

TodosApp – FireStore – React Native

Getting Started React Native Firebase : <https://rnfirebase.io/>

Authentication: <https://rnfirebase.io/auth/usage>

Cloud Firestore: <https://rnfirebase.io/firestore/usage>

Cloud Storage: <https://rnfirebase.io/storage/usage>

1. Install via NPM

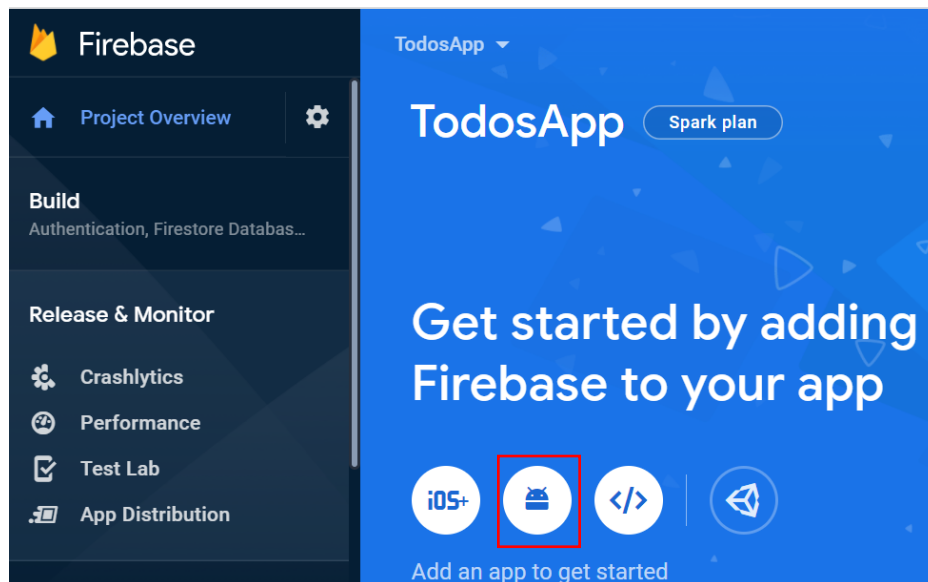
```
# Using npm
npm install --save @react-native-firebase/app

# Using Yarn
yarn add @react-native-firebase/app
```

2. Android Setup

Firebase console: <https://console.firebase.google.com/>

Create Project: TodosApp -> Create app Android



`/android/app/src/main/AndroidManifest.xml`

```
TodosApp > android > app > src > main > AndroidManifest.xml
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.todosapp">
2 |
```

Generate a certificate

```
> cd Android
```

```
> ./gradlew signingReport
```

Copy both 'SHA1' and 'SHA-256' keys that belong to the 'debugAndroidTest' variant key option. Then, you can add those keys to the 'SHA certificate fingerprints' on your app in Firebase console.

```
-----
Variant: debugAndroidTest
Config: debug
Store: E:\BAIGIANG\ReactNative\TodosApp\android\app\debug.keystore
Alias: androiddebugkey
MD5: 20:F4:61:48:B7:2D:8E:5E:5C:A2:3D:37:A4:F4:14:90
SHA1: 5E:8F:16:06:2E:A3:CD:2C:4A:0D:54:78:76:BA:A6:F3:8C:AB:F6:25
SHA-256: FA:C6:17:45:DC:09:03:78:6F:B9:ED:E6:2A:96:2B:39:9F:73:48:F0:BB:6F:89:9B:83:32:66:75:91:03:3B:9C
Valid until: Wednesday, May 1, 2052
-----
```

Create app Android

1

Register app

Android package name ⓘ

com.todosapp

App nickname (optional) ⓘ

My Android App

Debug signing certificate SHA-1 (optional) ⓘ

5E:8F:16:06:2E:A3:CD:2C:4A:0D:54:78:76:BA:A6:F3:8C:AB:F6:25

Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

Register app

Download the `google-services.json` file and place it inside of your project at the following location: `/android/app/google-services.json`

```
TodosApp > android > build.gradle
8      targetSdkVersion = 30
9    }
10   repositories {
11       google()
12       mavenCentral()
13       jcenter()
14   }
15   dependencies {
16       classpath("com.android.tools.build:gradle:4.1.0")
17       classpath 'com.google.gms:google-services:4.3.10'
18   // NOTE: Do not place your application dependencies here; they belong
19   // in the individual module build.gradle files
```

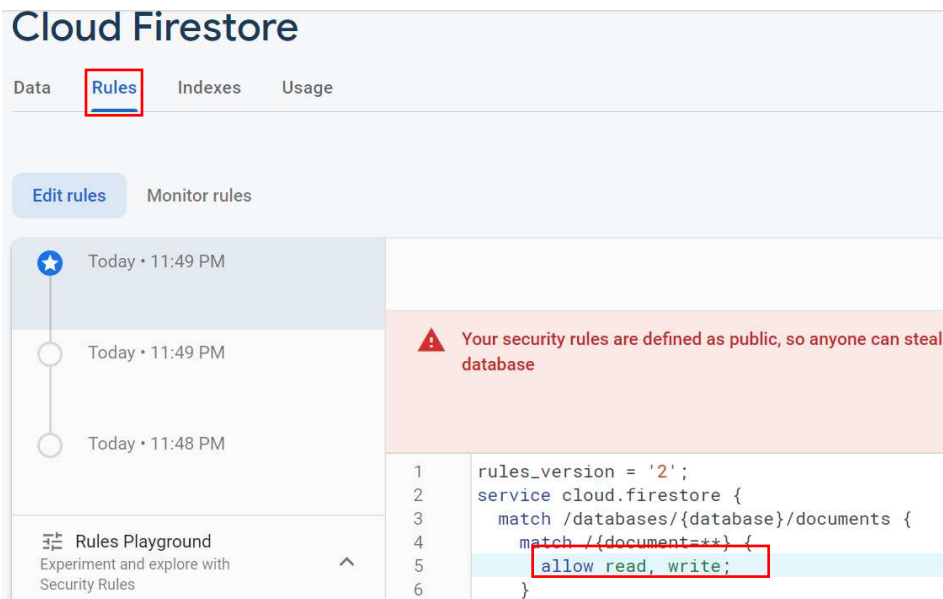
```
classpath 'com.google.gms:google-services:4.3.10'
```

```
TodosApp > android > app > build.gradle
1   apply plugin: "com.android.application"
2   apply plugin: 'com.google.gms.google-services'
3   import com.android.build.OutputFile
4
5   /**
6    * The react.gradle file registers a task for each build variant (e.g. bundleDebugJsAndAssets
7    * and bundleReleaseJsAndAssets).
8    * These basically call `react-native bundle` with the extra flag --bundle-output=  

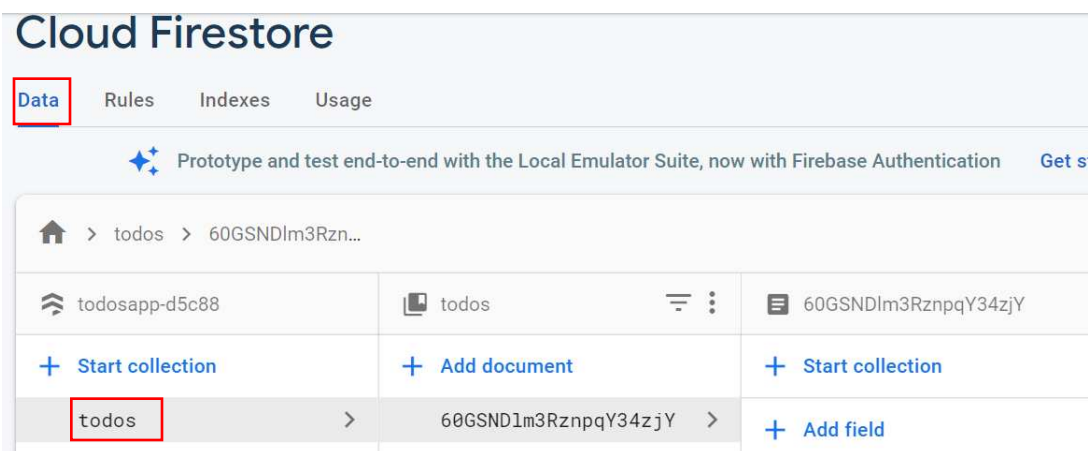
9    * cycle. By default, bundleDebugJsAndAssets is enabled, generating a bundle for the
```

```
apply plugin: 'com.google.gms.google-services'
```

Create Firestore Database



Create node root names todos



3. iOS Setup: <https://rnfirebase.io/>

4. Cài các thư viện sử dụng firestore và thiết kế giao diện sử dụng `react-native-paper`

Firestore: <https://rnfirebase.io/firestore/usage>

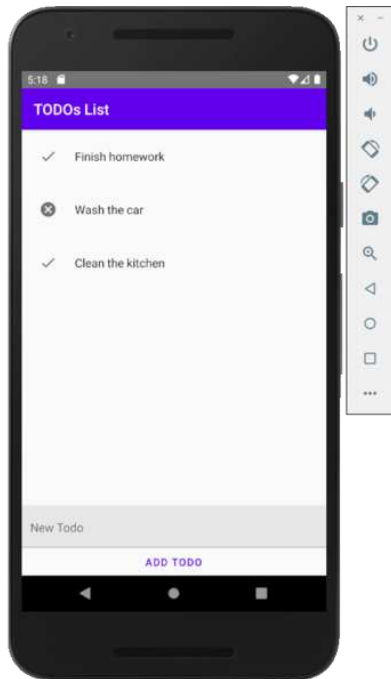
React-native-paper: <https://callstack.github.io/react-native-paper/getting-started.html>

```
"dependencies": {
  .....
  "@react-native-firebase/app":"13.0.1",
  "@react-native-firebase/firestore":"13.0.1",
  "react-native-paper":"4.10.1",
```

```
"react-native-vector-icons": "9.0.0"
},
```

>npm install

5. Code ứng dụng



Bước 1. Code file chạy App.js

```
import React from 'react';

function App() {
  return null;
}

export default App;
```

Bước 2. Tạo tham chiếu (ref) đến collection 'todos' của file store firestore

```
import React from 'react';
import firestore from '@react-native-firebase/firestore';

function App() {
  const ref = firestore().collection('todos');
  return null;
}
```

```
}  
  
export default App;
```

Bước 3. Giao diện cho ứng dụng

- Sử dụng các component: View, ScrollView, Text của react-native
- **AppBar, TextInput, Button** của **react-native-paper**

```
import React from 'react';  
import firestore from '@react-native-firebase/firestore';  
import { View, ScrollView, Text } from 'react-native';  
import { AppBar, TextInput, Button } from 'react-native-paper';  
function App() {  
  const ref = firestore().collection('todos');  
  return (  
    <View style={{flex:1}}>  
      <AppBar>  
        <AppBar.Content title={'TODOs List'} />  
      </AppBar>  
      <ScrollView style={{flex: 1}}>  
        <Text>List of TODOs!</Text>  
      </ScrollView>  
      <TextInput label={'New Todo'} onChangeText={() => {}} />  
      <Button onPress={() => {}}>Add TODO</Button>  
    </View>  
  );  
}
```

```
export default App;
```

Bước 4. Tạo State todo với phương thức setTodo để kết nối với giá trị text input

```
function App() {  
  const [todo, setTodo] = React.useState('');  
  const ref = firestore().collection('todos');  
  return (  
    <View style={{flex:1}}>  
      { /* ... */ }  
      <TextInput label={'New Todo'} value={todo} onChangeText={(text) => setTodo(text)} />  
    </View>  
  );  
}
```

```

    { /*... */ }
  </View>
);
}

```

Bước 5. Tạo function **addTodo** cho phép thêm mới 1 TODOS

```

function App() {
  const [ todo, setTodo ] = React.useState('');
  const ref = firestore().collection('todos');
  // ...
  async function addTodo() {
    await ref.add({
      title: todo,
      complete: false,
    });
    setTodo('');
  }
  // ...
}

```

Bước 6. Cập nhật sự kiện onPress cho button

```

<Button onPress={addTodo}>Add TODO</Button>

```

Khi thêm mới một todos sẽ thêm trên firestore, kiểm tra [Firestore](#) Console

Bước 7. Đọc tập TODOS firestore cập nhật ngược lên ScrollView

- Tạo State **todos** với phương thức **setTodos** để lưu trữ danh sách đọc được là array
- Và 'loading' state có giá trị mặc định là true (cho biết người dùng kết nối với firestore lần đầu tiên và việc tải dữ liệu chưa hoàn thành.

```

const [ loading, setLoading ] = React.useState(true);
const [ todos, setTodos ] = React.useState([]);

```

Bước 8. Tạo hàm để đọc dữ liệu từ firestore – hook react native

Sử dụng useEffect của React : <https://reactjs.org/docs/hooks-effect.html>

```

React.useEffect(() => {
  return ref.onSnapshot(querySnapshot => {
    const list = [];
    querySnapshot.forEach(doc => {
      const { title, complete } = doc.data();
      list.push({
        id: doc.id,
        title,
        complete,
      });
    });
  });

  setTodos(list);

  if (loading) {
    setLoading(false);
  }
});
});

```

Nếu đang trạng thái load return null

```

if (loading) {
  return null; // or a spinner
}

return (
  <View style={{flex:1}}>
    <AppBar>

```

Bước 9. Tạo component todo.js hiển thị todo trong thư mục gốc

```

import React from 'react';
import firestore from '@react-native-firebase/firestore';
import { List } from 'react-native-paper';

function Todo({ id, title, complete }) {
  async function toggleComplete() {
    await firestore()
      .collection('todos')
      .doc(id)
      .update({
        complete: !complete,

```



```

    });
  }

  return (
    <List.Item
      title={title}
      onPress={() => toggleComplete()}
      left={props => (
        <List.Icon {...props} icon={complete ? 'check' : 'cancel'} />
      )}
    />
  );
}

export default Todo;

```

Bước 10. Cấu hình lại FlatList trong App.js

```

import React from 'react';
import firestore from '@react-native-firebase/firestore';
import { FlatList, View, ScrollView, Text } from 'react-native';
import { AppBar, TextInput, Button } from 'react-native-paper';
import Todo from './todo'
function App() {
  const [todo, setTodo] = React.useState('');
  const [ loading, setLoading ] = React.useState(true);
  const [ todos, setTodos ] = React.useState([]);
  const ref = firestore().collection('todos');
  async function addTodo() {
    await ref.add({
      title: todo,
      complete: false,
    });
    setTodo('');
  }
  // mục đích useEffect để quản lý vòng đời của của một component voi react-hood
  React.useEffect(() => {
    return ref.onSnapshot(querySnapshot => {
      const list = [];
      querySnapshot.forEach(doc => {
        const { title, complete } = doc.data();
        list.push({
          id: doc.id,

```

```

        title,
        complete,
    });
});

setTodos(list);

if (loading) {
    setLoading(false);
}
});
});

if (loading) {
    return null; // or a spinner
}

return (
    <View style={{flex:1}}>
        <AppBar>
            <AppBar.Content title={'TODOs List'} />
        </AppBar>
        <FlatList
            style={{flex: 1}}
            data={todos}
            keyExtractor={(item) => item.id}
            renderItem={({ item }) => <Todo {...item} />}
        />
        <TextInput label={'New Todo'} value={todo} onChangeText={({text}) => setTodo(
text)} />
        <Button onPress={addTodo}>Add TODO</Button>
    </View>
);
}

export default App;

```