

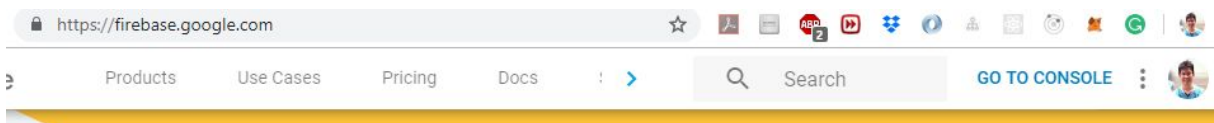
Lab 8 - Firebase (Web Infrastructure)

Firebase realtime database

Firebase เป็น platform ที่รวบรวมเครื่องมือต่าง ๆ ในส่วนของ backend มากมาย เช่น authentication, database, hosting, analytic ฯลฯ ในการทดลองนี้ จะเจาะจงในส่วนของฐานข้อมูลแบบ realtime และ hosting ซึ่งมีวิธีการดังนี้

Firebase console

1. ไปที่ firebase.google.com แล้ว เลือก Go to Console

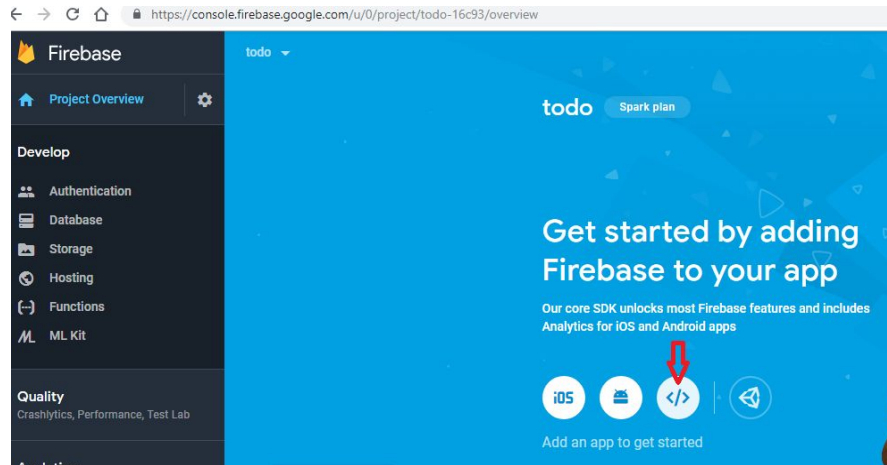


2. เลือก add project พร้อมกำหนดชื่อ และ เลือกยอมรับเงื่อนไข

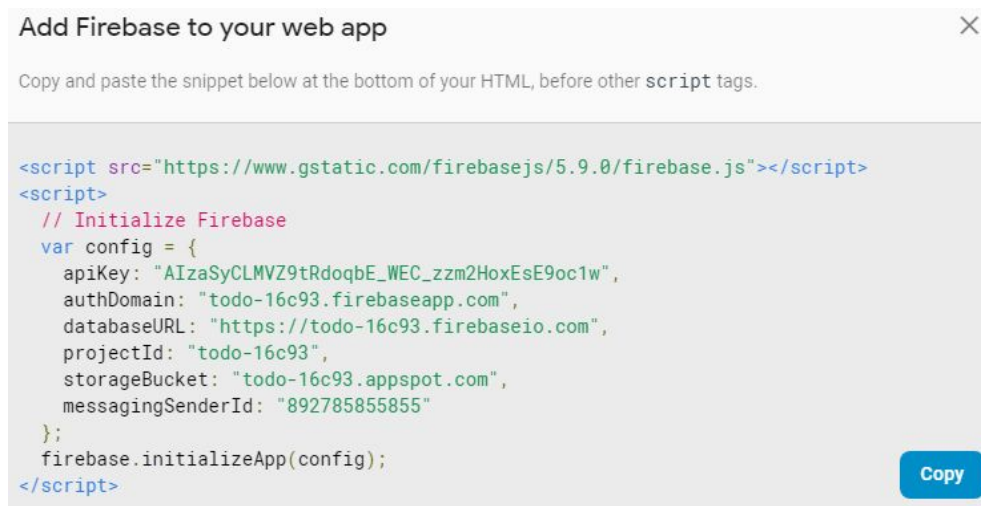
A screenshot of the 'Add a project' dialog in the Firebase console. The dialog has a title bar with a close button. It contains the following fields and options:

- Project name:** A text input field with the value 'todo'.
- Project ID:** A text input field with the value 'todo-16c93' and a pencil icon for editing.
- Locations:** A text input field with the value 'United States (Analytics)' and a pencil icon for editing. Below it, 'nam5 (us-central) (Cloud Firestore)' is listed.
- Analytics sharing:** A checkbox labeled 'Use the default settings for sharing Google Analytics for Firebase data' is checked. Below it, a list of five sharing options is shown, each with a checkmark:
 - ✓ Share your Analytics data with all Firebase features
 - ✓ Share your Analytics data with Google to improve Google Products and Services
 - ✓ Share your Analytics data with Google to enable technical support
 - ✓ Share your Analytics data with Google to enable Benchmarking
 - ✓ Share your Analytics data with Google Account Specialists
- Terms:** A checkbox labeled 'I accept the [controller-controller terms](#). This is required when sharing Analytics data to improve Google Products and Services. [Learn more](#)' is checked.
- Buttons:** At the bottom right, there are two buttons: 'Cancel' and 'Create project'.

3. จะได้หน้า dashboard จากนั้น เลือก web application



4. จะได้ source code ดังนี้ ให้ คัดลอกเก็บไว้



Create react todo app

1. สร้าง react project: `$ npx create-react-app todo`
2. ติดตั้ง firebase library: `$ npm i --save firebase`

และกำหนด App.js ดังนี้

App.js

```
import React, { Component } from 'react';
import firebase from 'firebase';

class App extends Component {

  constructor(props) {
    super(props)
```

```

    this.state = {
      tasks:
        [ {id:1, name:'Running'},
          {id:2, name:'Write React program'}
        ],
      task:''
    }

    var config = {
      apiKey: "AIzaSyCLMVZ9tRdoqbE_WEC_zzm2HoxEsE9oc1w",
      authDomain: "todo-16c93.firebaseio.com",
      databaseURL: "https://todo-16c93.firebaseio.com",
      projectId: "todo-16c93",
      storageBucket: "todo-16c93.appspot.com",
      messagingSenderId: "892785855855"
    };
    firebase.initializeApp(config);
  }

  renderTasks = () => {
    if (this.state.tasks.length !== 0)
      return this.state.tasks.map( (task,index) => {
        return (<li key={index}> {task.id}: {task.name} </li>)
      })
  }

  addTask = () => {
    let last = this.state.tasks.length
    let newTask = {id:this.state.tasks[last-1].id+1, name:this.state.task}
    this.setState({
      tasks: [...this.state.tasks, newTask]
    })
  }

  }

  handleChange = (e) => this.setState({task:
e.target.value})

  render() {
    return (
      <div style={{margin:'40px'}}>
        <h2>Todo: </h2>
        <input type="text" name="task"
onChange={this.handleChange} />
        <button onClick={this.addTask}>Submit </button> <br/>
        { this.renderTasks() }
      </div>
    );
  }
}

export default App;
```



Todo:

- 1: Running
- 2: Write React program

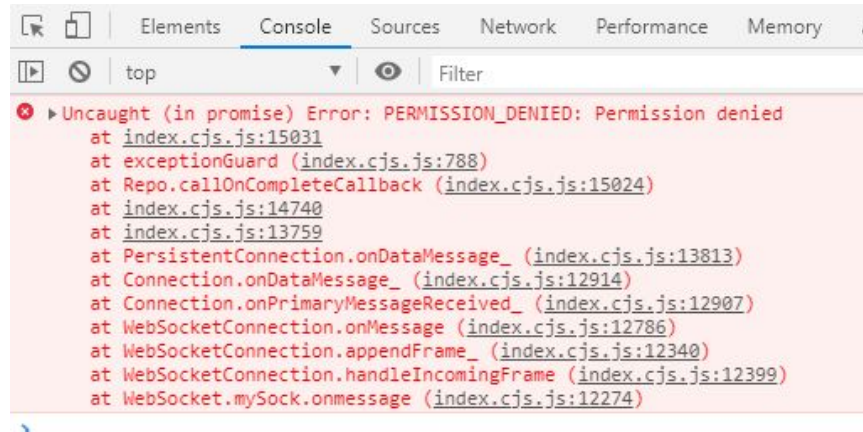
จากนั้นทดสอบการทำงานผ่าน <http://localhost:3000/>

ใน method addTask เพิ่ม code ต่อไปนี้

```
let dbCon = firebase.database().ref('/tasks');
```

```
dbCon.push({tasks: this.state.tasks});
```

เมื่อทดลองเรียกจะติด Permission ต้องไปกำหนดเพิ่มเติม

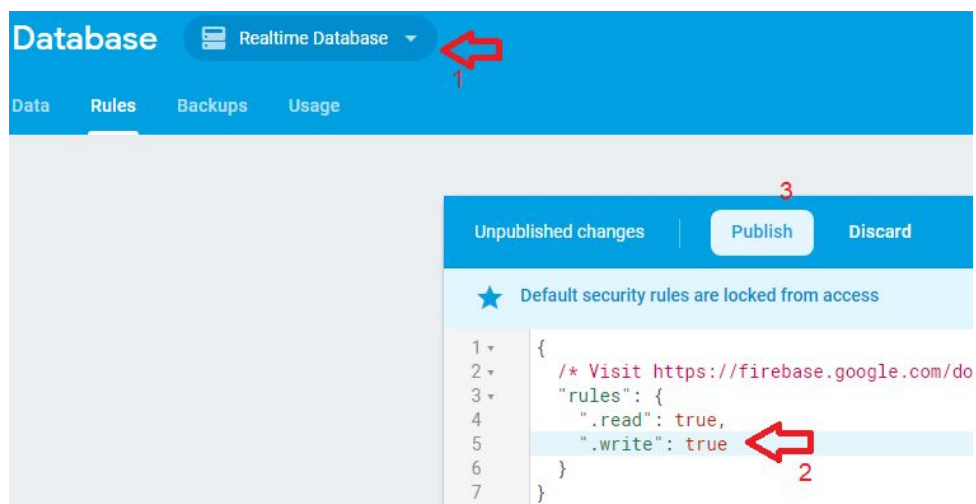
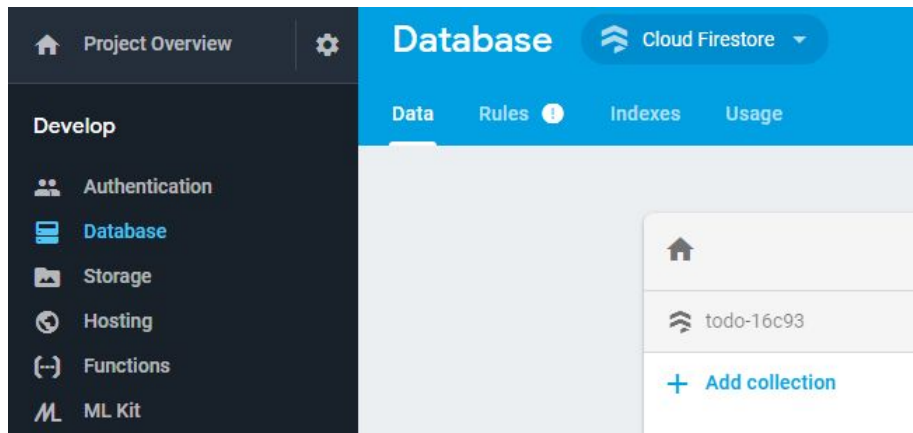


Database permission setting

กำหนดกฎการเข้าถึงฐานข้อมูลอย่างง่าย (สำหรับ development environment) เลือกที่ Database และ start in test mode จากนั้นกดปุ่ม Enable



จะได้ฐานข้อมูลมาดังนี้



Todo List CRUD example

ทดลองเขียนฟังก์ชันเพิ่มเติม เพื่อแก้ไขข้อมูลใน firebase

App.js

```
import React, {Component} from 'react';
import firebase from 'firebase';

class App extends Component {

  constructor(props) {
    super(props)
    this.state = {
      tasks: [
        {id: 1, name: 'Running at Damn'},
        {id: 2, name: 'Write React program'}
      ],
      task: ''
    }
  }
}
```

```

    }

    let config = {
      apiKey: "AIzaSyCLMVZ9tRdoqbE_WEC_zzm2HoxEsE9oc1w",
      authDomain: "todo-16c93.firebaseio.com",
      databaseURL: "https://todo-16c93.firebaseio.com",
      projectId: "todo-16c93",
      storageBucket: "todo-16c93.appspot.com",
      messagingSenderId: "892785855855"
    };

    if (firebase.apps.length === 0) firebase.initializeApp(config)

    console.log('firebase: ', firebase.database())
    console.log('firebase: ', firebase.app().name)

    let myapp = firebase.database().ref('/');
    let tasksChild = myapp.child('/taskTable')
    tasksChild.remove()
    tasksChild.set({tasks: this.state.tasks})
    myapp.on('value', snapshot => {
      console.log('task0: ', snapshot.val())
    });
  }

  removeTask = (id) => {
    let array = [...this.state.tasks]; // make a separate copy of the array
    let index = array.findIndex((task) => task.id === id)
    array.splice(index, 1)
    this.setState({tasks: array})
    console.log('update states tasks:', this.state.tasks)

    let tasksChild = firebase.database()
      .ref('/')
      .child('/taskTable/tasks/' + (id - 1))
    tasksChild.remove()
      .then(() => console.log("Remove success: "))
      .catch((err) => console.log("Remove failed: " + err))
  }

  addTask = () => {
    let lastItem = this.state.tasks[this.state.tasks.length - 1]
    let newTask = {id: lastItem.id + 1, name: this.state.task}
    this.setState({
      tasks: [...this.state.tasks, newTask]
    })

    let tasksChild = firebase.database()
      .ref('/')
      .child('/taskTable/tasks/' + lastItem.id)
    tasksChild.set(newTask)
      .then(() => console.log("Add successfully: "))
      .catch((err) => console.log("Remove failed: " + err))
  }

  editTask = (id) => {
    let index = this.state.tasks.findIndex((task) => task.id === id)
    this.setState({task: this.state.tasks[index].name})
  }

  updateTask = (id) => {
    console.log('update states tasks:', this.state.tasks)
    let array = [...this.state.tasks]
    let index = array.findIndex((task) => task.id === id)
    array[index].name = this.state.task
  }

```

```

        this.setState({tasks: array})

        let tasksChild = firebase.database()
            .ref('/')
            .child('/taskTable/tasks/' + (id - 1))
        tasksChild.set({id: array[index].id, name: this.state.task})
            .then(() => console.log("Update successfully: "))
            .catch((err) => console.log("Update failed: " + err))
    }

    handleChange = (e) => this.setState({task: e.target.value})

    renderTasks = () => {
        if (this.state.tasks.length !== 0)
            return this.state.tasks.map((task, index) => {
                return (
                    <tr key={index}>
                        <td>{task.id}</td>
                        <td> {task.name}</td>
                        <td>
                            <button onClick={() => this.editTask(task.id)}> Get</button>
                        </td>
                        <td>
                            <button onClick={() => this.updateTask(task.id)}> Update</button>
                        </td>
                        <td>
                            <button onClick={() => this.removeTask(task.id)}> Delete</button>
                        </td>
                    </tr>
                )
            })
    }

    render() {
        return (
            <div style={{margin: '40px'}}>
                <h2>Todo: </h2>
                <input type="text" name="task" onChange={this.handleChange}
value={this.state.task}/>
                <button onClick={this.addTask}>Add</button>
                <br/><br/>
                <table>
                    <thead>
                        <tr>
                            <th>ID</th>
                            <th>Task</th>
                            <th colSpan={3}>Action</th>
                        </tr>
                    </thead>
                    <tbody>
                        {this.renderTasks()}
                    </tbody>
                </table>
            </div>
        );
    }
}

export default App;

```

8.1 จงพัฒนาแอปพลิเคชันที่มีการเก็บข้อมูลลงใน Firebase มา 1 อย่าง (CRUD)

Firebase deployment

เป็นการนำแอปพลิเคชันวางบนเครื่องแม่ข่ายของ firebase เพื่อเข้าใช้งานจากเครื่องแม่ข่าย

Install Firebase tool

โดยใช้คำสั่ง `npm i -g firebase-tools`

Build react application สำหรับ production

โดยใช้คำสั่ง `npm run build` หรือ `yarn build`

Firebase login

เพื่อเชื่อมต่อคอมพิวเตอร์กับ firebase account โดยใช้คำสั่ง `firebase login` แล้ว firebase จะ redirect ไปยังหน้า page ของ Google account และให้กด Allow เพื่อให้สิทธิ์กับ firebase CLI เมื่อให้สิทธิ์เรียบร้อยแล้ว ที่หน้า console ก็จะแจ้งว่า login สำเร็จ

Firebase init

หลังจากนั้นสั่ง `firebase init` ให้ตอบ yes เพื่อดำเนินการ พร้อมกับเลือก Hosting

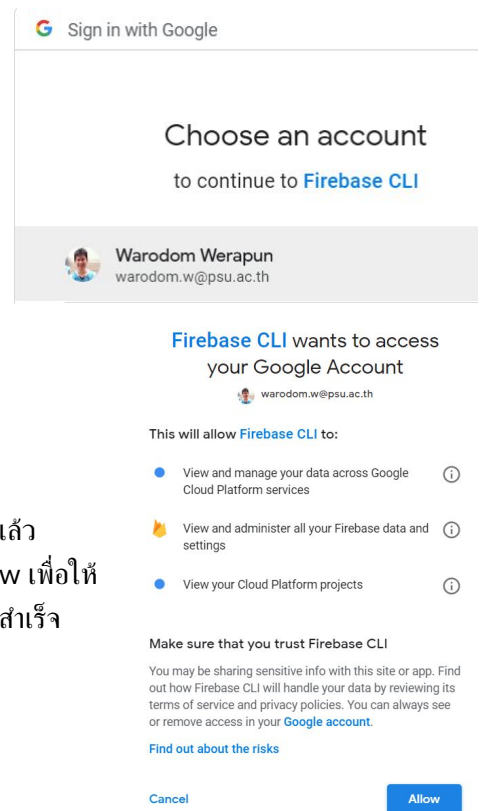
```
F:\nodejs\block\todo>firebase init

#####
##          ##          ##          ##          ##          ##          ##          ##
#####
##          ##          ##          ##          ##          ##          ##          ##
#####
##          ##          ##          ##          ##          ##          ##          ##
##          ##          ##          ##          ##          ##          ##          ##
#####

You're about to initialize a Firebase project in this directory:

  F:\nodejs\block\todo

? Are you ready to proceed? Yes
? Which Firebase CLI features do you want to setup for this folder? Press Space to select features,
then Enter to confirm your choices.
  ( ) Database: Deploy Firebase Realtime Database Rules
  ( ) Firestore: Deploy rules and create indexes for Firestore
  ( ) Functions: Configure and deploy Cloud Functions
> (*) Hosting: Configure and deploy Firebase Hosting sites
  ( ) Storage: Deploy Cloud Storage security rules
```



จากนั้นเลือก project ที่จะ deploy ให้เลือก project ที่พัฒนามา (ตัวอย่างคือ todo-16c93) (หากต้องการสร้าง project ใหม่ จะต้องมีการสร้าง project นั้นจาก <https://console.firebase.google.com> แล้วสั่ง add project โดยใช้คำสั่ง firebase use --add เพิ่มก่อน)

```
=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Select a default Firebase project for this directory:
  [don't setup a default project]
  guessnumber-5f301 (guessNumber)
  simpleapi-56f74 (SimpleApi)
  test-a4e15 (test)
> todo-16c93 (todo)
  [create a new project]
```

กำหนด public directory และ index file

```
=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? build
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? File build/index.html already exists. Overwrite? (y/N) n_
```

เท่านั้นเป็นอันเรียบร้อย

```
? What do you want to use as your public directory? build
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? File build/index.html already exists. Overwrite? No
i Skipping write of build/index.html

i Writing configuration info to firebase.json...
i Writing project information to .firebaseerc...

+ Firebase initialization complete!
```

Firestore deploy

สั่ง deploy แอปพลิเคชัน โดย firebase deploy เมื่อ deploy
เรียบร้อยแล้ว ก็จะสามารถเข้าแอปพลิเคชันได้จาก
<https://todo-16c93.firebaseio.com/>

8.2 ให้นำแอปพลิเคชันที่พัฒนาขึ้น firebase hosting

← → ↺ ⌂ 🔒 <https://todo-16c93.firebaseio.com/>

Todo:

Add

ID	Task	Action		
1	Running at Damn	Get	Update	Delete
2	Write React program	Get	Update	Delete