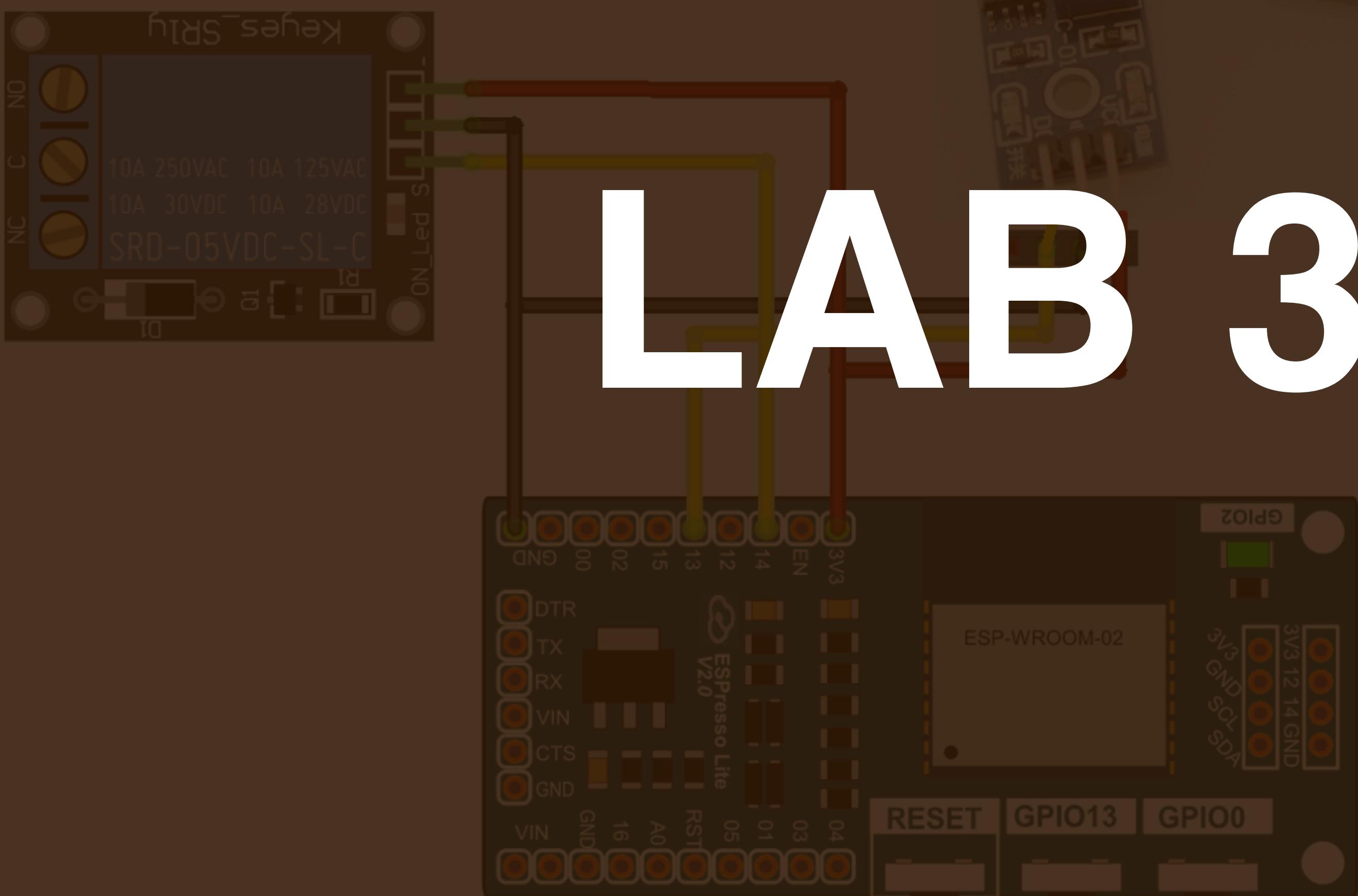
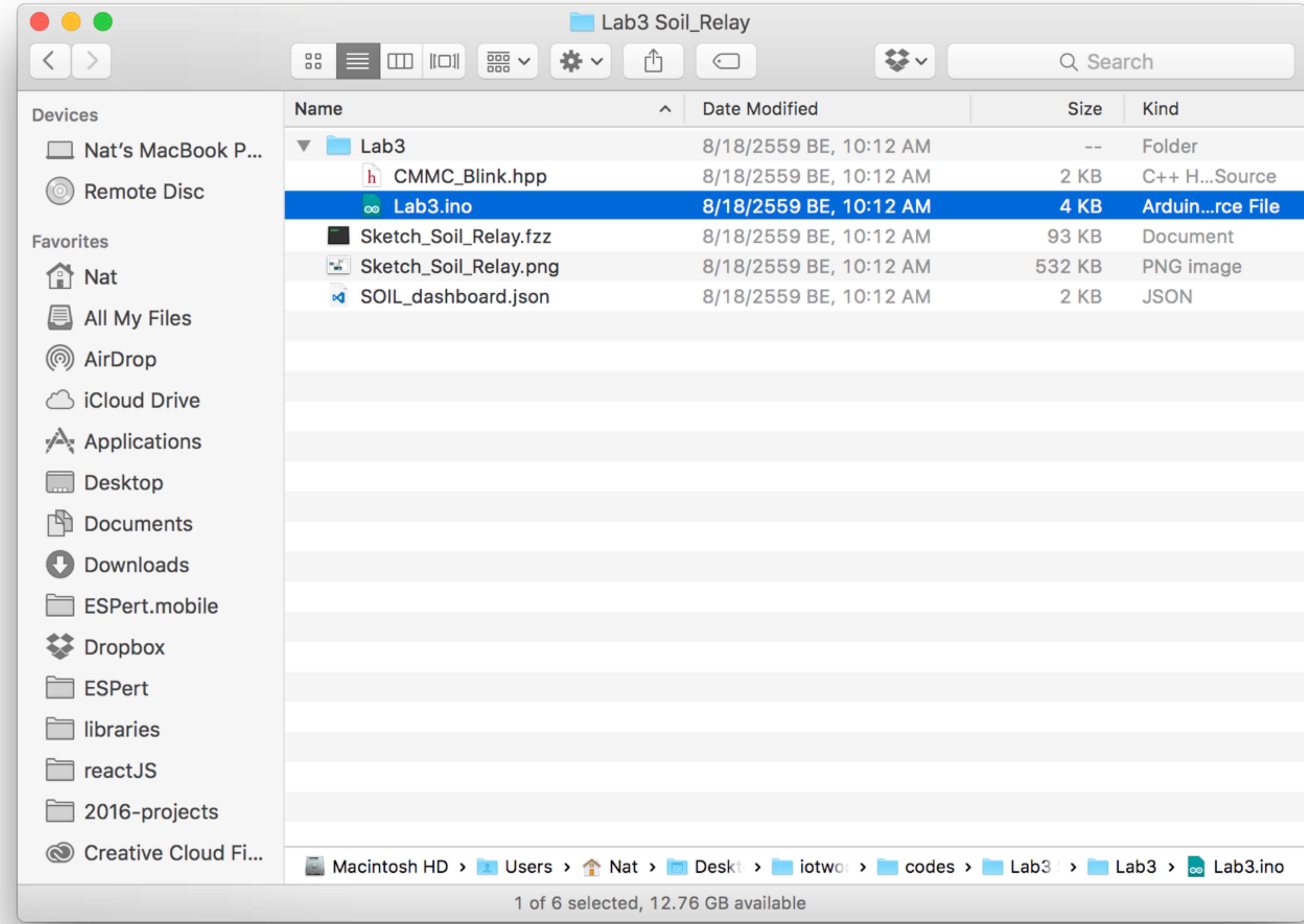


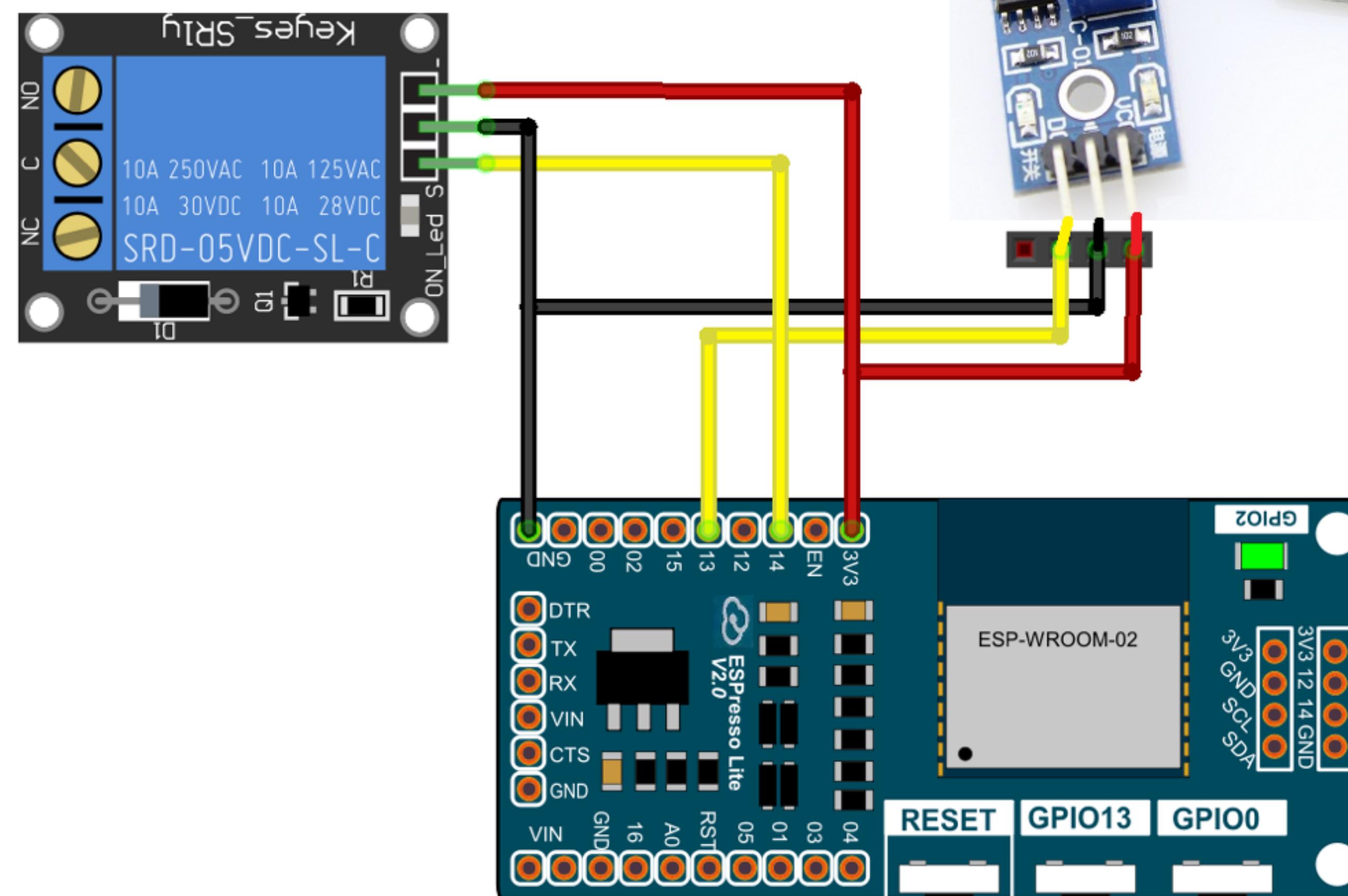
cmmc.io/day3

LAB 3



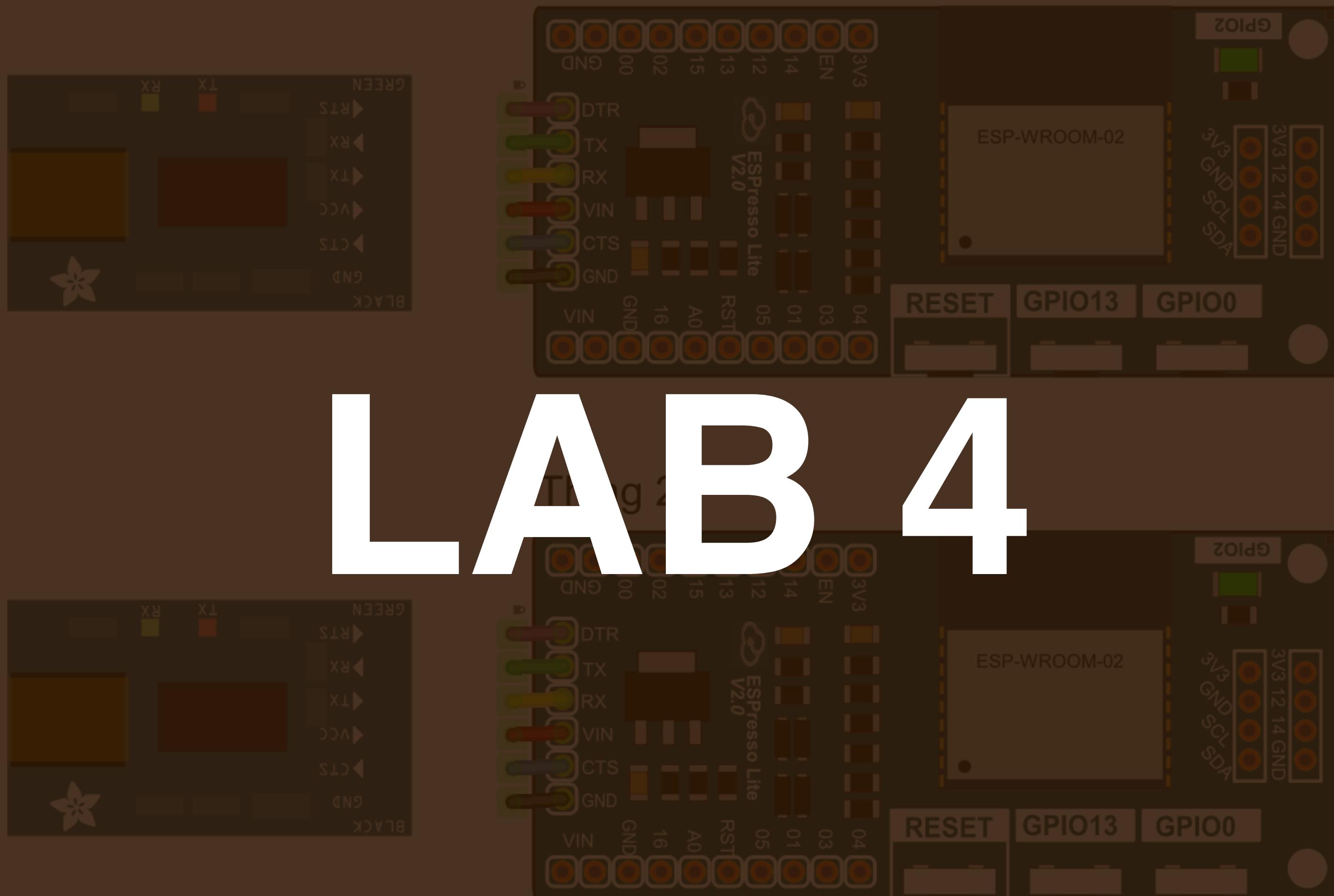
fritzing





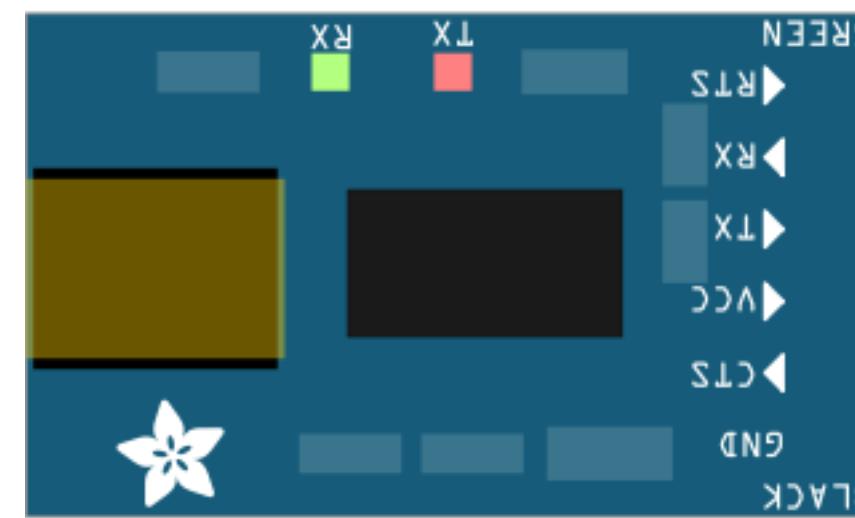
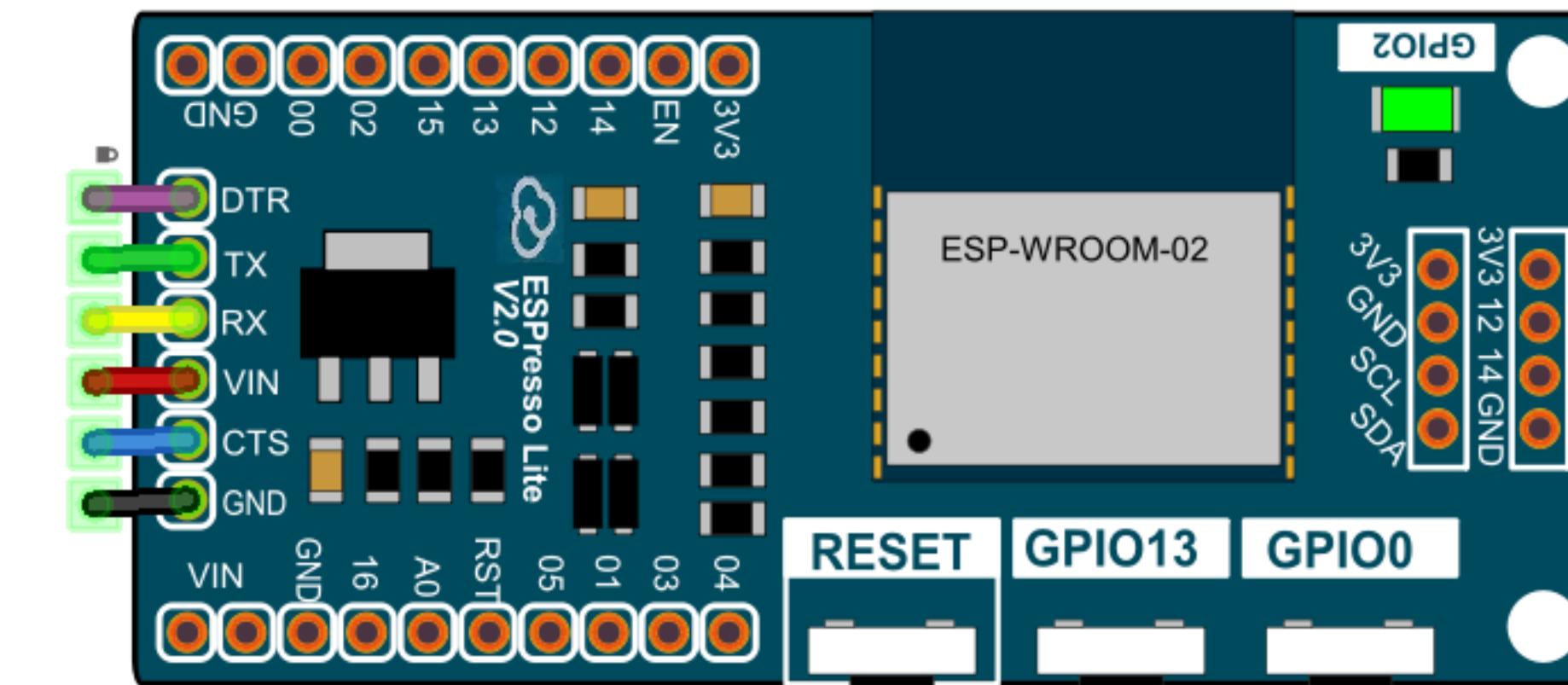
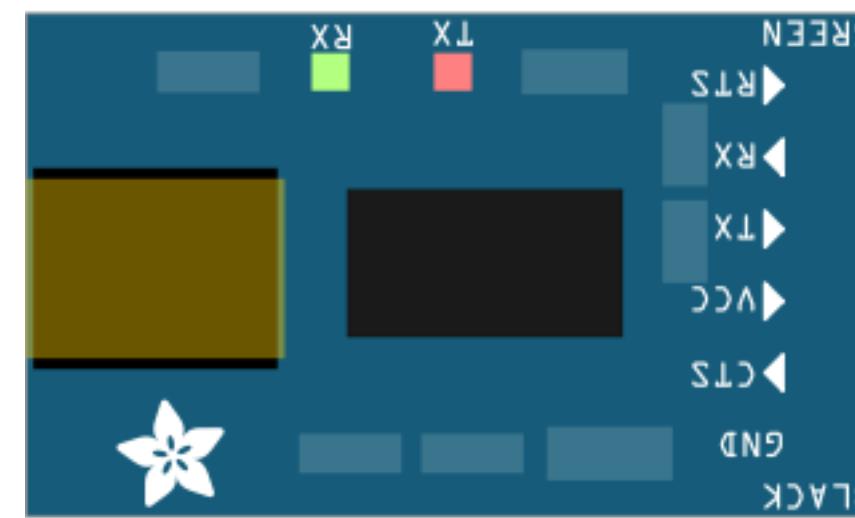
fritzing

CHIANG MAI
MAKERCLUB

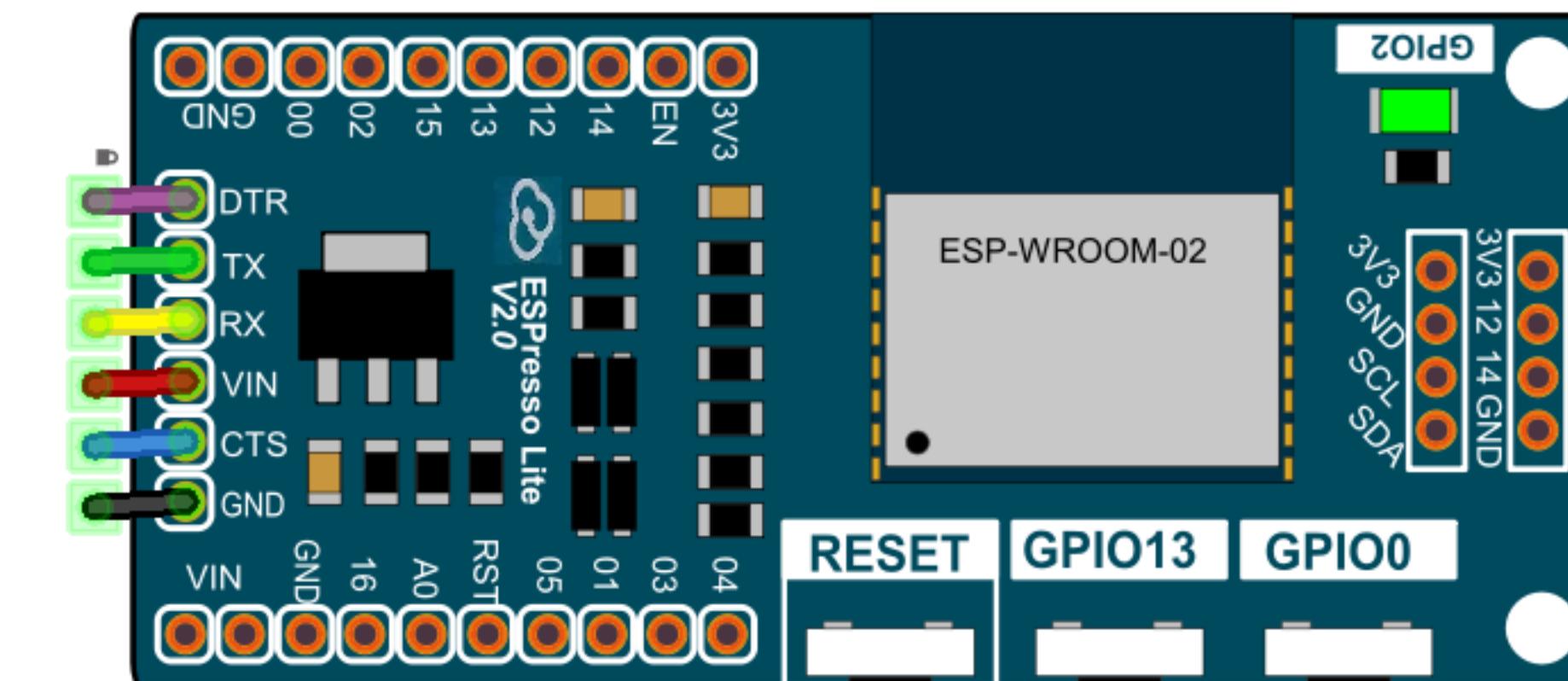


LAB 4

Thing 1



Thing 2



Chiang Mai Maker Club *fritzing*

Thing.chat

Lab4_thing1 | Arduino 1.6.9

```
10
11 #include <Arduino.h>
12 #include <ESP8266WiFi.h>
13 #include <MicroGear.h> // v 1.1.7
14 #include "CMMC_Blink.hpp"
15 CMMC_Blink blinker;
16
17 const char* ssid      = ""; // Change your ssid wifi
18 const char* password = ""; // Change your password wifi
19
20 // NETPIE.io : lab_device
21 #define APPID      ""
22 #define KEY        ""
23 #define SECRET     ""
24 #define ALIAS      "thing1"
25 #define CHAT_WITH  "thing2"
26
27 #define BUTTON 0
28
29 WiFiClient client;
30 MicroGear microgear(client);
31
32 void init_wifi();
33 void init_hardware() {
34   pinMode(BUTTON, INPUT);
35 }
```

Done Saving.

Sketch uses 300,245 bytes (28%) of program storage space. Maximum is 1,044,464 bytes.
Global variables use 41,836 bytes (51%) of dynamic memory, leaving 40,084 bytes for local variables.

Lab4_thing1 | Arduino 1.6.9

```
Lab4_thing1 § CMMC_Blink.hpp
40   serial.println("init done... ");
41 }
42
43 void loop() {
44   if (microgear.connected())
45   {
46     microgear.loop();
47
48     bool state = digitalRead(BUTTON);
49     Serial.println(state);
50     digitalWrite(BUTTON, !state);
51
52     if (state == 1) {
53       microgear.chat(CHAT_WITH, "ON");
54     }
55     else {
56       microgear.chat(CHAT_WITH, "OFF");
57     }
58     delay(200);
59   }
60
61   else
62   {
63     Serial.println("connection lost, reconnect...");
64     microgear.connect(APPID);
65   }
66
67 **** microgear Loop ****
Done Saving.
Using Library Flicker at version 1.0 in folder: /Users/Naresh/Library/Arduino15/packages/esp8266/hardware
Sketch uses 300,245 bytes (28%) of program storage space. Maximum is 1,044,464 bytes.
Global variables use 41,836 bytes (51%) of dynamic memory, leaving 40,084 bytes for local variables.
```

Lab4_thing1 | Arduino 1.6.9

```
Lab4_thing1 § CMMC_Blink.hpp
40     Serial.println("init done..."); 
41 }
42
43 void loop() {
44     if (microgear.connected())
45     {
46         microgear.loop();
47
48         bool state = digitalRead(BUTTON);
49         Serial.println(state);
50         digitalWrite(BUTTON, !state);
51         if (state == 1) {
52             microgear.chat(CHAT_WITH, "ON");
53         }
54         else {
55             microgear.chat(CHAT_WITH, "OFF");
56         }
57         delay(200);
58     }
59     else
60     {
61         Serial.println("connection lost, reconnect...");
62         microgear.connect(APPID);
63     }
64 }
65
66 //***** microgear loop *****
```

Done Saving.

Sketch uses 300,245 bytes (28%) of program storage space. Maximum is 1,044,464 bytes.
Global variables use 41,836 bytes (51%) of dynamic memory, leaving 40,084 bytes for local variables.

React Native

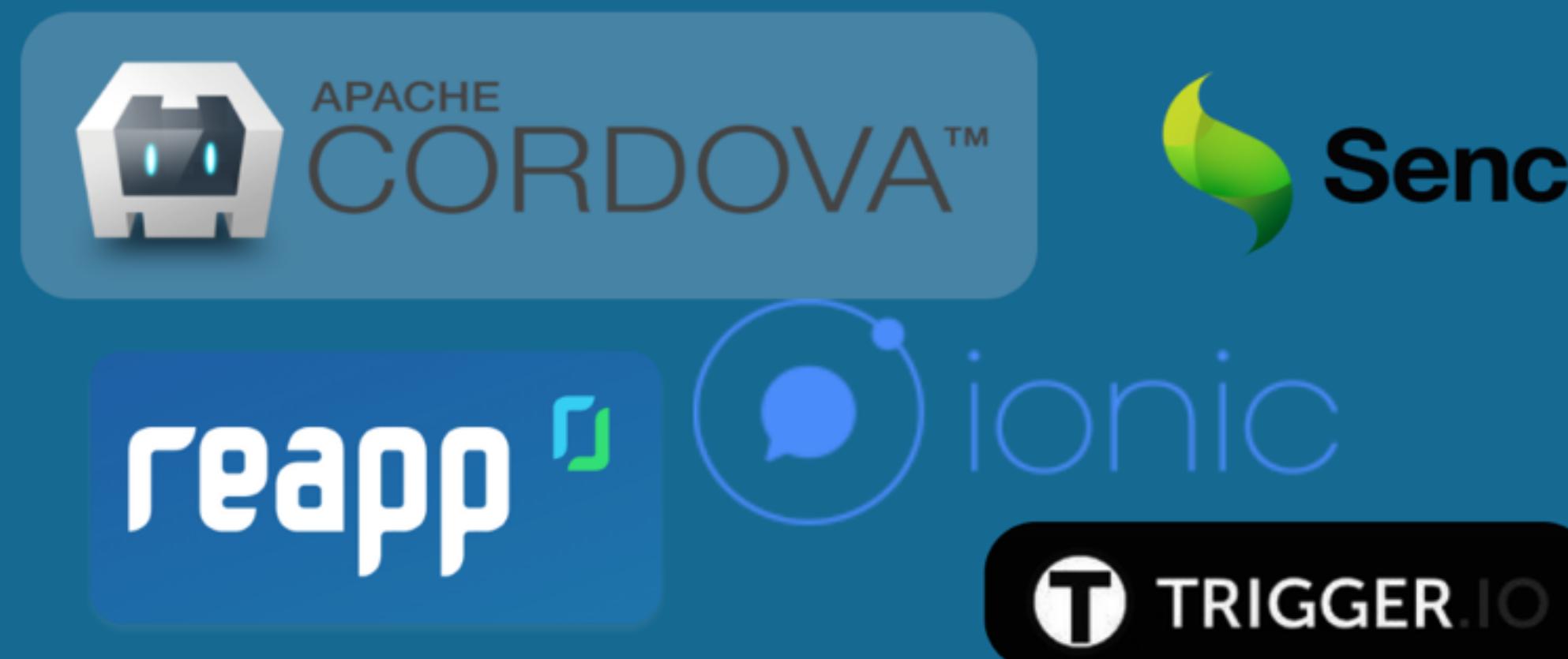
Chiang Mai Maker Club

Mobile Development Options

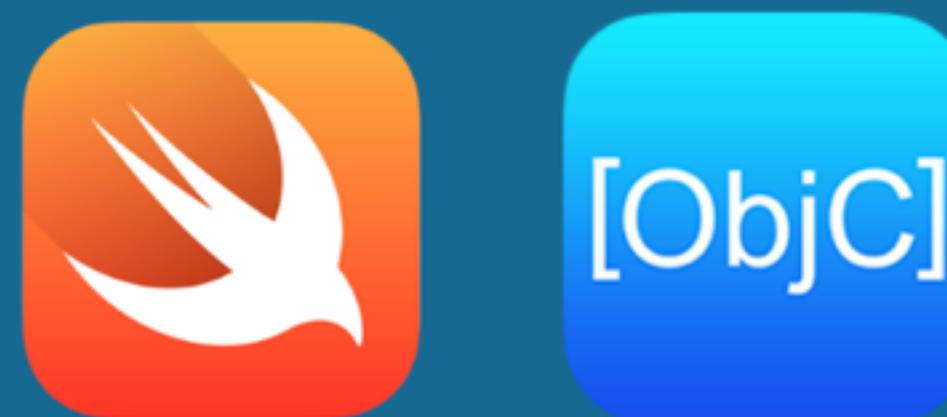
- Native
- Hybrid
- HTML5

Tools to build mobile apps

Webviews & basic JS bridge (& hybrid)



Native



Compile to Native binary



Native components via JS bridge



techcrunch.com

My Sites Reader

TC News Startups Mobile Gadgets Enterprise Trending ↗ Pokemon Tesla Amazon

Mark Zuckerberg: Our Biggest Mistake Was Betting Too Much On HTML5

Posted Sep 11, 2012 by Drew Olanoff (@drew)



[Next Story](#)



Today, Mark Zuckerberg revealed that Facebook's mobile strategy relied too much on HTML5, rather than native applications.

Not only was this a big mistake with mobile, but Zuckerberg says that its biggest mistake period was the focus on HTML5. This is the first time that the Facebook CEO has openly admitted this, but things are looking good for the new iOS native app. According to Zuckerberg, people are consuming twice as many feed stories since the update to the new iOS app, which is great.

The first half year has been a little bit slow on product, but for the next six months I expect a lot of really cool stuff.

This "really cool stuff" will probably have monetization in mind, as it's very clear that mobile path to ad revenue for the company.

Follow ...

GitHub, Inc.

This repository Search

Pull requests Issues Gist

Watch 2,462 Star 34,931 Fork 7,445

Code Issues 921 Pull requests 206 Wiki Pulse Graphs

Branch: master

Commits on Feb 13, 2015

- Update react-docgen**
fkling committed on Feb 13, 2015
- Adding react-docgen for documentation generation**
fkling committed on Feb 13, 2015
- Vj setup twitter feed and google analytics**
vjeux committed on Feb 13, 2015

Commits on Feb 12, 2015

- updated copyright year**
galenandrew committed on Feb 12, 2015
- Vj Initial version of the website and docs.**
vjeux committed on Feb 12, 2015
- Vj [ReactNative] Add warning when using onScroll without throttleScrollIC...**
vjeux committed on Feb 12, 2015

CHIANG MAI MAKERCLUB

The screenshot shows a Mac OS X desktop environment with a browser window open to the React Native documentation at facebook.github.io/react-native/docs/getting-started.html. The browser's address bar also displays the URL. The React Native logo is visible in the top left of the page header.

THE BASICS

- Getting Started
- Tutorial
- Props
- State
- Style
- Height and Width
- Layout with Flexbox
- Handling Text Input
- Using a ScrollView
- Using a ListView
- Networking
- More Resources

GUIDES

- Integration With Existing Apps
- Colors
- Images
- Handling Touches
- Animations
- Accessibility
- Timers
- Direct Manipulation
- Debugging
- Testing
- JavaScript Environment
- Navigation
- Performance
- Upgrading
- Platform Specific Code
- Gesture Responder System

GUIDES (IOS)

- Native Modules
- Native UI Components
- Linking Libraries
- Running On Device
- Running On Simulator
- Communication between native and React Native

GUIDES (ANDROID)

- Native Modules
- Native UI Components
- Running On Device

Getting Started

Welcome to React Native! This page will help you install React Native on your system, so that you can build apps with it right away. If you already have React Native installed, you can skip ahead to the [Tutorial](#).

The instructions are a bit different depending on your development operating system, and whether you want to start developing for iOS or Android. If you want to develop for both iOS and Android, that's fine - you just have to pick one to start with, since the setup is a bit different.

Mobile OS: [iOS](#) [Android](#) Development OS: [Mac](#) [Linux](#) [Windows](#)

Dependencies for Windows + Android

You will need node.js, the React Native command line tools, Watchman, and Android Studio.

We recommend installing node.js and Python2 via [Chocolatey](#), a popular package manager for Windows. Open a Command Prompt as Administrator, then run:

```
choco install nodejs.install  
choco install python2
```

Node comes with npm, which lets you install the React Native command line interface.

```
npm install -g react-native-cli
```

Download and install [Android Studio](#).

If you plan to make changes in Java code, we recommend [Gradle Daemon](#) which speeds up the build.

Testing your React Native Installation

Use the React Native command line tools to generate a new React Native project called "AwesomeProject", then run `react-native run-android` inside the newly created folder.

```
react-native init AwesomeProject  
cd AwesomeProject  
react-native run-android
```

If everything is set up correctly, you should see your new app running in your Android emulator shortly.

```
1 import React, { Component, } from 'react';
2 import {
3   AppRegistry,
4   StyleSheet,
5   Text,
6   View,
7 } from 'react-native';
8
9 class App extends Component {
10   render() {
11     return (
12       <View style={styles.container}>
13         <Text style={styles.welcome}>
14           Welcome to React Native!
15         </Text>
16       </View>
17     );
18   }
19 }
20
21 const styles = StyleSheet.create({
22   container: {
23     flex: 1,
24     justifyContent: 'center',
25     alignItems: 'center',
26     backgroundColor: '#F5FCFF',
27   },
28   welcome: {
29     fontSize: 20,
30     textAlign: 'center',
31     margin: 10,
32   },
33 });
34
35 AppRegistry.registerComponent('App', () => App);
```

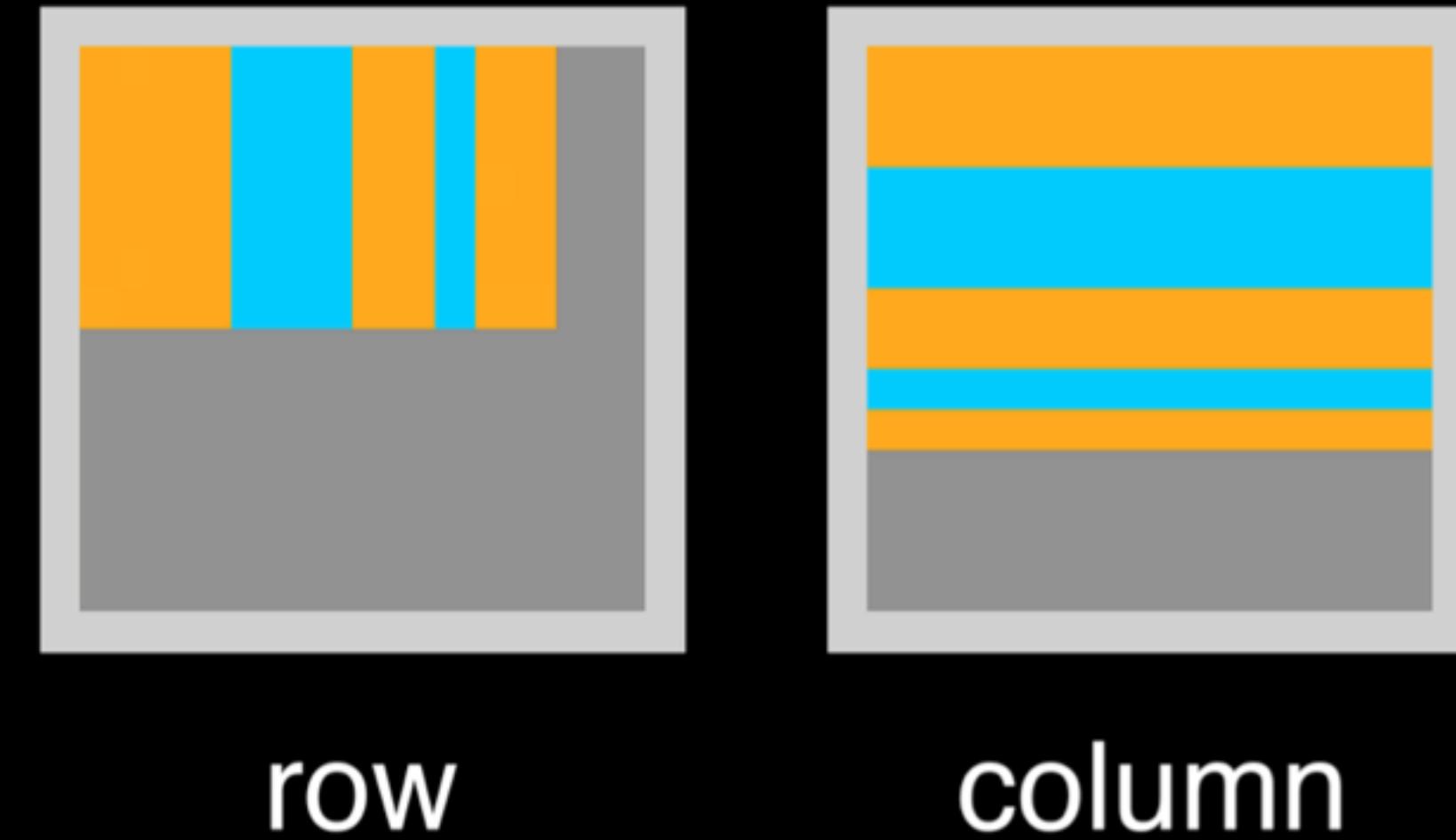


<http://dabbott.github.io/react-native-web-player/>

FLEX DIRECTION

`flexDirection: 'row' | 'column' (default)`

This defines the main axis along which subviews are stacked.



JUSTIFY CONTENT

```
justifyContent: 'flex-start' (default) | 'center' | 'flex-end' |  
'space-between' | 'space-around'
```

This defines how views align along the main axis.



flex-start



center



flex-end



space-between

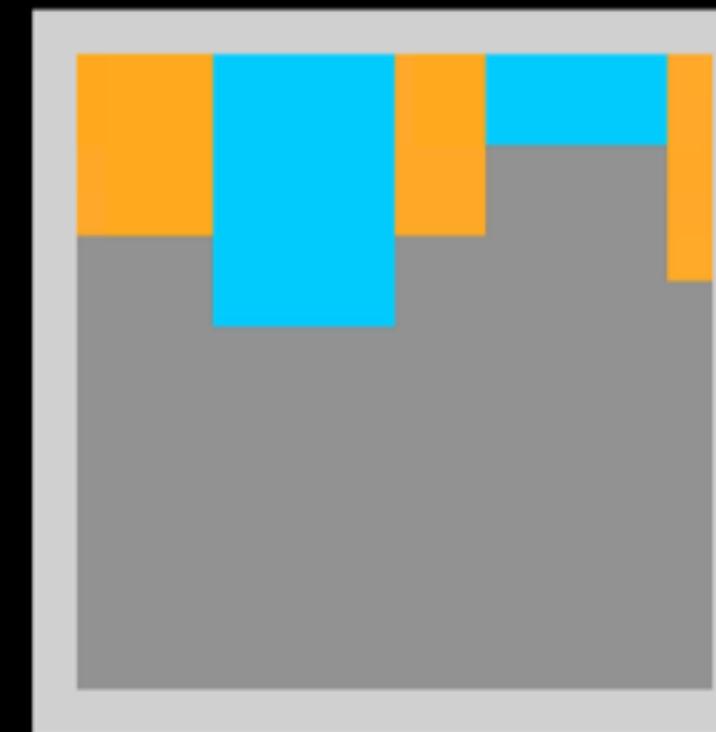


space-around

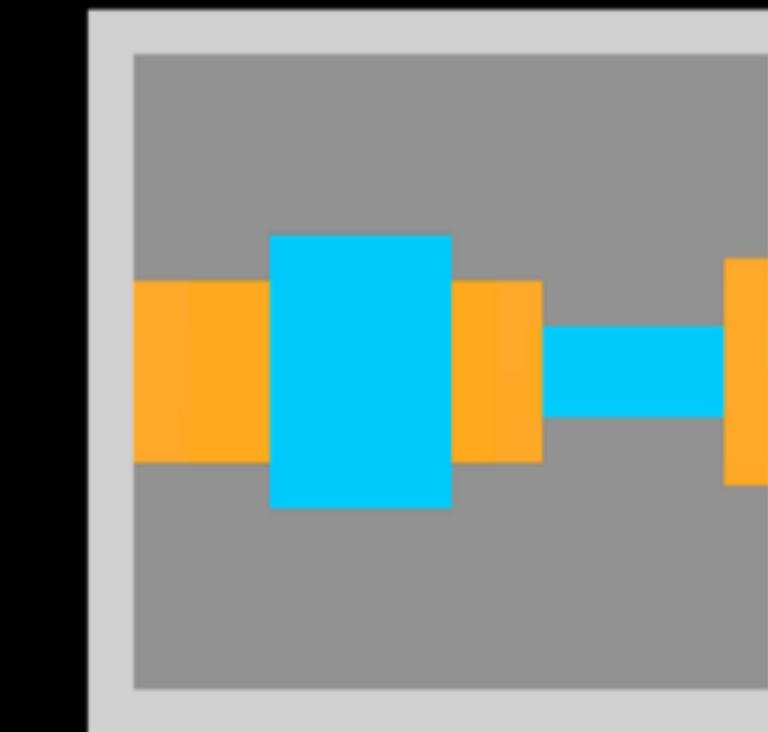
ALIGN ITEMS

alignItems: 'flex-start' (default) | 'center' | 'flex-end'

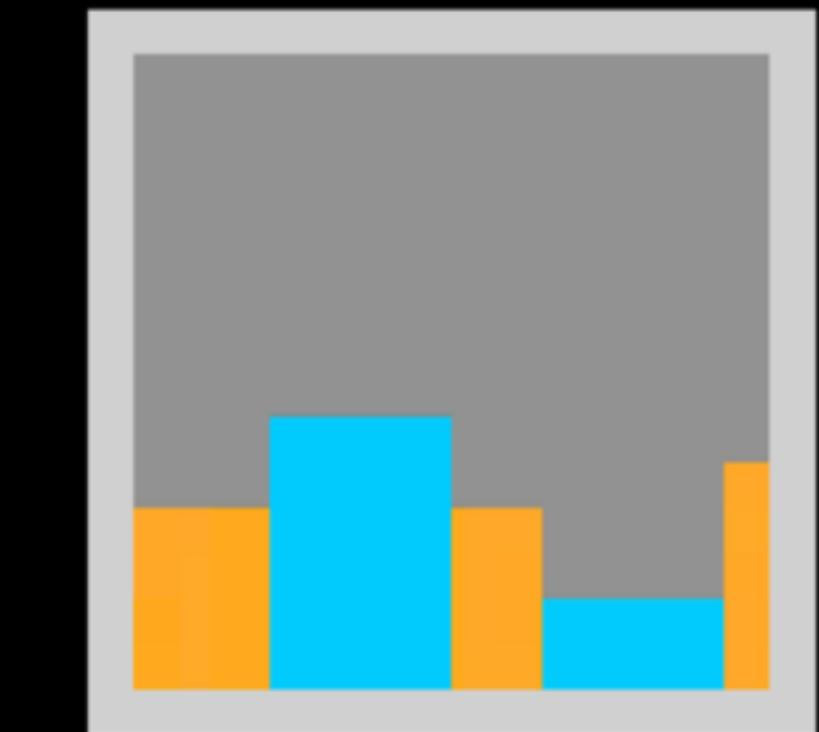
This defines how views align along the cross axis.



flex-start



center



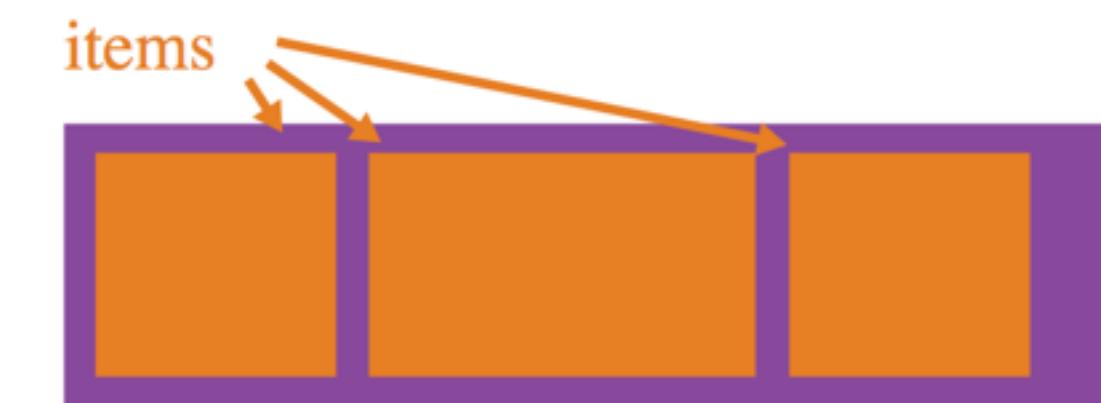
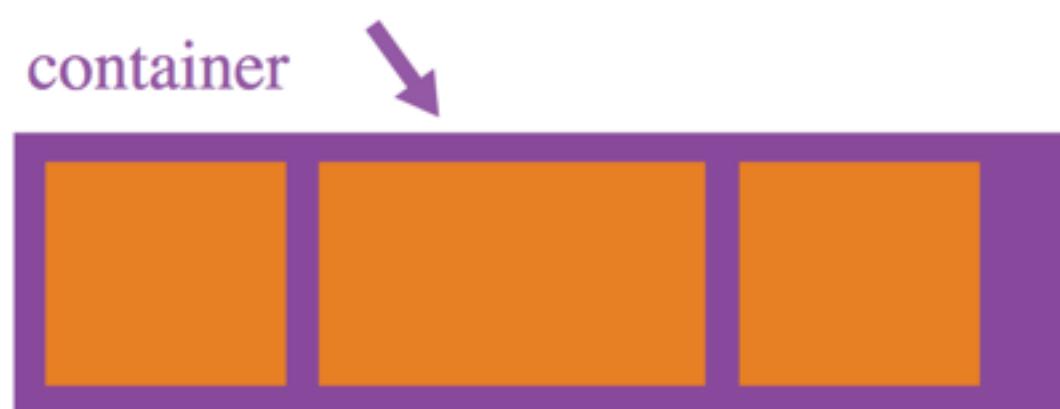
flex-end

A Complete Guide to Flexbox

BY [CHRIS COYIER](#) LAST UPDATED ON MAY 18, 2016

► Background

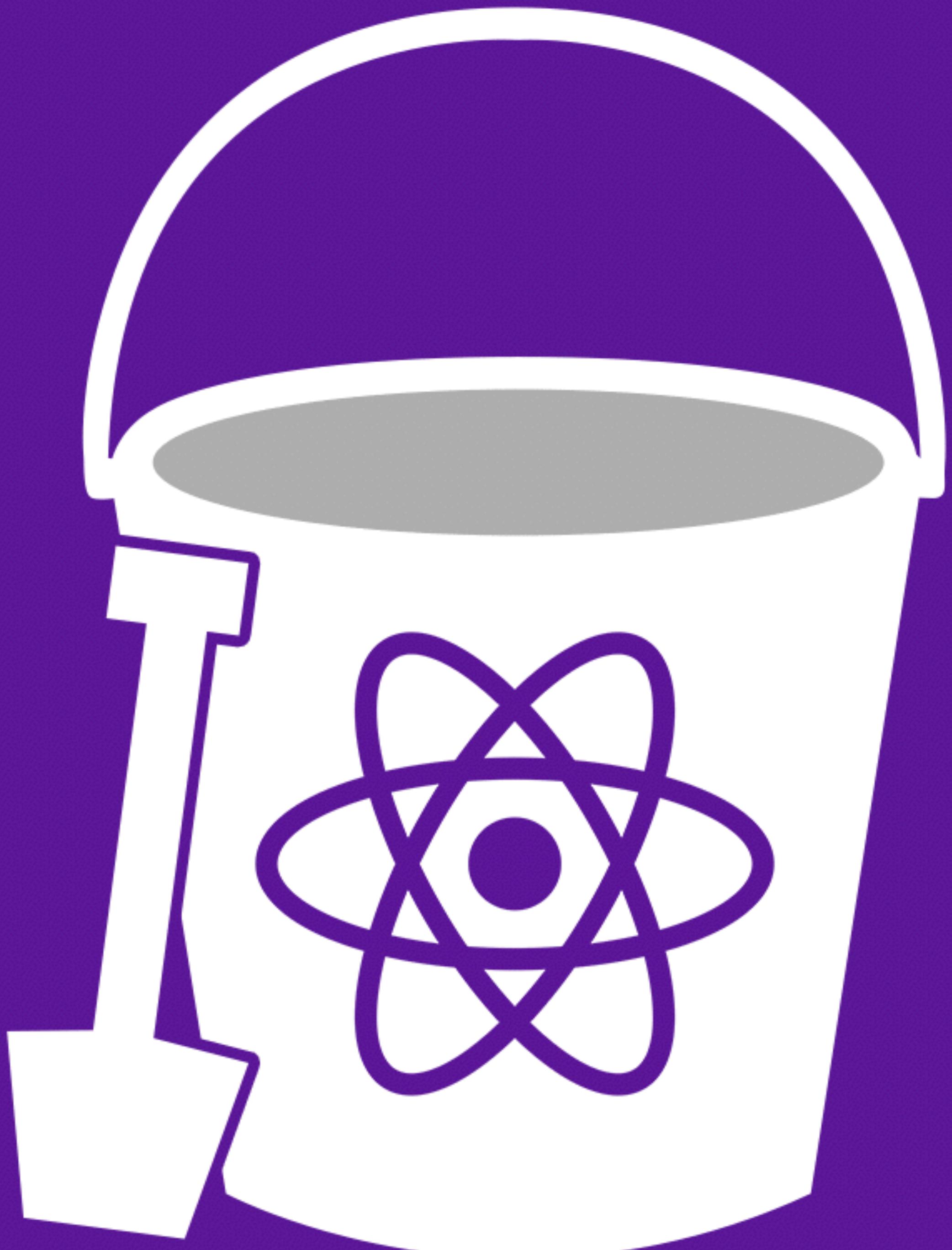
► Basics & Terminology



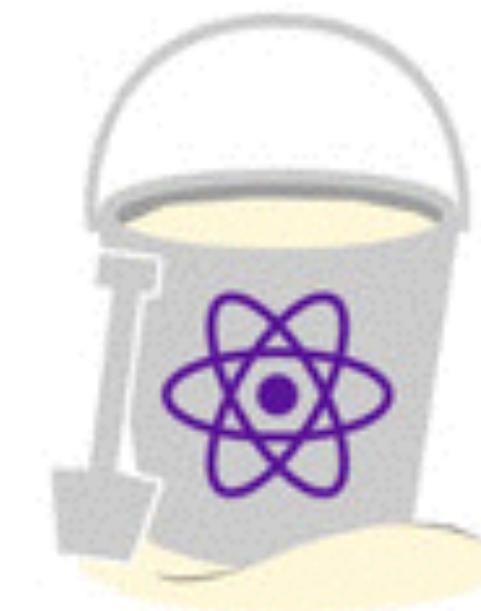
**Properties for the
Parent**

**Properties for the
Children**

<http://rnplay.org>



React Native Playground



Run React Native apps from
rnplay.org directly on your device.

Get started by trying apps on the *Explore* tab, or
login to your rnplay.org account in *My Apps*.

How do I exit a loaded app?

Tap and hold the screen with two fingers
simultaneously for 1.5 seconds.

What does target mean?

The target React Native version for an app. An
app might not behave if the target is different than
the our bundled React Native version.



Explore



My Apps



Direct URL



Scan Code



About



CHIANG MAI
MAKERCLUB

rnplay.org

Commits · facebook/react-native Layout with Flexbox – React Native | A framework for building... WorkShopNETPIE-ReactNativeSwitch by compilerexe

Files index.ios.js

index.android.js

index.ios.js

```
1 import React, { Component } from 'react';
2 import {
3   AppRegistry,
4   StyleSheet,
5   Text,
6   TextInput,
7   Switch,
8   View
9 } from 'react-native';
10
11 var WorkShopNETPIE = {
12   getInitialState() {
13     return {
14       topic: 'React Native'
15     }
16   },
17   onChangeTopic(event) {
18     this.setState({topic: event.nativeEvent.text});
19     console.log(this.state.topic);
20   }
21 };
22
23 render() {
24   var onChangeTopic = function(event) {
25     this.setState({topic: event.nativeEvent.text});
26     console.log(this.state.topic);
27   }.bind(this);
28 }
```

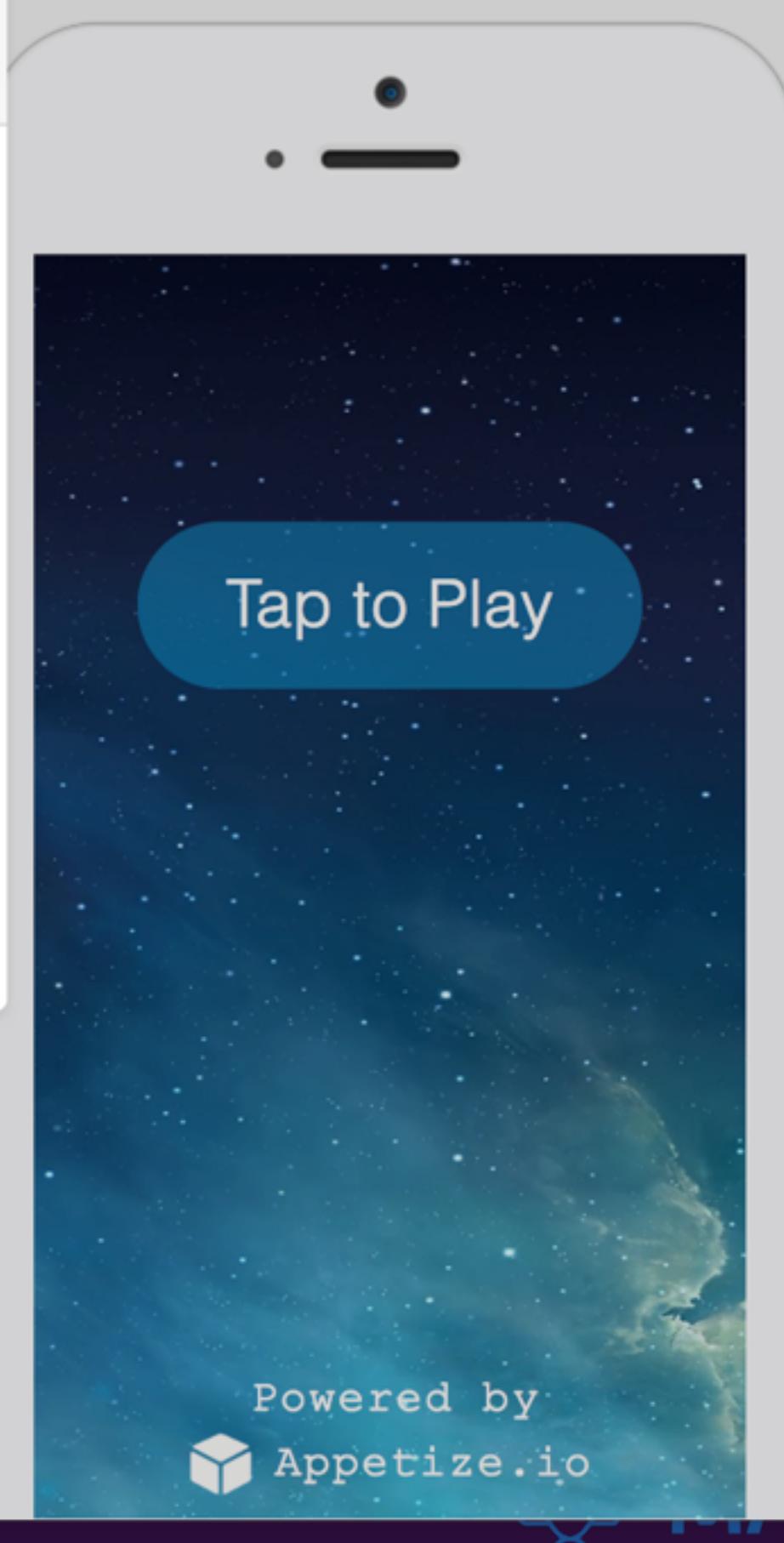
Scan this with our iOS app! [Download on the App Store](#)



Run on your device

VERSION 0.24.1

iOS Android



Powered by Appetize.io

Console Logs

REACT NATIVE!

<http://dabbott.github.io/react-native-web-player/>

<http://bit.ly/react-button>

<http://bit.ly/react-layout>

The screenshot shows a web browser window with the URL gist.github.com in the address bar. The main content is a GitHub Gist titled "NAzT / rn-layout-basic-step1.js". The code in the gist is as follows:

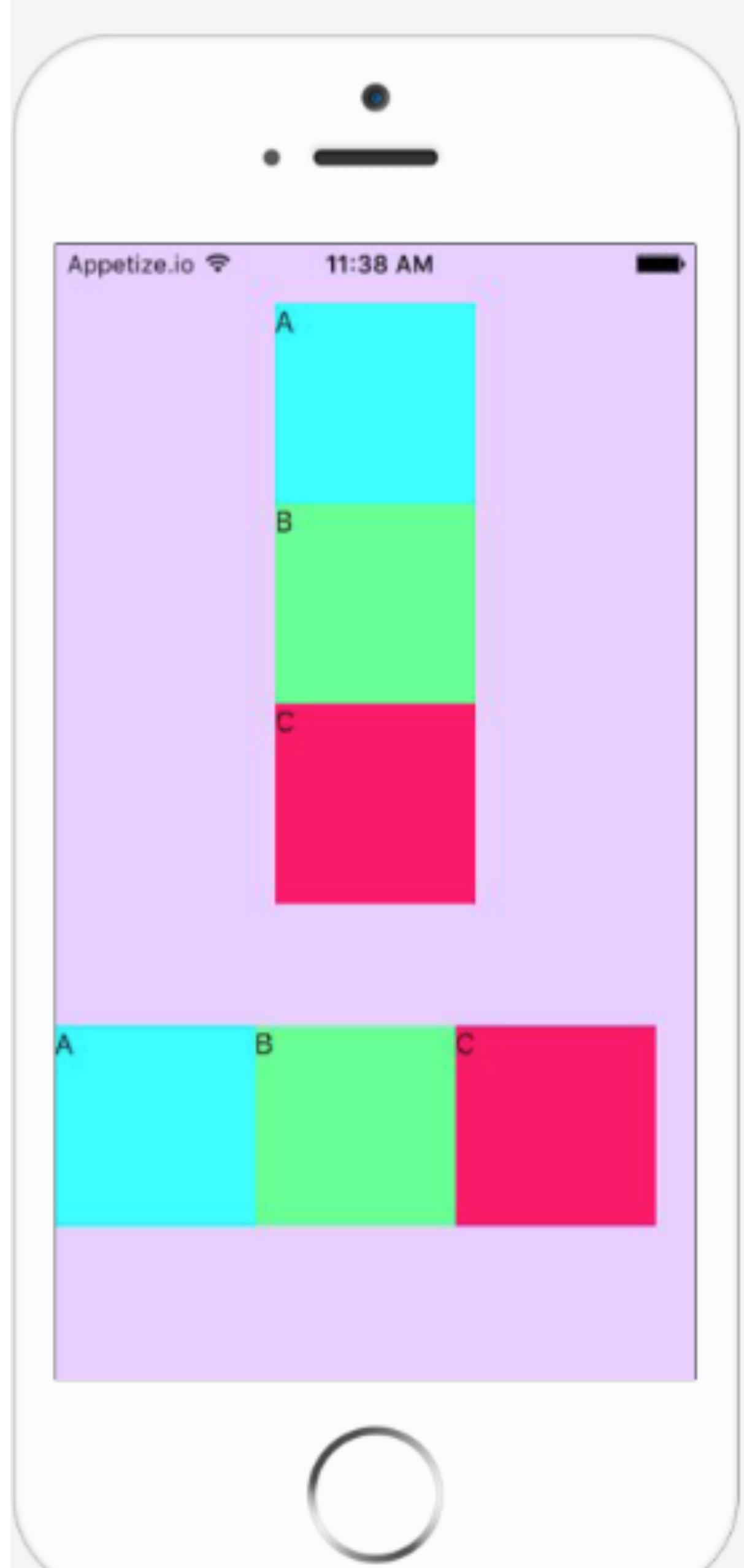
```
// http://dabbott.github.io/react-native-web-player/
import React, { Component } from 'react';
import { AppRegistry, View, Text } from 'react-native';

class FlexDirectionBasics extends Component {
  render() {
    <View style={{flex: 1}}>
      <View style={{flex: 1, flexDirection: 'row'}>
        <View style={{width: 50, height: 50, backgroundColor: 'powderblue'}} />
        <View style={{width: 50, height: 50, backgroundColor: 'skyblue'}} />
        <View style={{width: 50, height: 50, backgroundColor: 'steelblue'}} />
      </View>
      <View style={{flex: 1, flexDirection: 'row'}>
        <Text>Hello</Text>
      </View>
    </View>
  }
}

AppRegistry.registerComponent('FlexDirectionBasics', () => FlexDirectionBasics);
```

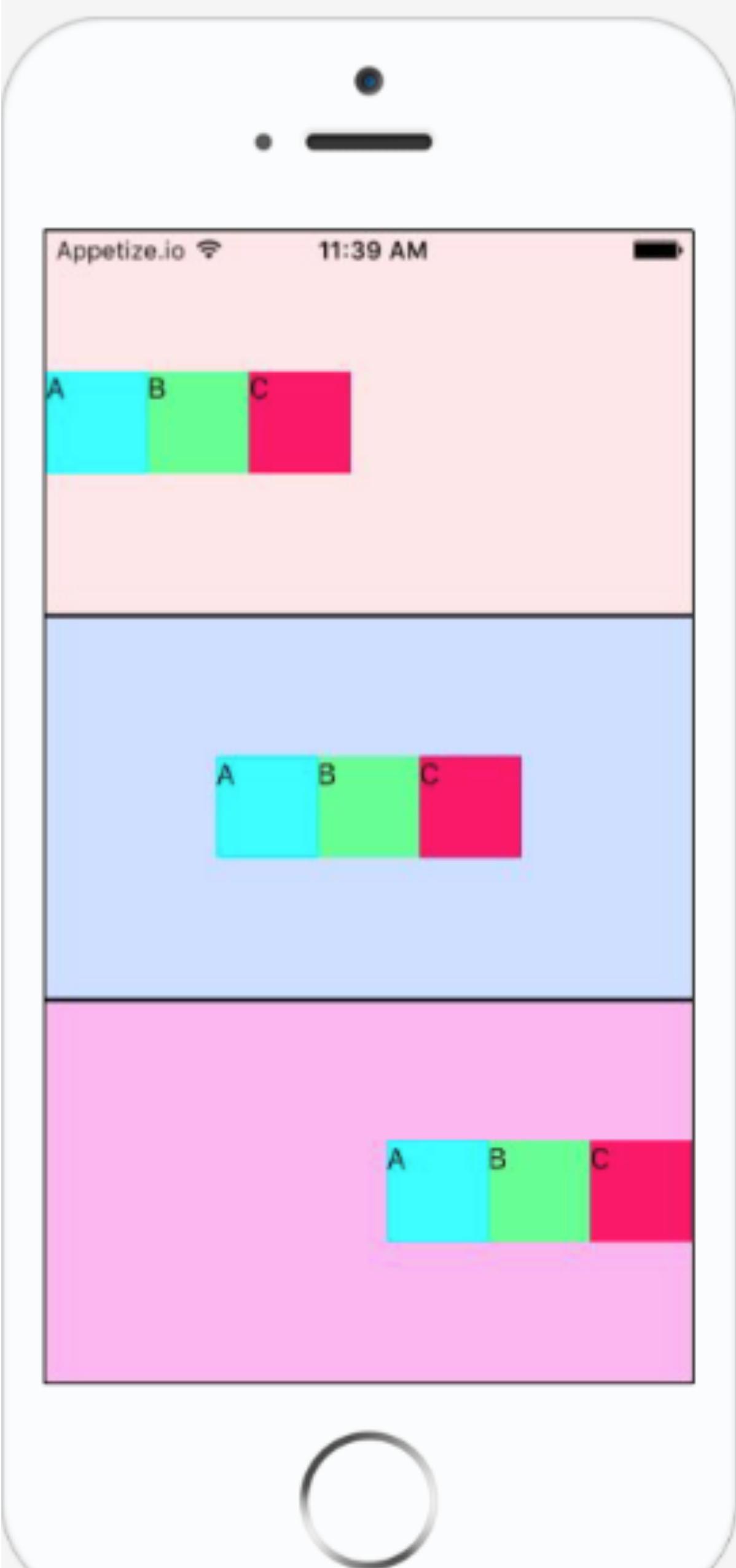
<https://facebook.github.io/react-native/docs/flexbox.html>

Examples



<https://rnplay.org/apps/nkersg>

React NativeFlexDirection-Tutorial



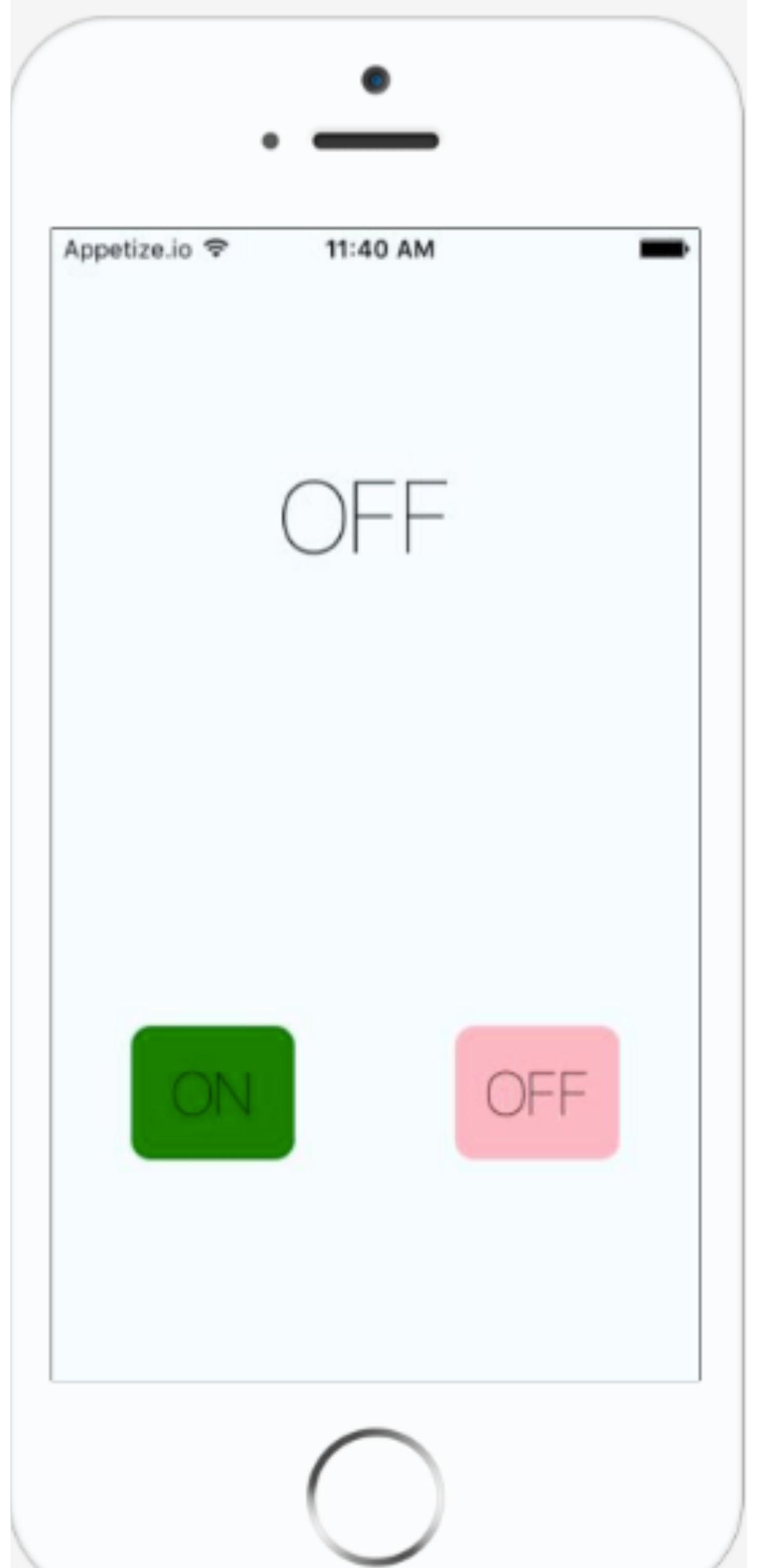
<https://rnplay.org/apps/AMMbrg>

React Native-JustifyContent



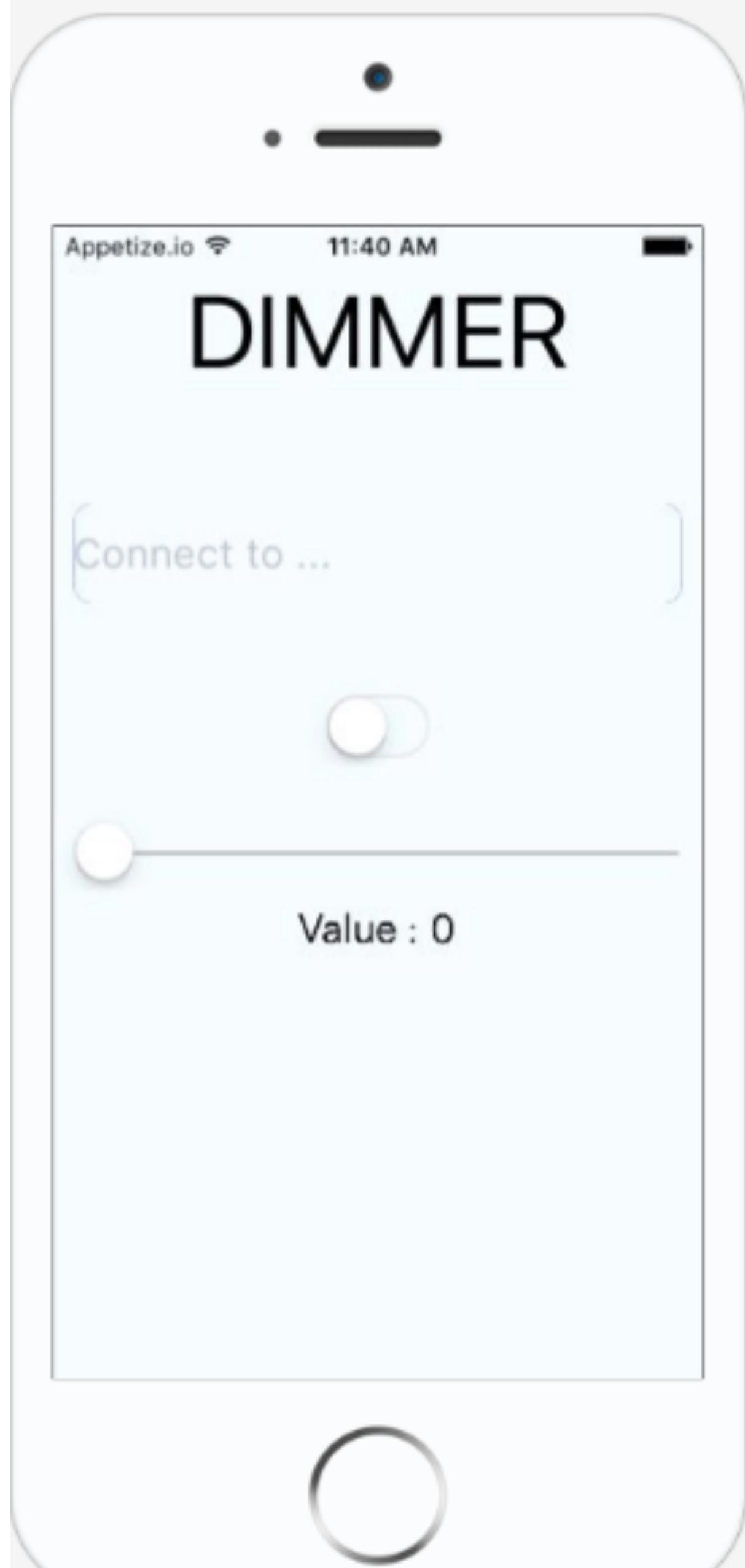
<https://rnplay.org/apps/iniN5g>

React Native-Slider



<https://rnplay.org/apps/HHz7IA>

React Native-OnOffButton



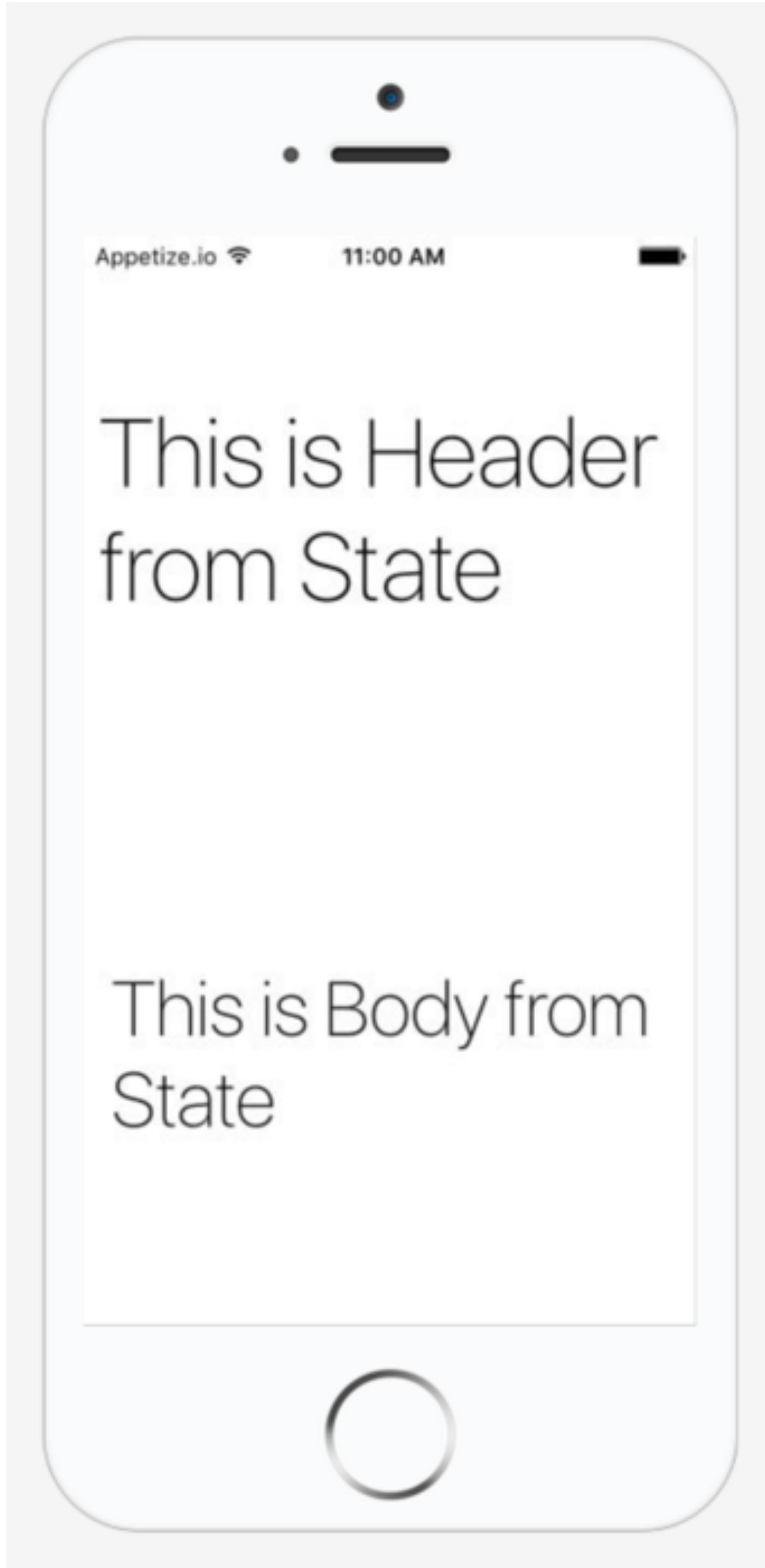
<https://rnplay.org/apps/UI2eHw>

React Native-Dimmer



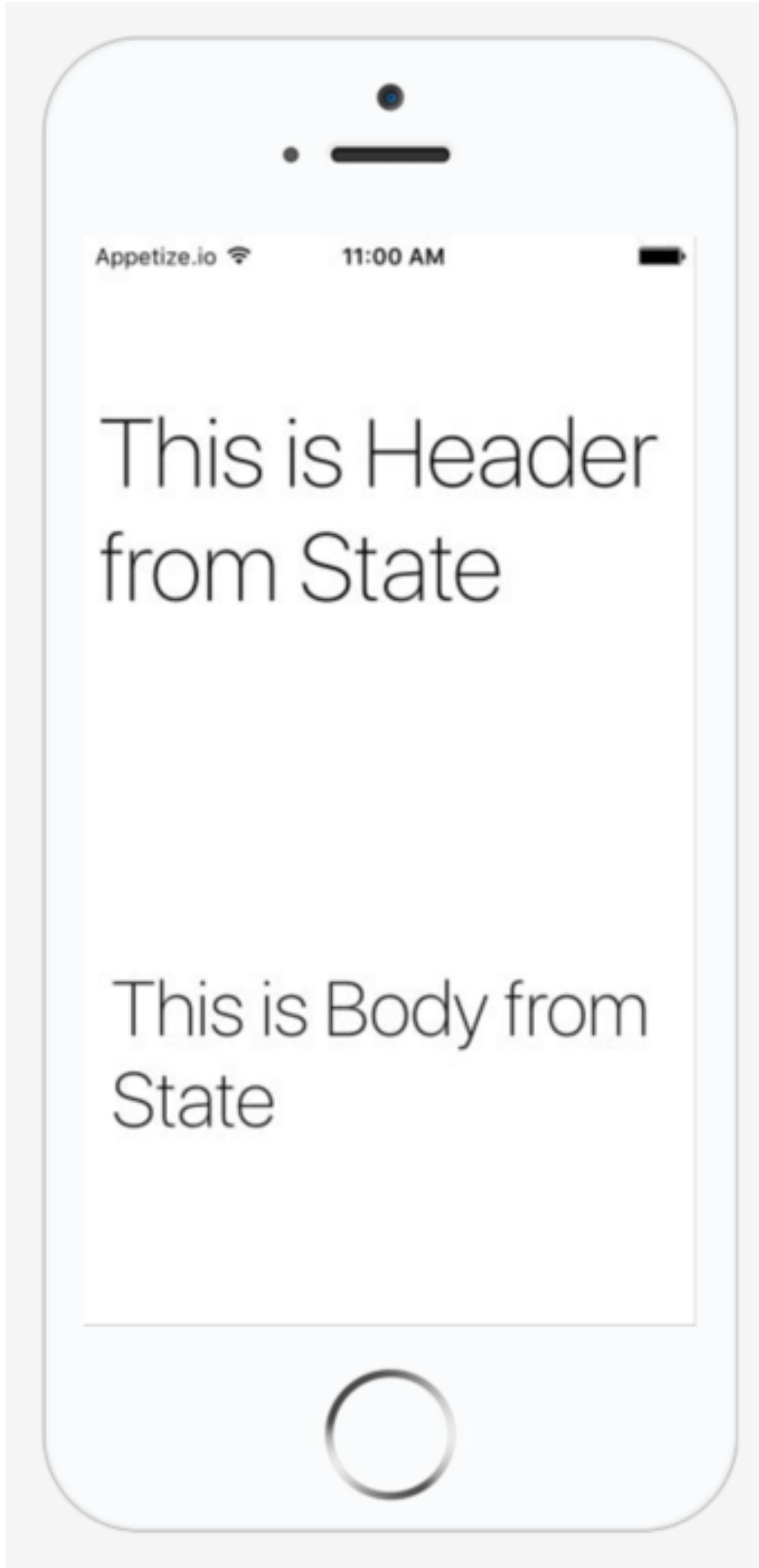
<https://rnplay.org/apps/XdYffw>

React Native-RealTimeSensor



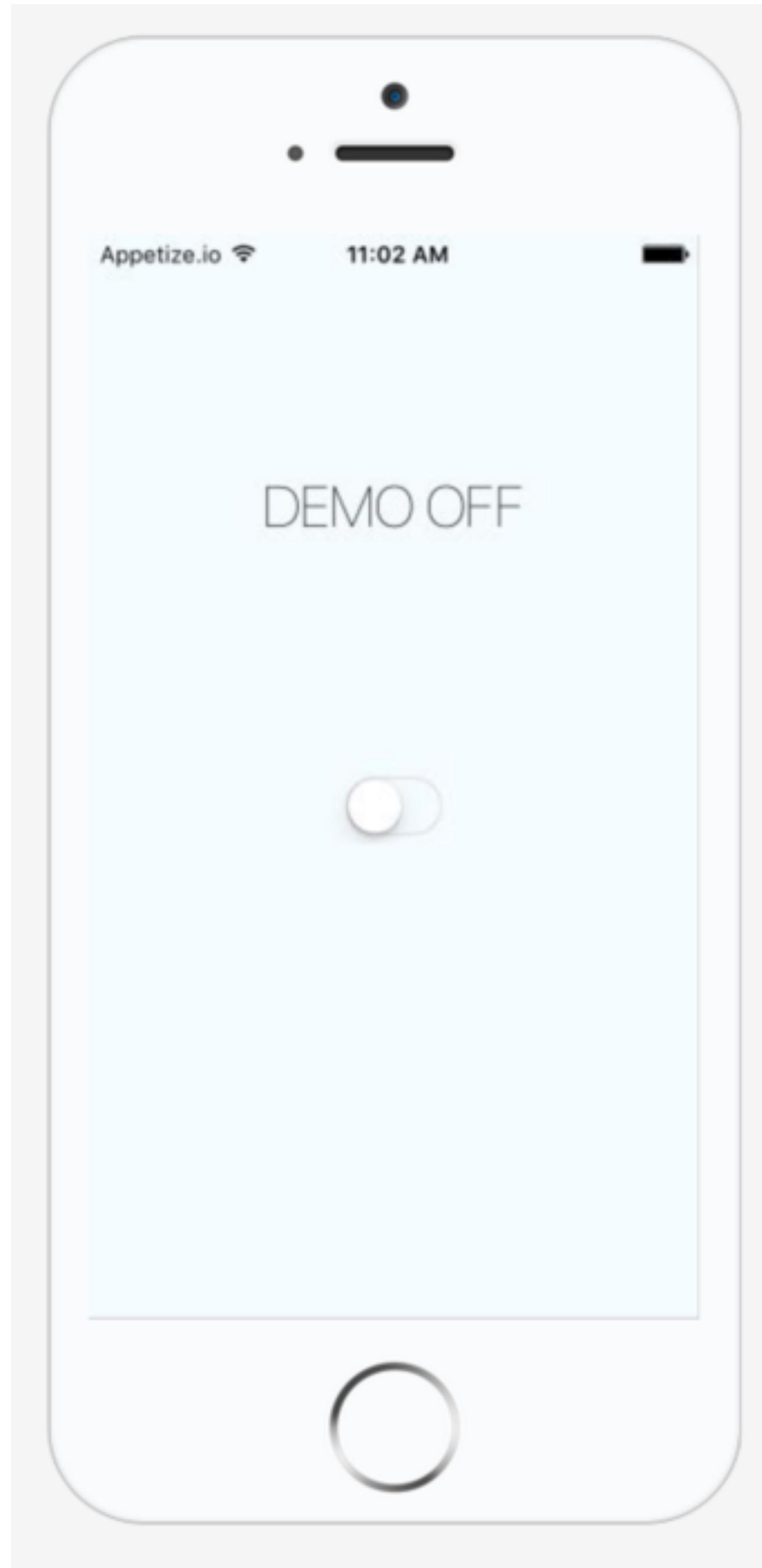
<https://rnplay.org/apps/a-63lg>

React Native-Tutorial Layout



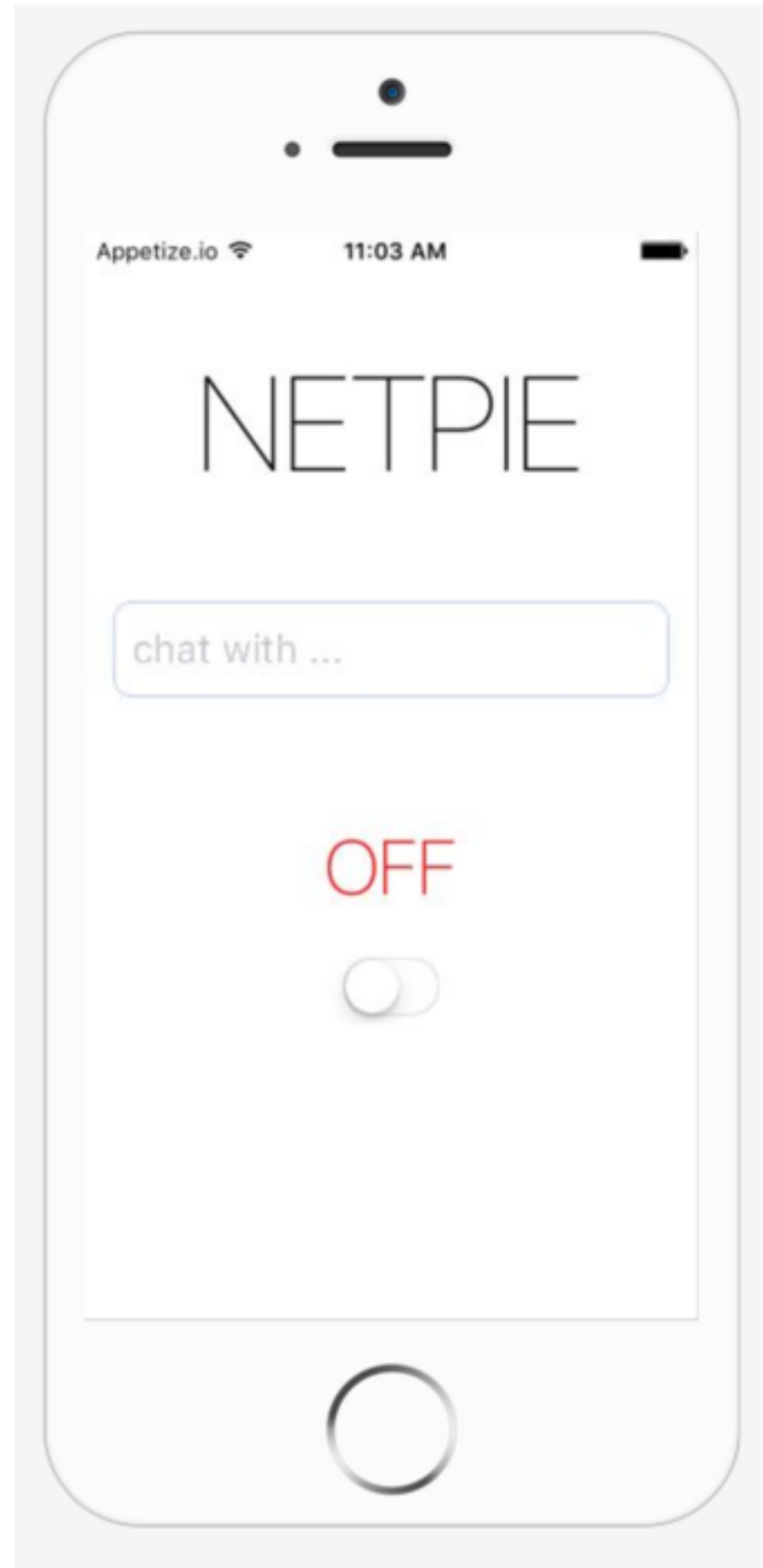
<https://rnplay.org/apps/CVt-ZQ>

React Native - Tutorial State



<https://rnplay.org/apps/q0FriQ>

React Native - Tutorial Switch



<https://rnplay.org/apps/tqyBVQ>

WorkShopNETPIE-ReactNativeSwitch



<https://rnplay.org/apps/0VjZWw>

React Native - Tutorial Message & State