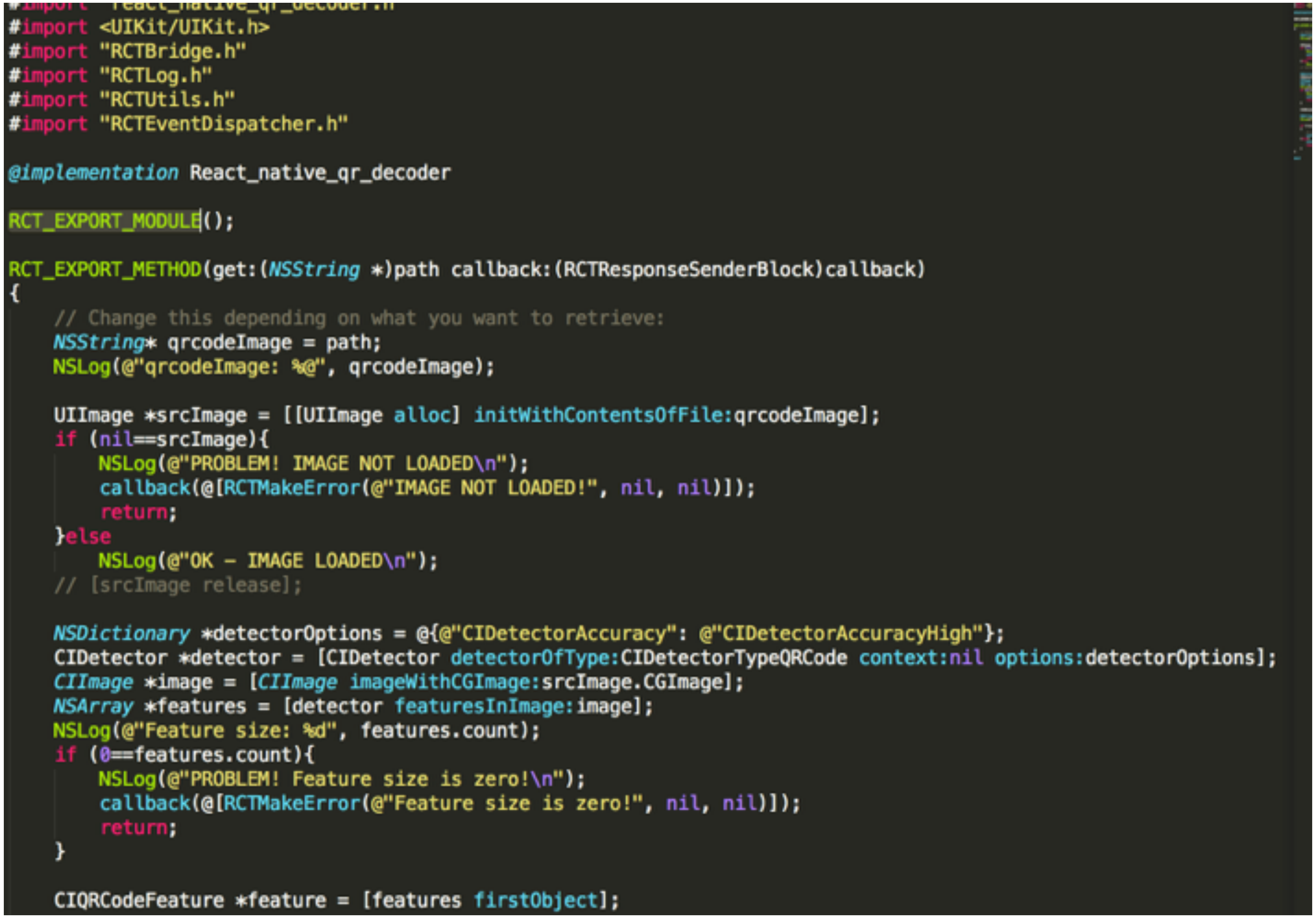
编写iOS端

In Xcode, use File > New > File... to create a new Cocoa Touch class that extends NSObject.



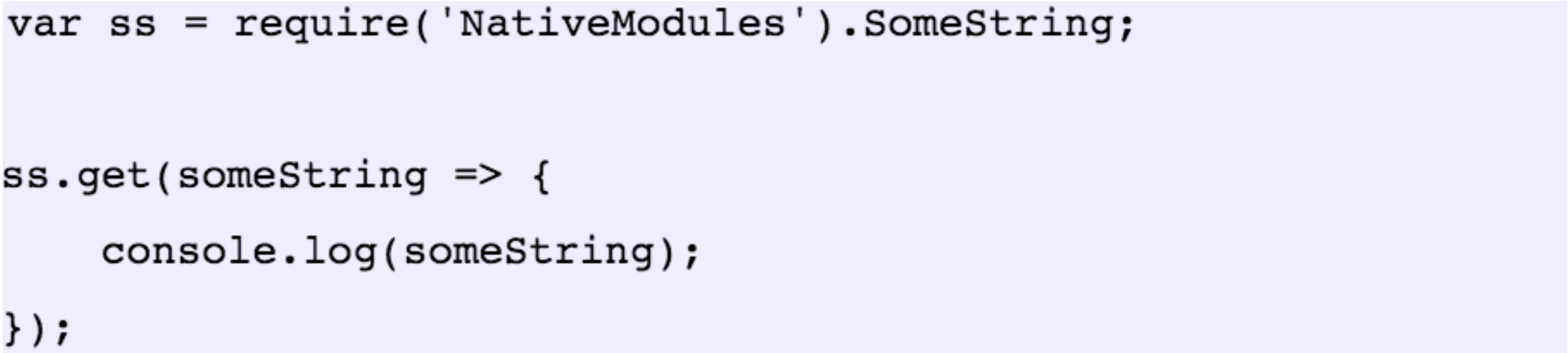
```
//  SomeString.m
#import "SomeString.h"

@implementation SomeString

RCT_EXPORT_MODULE();

RCT_EXPORT_METHOD(get:(RCTResponseSenderBlock)callback)
{
  // Change this depending on what you want to retrieve:
  NSString* someString = @"something";

  callback(@[someString]);
}

@end
```

rn组件的callback



```
#import "react_native_qr_decoder.h"
#import <UIKit/UIKit.h>
#import "RCTBridge.h"
#import "RCTLog.h"
#import "RCTUtils.h"
#import "RCTEventDispatcher.h"

@implementation React_native_qr_decoder

RCT_EXPORT_MODULE();

RCT_EXPORT_METHOD(get:(NSString *)path callback:(RCTResponseSenderBlock)callback)
{
    // Change this depending on what you want to retrieve:
    NSString* qrcodeImage = path;
    NSLog(@"qrcodeImage: %@", qrcodeImage);

    UIImage *srcImage = [[UIImage alloc] initWithContentsOfFile:qrcodeImage];
    if (nil==srcImage){
        NSLog(@"PROBLEM! IMAGE NOT LOADED\n");
        callback(@[RCTMakeError(@"IMAGE NOT LOADED!", nil, nil)]);
        return;
    }else
        NSLog(@"OK - IMAGE LOADED\n");
    // [srcImage release];

    NSDictionary *detectorOptions = @{@"CIDetectorAccuracy": @"CIDetectorAccuracyHigh"};
    CIDetector *detector = [CIDetector detectorOfType:CIDetectorTypeQRCode context:nil options:detectorOptions];
    CIImage *image = [CIImage imageWithCGImage:srcImage.CGImage];
    NSArray *features = [detector featuresInImage:image];
    NSLog(@"Feature size: %d", features.count);
    if (0==features.count){
        NSLog(@"PROBLEM! Feature size is zero!\n");
        callback(@[RCTMakeError(@"Feature size is zero!", nil, nil)]);
        return;
    }

    CIQRCodeFeature *feature = [features firstObject];
```

rn调用组件



```
var ss = require('NativeModules').SomeString;


ss.get(someString => {

    console.log(someString);

});
```

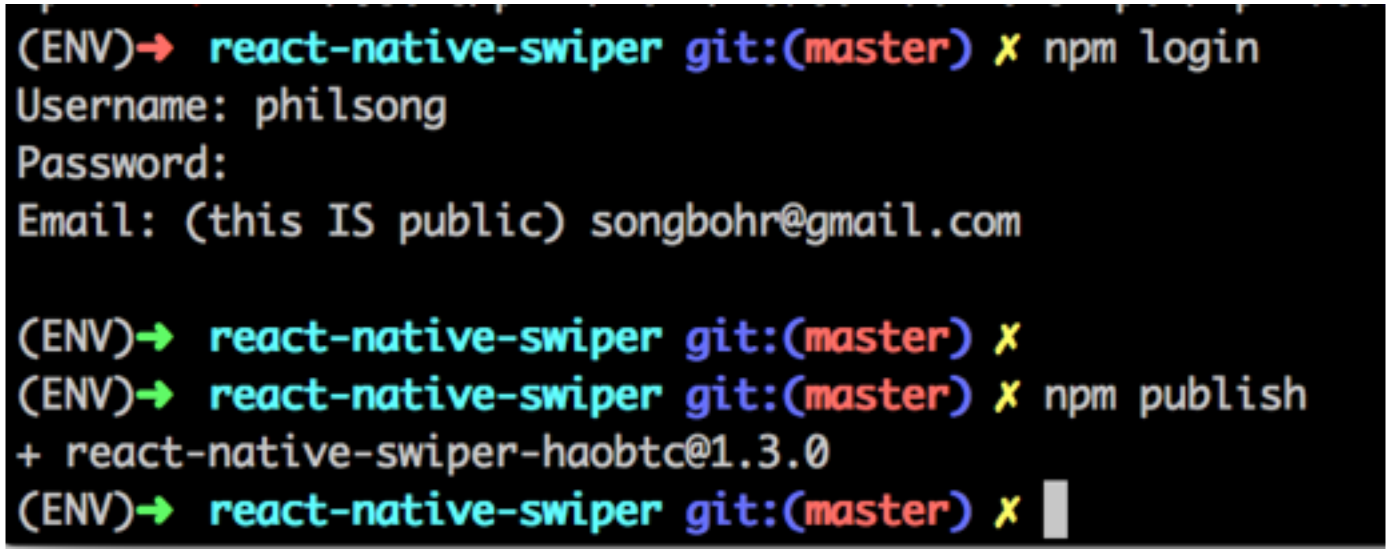
发布组件

$ npm adduser

用于在npmjs.com注册一个用户。

$ npm login

$ npm publish



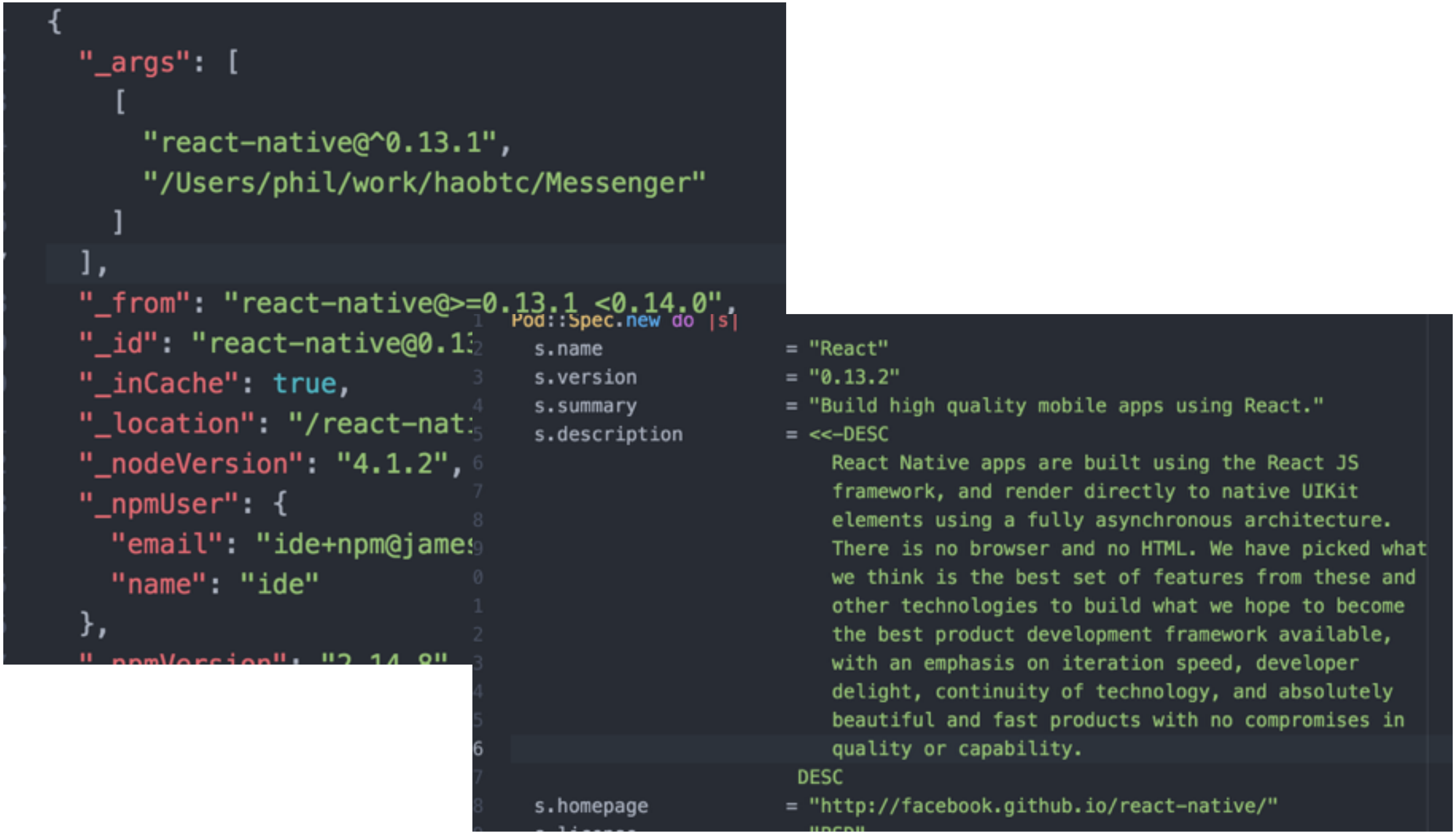
```
(ENV)➜  react-native-swiper git:(master) ✗ npm login
Username: philsong
Password:
Email: (this IS public) songbohr@gmail.com

(ENV)➜  react-native-swiper git:(master) ✗
(ENV)➜  react-native-swiper git:(master) ✗ npm publish
+ react-native-swiper-haobtc@1.3.0
(ENV)➜  react-native-swiper git:(master) ✗ ▌
```

https://www.npmjs.com/package/react-native-qr-decoder

https://www.npmjs.com/~philsong

注意的几点问题

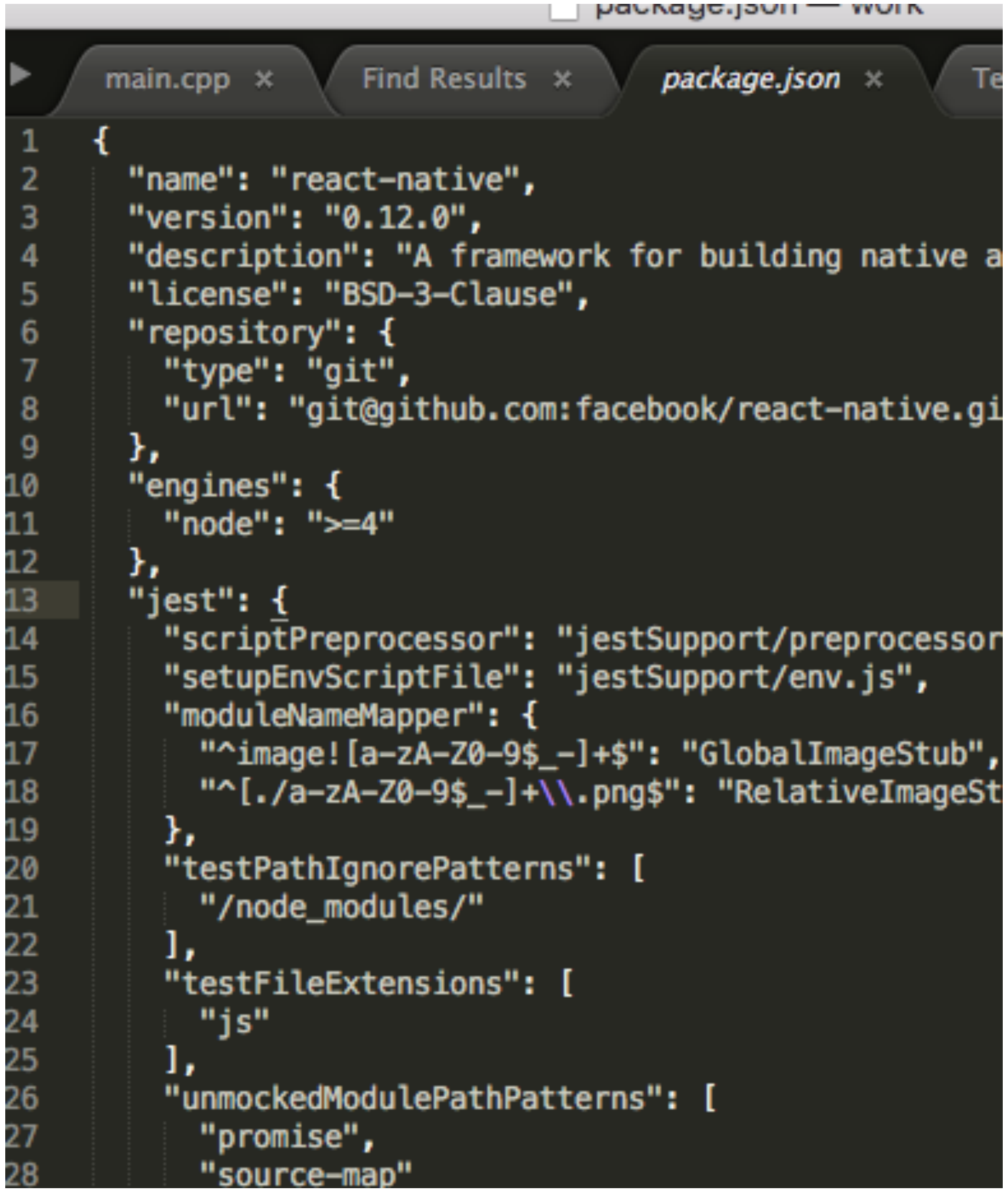
RN版本问题



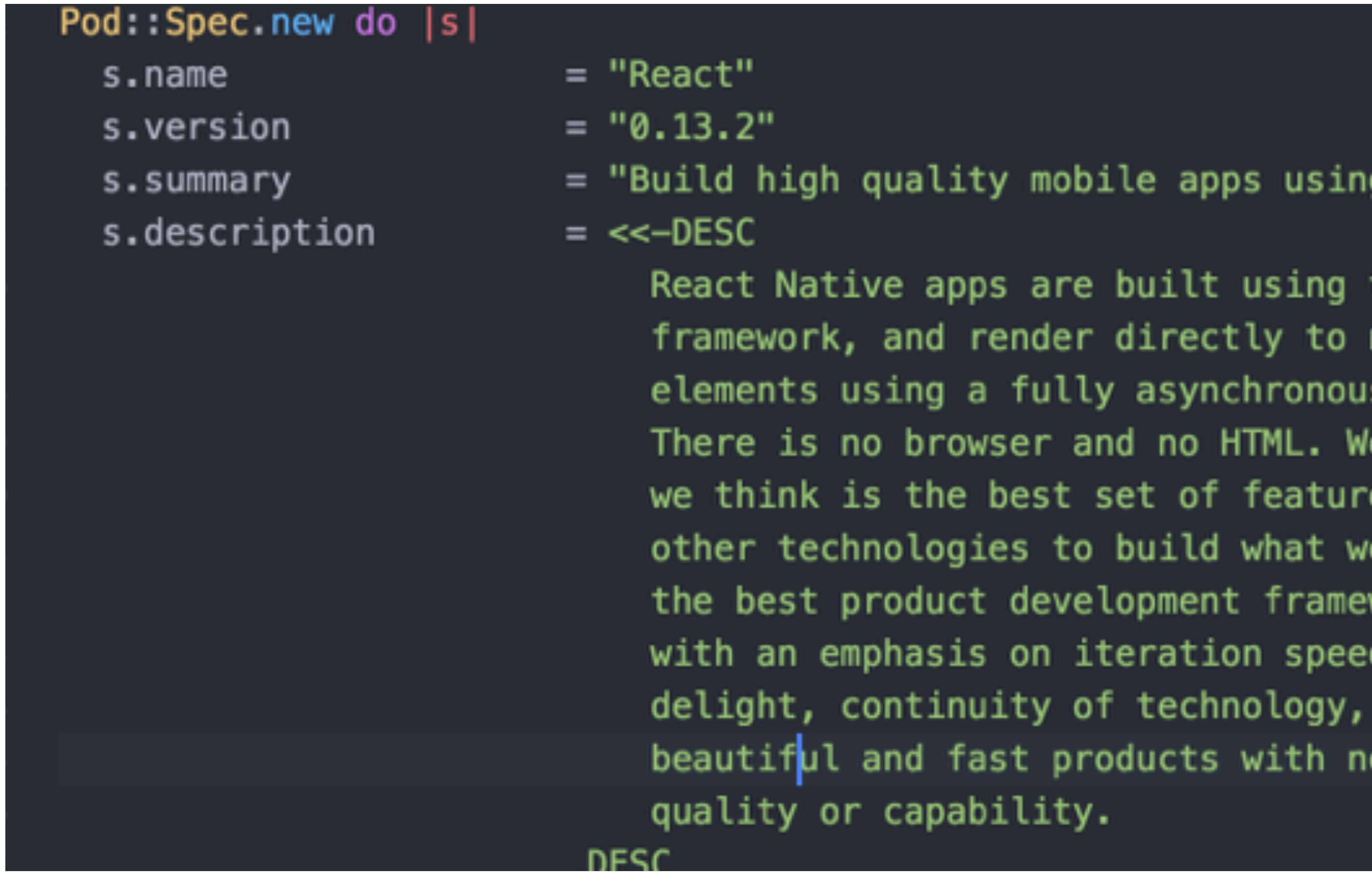
```
{
  "_args": [
    [
      "react-native@^0.13.1",
      "/Users/phil/work/haobtc/Messenger"
    ]
  ],
  "_from": "react-native@>=0.13.1 <0.14.0",
  "_id": "react-native@0.1
  "_inCache": true,
  "_location": "/react-nat
  "_nodeVersion": "4.1.2",
  "_npmUser": {
    "email": "ide+npm@james
    "name": "ide"
  },
  "_npmVersion": "2.14.8",
```

```
Pod::Spec.new do |s|
  s.name                 = "React"
  s.version              = "0.13.2"
  s.summary              = "Build high quality mobile apps using React."
  s.description          = <<-DESC
                             React Native apps are built using the React JS
                             framework, and render directly to native UIKit
                             elements using a fully asynchronous architecture.
                             There is no browser and no HTML. We have picked what
                             we think is the best set of features from these and
                             other technologies to build what we hope to become
                             the best product development framework available,
                             with an emphasis on iteration speed, developer
                             delight, continuity of technology, and absolutely
                             beautiful and fast products with no compromises in
                             quality or capability.
                          DESC
  s.homepage             = "http://facebook.github.io/react-native/"
  s.license              = "BSD"
```

RN版本问题



```
{
  "name": "react-native",
  "version": "0.12.0",
  "description": "A framework for building native a
  "license": "BSD-3-Clause",
  "repository": {
    "type": "git",
    "url": "git@github.com:facebook/react-native.gi
  },
  "engines": {
    "node": ">=4"
  },
  "jest": {
    "scriptPreprocessor": "jestSupport/preprocessor
    "setupEnvScriptFile": "jestSupport/env.js",
    "moduleNameMapper": {
      "^image![a-zA-Z0-9$_-]+$": "GlobalImageStub",
      "^[./a-zA-Z0-9$_-]+\\.png$": "RelativeImageSt
    },
    "testPathIgnorePatterns": [
      "/node_modules/"
    ],
    "testFileExtensions": [
      "js"
    ],
    "unmockedModulePathPatterns": [
      "promise",
      "source-map"
```



```
Pod::Spec.new do |s|
  s.name                = "React"
  s.version             = "0.13.2"
  s.summary             = "Build high quality mobile apps usin
  s.description         = <<-DESC
                             React Native apps are built using
                             framework, and render directly to
                             elements using a fully asynchronou
                             There is no browser and no HTML. W
                             we think is the best set of featur
                             other technologies to build what w
                             the best product development frame
                             with an emphasis on iteration spee
                             delight, continuity of technology,
                             beautiful and fast products with n
                             quality or capability.
                          DESC
```

组件升级

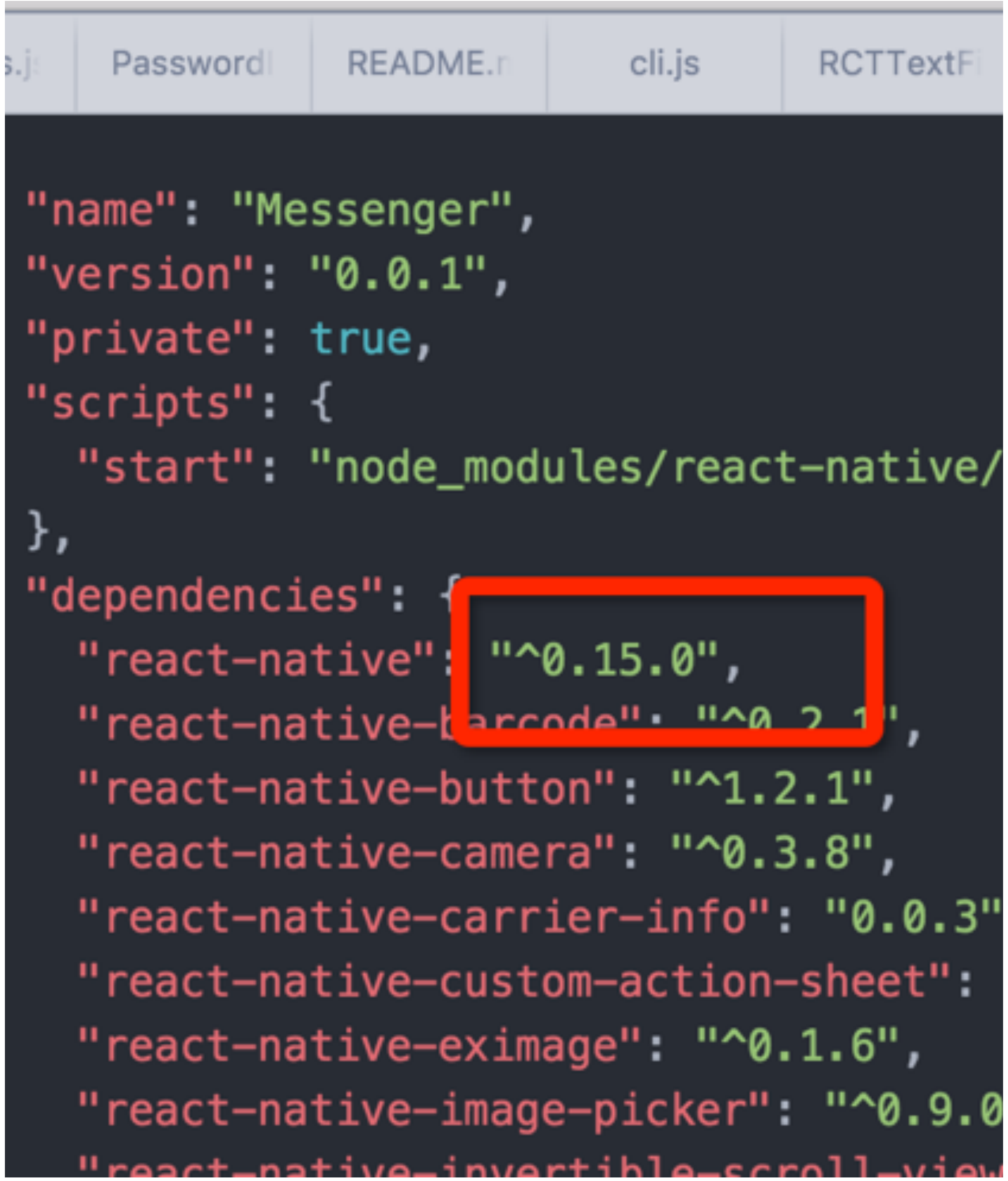
https://www.npmjs.com/package/react-native

package.son 修改成最新版本号

$ npm install

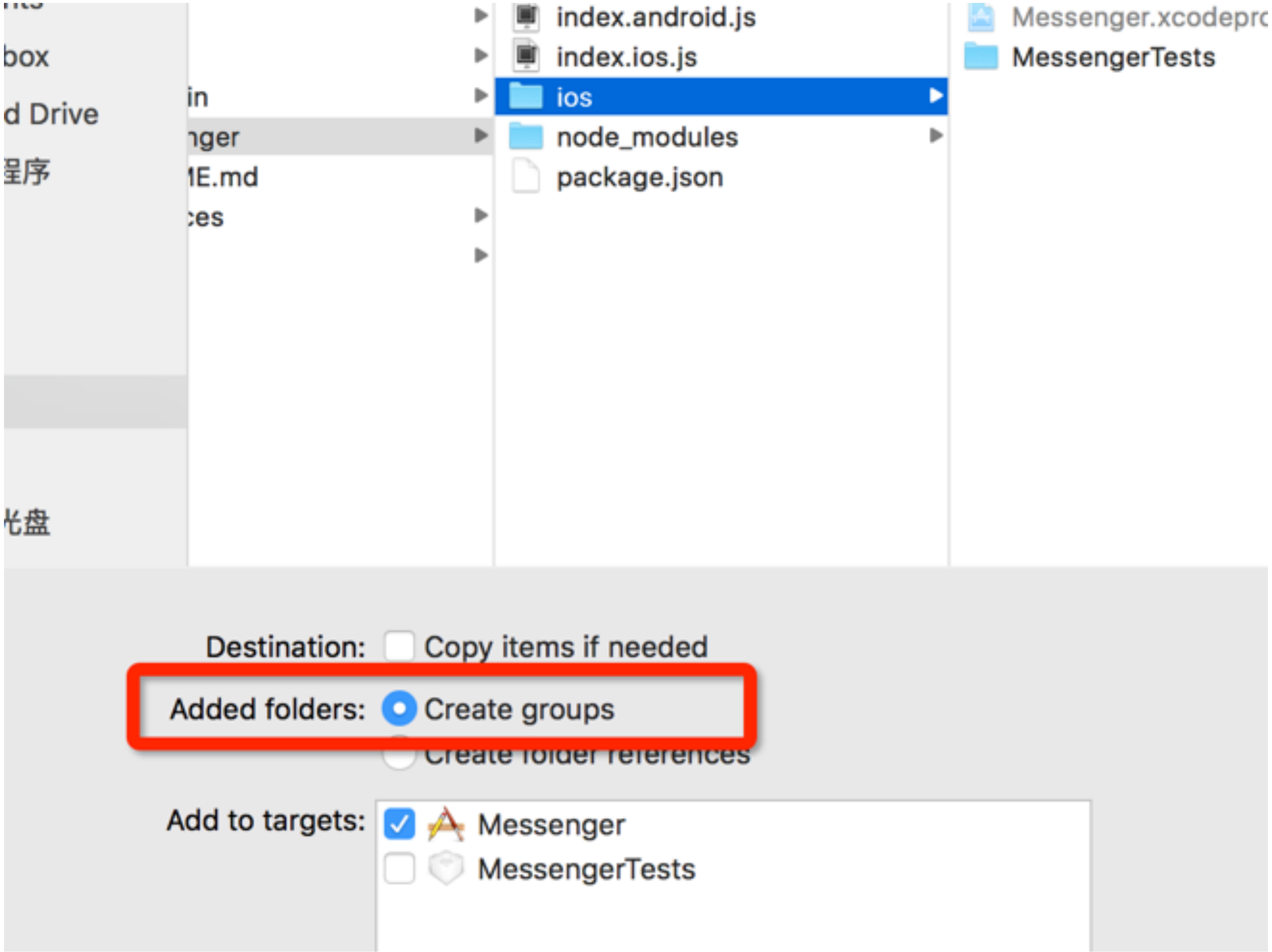
同时升级react-native init的模版

$ react-native upgrade



```
"name": "Messenger",
"version": "0.0.1",
"private": true,
"scripts": {
  "start": "node_modules/react-native/
},
"dependencies": {
  "react-native": "^0.15.0",
  "react-native-barcode": "^0.2.1",
  "react-native-button": "^1.2.1",
  "react-native-camera": "^0.3.8",
  "react-native-carrier-info": "0.0.3"
  "react-native-custom-action-sheet":
  "react-native-eximage": "^0.1.6",
  "react-native-image-picker": "^0.9.0
  "react-native-invertible-scroll-view
```

源码组件添加



Destination: ☐ Copy items if needed

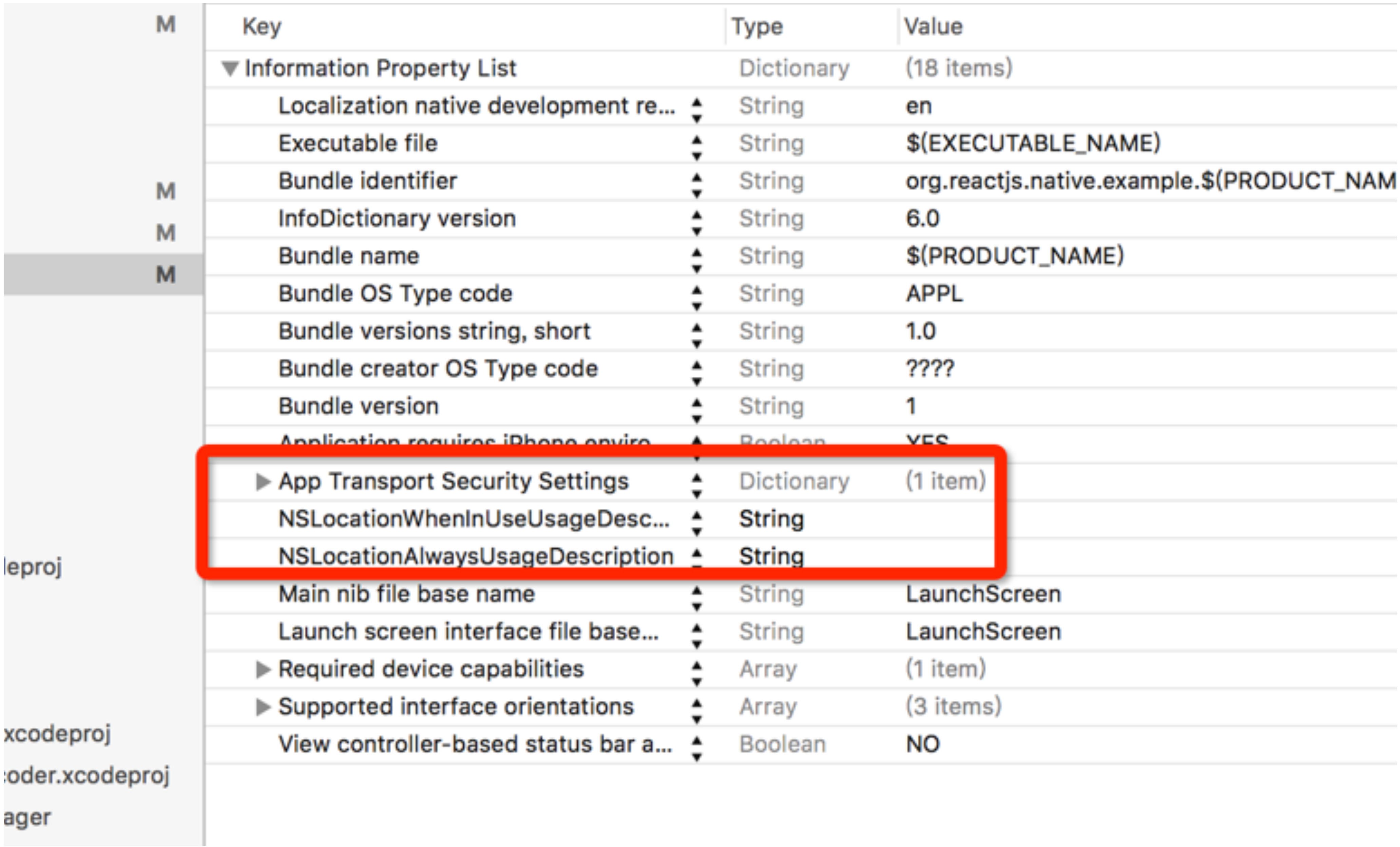
Added folders: ● Create groups

○ Create folder references

Add to targets: ☑ Messenger

☐ MessengerTests

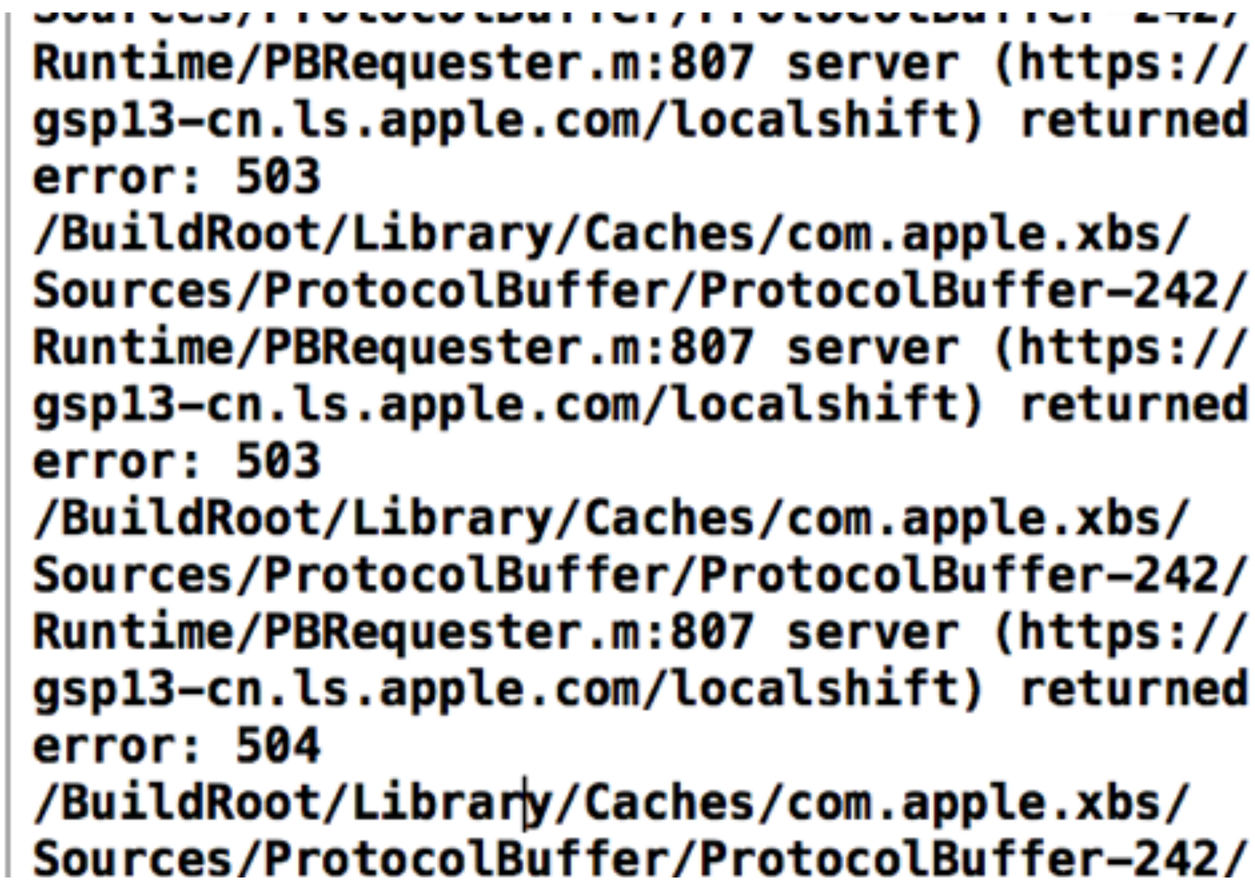
info.list



Key	Type	Value
▼ Information Property List	Dictionary	(18 items)
Localization native development re...	String	en
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	org.reactjs.native.example.$(PRODUCT_NAM
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1
Application requires iPhone enviro...	Boolean	YES
▶ App Transport Security Settings	Dictionary	(1 item)
NSLocationWhenInUseUsageDesc...	String	
NSLocationAlwaysUsageDescription	String	
Main nib file base name	String	LaunchScreen
Launch screen interface file base...	String	LaunchScreen
▶ Required device capabilities	Array	(1 item)
▶ Supported interface orientations	Array	(3 items)
View controller-based status bar a...	Boolean	NO

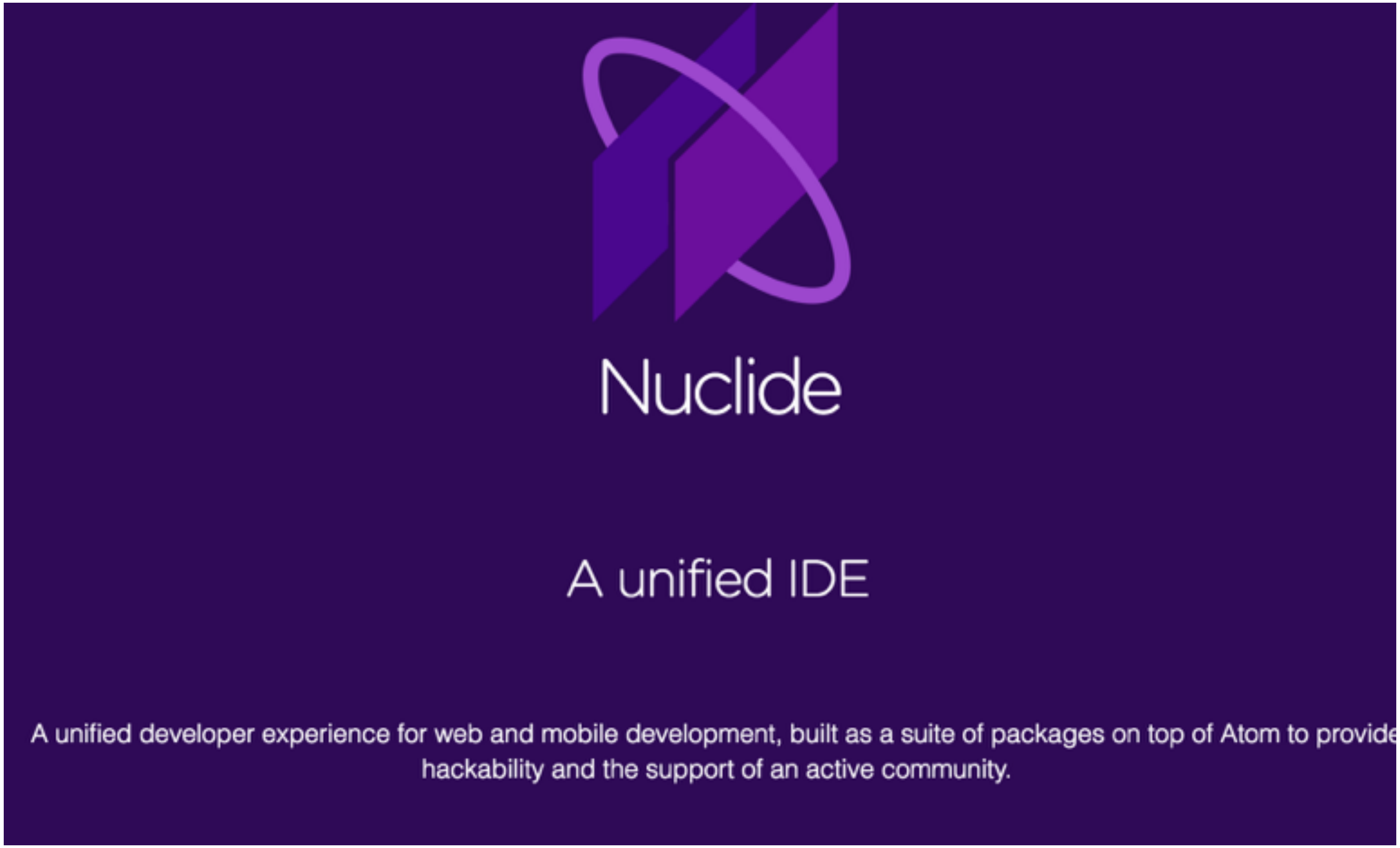
Location 反查服务

原理是向谷歌服务器请求的，由谷歌服务器完成这个这个命令，但是谷歌服务器会做一个判断，60s内不能进行重复的请求，如果重复请求的次数过多，谷歌会认为你是在测试，这样谷歌回终止你的服务.



```
Sources/ProtocolBuffer/ProtocolBuffer-242/
Runtime/PBRequester.m:807 server (https://
gsp13-cn.ls.apple.com/localshift) returned
error: 503
/BuildRoot/Library/Caches/com.apple.xbs/
Sources/ProtocolBuffer/ProtocolBuffer-242/
Runtime/PBRequester.m:807 server (https://
gsp13-cn.ls.apple.com/localshift) returned
error: 503
/BuildRoot/Library/Caches/com.apple.xbs/
Sources/ProtocolBuffer/ProtocolBuffer-242/
Runtime/PBRequester.m:807 server (https://
gsp13-cn.ls.apple.com/localshift) returned
error: 504
/BuildRoot/Library/Caches/com.apple.xbs/
Sources/ProtocolBuffer/ProtocolBuffer-242/
```

SVGeocoder http://samvermette.com/164



Nuclide

A unified IDE

A unified developer experience for web and mobile development, built as a suite of packages on top of Atom to provide hackability and the support of an active community.

rn 编辑器Atom

```
apm install nuclide-installer
```

http://nuclide.io

Questions?

Thanks