Mastering CRUD Operations with Next.js and Firebase

Ryan Adhitama Putra · Follow

Published in Stackademic · 7 min read · Feb 26, 2024

84 2

Embark on a journey of web development mastery with this tutorial. Learn to seamlessly integrate Firebase into your Next.js project, unlocking the power of real-time data manipulation.

Whether you're a beginner or an experienced developer, dive into this hands-on guide to elevate your proficiency in Next.js and Firebase, and build applications that stand out in today's digital landscape.

Requirement

Before we go so far, let's prepare some software we need.

Software

- Yarn (https://yarnpkg.com)
- Visual Studio Code (https://code.visualstudio.com/download)
- Firebase

1. Setup Project

Run the following command in your terminal/cmd

```
$ yarn create next-app next-firebase-crud
```

- Would you like to use TypeScript? **No**
- Would you like to use ESLint? **Yes**
- Would you like to use Tailwind CSS? **Yes**

- Would you like to use `src/` directory? **Yes**
- Would you like to use App Router? (recommended)? **No**
- Would you like to customize the default import alias (@/*)? **Yes**

2. Install Dependencies

You now have a new directory called **next-firebase-crud** Let's `cd` into it:

```
$ cd next-firebase-crud
```

Next, install dependencies

```
$ yarn add firebase
```

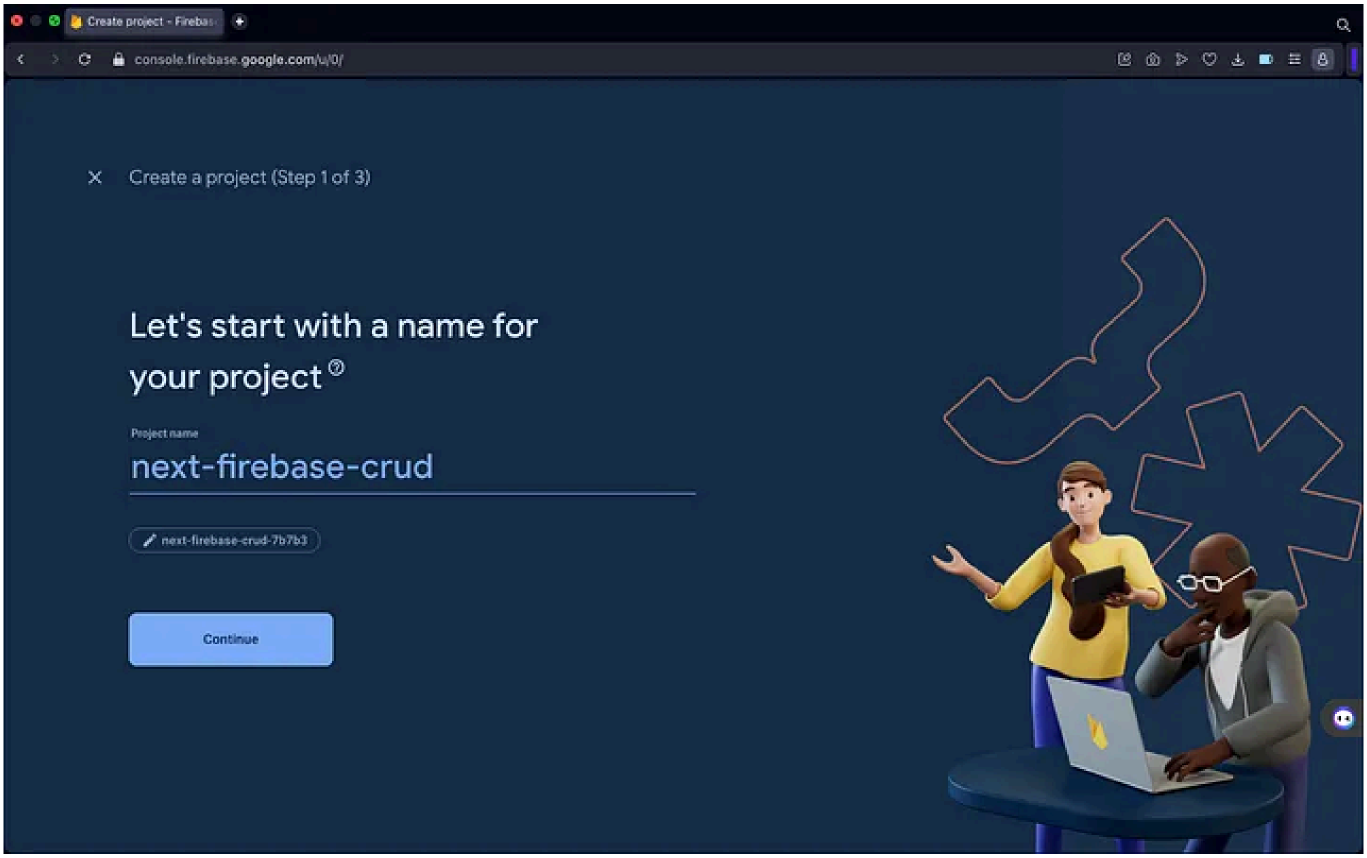
Explanation :

- **firebase :** Firebase JavaScript library for web and Node.js.

3. Firebase

Go to Console https://console.firebase.google.com/u/0/

Type your project name. In this tutorial, we create project named **next-firebase-crud**



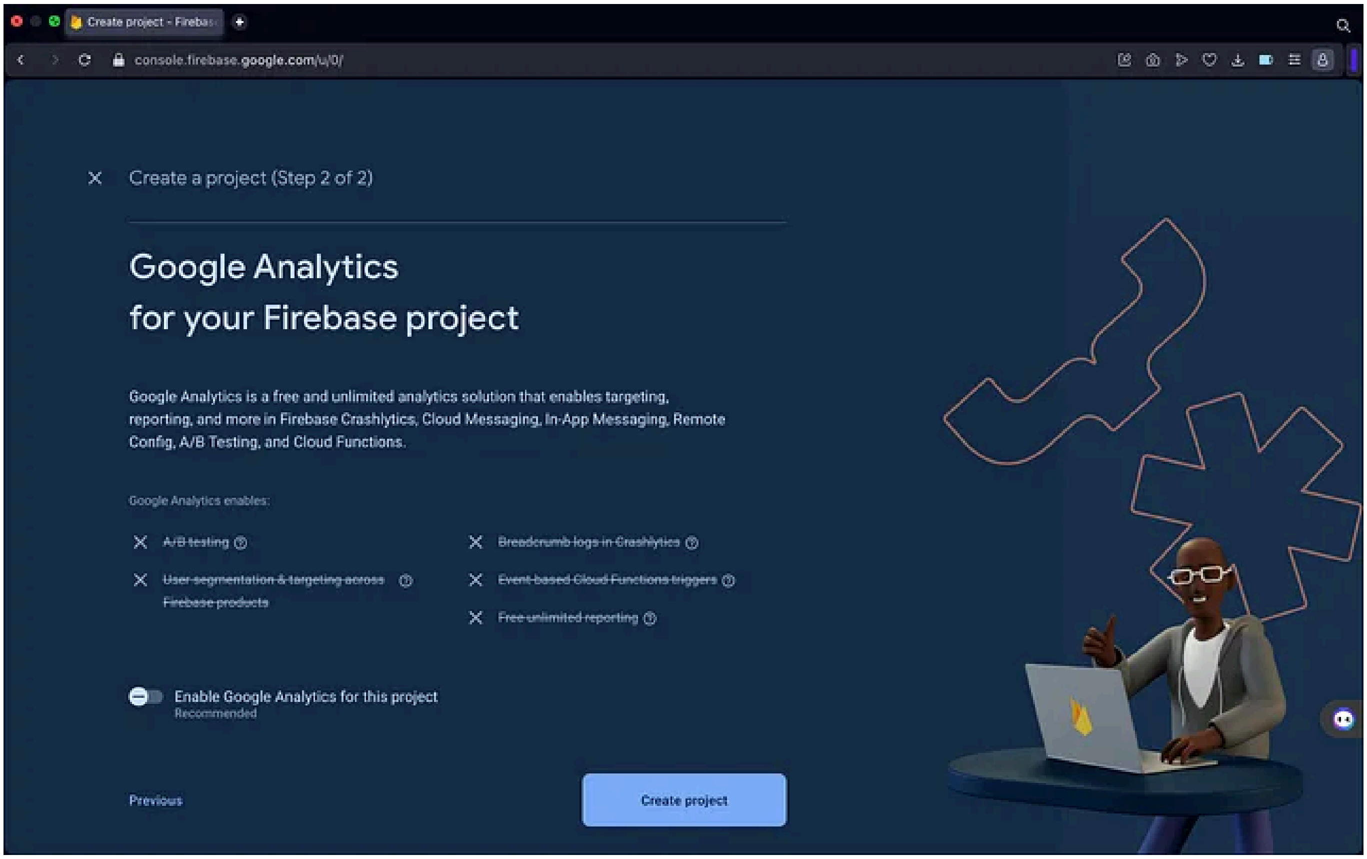
Create a project (Step 1 of 3)

Let's start with a name for your project

Project name: next-firebase-crud

next-firebase-crud-7b7b3

Disable google analytics, click create project



Create a project (Step 2 of 2)

Google Analytics for your Firebase project

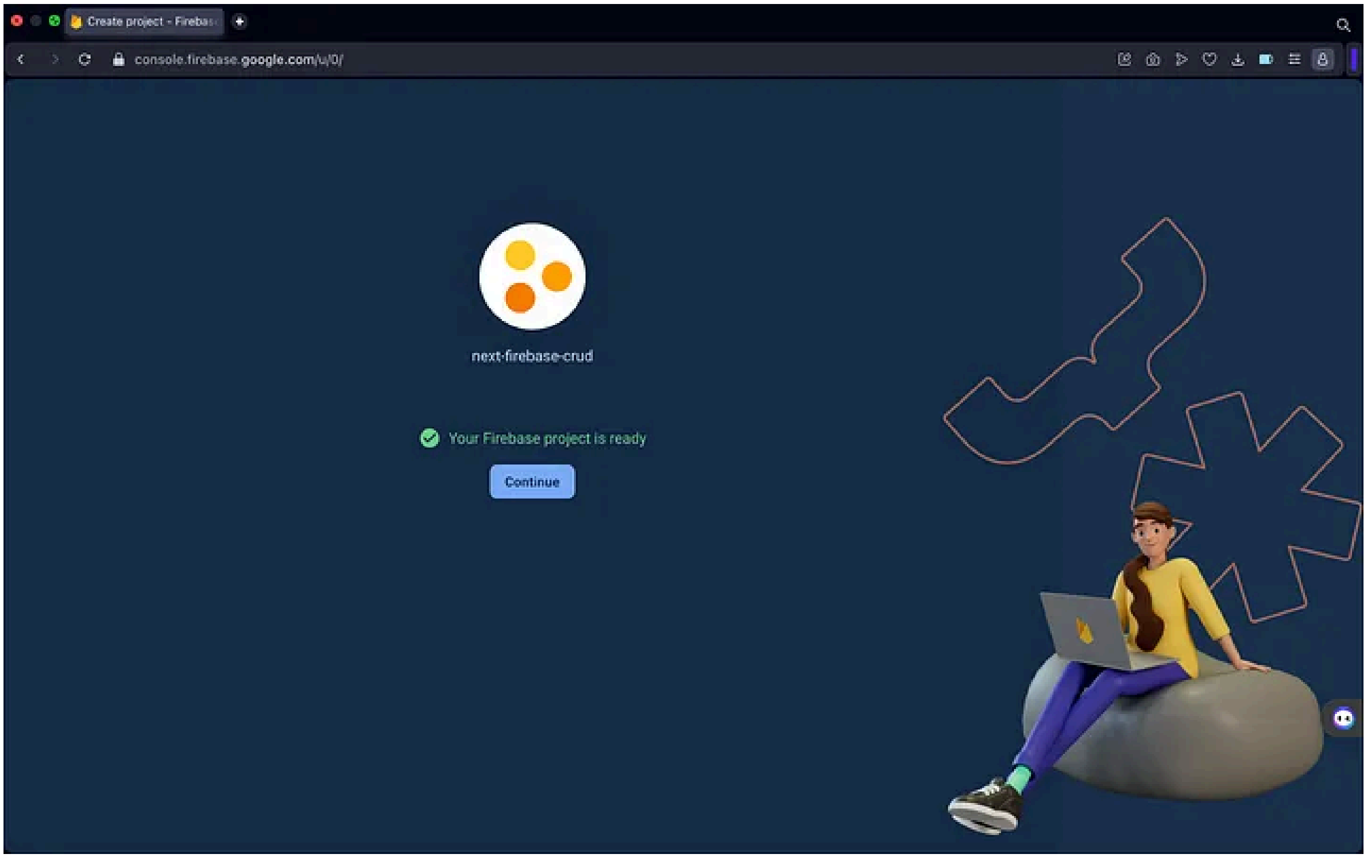
Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.

Google Analytics enables:

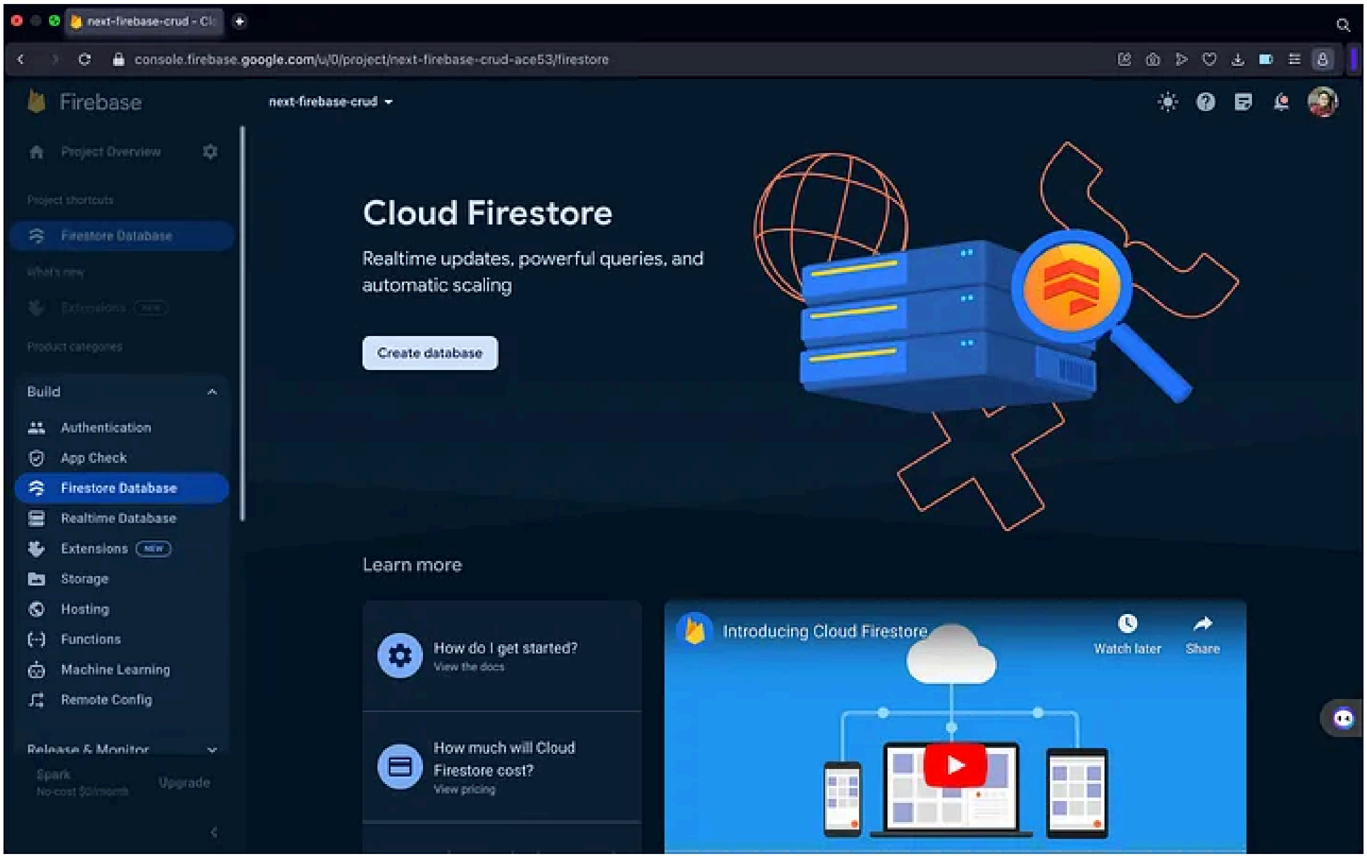
- ✗ ~~A/B testing~~
- ✗ ~~User segmentation & targeting across Firebase products~~
- ✗ ~~Breadcrumb logs in Crashlytics~~
- ✗ ~~Event-based Cloud Functions triggers~~
- ✗ ~~Free unlimited reporting~~

○ Enable Google Analytics for this project
Recommended

Click continue


Create project - Firebase
console.firebase.google.com/u/0/
next-firebase-crud
Your Firebase project is ready
Continue

Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform. Go to cloud firestore, and create database



Cloud Firestore

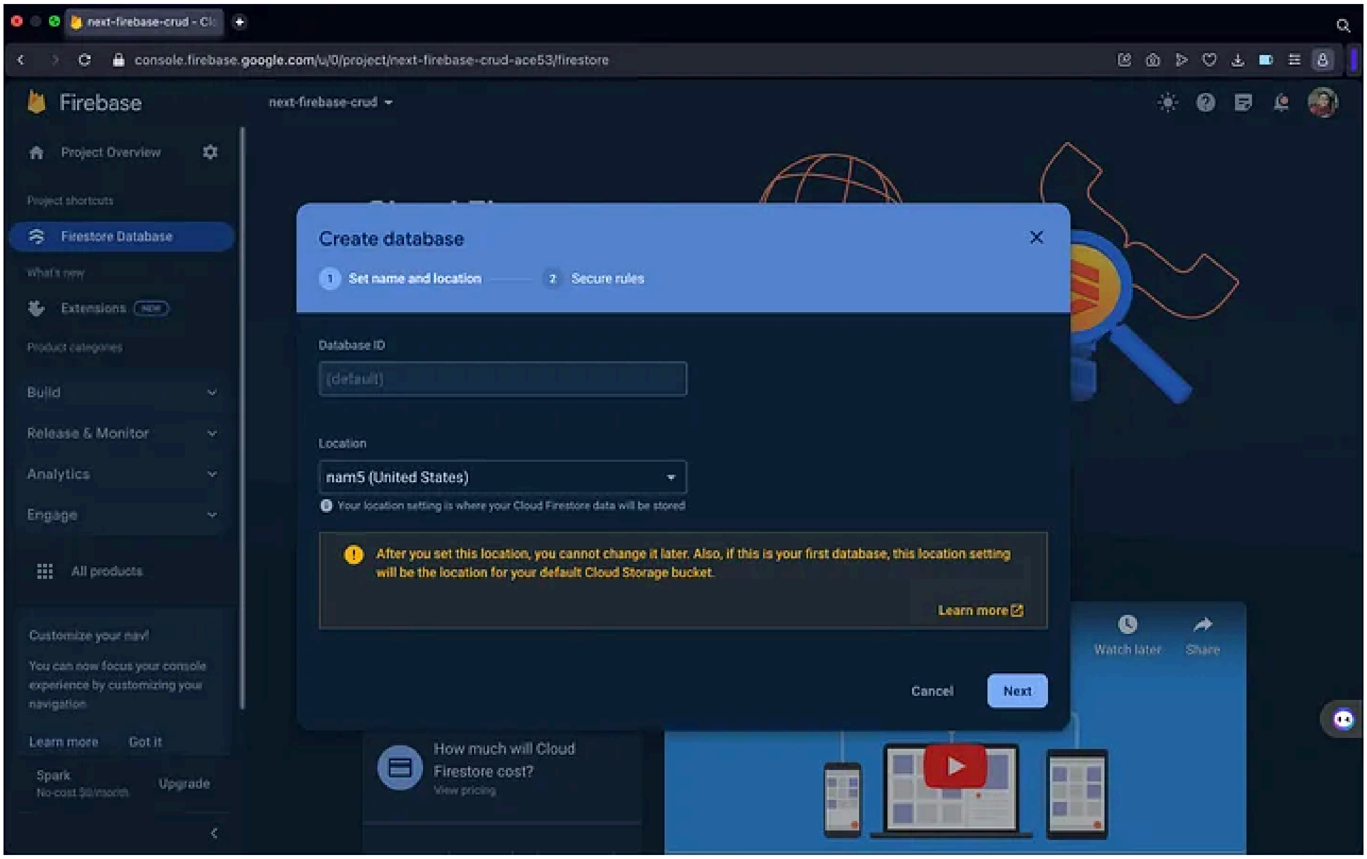
Realtime updates, powerful queries, and automatic scaling

Create database

Learn more

How do I get started?
View the docs

How much will Cloud Firestore cost?
View pricing



Create database

1 Set name and location

2 Secure rules

Database ID

(default)

Location

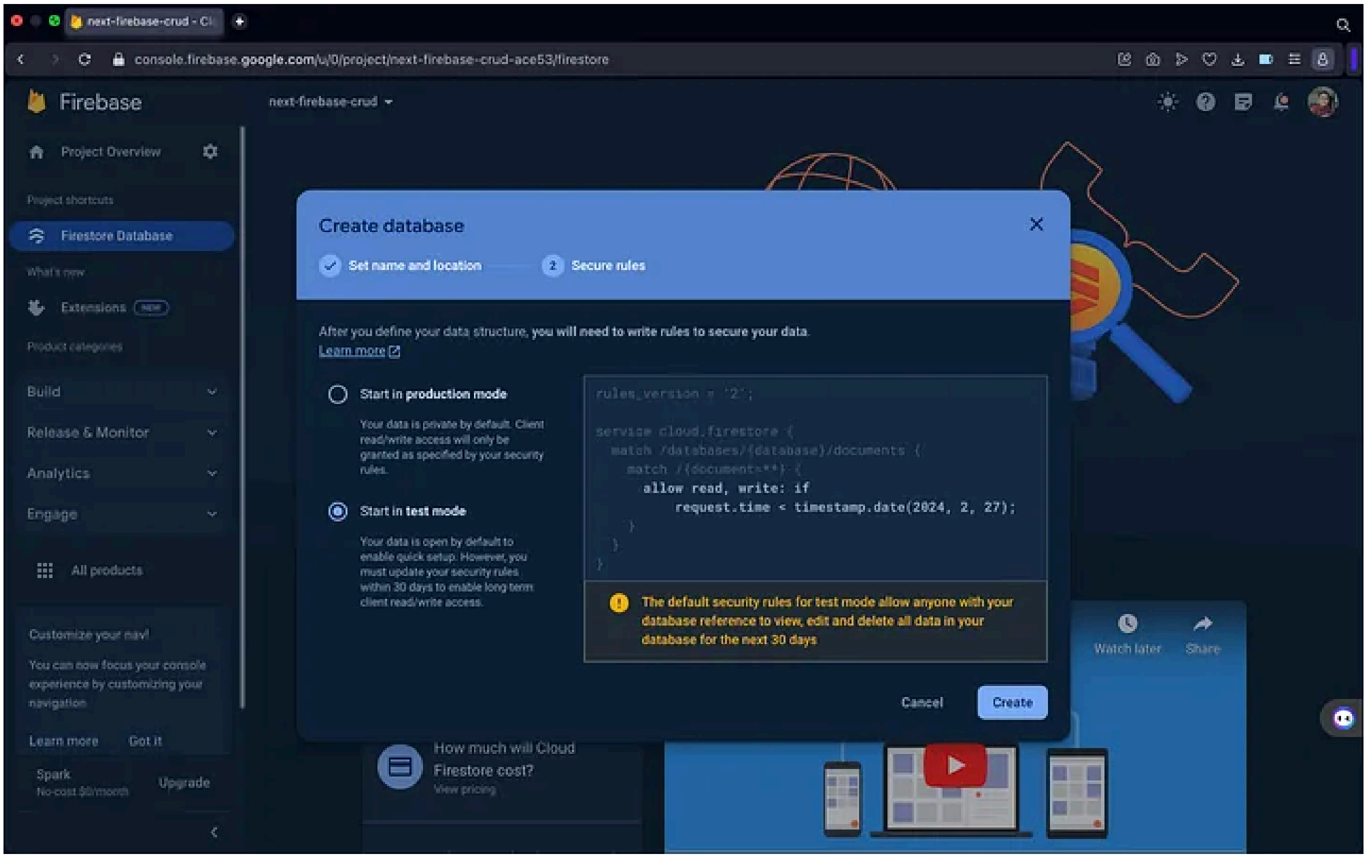
nam5 (United States)

Your location setting is where your Cloud Firestore data will be stored

After you set this location, you cannot change it later. Also, if this is your first database, this location setting will be the location for your default Cloud Storage bucket.

Learn more

Cancel Next



Create database

✓ Set name and location — 2 Secure rules

After you define your data structure, you will need to write rules to secure your data.
Learn more

○ Start in **production mode**

Your data is private by default. Client read/write access will only be granted as specified by your security rules.

● Start in **test mode**

Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long term client read/write access.

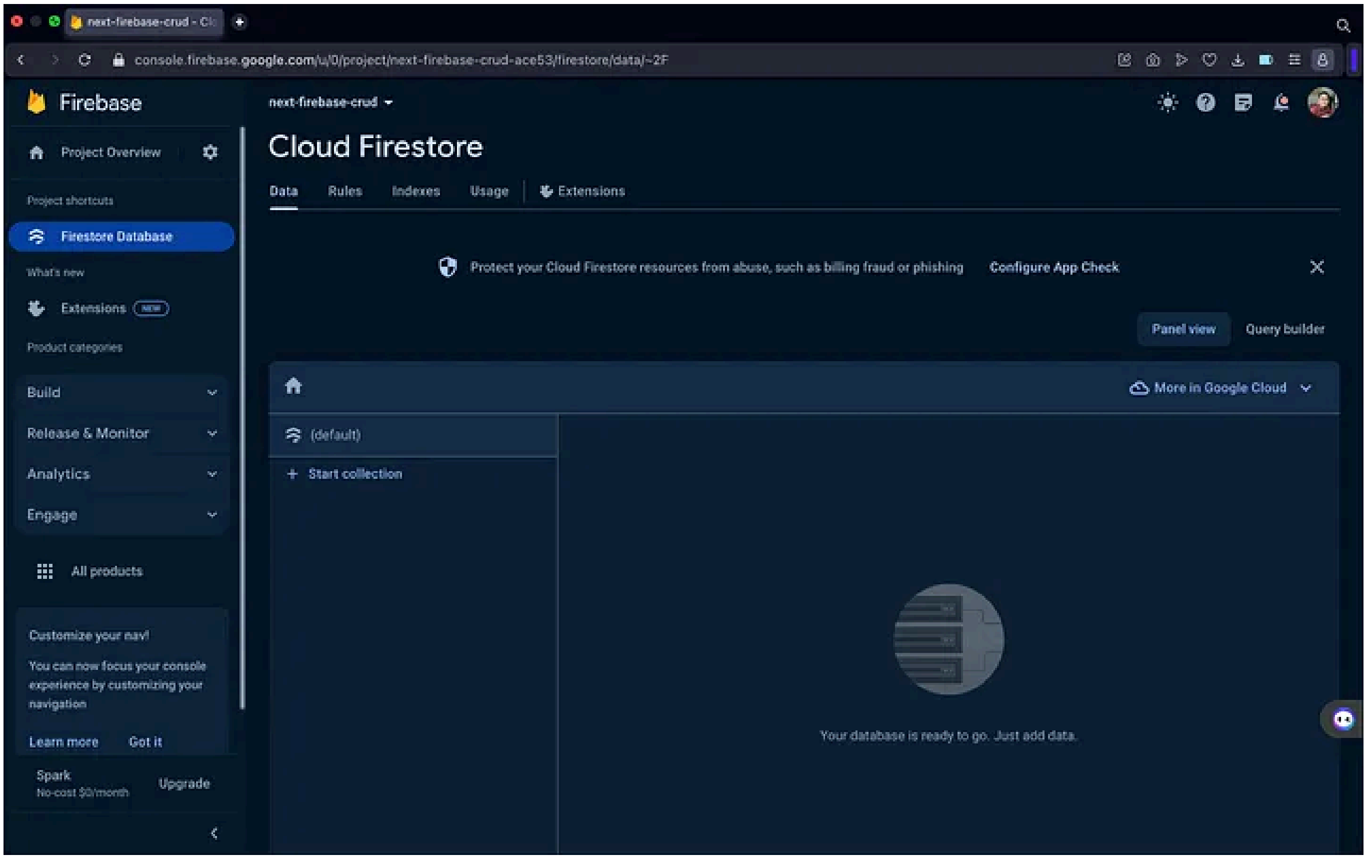
```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
          request.time < timestamp.date(2024, 2, 27);
    }
  }
}
```

The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

Cancel Create

Then, your cloud firestore will looks like this.



Cloud Firestore

Data | Rules | Indexes | Usage | Extensions

Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing — Configure App Check

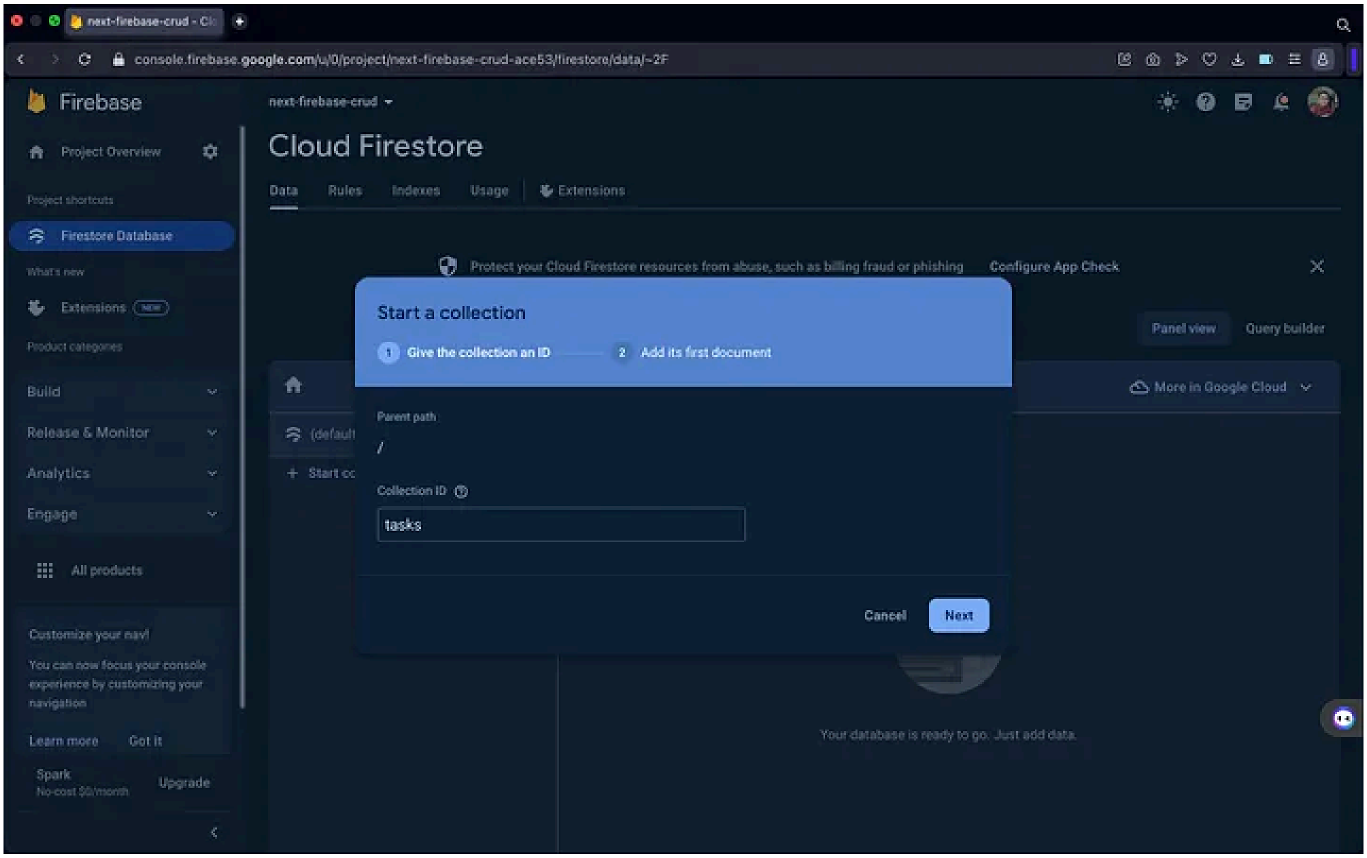
Panel view | Query builder

More in Google Cloud

(default)

+ Start collection

Your database is ready to go. Just add data.



Start a collection

1 Give the collection an ID — 2 Add its first document

Parent path

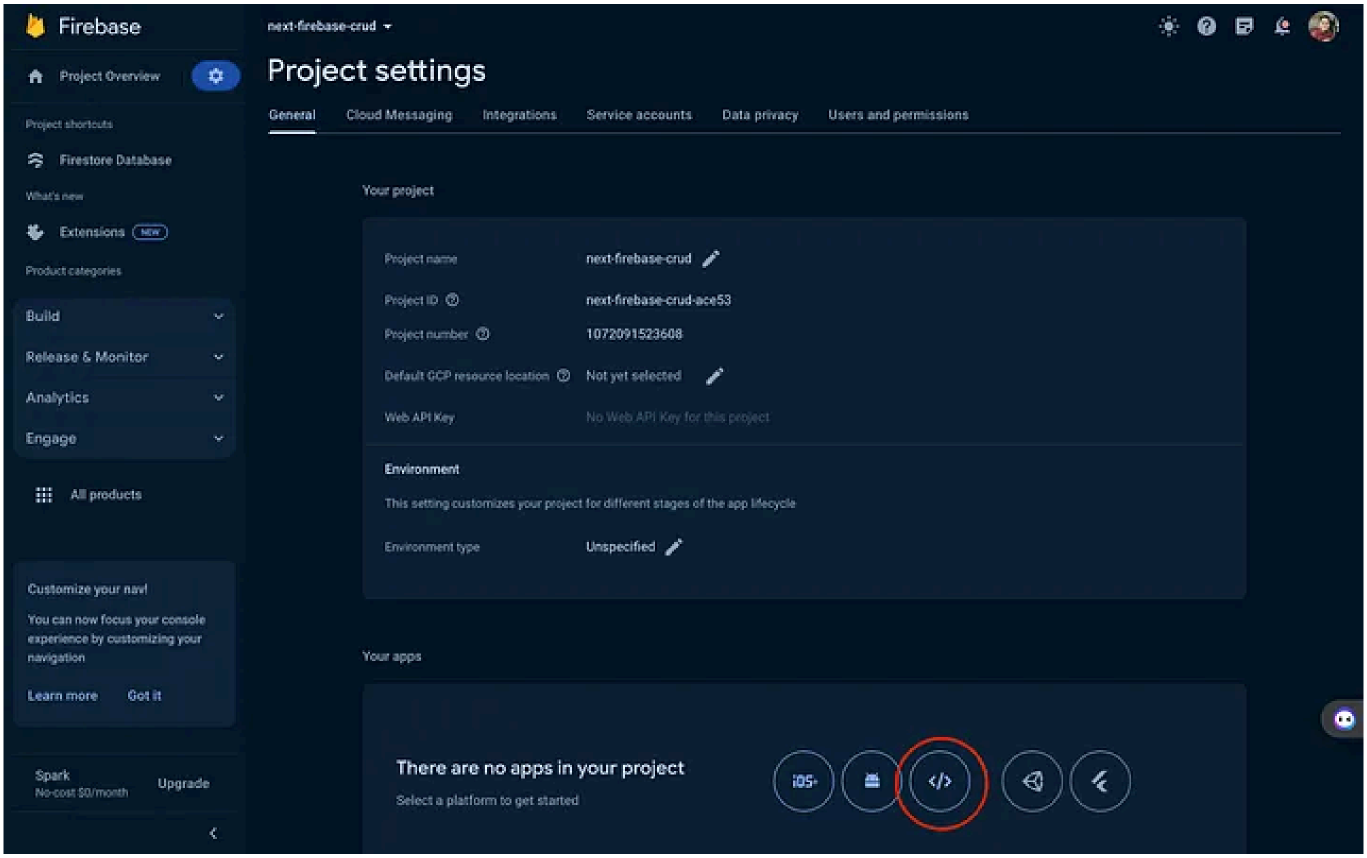
/

Collection ID

tasks

4. Register Firebase SDK App

Go to project setting, then choose web app.



Project settings

Your project

Project name next-firebase-crud

Project ID next-firebase-crud-ace53

Project number 1072091523608

Default GCP resource location Not yet selected

Web API Key No Web API Key for this project

Environment

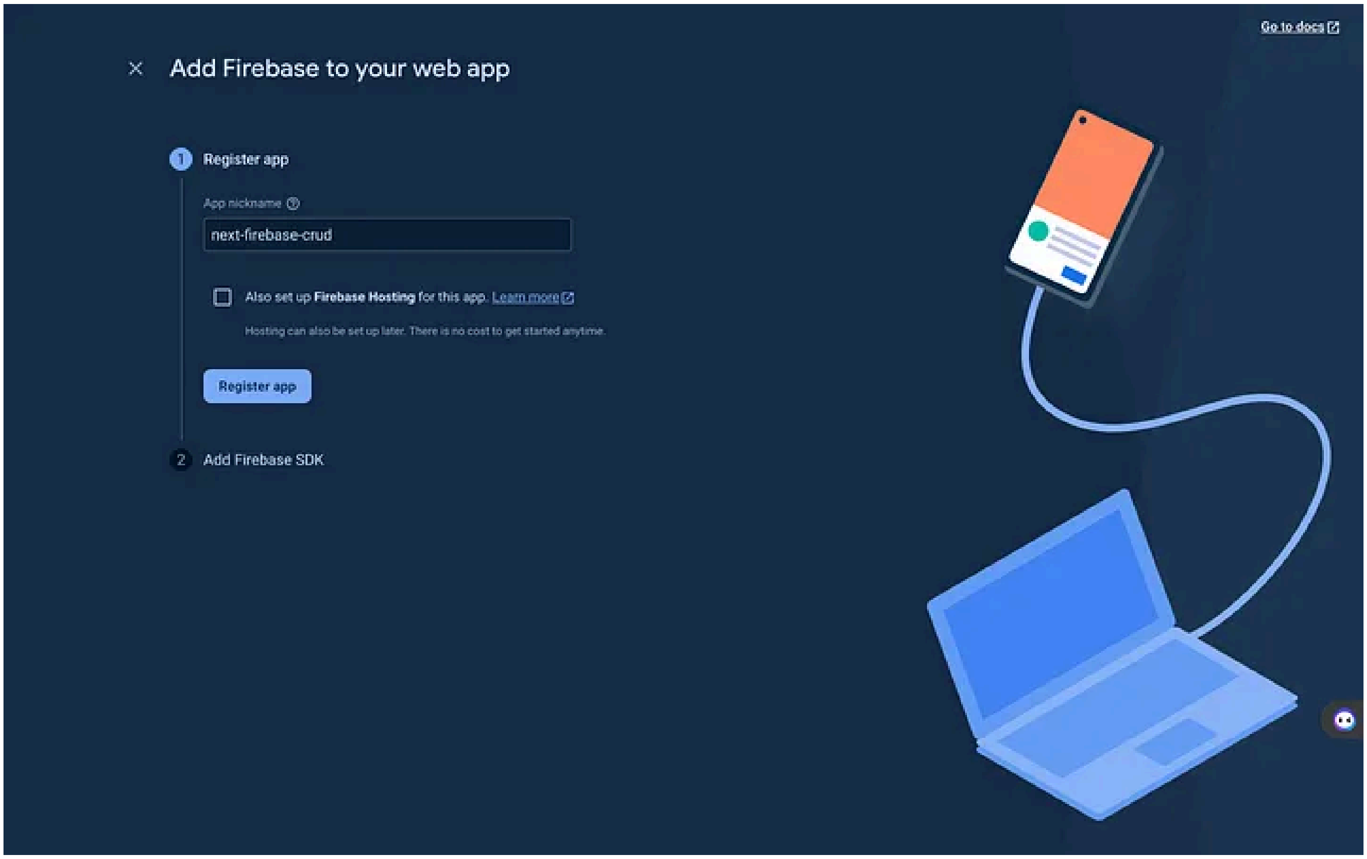
This setting customizes your project for different stages of the app lifecycle

Environment type Unspecified

Your apps

There are no apps in your project

Select a platform to get started



Add Firebase to your web app

1 Register app

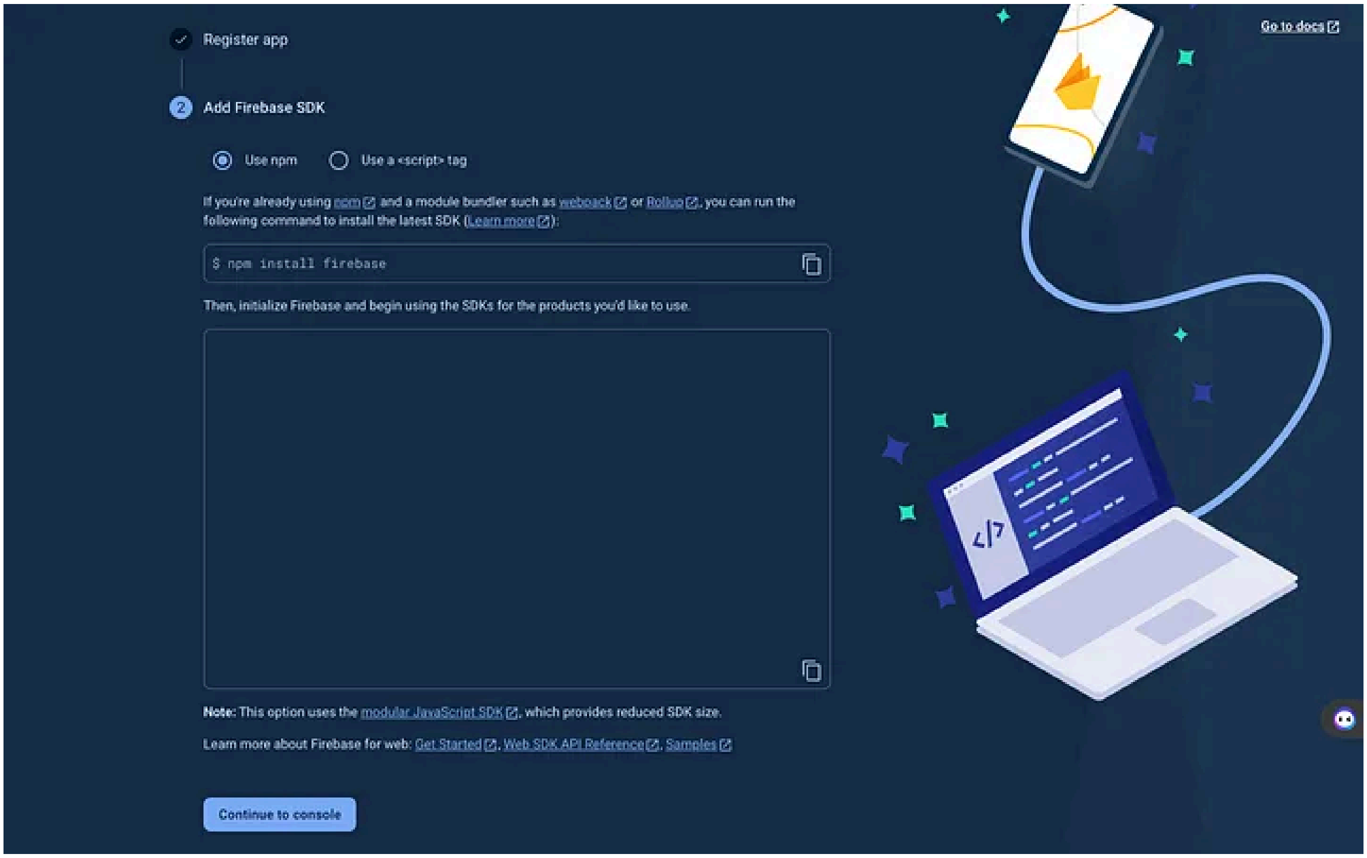
App nickname

next-firebase-crud

□ Also set up **Firebase Hosting** for this app. Learn more

Hosting can also be set up later. There is no cost to get started anytime.

2 Add Firebase SDK



Register app

2 Add Firebase SDK

● Use npm ○ Use a <script> tag

If you're already using npm and a module bundler such as webpack or Rollup, you can run the following command to install the latest SDK (Learn more):

```
$ npm install firebase
```

Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

Note: This option uses the modular JavaScript SDK, which provides reduced SDK size.

Learn more about Firebase for web: Get Started, Web SDK API Reference, Samples

5. Configure Environment Variable

Let’s create **.env** file, and copy-paste this code below and define variable value based on firebase config.

```
NEXT_PUBLIC_FIREBASE_API_KEY=api-key
NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN=auth-domain
NEXT_PUBLIC_FIREBASE_PROJECT_ID=project-id
NEXT_PUBLIC_FIREBASE_STORAGE_BUCKET=storage-bucket
NEXT_PUBLIC_FIREBASE_MESSAGING_SENDER_ID=sender-id
NEXT_PUBLIC_FIREBASE_APP_ID=app-id
```

6. Create Config File

Then create firebase configuration file **src/config/firebase.js**, and copy paste code below.

```
// src/config/firebase.js

// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: process.env.NEXT_PUBLIC_FIREBASE_API_KEY,
  authDomain: process.env.NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN,
  projectId: process.env.NEXT_PUBLIC_FIREBASE_PROJECT_ID,
  storageBucket: process.env.NEXT_PUBLIC_FIREBASE_STORAGE_BUCKET,
  messagingSenderId: process.env.NEXT_PUBLIC_FIREBASE_MESSAGING_SENDER_ID,
  appId: process.env.NEXT_PUBLIC_FIREBASE_APP_ID,
};

// Initialize Firebase
let app = getApps().length === 0 ? initializeApp(firebaseConfig) : getApps()[0];

export default app;
```

7. Create Page to List All Tasks

Update page **src/pages/index.js** and copy paste code below.

```
import { db } from "@/config/firebase";
import { collection, getDocs } from "firebase/firestore";
import Head from "next/head";
import Link from "next/link";
import { useEffect, useState } from "react";

export default function Home() {
  const [tasks, setTasks] = useState([]);

  const getTasks = async () => {
    const col = collection(db, "tasks");
    const snapshot = await getDocs(col);
    setTasks(snapshot.docs.map(doc => {
      return {
        id: doc.id,
        ...doc.data()
      }
    }));
  }

  useEffect(() => {
    getTasks()
  }, [])

  return (
    <>
      <div className="container mx-auto mt-8 max-w-[560px]">
        <div className="flex justify-between items-center pb-4 border-b border-d
          <h1 className="text-3xl font-semibold">Tasks</h1>
          <Link
            className="bg-green-600 hover:bg-opacity-80 text-white rounded-lg px
            href="/create"
          >
            Create New
          </Link>
        </div>
        <ul>
          {tasks.map((task) => (
            <li key={task.id} className="py-2 flex justify-between w-full">
              <span>
                <strong>{task.name}</strong> - {task.description}
              </span>
              <span className="flex gap-2">
                <Link className="text-blue-700 underline hover:no-underline" hre
                <Link className="text-red-500 underline hover:no-underline" href
              </span>
            </li>
          ))}
          {tasks?.length < 1 && <div className="py-2">No data</div>}
        </ul>
      </div>
      <Head>
        <title>Task</title>
      </Head>
    </>
  );
}
```

8. Create Page to Create New Tasks

Create page **src/pages/create.jsx** and copy paste code below.

```
import { db } from "@/config/firebase";
import { addDoc, collection, doc, setDoc } from "firebase/firestore";
import Head from "next/head";
import { useRouter } from "next/router";
import { useEffect, useState } from "react";

export default function Create() {
  const router = useRouter();
  const [task, setTask] = useState({
    name: "",
    description: "",
  });

  const onChange = (e) => {
    setTask({ ...task, [e.target.name]: e.target.value });
  };


  const handleCreate = async () => {
    const col = collection(db, "tasks");
    try {
      addDoc(col, {
        name: task.name,
        description: task.description,
      });
      router.push("/");
    } catch (error) {}
  };

  return (
    <>
      <>
        <div className="container mx-auto mt-8 max-w-[560px]">
          <div className="flex justify-between items-center pb-4 border-b border
            <h1 className="text-3xl font-semibold">Create Task</h1>
          </div>
          <form>
            <div className="mb-4">
              <label>Title</label>
              <input
                className="mt-1 px-4 py-2 border border-gray-300 rounded-md bloc
                type="text"
                name="name"
                value={task?.name}
                onChange={onChange}
              />
            </div>
            <div className="mb-4">
              <label>Description</label>
              <input
                className="mt-1 px-4 py-2 border border-gray-300 rounded-md bloc
                type="text"
                name="description"
                value={task?.description}
                onChange={onChange}
              />
            </div>
            <button
              className="bg-green-600 hover:bg-opacity-80 text-white rounded-lg
              type="button"
              onClick={handleCreate}
            >
              Create Task
            </button>
          </form>
        </div>
        <Head>
          <title>Create Task</title>
        </Head>
      </>
    </>
  );
}
```

9. Create Page to Edit Tasks

Create page **src/pages/[id]/edit.jsx** and copy paste code below.

```
import { db } from "@/config/firebase";
import { doc, getDoc, setDoc } from "firebase/firestore";
import Head from "next/head";
import { useRouter } from "next/router";
import { useEffect, useState } from "react";

const Edit = () => {
  const router = useRouter();
  const { id } = router.query;

  const [task, setTask] = useState({
    name: "",
    description: "",
  });

  const onChange = (e) => {
    setTask({ ...task, [e.target.name]: e.target.value });
  };

  useEffect(() => {
    const fetchTask = async () => {
      const ref = doc(db, "tasks", id);
      const snapshot = await getDoc(ref);
      setTask({ id: snapshot.id, ...snapshot.data() });
    };

    if (id) {
      fetchTask();
    }
  }, [id]);

  const handleUpdate = async () => {
    const ref = doc(db, "tasks", id);
    setDoc(ref, {
      name: task.name,
      description: task.description,
    })
      .then(() => {
        router.push("/");
      })
      .catch((error) => {
        console.log(error);
      });
  };

  return (
    <>
      <div className="container mx-auto mt-8 max-w-[560px]">
        <div className="flex justify-between items-center pb-4 border-b border-d
          <h1 className="text-3xl font-semibold">Edit Task</h1>
        </div>
        <form>
          <div className="mb-4">
            <label>Title</label>
            <input
              className="mt-1 px-4 py-2 border border-gray-300 rounded-md block
              type="text"
              name="name"
              value={task?.name}
              onChange={onChange}
            />
          </div>
          <div className="mb-4">
            <label>Description</label>
            <input
              className="mt-1 px-4 py-2 border border-gray-300 rounded-md block
              type="text"
              name="description"
              value={task?.description}
              onChange={onChange}
            />
          </div>
          <button
            className="bg-green-600 hover:bg-opacity-80 text-white rounded-lg px
            type="button"
            onClick={handleUpdate}
          >
            Edit Task
          </button>
        </form>
      </div>
      <Head>
        <title>Edit Task</title>
      </Head>
    </>
  );
};

export default Edit;
```

10. Create Page to Delete Tasks

Create page **src/pages/[id]/delete.jsx** and copy paste code below.

```
import { db } from "@/config/firebase";
import { deleteDoc, doc, getDoc } from "firebase/firestore";
import Head from "next/head";
import Link from "next/link";
import { useRouter } from "next/router";
import { useEffect, useState } from "react";
```

```
const Delete = () => {
  const router = useRouter();
  const { id } = router.query;

  const [task, setTask] = useState({
    title: "",
    description: "",
  });

  useEffect(() => {
    const fetchTask = async () => {
      const ref = doc(db, "tasks", id);
      const snapshot = await getDoc(ref);
      setTask({ id: snapshot.id, ...snapshot.data() });
    };

    if (id) {
      fetchTask();
    }
  }, [id]);

  const handleDelete = async () => {
    const ref = doc(db, "tasks", id);
    deleteDoc(ref)
      .then(() => {
        router.push("/");
      })
      .catch((error) => {
        console.log(error);
      });
  };

  return (
    <>
      <div className="container mx-auto mt-8 max-w-[560px]">
        <div className="flex justify-between items-center pb-4 border-b border-d
          <h1 className="text-3xl font-semibold">Delete Task</h1>
        </div>
        <form>
          <div className="my-12">
            Are you sure to delete <strong>{task?.name}</strong>?
          </div>
          <div className="flex w-full gap-2">
            <Link
              href="/"
              className="text-center bg-gray-300 hover:bg-opacity-80 text-black
            >
              Cancel
            </Link>
            <button
              className="bg-red-500 hover:bg-opacity-80 text-white rounded-lg px
              type="button"
              onClick={handleDelete}
            >
              Delete
            </button>
          </div>
        </form>
      </div>
      <Head>
        <title>Delete Task</title>
      </Head>
    </>
  );
};

export default Delete;
```

Don’t forget clear some css variable at **src/styles/globals.css.** And your globals.css will look like this

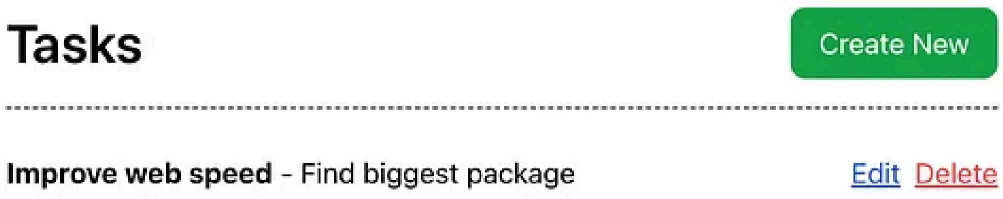
```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

Result

Run the following command

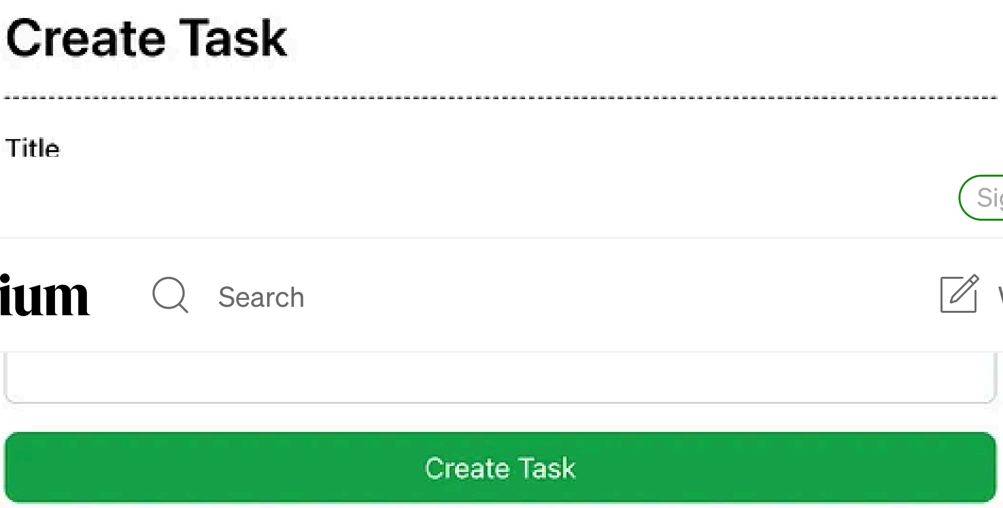
```
$ yarn dev
```

Then your result will looks like this

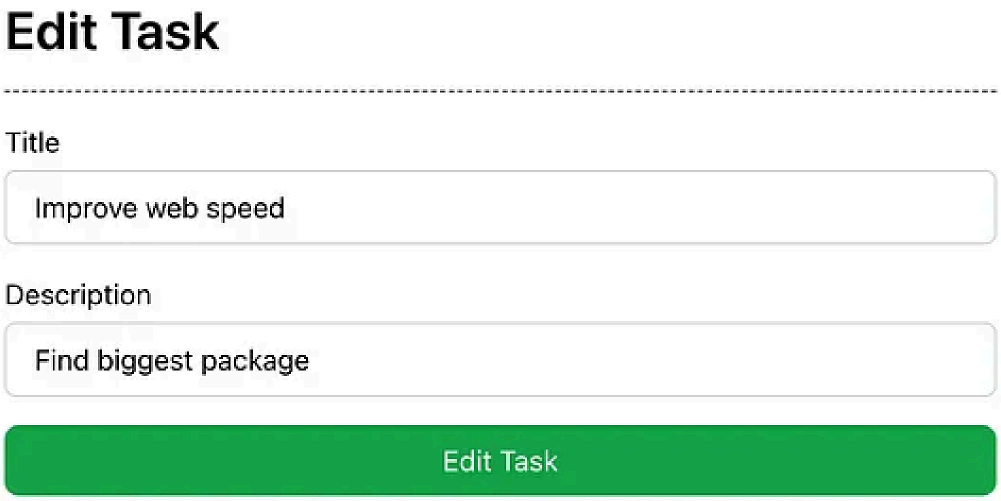


Tasks

Improve web speed - Find biggest package

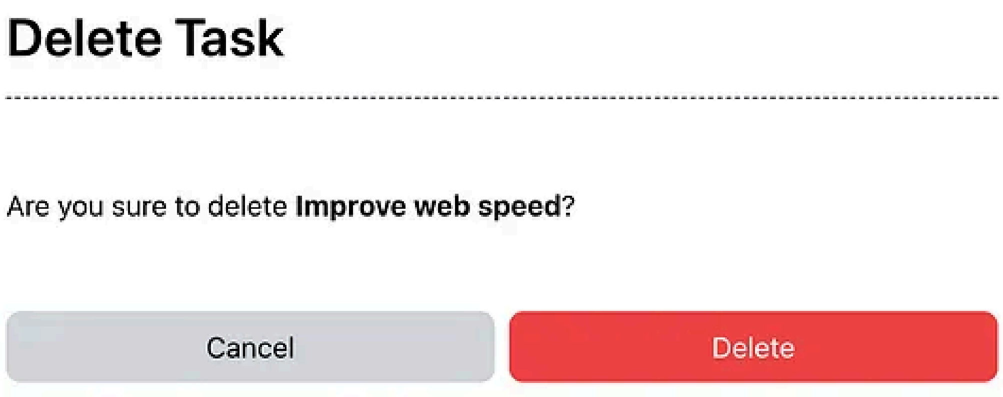

Create Task

Title


Edit Task

Title: Improve web speed

Description: Find biggest package


Delete Task

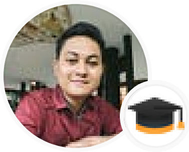
Are you sure to delete **Improve web speed**?

Stackademic 🎓

Thank you for reading until the end. Before you go:

- Please consider **clapping** and **following** the writer! 👏
- Follow us X | LinkedIn | YouTube | Discord
- Visit our other platforms: In Plain English | CoFeed | Venture | Cubed
- More content at Stackademic.com

Firebase Crud Nextjs React Tailwind



Written by Ryan Adhitama Putra

Follow

320 Followers · Writer for Stackademic

Software Engineer ; Tech Enthusiast ;

More from Ryan Adhitama Putra and Stackademic

Ryan Adhitama Putra in Stackademic

NextJS with Stripe Payment Integration (Part 1)

Stripe is a popular payment processing platform that allows businesses to accept...

4 min read · Nov 11, 2023

62 3

Dylan Cooper in Stackademic

Mojo, 90,000 Times Faster Than Python, Finally Open Sourced!

On March 29, 2024, Modular Inc. announced the open sourcing of the core components o...

✦ · 10 min read · Apr 8, 2024

3.7K 26

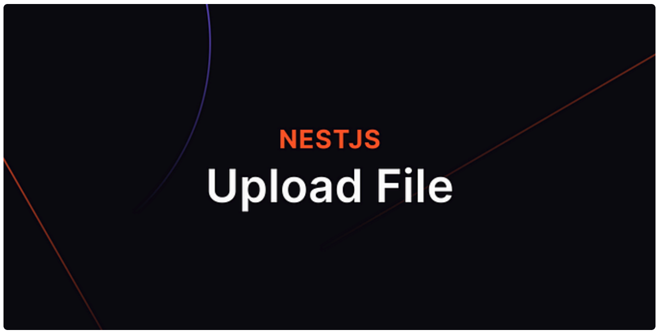
Jason Roell in Stackademic

Ultimate Python Cheat Sheet: Practical Python For Everyday…

(My Other Ultimate Guides)

34 min read · Jan 29, 2024

4.3K 38



Ryan Adhitama Putra in Bits and Pieces

Upload File with NestJS

How to upload file to local folder and online storage

6 min read · May 28, 2023

18

See all from Ryan Adhitama Putra

See all from Stackademic

Recommended from Medium

Stanislav Sopov

Sending Emails with Firebase

A Complete Guide Using Cloud Functions and Nodemailer

6 min read · Apr 13, 2024

10

Nikhil Chapre in JavaScript in Plain English

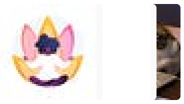
Client Side Form Validation with Zod & useFormState() in Next.js 14

Authentication in Web Apps is a crucial aspect for securing any Web Application an...

8 min read · Feb 26, 2024

189

Lists

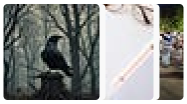


Stories to Help You Grow as a Software Developer

19 stories · 1027 saves

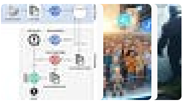
General Coding Knowledge

20 stories · 1184 saves



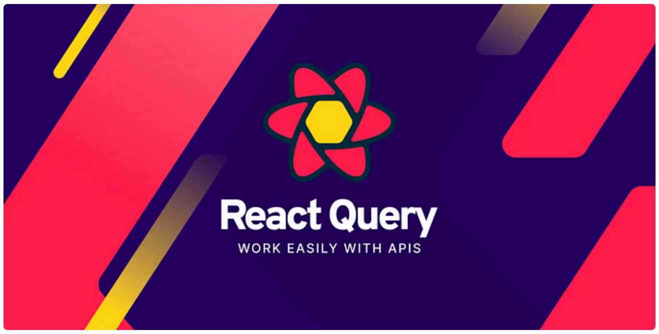
Medium's Huge List of Publications Accepting...

302 stories · 2593 saves



Natural Language Processing

1424 stories · 921 saves



Carina González

React Query: Integration with Next JS, are you ready?

Hello developers! Get ready for an insightful dive into the world of React Query. In this blo...

5 min read · Dec 29, 2023

Rishi Pardeshi

Authentication with NextJS 14 and Next Auth

Hey viewers! My self Rishi. Let's direct jump into the topic, today we will see how to set u...

8 min read · Dec 24, 2023

52 2

Ingrid Ruiz

How create Firebase Realtime Database and Hosting on a React…

In this upcoming article, I will present a step by step guide on how to use Realtime…

3 min read · Dec 12, 2023

2

122 7

Santosh Yadav

Create CRUD API in Next.Js

To Create An API in NEXT.JS. We should First Create a fresh Project.

7 min read · Jan 13, 2024

4

See more recommendations