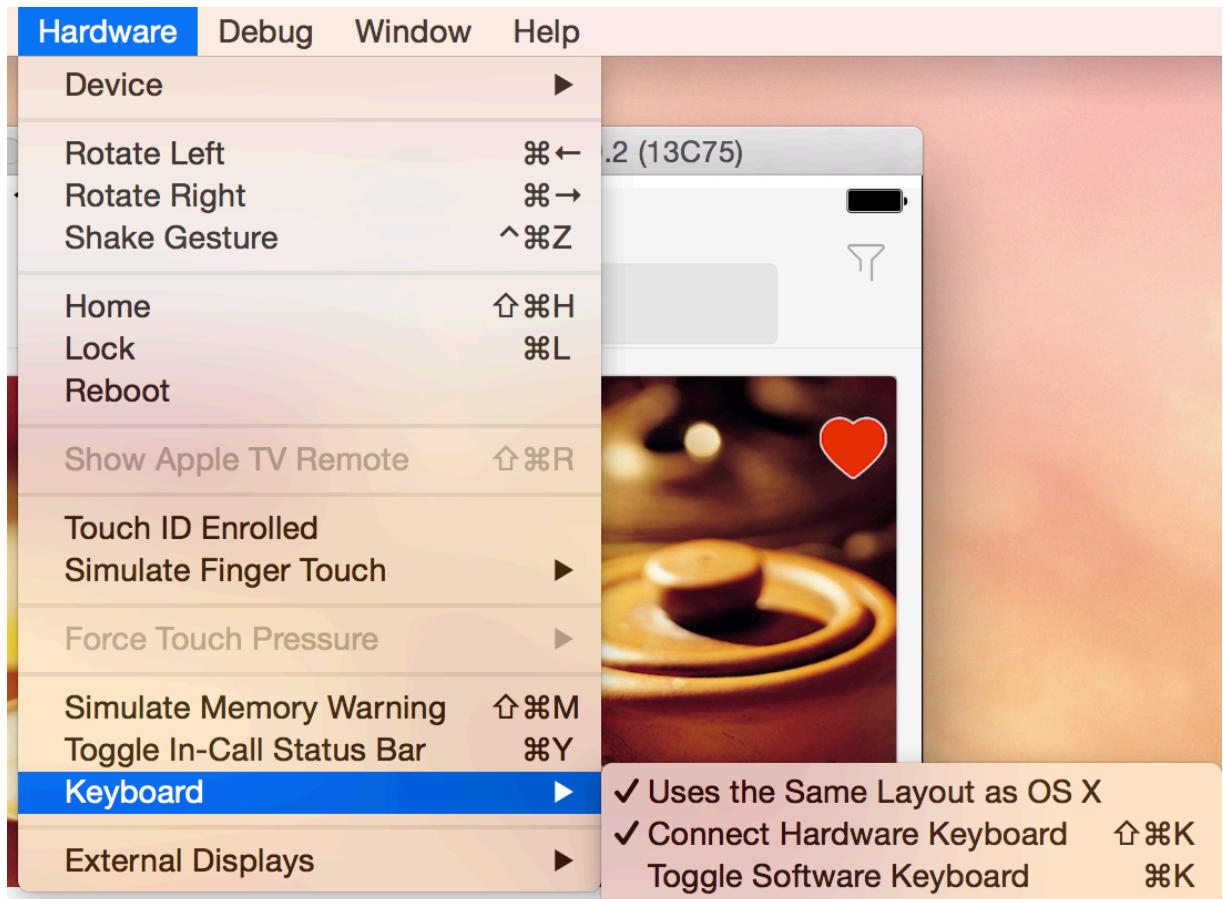


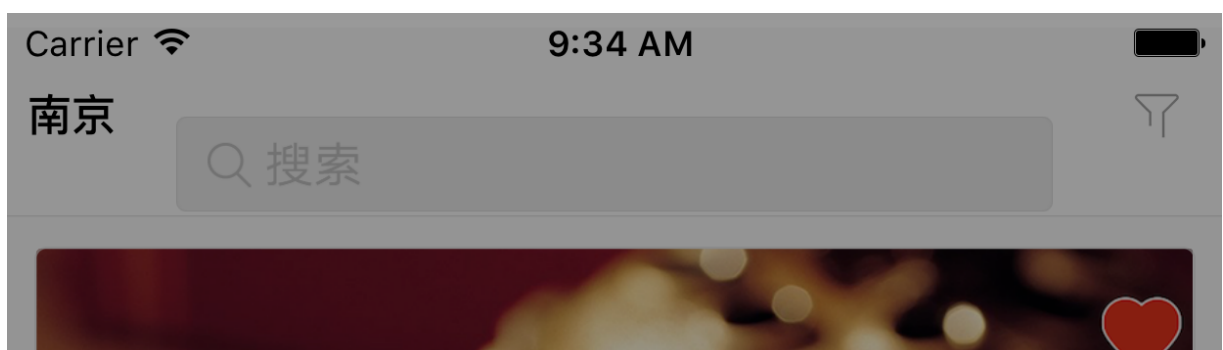
React-Native之调试篇

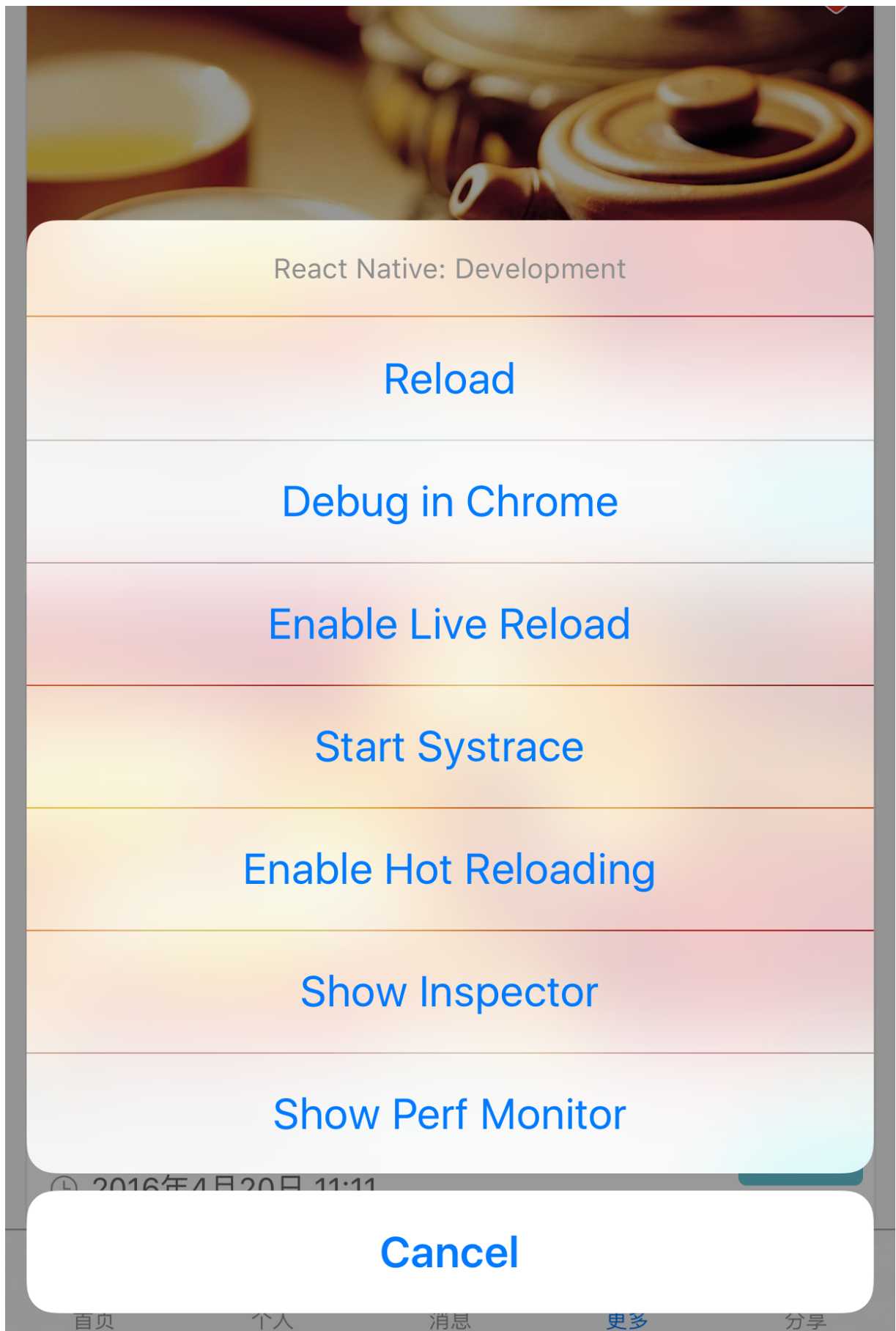
模拟器调试、断点跟踪js代码

1. 在键盘中按 `CMD+R` 可以刷新react页面，可以实时更新js代码。如果键盘没有响应，请确保模拟器中选中了 `Connect Hardware keyboard` （如下图）



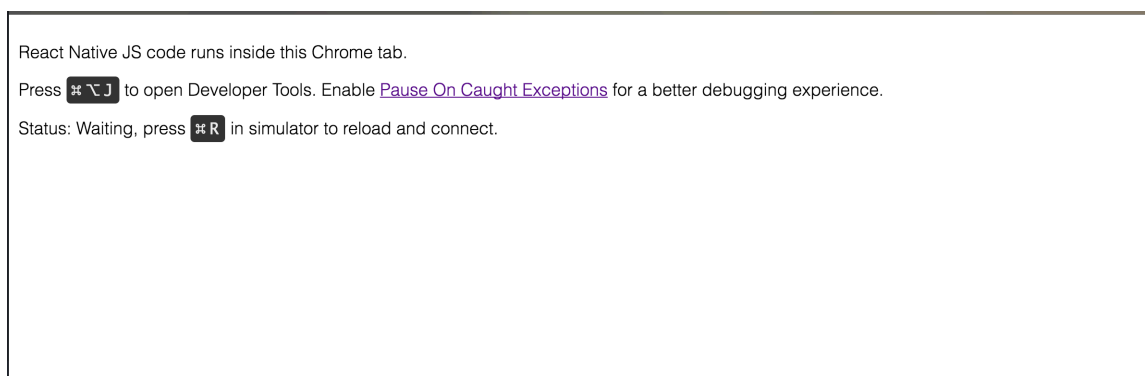
2. `CMD+D` 可以打开模拟器上的选择调试页面。
 - `Reload` 刷新页面
 - `Debug in Chrome`，使用谷歌浏览器调试react-native。谷歌浏览器内置V8 JavaScript引擎，与iOS内置的JavaScriptCore引擎差别很小。
 - `Show inspector` 类似于iOS的Flex，可以用于UI调试





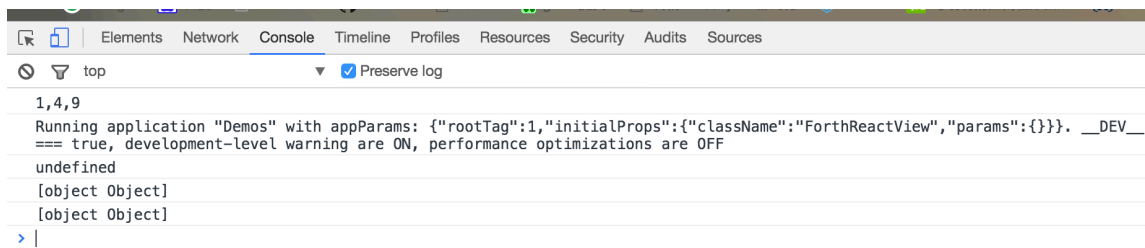
3. `Debug in Chrome` 使用谷歌浏览器断点调试

- 在上图2中选择 **Debug in Chrome** ，然后按照在下图谷歌浏览器中的提示： **Press CMD+Option+J** 快捷键，打开react调试窗口。



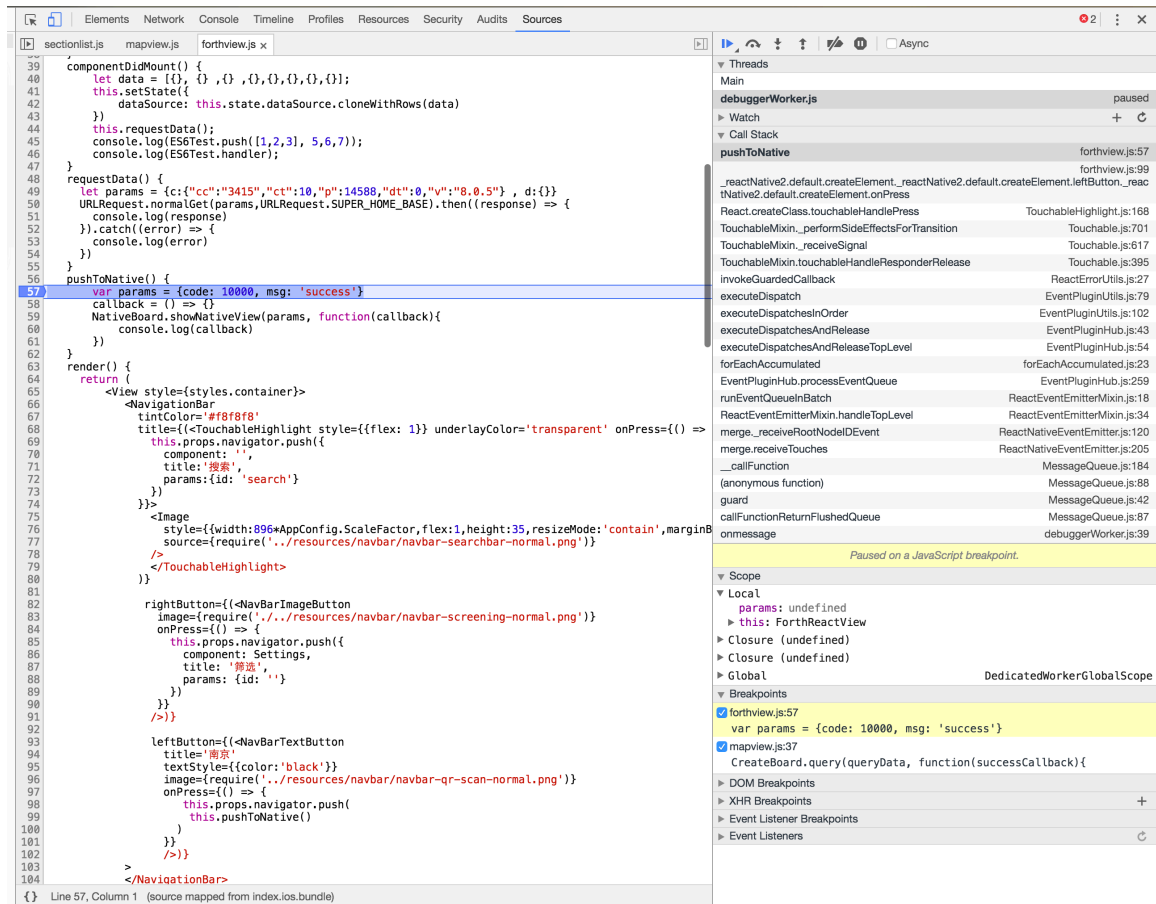
- 调试窗口是这样的

*Sources*模块可以在运行成功时 按**CMD+P** 快捷键打开你的js文件； *Console*模块可以在下面输入**console.log()**打印参数的值



- 断点调试堆栈

从下面的调试界面可以看到，右边是**pushToNative**方法的调用栈，左边还可以设置断点，右上方可以调试跟踪断点



◦ over

4. 查看报错信息

仔细查看react-native的报错信息，发现连行（207：12）都显示出来了，是不是很方便？

**SyntaxError /Users/schiller/iOS/MyGLDemos/
reactjs/forth/forthview.js: Unexpected token
(207:12)**

RCTFatal + 104

**-[RCTBatchedBridge stopLoadingWithError:] +
745**

**___25-[RCTBatchedBridge start]_block_invoke_2
+ 67**

dispatch call block and release + 12

```
_dispatch_client_callout + 8

_dispatch_main_queue_callback_4CF + 1738

__CFRunLoop_IS_SERVICING_THE_MAIN_DISPATCH_QUEUE__ + 9

__CFRunLoopRun + 2073

CFRunLoopRunSpecific + 488

GSEventRunModal + 161

UIApplicationMain + 171

main + 111
```

Dismiss (ESC)

Reload JS (⌘R)

5. over

运行别人的项目

1. 运行别人开源react-native项目，要执行 `npm install` 导入项目所需的依赖库，项目具体的依赖库可以参考项目中的 `package.json` 文件，该文件类似iOS的 `podfile`。

使用纯代码调试

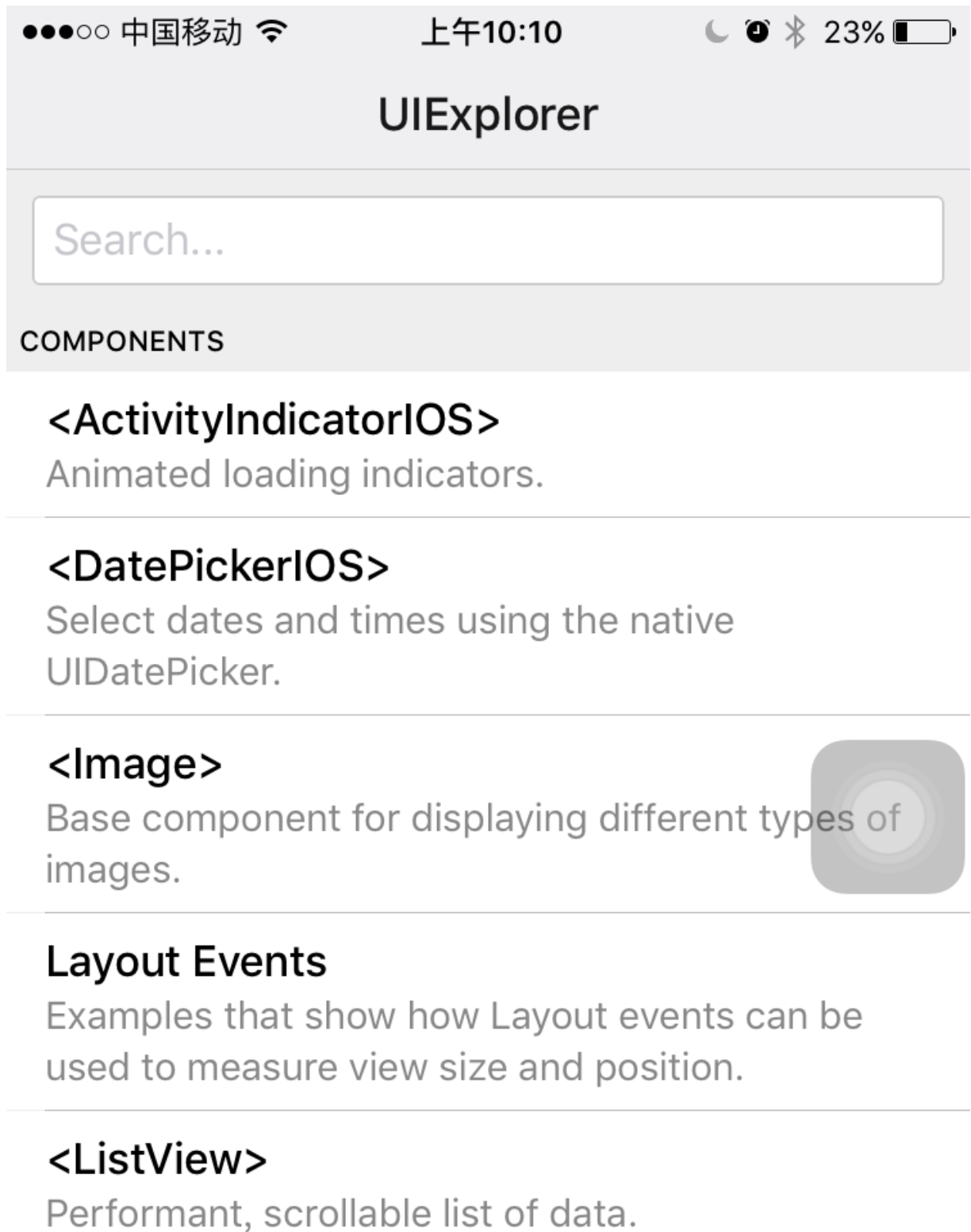
1. 可以用 `console.log()` 或者 `alert()` 直接打印参数值。
2. 也可以直接查看Xcode控制台，`console.log()`打印的日志信息会显示在Xcode控制台上。

在真机上调式

我们尝试在真机上跑Facebook官方测试项目：<https://github.com/facebook/react-native>

1. 真机上运行Facebook的UIExplorer项目

UIExplorer项目含有比较全面的react控件示例和相关API示例，可以说是一个很不错的学习参考文档



<ListView> - Grid Layout

Flexbox grid layout.

<ListView> - Paging

Floating headers & layout animations.

<MapView>

Base component to display maps

2. 在模拟器上运行该项目，找到文件夹里的 `Example-->UIExplorer` 文件夹（这个项目比较乱）。

```
$ cd /UIExplorer
$ npm install //必须执行这个命令，需要导入项目的依赖

#模拟器调试时提示polifill报错或者其它报错，可能是该项目所需要的依赖包没有
导入的原因，比如babel转码等依赖包
http://es6.ruanyifeng.com/#docs/intro

#在AppDelegate中查看这段代码，根据sourceURL加载你的js代码，

- (NSURL *)sourceURLForBridge:(__unused RCTBridge *)bridge
{
    ...
    // option1 ,这是在模拟器上调试的地址，链接到你的js代码，
    sourceURL = [NSURL
URLWithString:@"http://localhost:8081/Examples/UIExplorer/UIExplorer
App.ios.bundle?platform=ios&dev=true"];
    //option2, 这是将js文件打包成bundle，以bundle形式加载
    /**
     * OPTION 2
     * Load from pre-bundled file on disk. To re-generate the static
bundle, `cd`
     * to your Xcode project folder and run
     *
     * $ curl
'http://localhost:8081/Examples/UIExplorer/UIExplorerApp.ios.bundle?
platform=ios' -o main.jsbundle
```

```
*
* then add the `main.jsbundle` file to your project and
uncomment this line:
*/
// sourceURL = [[NSBundle mainBundle] URLForResource:@"main"
withExtension:@"jsbundle"];
...
return sourceURL;
}

#注意：如果要在真机上调试，需要注释掉 option1,打开option2。然后按照提示将
js代码打成bundle并拖到你的项目中。
$ cd /UIexplorer
$ curl
'http://localhost:8081/Examples/UIExplorer/UIExplorerApp.ios.bundle?
platform=ios' -o main.jsbundle
# add the main.jsbundle file to your project and uncomment ...

#如果你出现 报 'no url script' 错误，很可能是你上面步骤不对哦。
```

3. over

react-native调试原理

1. 这一块还有待深入研究，从sourceURL来看，应该是在项目中使用node创建了一个小型调试服务器，可以实时加载js代码。