

React Native Check TextInput is Empty or Not

Contents

- 1 Introduction
- 2 Check TextInput Using
- 3 To Make a React Native App
- 4 Code
 - 4.1 App.js
- 5 To Run the React Native App
- 6 Output Screenshots
- 7 Output in Online Emulator

Introduction

This is an example of React Native Check TextInput is Empty or Not. As the topic name describes itself, we will check the TextInput on a click of the Button. This is very useful if you are making any form that has some mandatory fields.

Check TextInput Using

```
1 if (!textInputName.trim()) {  
2   alert('Please Enter Name');  
3   return;  
4 }
```

In this example, we will check whether both inputs filled or not. So let's get started.

To Make a React Native App



Getting started with React Native will help you to know more about the way you can make a React Native project. We are going to use react-native init to make our React Native App. Assuming that you have node installed, you can use npm to install the **react-native-cli** command line utility. Open the terminal and go to the workspace and run

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

Run the following commands to create a new React Native project

```
react-native init ProjectName
```

If you want to start a new project with a specific React Native version, you can use the **-version** argument:

```
react-native init ProjectName --version X.XX.X
```

```
react-native init ProjectName --version react-native@next
```

This will make a project structure with an index file named App.js in your project directory.

Code

Open App.js in any code editor and replace the code with the following code

App.js

```
1 //React Native Check TextInput is Empty or Not
2 //https://aboutreact.com/react-native-check-textinput-is-empty-
  or-not/
3
4 //import React in our code
5 import React, {useState} from 'react';
```

```

6
7 //import all the components we are going to use
8 import {
9   SafeAreaView,
10  StyleSheet,
11  TextInput,
12  View,
13  Button
14} from 'react-native';
15
16const App = () => {
17  const [textInputName, setTextInputName] = useState('');
18  const [textInputEmail, setTextInputEmail] = useState('');
19
20  const checkTextInput = () => {
21    //Check for the Name TextInput
22    if (!textInputName.trim()) {
23      alert('Please Enter Name');
24      return;
25    }
26    //Check for the Email TextInput
27    if (!textInputEmail.trim()) {
28      alert('Please Enter Email');
29      return;
30    }
31    //Checked Successfully
32    //Do whatever you want
33    alert('Success');
34  };
35
36  return (
37    <SafeAreaView style={{flex: 1}}>
38      <View style={styles.container}>
39        <TextInput
40          placeholder="Enter Name"
41          onChangeText={
42            (value) => setTextInputName(value)
43          }
44          underlineColorAndroid="transparent"
45          style={styles.textInputStyle}
46        />
47        <TextInput
48          placeholder="Enter Email"
49          onChangeText={
50            (value) => setTextInputEmail(value)
51          }
52          underlineColorAndroid="transparent"
53          style={styles.textInputStyle}
54        />
55        <View style={{marginTop: 15}}>
56          <Button
57            title="Submit"
58            onPress={checkTextInput}
59            color="#606070"
60          />

```

```
61     </View>
62   </View>
63 </SafeAreaView>
64 );
65 };
66
67 const styles = StyleSheet.create({
68   container: {
69     flex: 1,
70     padding: 35,
71   },
72   textInputStyle: {
73     width: '100%',
74     height: 40,
75     paddingHorizontal: 5,
76     borderWidth: 0.5,
77     marginTop: 15,
78   },
79 });
80
81 export default App;
```

To Run the React Native App

Open the terminal again and jump into your project using.

```
cd ProjectName
```

To run the project on an Android Virtual Device or on real debugging device

```
react-native run-android
```

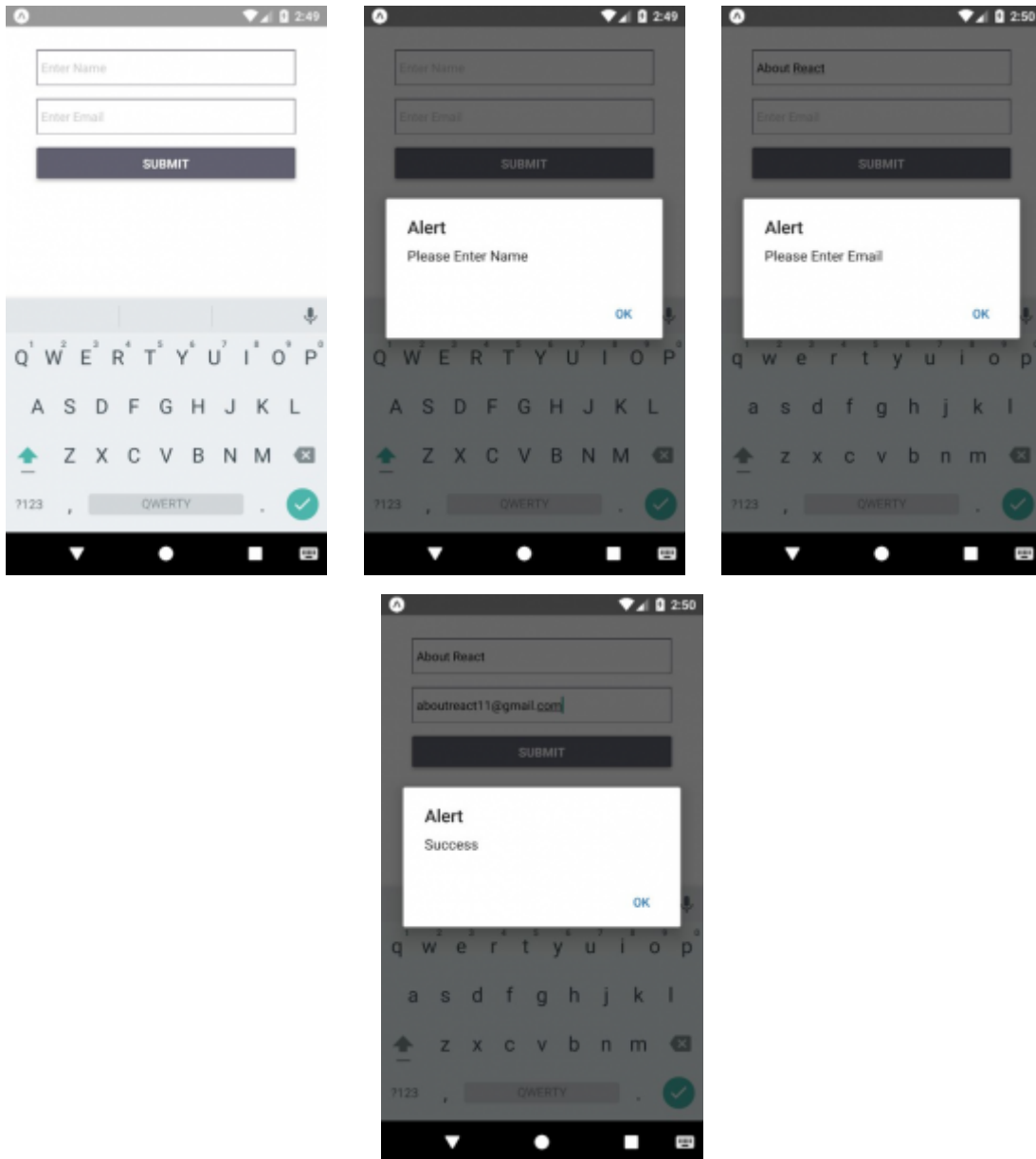
or on the iOS Simulator by running (macOS only)

```
react-native run-ios
```

[Download Source Code](#)



Output Screenshots



Output in Online Emulator



Check Empty TextInput Example

```
//React Native Check TextInput is Empty or Not
//https://aboutreact.com/react-native-check-textinput-is-empty-or-not/

//import React in our code
import React, {useState} from 'react';

//import all the components we are going to use
import {SafeAreaView, StyleSheet, TextInput, View, Button} from 'react-native';

const App = () => {
  const [textInputName, setTextInputName] =
    useState('');
  const [textInputEmail, setTextInputEmail] =
    useState('');

  const checkTextInput = () => {
    //Check for the Name TextInput
    if (!textInputName.trim()) {
      alert('Please Enter Name');
      return;
    }
    //Check for the Email TextInput
  }
}
```

Preview

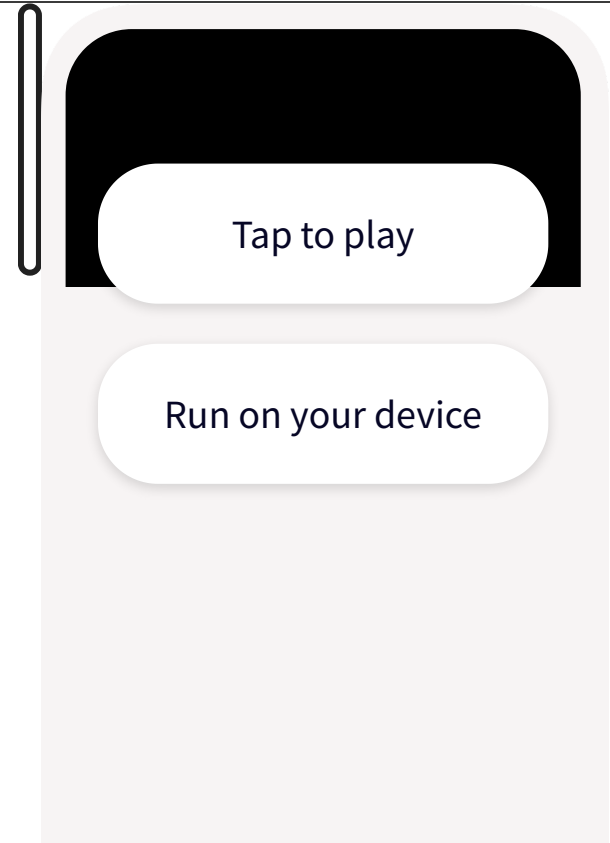


My Device

iOS

Android

Web



This is how you can check TextInput in empty or not. If you have any doubts or you want to share something about the topic you can comment below or [contact us here](#). There will be more posts coming soon. Stay tuned!

Hope you liked it. 😊

< [Image Icon in React Native TextInput](#)

> [React Native Enable Disable TextInput Programmatically](#)

1 thought on “React Native Check TextInput is Empty or Not”



Hridey



Why have you used trim ?

Reply

Leave a Comment

Name *

Email *

Website

Post Comment

This site uses Akismet to reduce spam. [Learn how your comment data is processed.](#)

Basics

Hello World Program

Online Emulator / Simulator

RN Component

Basic React Native Components

React native Comments

React Native View



React Native Text

React Native State

React Native Props

React Native Stylesheet

React Native SafeAreaView

React Native TextInput

React Native Image

React Native ScrollView

React Native Button

React Native Touchable

React Native Alert

React Native Modal

React Native Picker

React Native Switch

React Native Slider

React Native WebView

React Native FlatList

Request Focus and Keyboard Avoiding View

React Native StatusBar

React Native ActivityIndicator

Overlay ActivityIndicator

Lifecycle of React Native

Debug App in Real Device

Disable Yellow Warning Box

Run App On Real Device (USB)

Run App On Real Device (Wi-Fi)

generate Android Debug APK

Change Default iOS Simulator

View Props

onPress on a View

Use of StyleSheet

Create Custom Shapes

Align View at Bottom



Load Different Dynamic StyleSheet

Hide Show Component

Text Props

Strikethrough Text

Underline Text

SuperScript and SubScript

Break Line/Multi-Line Text

TextInput Props

TextInput value is a Number or not

AutoComplete Input

Convert Upper/Lower Case

Only Numeric Value From TextInput

Remove TextInput Underline

TextInput with Image Icon

TextInput is Empty or Not

Enable/Disable TextInput

Buttons Props

onLongPress on TouchableOpacity

Button with Image Icon

Image Props

Image Transparency/Opacity

Pinch to Zoom Image

Image Border Radius

Circular Image (Border Radius)

ListView Props

Swipe Down to Refresh List

FlatList Pagination using Load More

Infinite Loading Listview (Pagination)

Empty List Message

ListView Header Footer

Add or Remove FlatList Item with Animation

ListView Scroll to Top/Bottom



Search Bar Filter

Use of ScrollView

Add Items to ScrollView using Loop

Scroll to a Specific Item in List View

Webview Props

Load Local HTML File / URL

Progress bar While Loading

[Move to Next Section >>](#)

