# Workshop: setup a domain and newsletter service from scratch for free with **YUNoHost** + ‘**mail server name**’ + **AMP** + **Node** + **React** + **Firebase**

1. **Build our own domain using YUNoHost. Get our own email and mail server.**
   * Install the web server, host a sample page to test it.
   * Install the mail server, authenticate it with Google, send out a sample mail.
2. **Build an email template using AMP.**
   * Make a template from which they will create their own.
   * Make a pdf that synthesises the different things you can do with AMP, and its syntax.
3. **Build a node application that sends out emails.**
   * How to integrate it to our server (NPM install + CRON job?)
   * How to insert the AMP template inside of it.
4. **Setup a front-end using React and Firebase.**
   * Use YUNoHost to host our web page built with React and Firebase.
   * Use Firebase to link our front-end to the back-end which is the node mailer.
   * Subscribe a sample email to the newsletter and do a firing test.

## How to setup Firebase with React:

Create a free account on firebase

Install the package in our project npm install --save firebase

Build a file called fire.js in our src folder

Paste the Config file in it. The config file can be found by following these steps:

<https://support.google.com/firebase/answer/7015592?hl=en>

<https://firebase.google.com/docs/web/setup>

<https://www.youtube.com/watch?v=qo7xtBCh18o>

## **Get config object for your web app**

To get the config object for a Firebase Web App:

1. Sign in to Firebase, then open your project.
2. Click the cog, then select **Project settings**.
3. In the **Your apps** card, select the nickname of the app for which you need a config object.
4. Select **Config** from the Firebase SDK snippet pane.
5. Copy the config object snippet, then add it to your app's HTML.

After that change the writing of the firebase app in the fire.js module like so:

import firebase from 'firebase';

const firebaseConfig = {

apiKey: "AIzaSyAuzdcOqv\_ZPNRaypeVUO4xCumQdrXmIc8",

authDomain: "lunar-chat-8337b.firebaseapp.com",

databaseURL: "https://lunar-chat-8337b.firebaseio.com",

projectId: "lunar-chat-8337b",

storageBucket: "lunar-chat-8337b.appspot.com",

messagingSenderId: "79233455457",

appId: "1:79233455457:web:b78c1fa2ba882f8a"

};

const firebaseApp = firebase.**initializeApp**(firebaseConfig);

export default firebaseApp;

Once this is done import it in App.js and run a console.log on it. Let’s go through each part of that console.log to further understand what exactly is going on here. Under \_proto\_ you can find all the services Firebase offers, but aren’t quite implemented yet.

Then go to the “database” tab and create a simple “realtime database” in **locked mode**.

Let’s do a little test with our database by adding data to it. First of all,

const foodRef = firebaseApp.**database**().**ref**('foods');

foodRef.push({

name: 'Pizza',

price: 10.00

})

console.log(firebaseApp.**auth**());

This will return an error of type “PERMISSION\_DENIED”, for the moment let’s change our database rules by read: true and write: true. These rules make it so everyone can read and write to our database, this is just a test, never do that ever.