

# User guide

## Vopti application for iOS devices

**Jordy Cabannes  
Summer Research Student  
Summer 2016**

## Table of contents

<b>I REQUIREMENTS</b>	<b>1</b>
<b>II INSTALLATION</b>	<b>2</b>
<b>III DEBUG</b>	<b>14</b>

# I Requirements

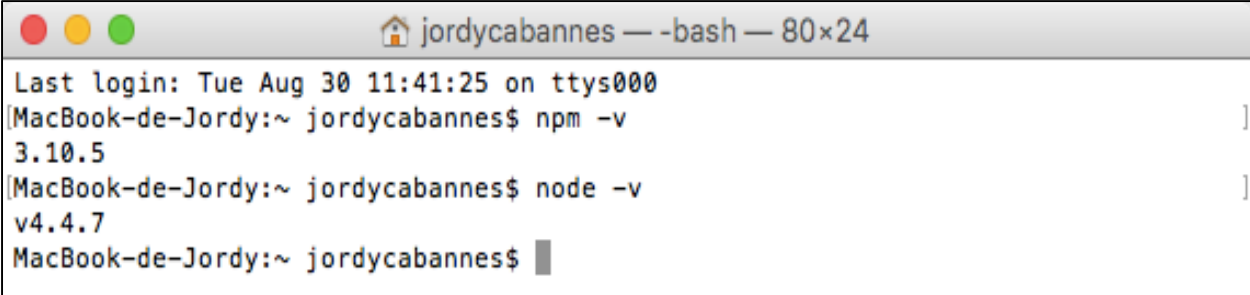
- 1- You need to install Xcode on your Apple computer, you can download it on the AppStore via the link : <https://itunes.apple.com/us/app/xcode/id497799835?mt=12>
- 2- You need to install node.js and npm. You can download them on <https://nodejs.org/en/>
- 3- When you have installed node.js and npm, to check that they are well installed, open a terminal and input

```
$ npm -v
```

and

```
$ node -v
```

You should have something like that

A screenshot of a macOS terminal window. The title bar shows a home icon, the username 'jordycabannes', and the shell '-bash' with a window size of '80x24'. The terminal content shows the last login time as 'Tue Aug 30 11:41:25 on ttys000'. The user runs 'npm -v' and the output is '3.10.5'. Then the user runs 'node -v' and the output is 'v4.4.7'. The prompt 'MacBook-de-Jordy:~ jordycabannes\$' is shown at the bottom with a cursor.

```
MacBook-de-Jordy:~ jordycabannes$ npm -v
3.10.5
MacBook-de-Jordy:~ jordycabannes$ node -v
v4.4.7
MacBook-de-Jordy:~ jordycabannes$
```

- 4- You have to install cordova. In a terminal just write

```
$ sudo npm install -g cordova
```

- 5- You should have an Apple developer account. If you do not have one, you can create it on : <https://idmsa.apple.com/IDMSWebAuth/login?appldKey=891bd3417a7776362562d2197f89480a8547b108fd934911bcbea0110d07f757&baseUrl=https://developer.apple.com/&path=%2Faccount%2F&rv=1>

## II Installation

- 1- Download the directory Vopti\_app from [https://github.com/JordyCabannes/Vopticonnect\\_application](https://github.com/JordyCabannes/Vopticonnect_application)
- 2- Unzip the file downloaded
- 3- Create a Cordova Project, go into your Desktop or your Documents folder thanks to a terminal

```
$ cd Documents
```

or

```
$ cd Desktop
```

and write

```
$ cordova create appName com.example.appName AppName
```

\**appName* is the name of the application folder

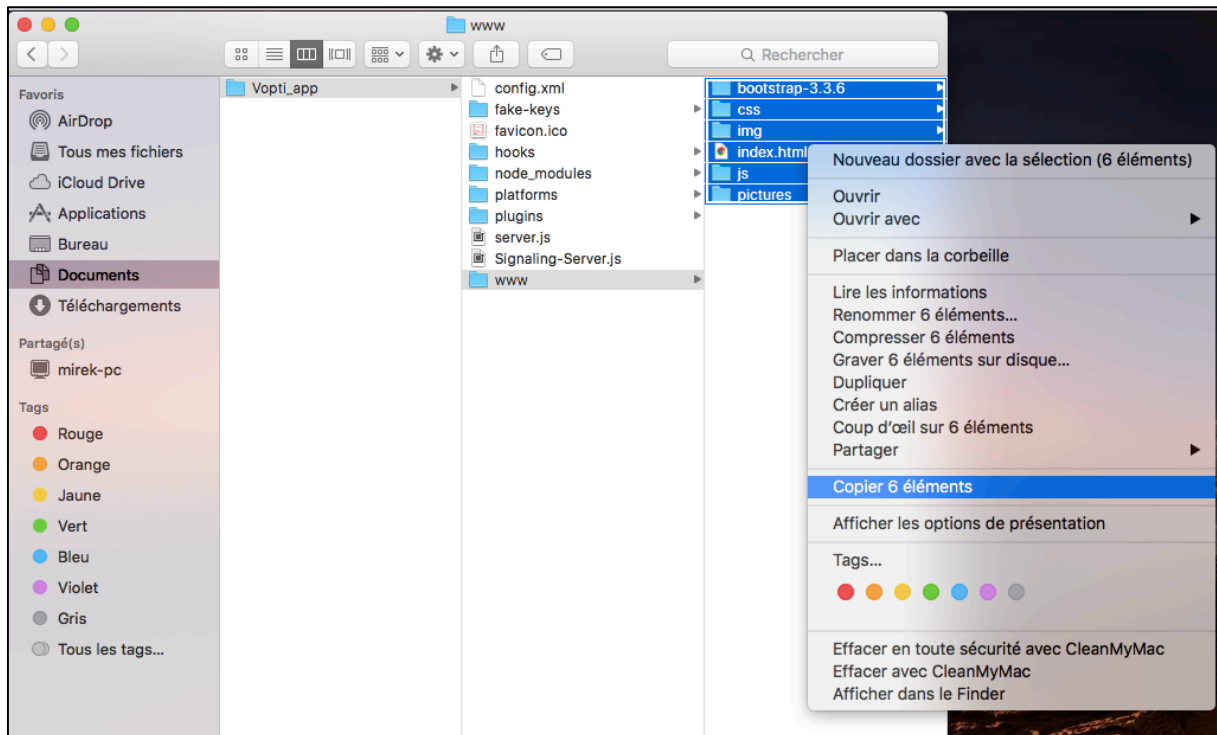
\* *com.example.appName* is the ID of the application

\**AppName* is the visible name of the application

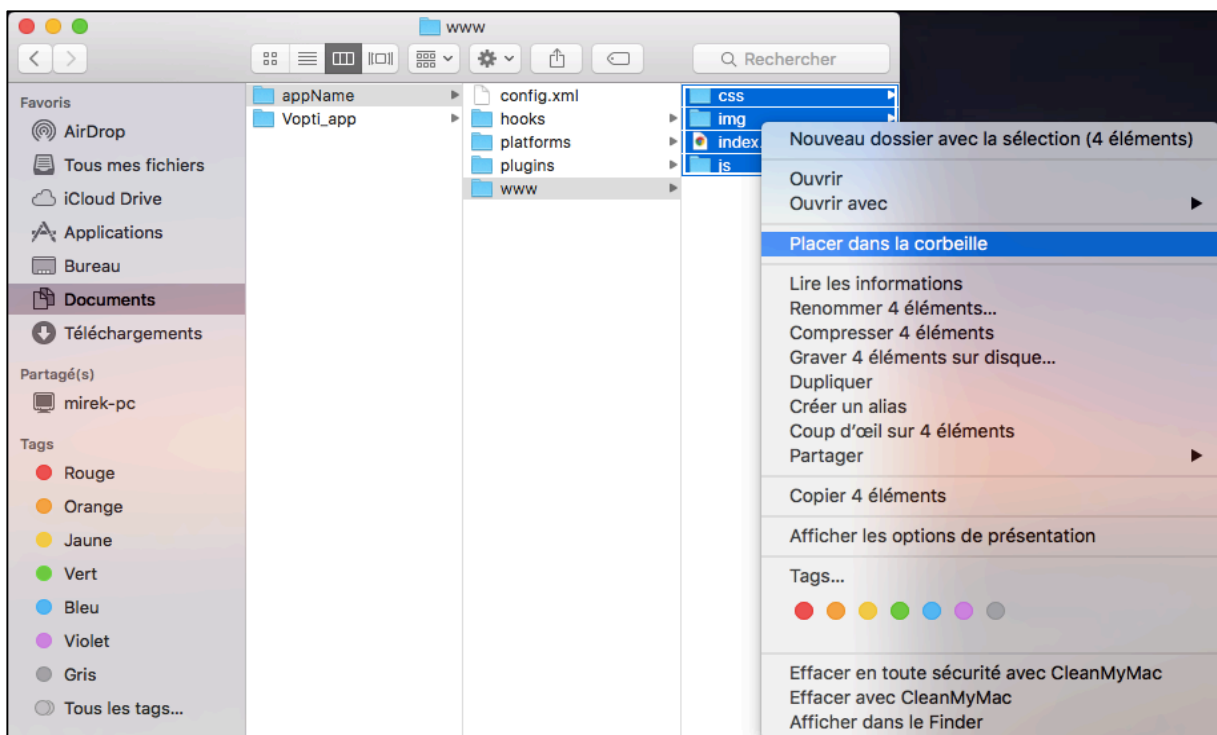
All the previous steps are explained at the address

<http://cordova.apache.org/docs/en/latest/guide/cli/index.html>

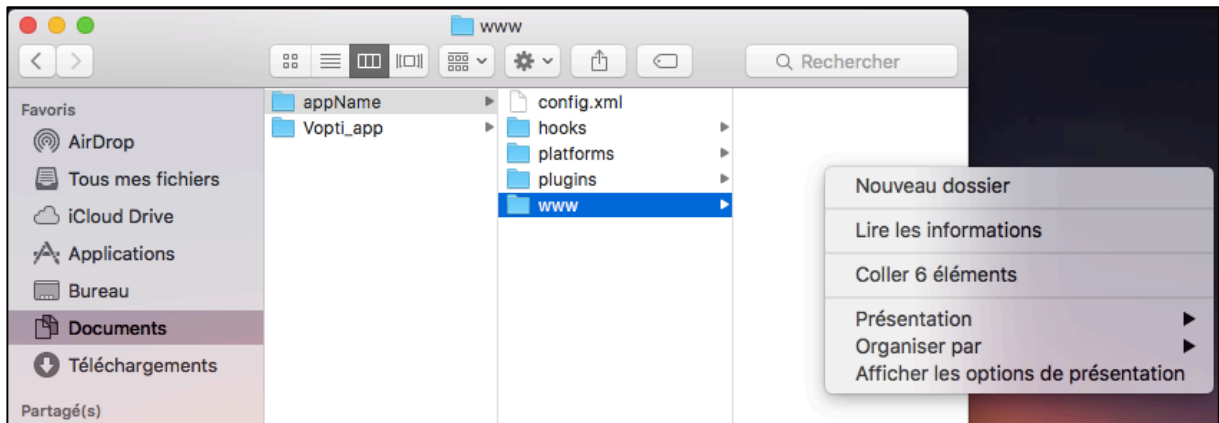
- 4- Now, with the Finder, go into the directory that you downloaded. Afterwards go into its directory "www" and copy all the files



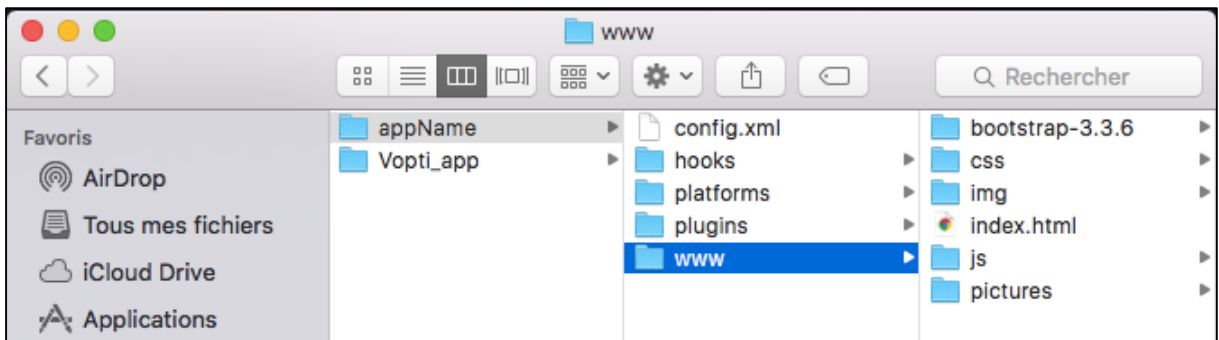
- 5- Go into your cordova project and delete all the files and repertories present in the www folder.



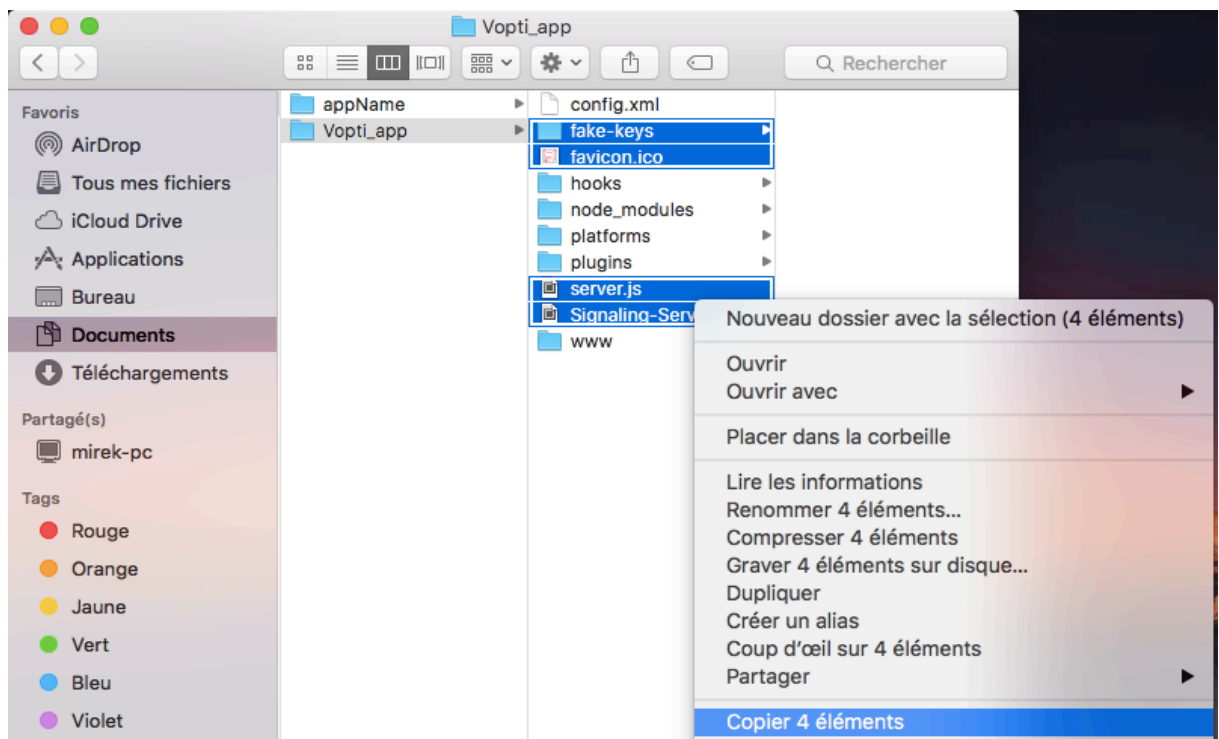
- 6- Now you can paste the files and repertories in this empty www folder.



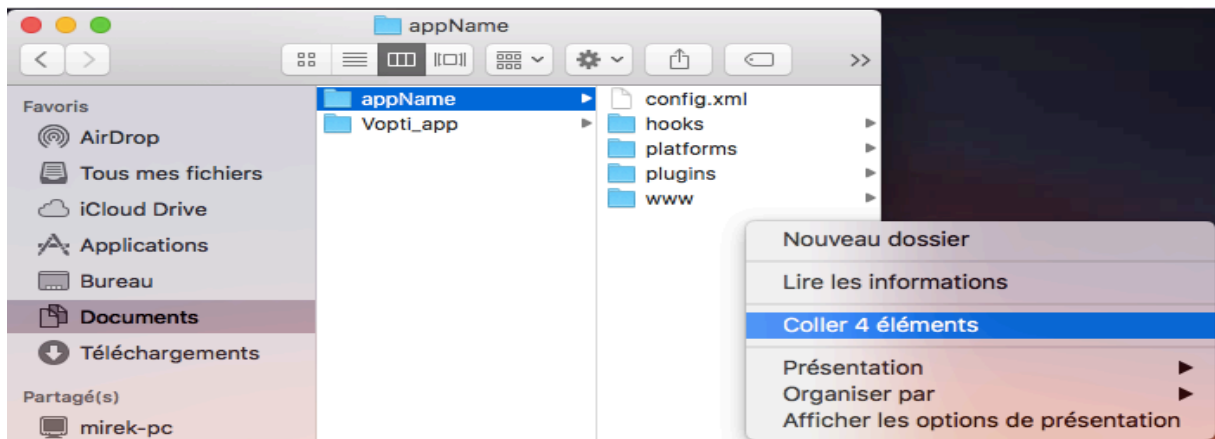
You will obtain something like this



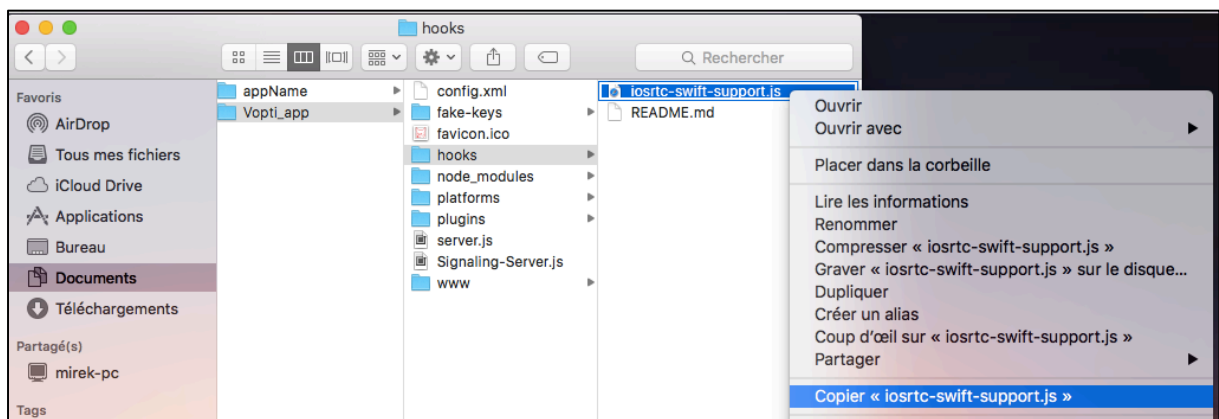
- 7- With the Finder, go into the directory that you downloaded. Afterwards copy the files `server.js`, `Signaling-Server.js`, `favicon.ico` and the repertory `fake-keys`.



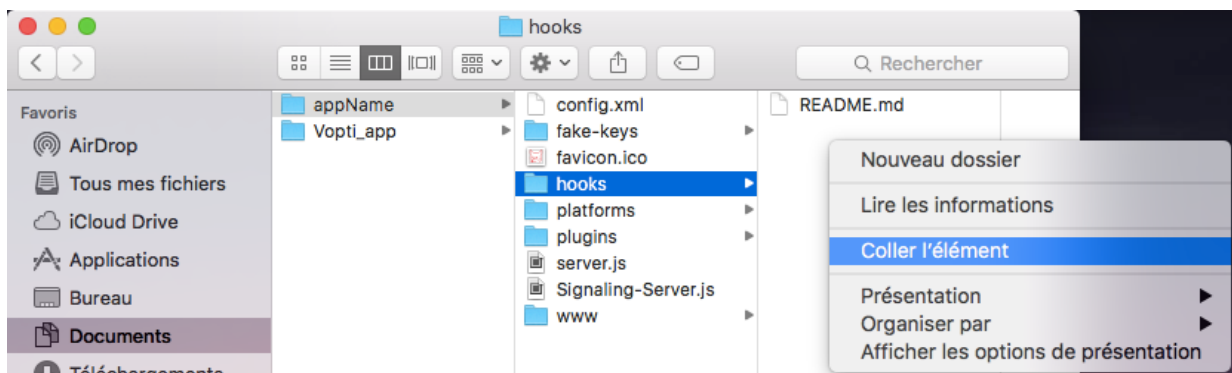
8- Paste them into your Cordova Project.



9- With the Finder, go into the directory that you downloaded. Afterwards go into hooks folder and copy the file iosrtc-swift-support.js



10- Paste this file into the hooks folder of your Cordova Project



11- Open the config.xml file of your project and add this line just below the line 24:

```
<hook type="after_platform_add" src="hooks/iosrtc-swift-support.js" />
```

You will get something like this



```
1 <?xml version='1.0' encoding='utf-8'?>
2 <widget id="com.VoptiConnect.VoptiApp" version="1.0.0" xmlns="http://www.w3.org/ns/widgets" xmlns:cdv="http://cordova.apache.org/
ns/1.0">
3   <name>VoptiApp</name>
4   <description>
5     A sample Apache Cordova application that responds to the deviceready event.
6   </description>
7   <author email="dev@cordova.apache.org" href="http://cordova.io">
8     Apache Cordova Team
9   </author>
10  <content src="index.html" />
11  <plugin name="cordova-plugin-whitelist" spec="1" />
12  <access origin="*" />
13  <allow-intent href="http://*/*" />
14  <allow-intent href="https://*/*" />
15  <allow-intent href="tel:*" />
16  <allow-intent href="sms:*" />
17  <allow-intent href="mailto:*" />
18  <allow-intent href="geo:*" />
19  <platform name="android">
20    <allow-intent href="market:*" />
21  </platform>
22  <platform name="ios">
23    <allow-intent href="itms:*" />
24    <allow-intent href="itms-apps:*" />
25    <hook type="after_platform_add" src="hooks/iosrtc-swift-support.js" />
26  </platform>
27 </widget>
28
```

12 – In a terminal, write ifconfig

```
$ ifconfig
```

and copy your IP address

```
MacBook-de-Jordy:Documents jordycabannes$ ifconfig
[lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=3<RXCSUM,TXCSUM>
    inet6 ::1 prefixlen 128
    inet 127.0.0.1 netmask 0xff000000
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
    nd6 options=1<PERFORMNUD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=27<RXCSUM,TXCSUM,VLAN_MTU,TSO4>
    ether d4:9a:20:f8:94:ac
    nd6 options=1<PERFORMNUD>
    media: autoselect
    status: inactive
en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 34:15:9e:95:84:d8
    inet6 fe80::3615:9eff:fe95:84d8%en1 prefixlen 64 scopeid 0x5
    inet 192 netmask 0xffffffff00 broadcast 192.168.0.255
    nd6 options=1<PERFORMNUD>
    media: autoselect
    status: active
p2p0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
    ether 06:15:9e:95:84:d8
    media: autoselect
    status: inactive
MacBook-de-Jordy:Documents jordycabannes$
```



- 13- Open the file server.js of your project and give your IP address to the variable serverIP

```
1 // Muaz Khan - www.MuazKhan.com
2 // MIT License - www.WebRTC-Experiment.com/licence
3 // Documentation - github.com/muaz-khan/RTCMultiConnection
4
5 var serverIP = "192.168.1.1";
6 var serverPORT = 443;
7
```

- 14- With the finder, go into the “www” folder and in the “js” folder. Open the file “index.js”. Go to the line 161, and give the value “https://your\_ip\_address” to the variable connection.socketURL.

- 15- In a terminal go to your Cordova project, and write:

```
$ cordova plugin add cordova-plugin-iosrtc
$ cordova plugin add cordova-plugin-whitelist
$ sudo npm init
```

For “sudo npm init”, you have to enter some information for your app. First of all, a “name” with lowercase. After you can press the “enter” button until the end of this “sudo npm init” command.

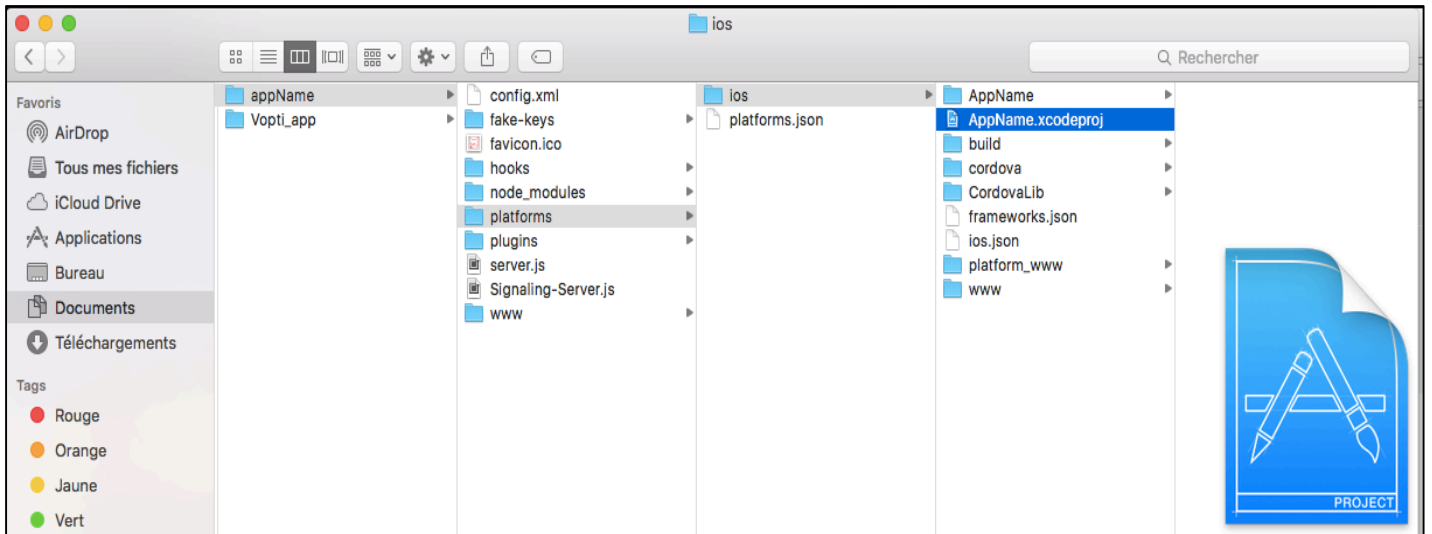
Afterwards, you can write these following command lines.

```
$ sudo npm install
$ sudo npm install Xcode
$ sudo npm install express
$ sudo npm install socket.io
```

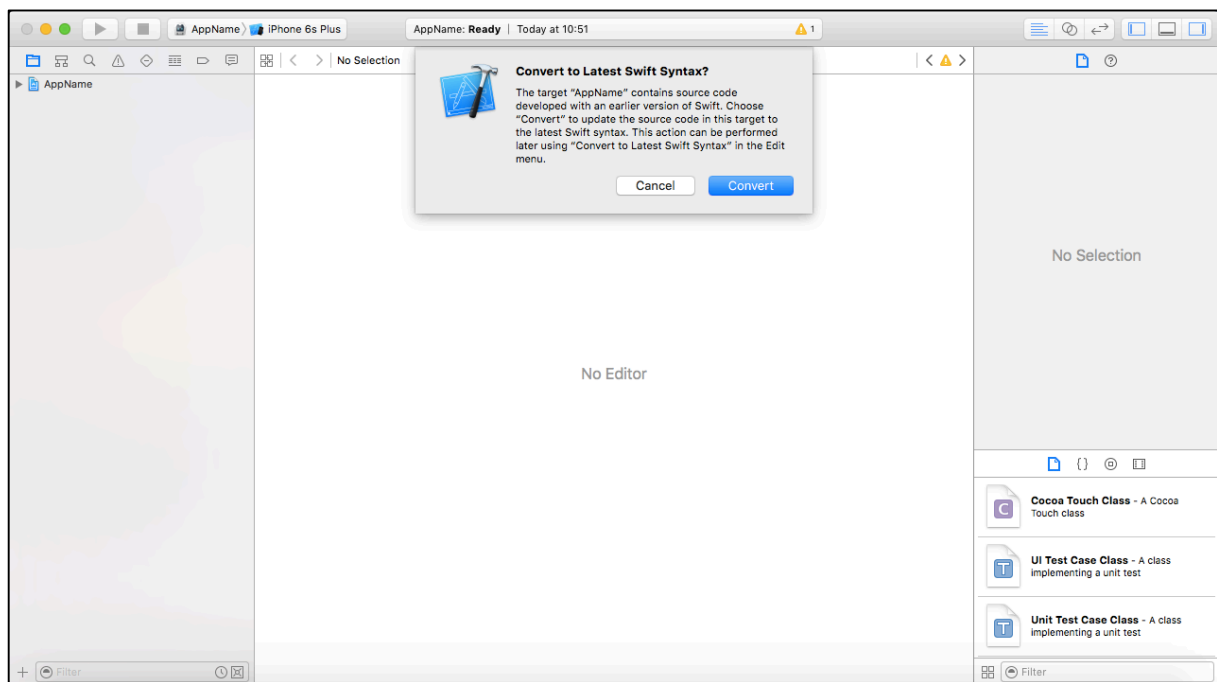
- 16- Now write

```
$ cordova platform add ios
$ cordova prepare
$ cordova build ios
```

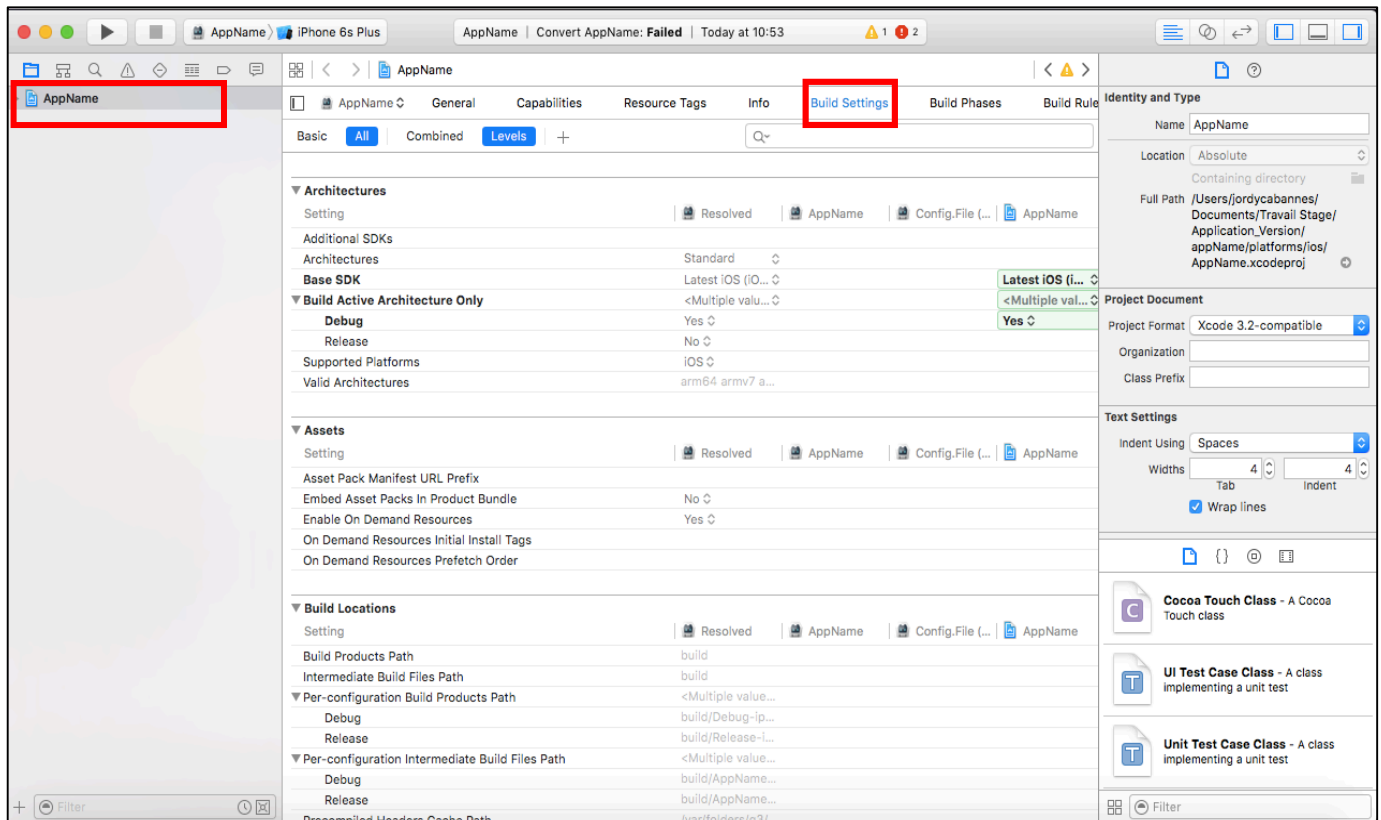
17- Go into the folder of your project, now if you open the platform repository you can see an ios repository. Open it and double click on the file “AppName.xcodeproj”



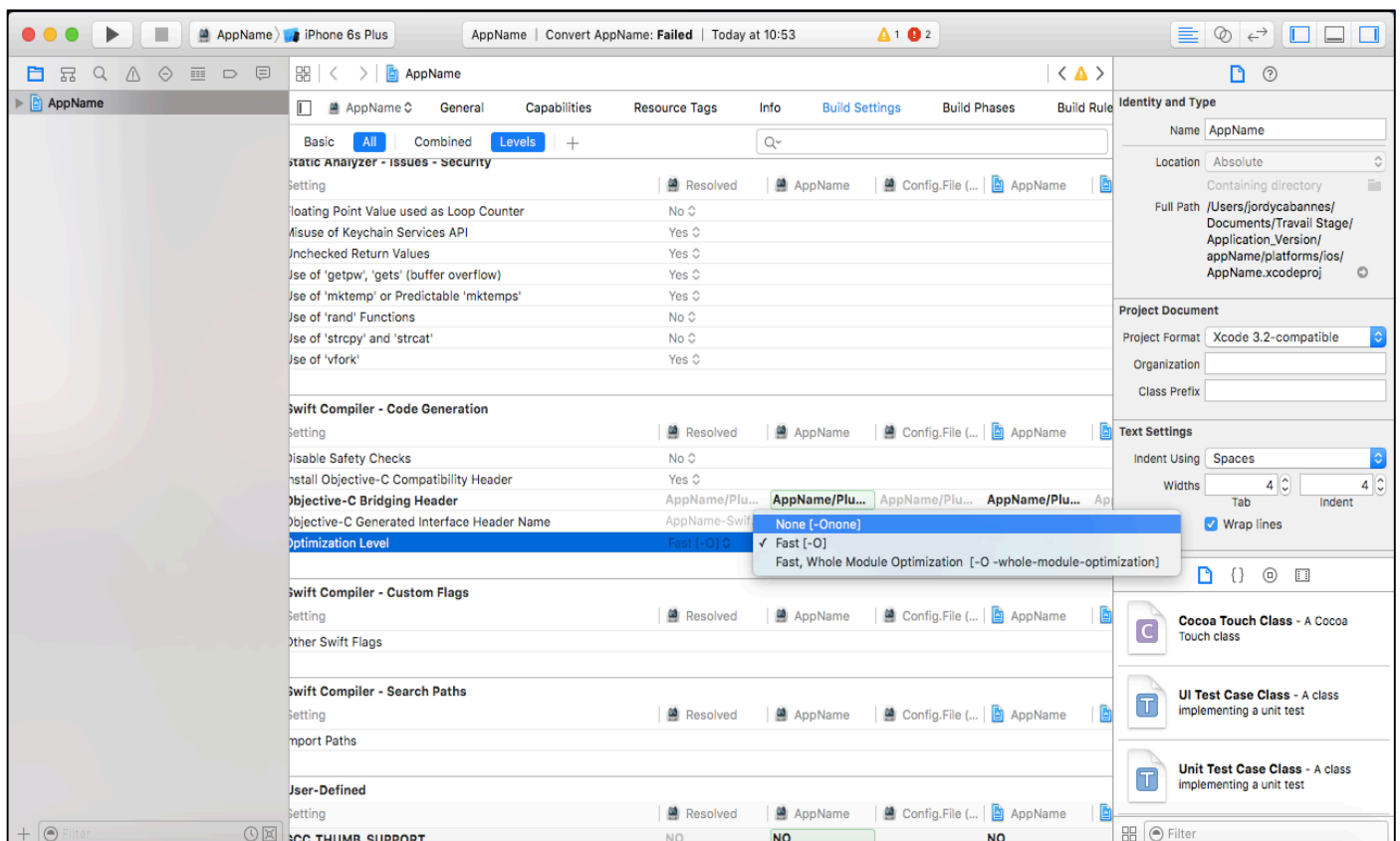
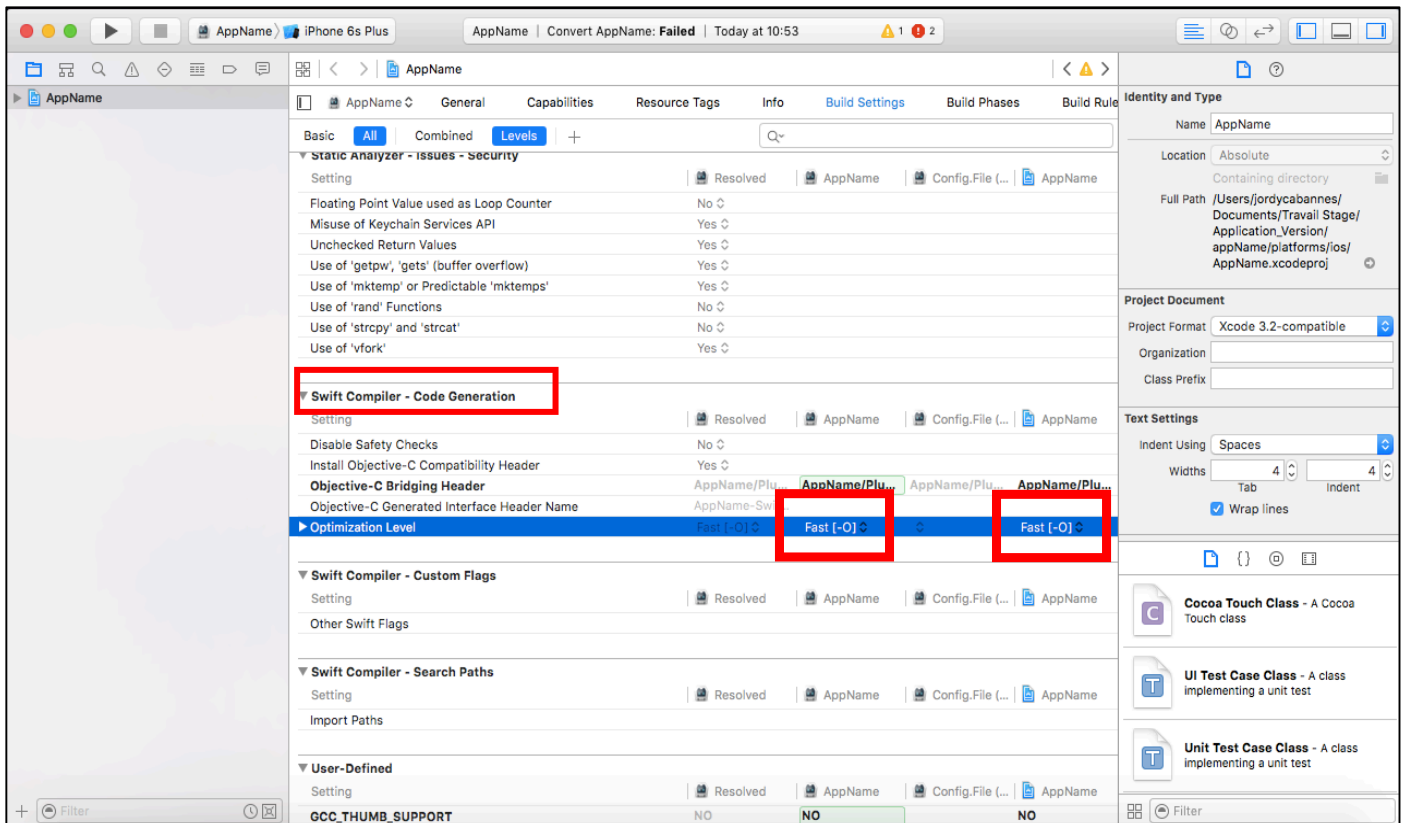
18- Now, Xcode should be open, you have to click on the button “Convert” of the pop-up which appears, then “Next”, “Next” again and “Done”.



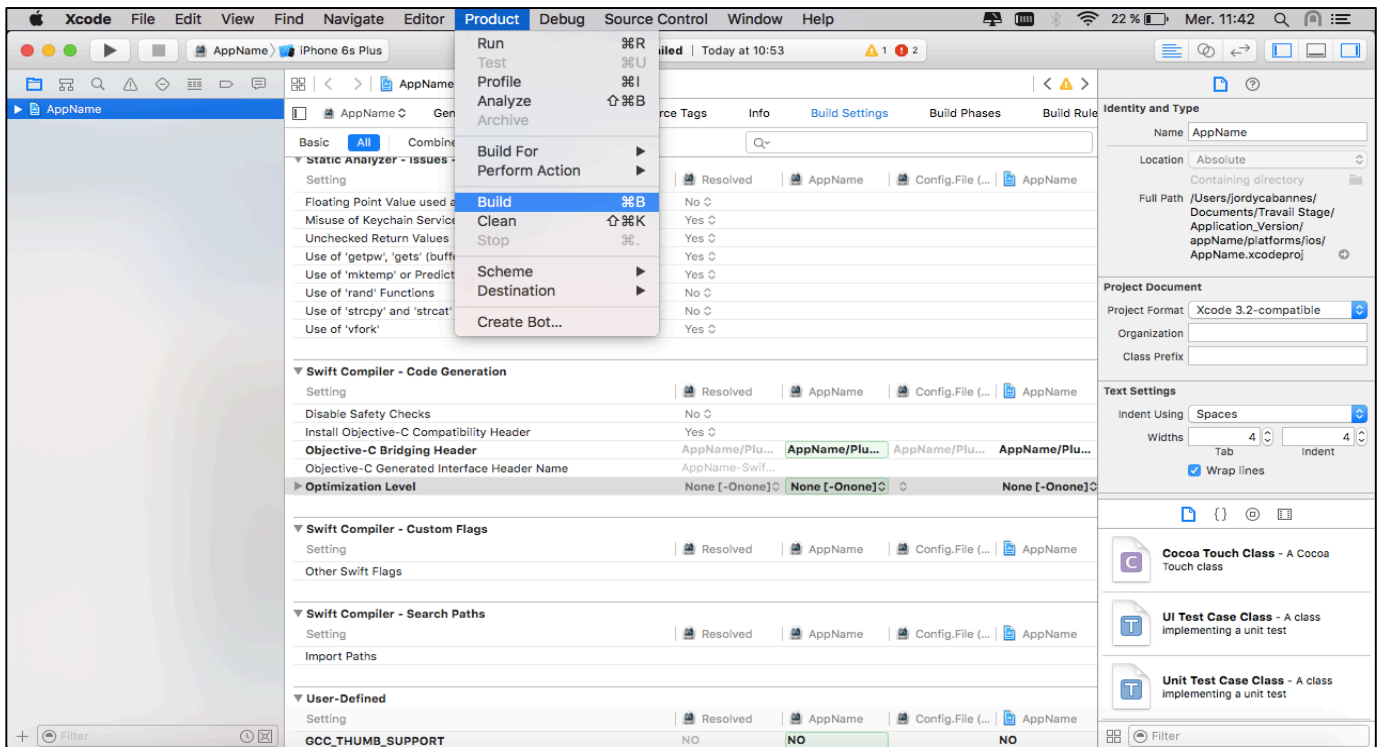
19- Click on the folder of your project on the right of the Xcode window. Now in the center of the window, click on Build Settings



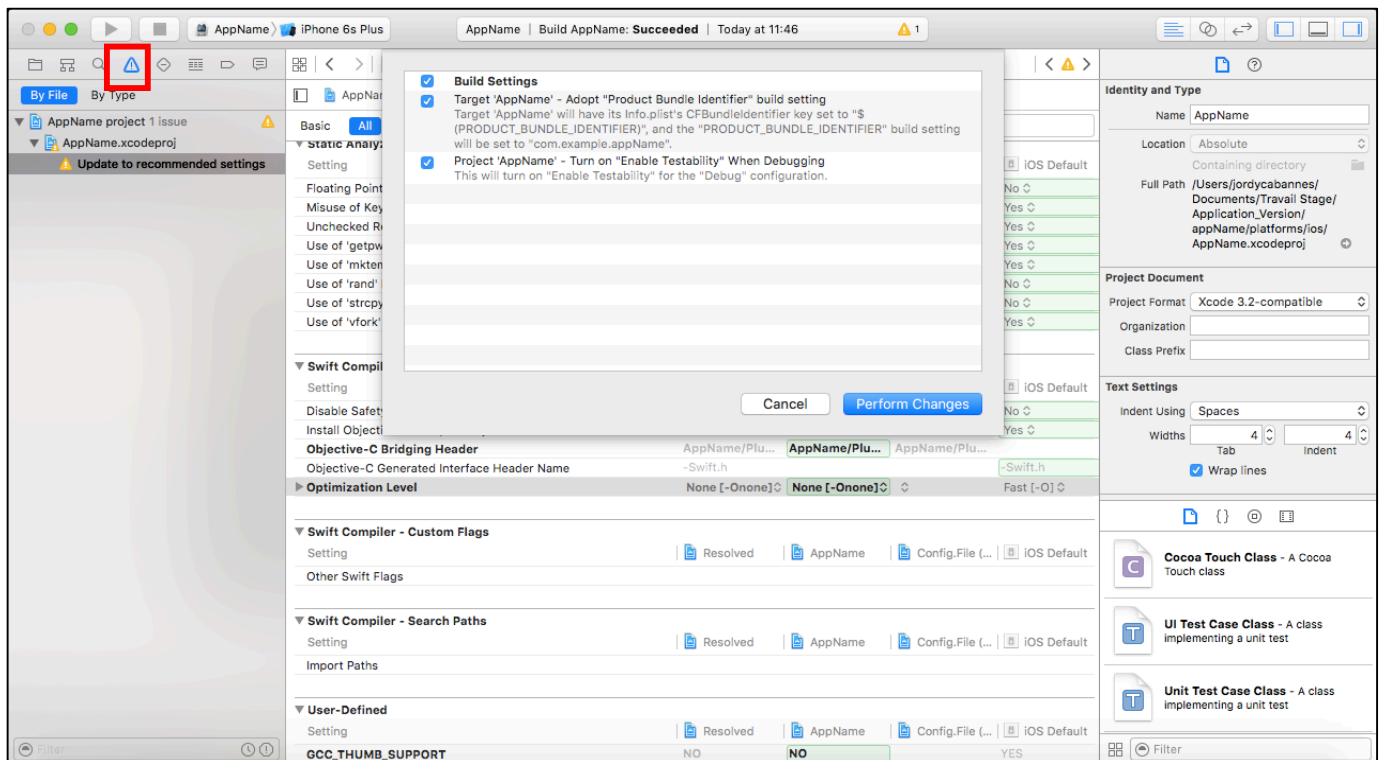
20- Scroll down until to see “Switch Compiler – Code Generation”. Then you will have to change the two “Fast[-Q]” options to “None[-Onone]”



21- In the task bar click on “Product” and “Build”.

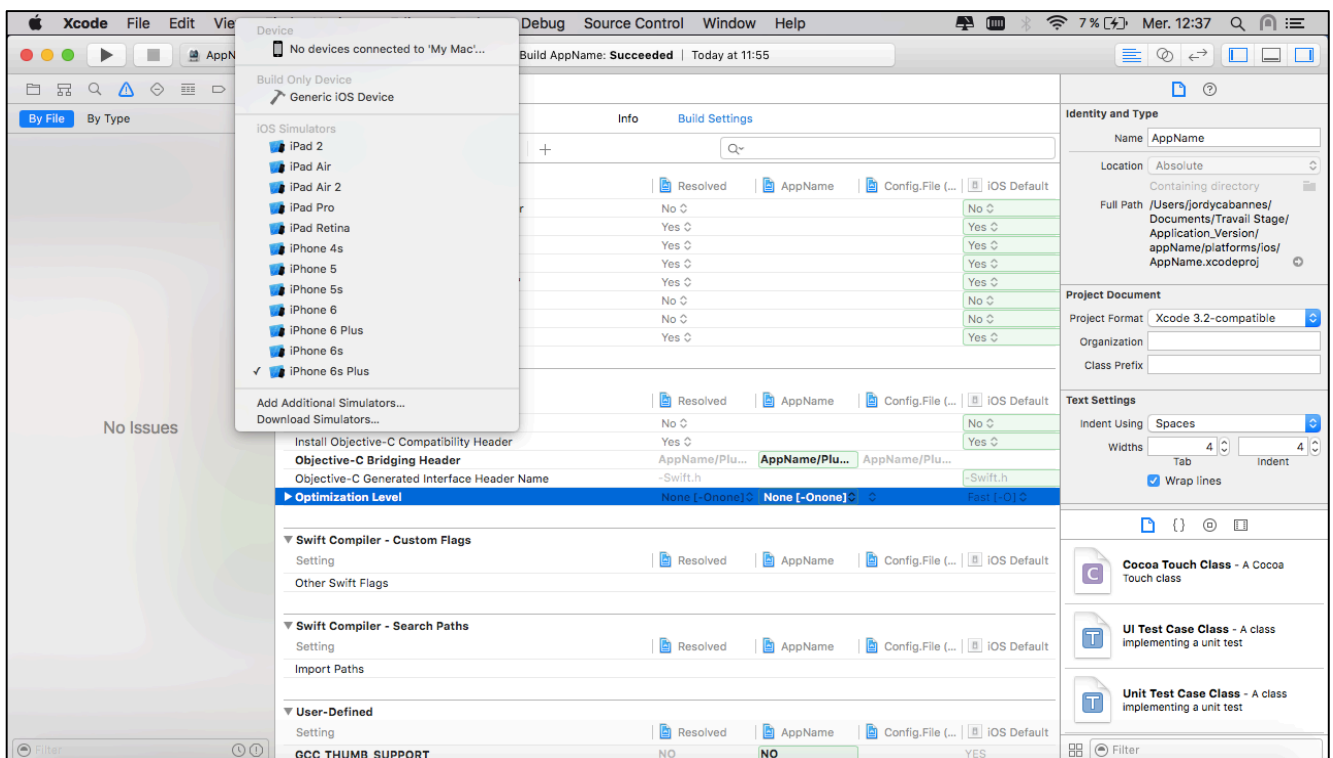
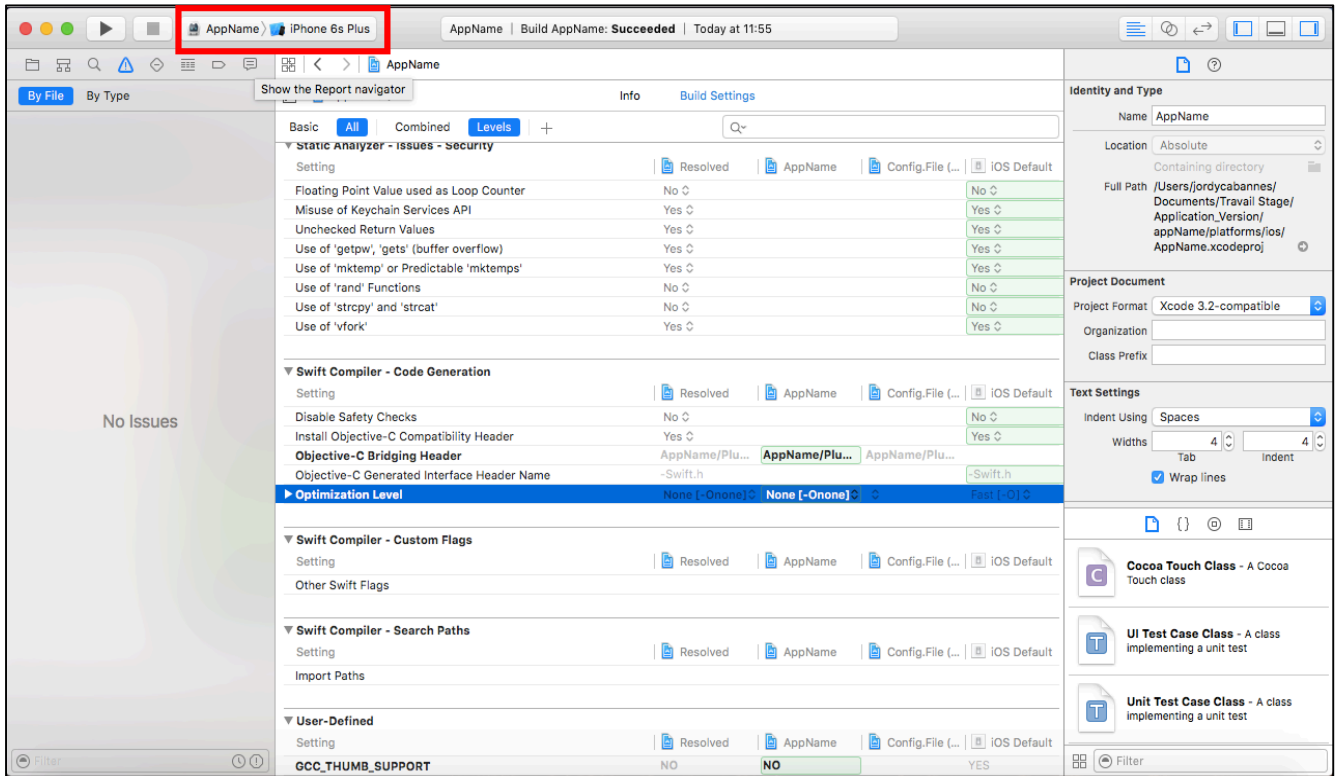


22- Click on the Warning icon. Then click on “Update to recommended settings”.  
Afterward click on the “Perform Changes” button of the pop-up which appears.



23- Another pop-up will appear, just press “enter”. After, in the task bar click on “Product” and “Build”.

24- Now for testing, choose the iOS device for the test in the dropdown menu or plug in your iOS device \* with USB wire and it will appear in the dropdown menu. After that you have to click on Play button to launch the simulator.



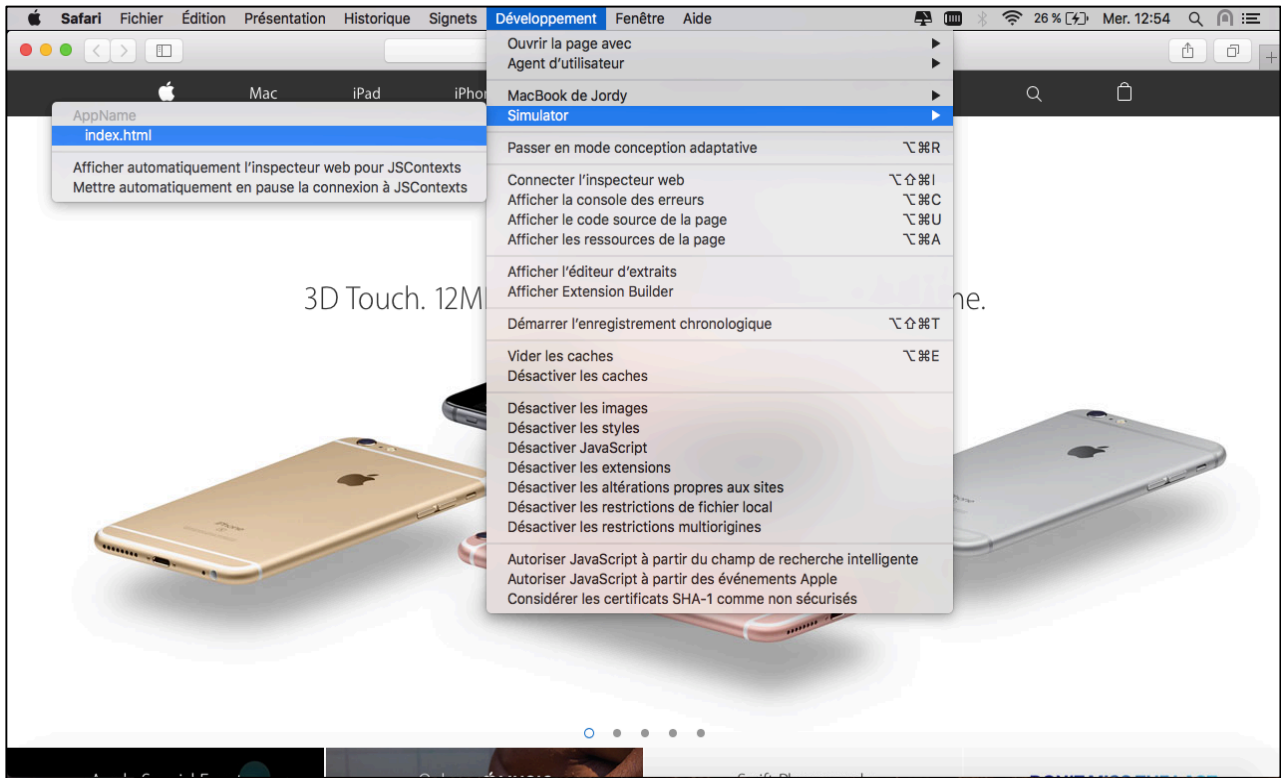
If you want to make any modifications in the code, you have to touch the files in the www folder of your project. And when you have finished the modifications, open a terminal, go into your cordova project with the terminal and write this command line.

```
$ cordova platform remove ios
```

Then you will have to repeat the steps 16 to 24 if you want to test again your application.

# III Debug

When you have launched your application, to debug and see if there are errors, you have to open Safari and in the task bar, click on “Development” and on “Simulator”/ your iOS device. Then click on “index.html”.



Then you will have the debug window for your application which is running on the Xcode simulator or on your iOS device

