Send Email With Node.JS From a React Website Using Nodemailer and Sendgrid

Bassit Owolabi · Follow

Published in The Startup · 5 min read · Oct 12, 2020

69 5

A while back I wrote an article on how to send e-mails directly from your

Re

n

find it here.

This tutorial isn't completely beginner-friendly, I expect you to have a basic understanding of how node works, a basic express setup and how to enable communication between an express server and a React website as the client.

Source code for this tutorial is available **here**

In this article I will show you how to:

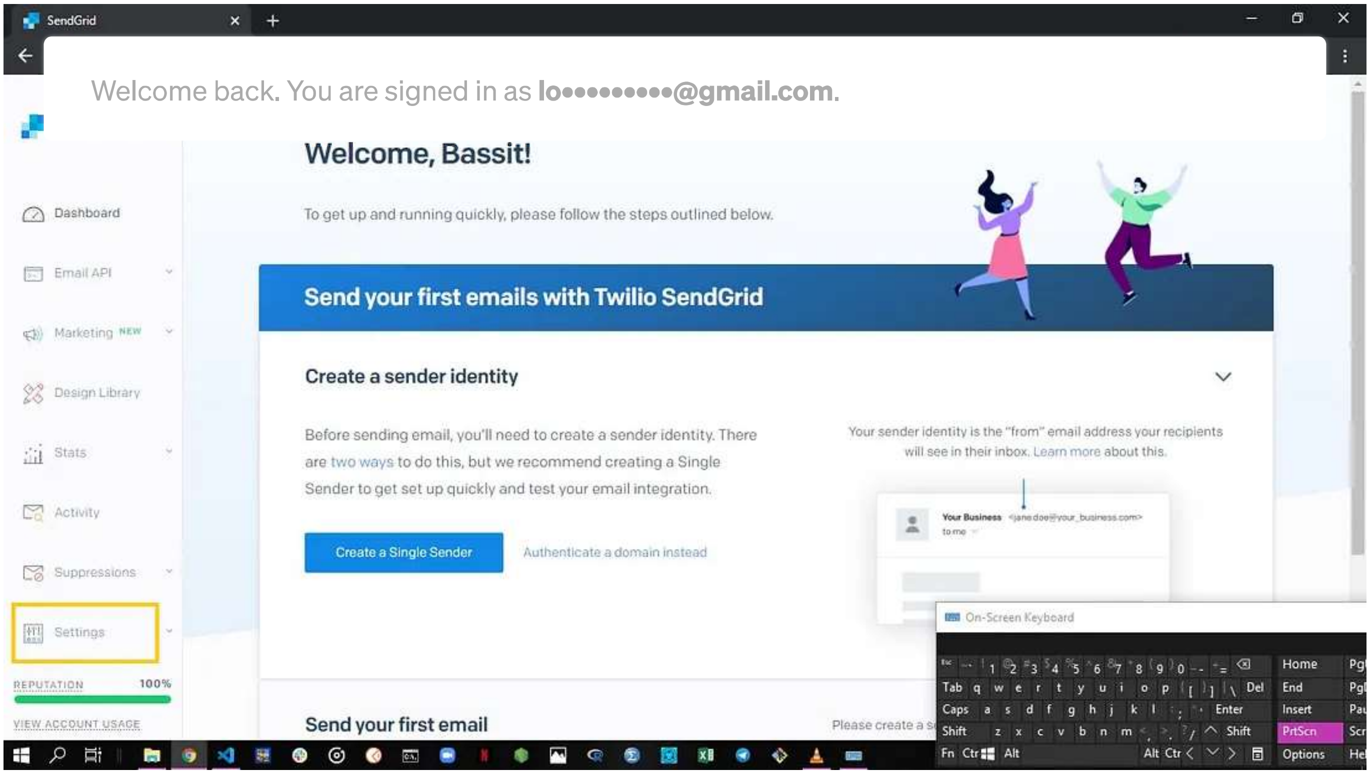
- configure nodemailer
- Send e-mail with node

1. **Configure sendgrid account**

Sendgrid is a cloud-based SMTP server that allows you to send emails without having to maintain email servers.

Head over to sendgrid, create an account using the free plan or log in if you already have an account.

After login, you will be greeted with a page as shown in the image below. Click on settings, I made a yellow box around it for easier identification.



Welcome back. You are signed in as **lo••••••••••@gmail.com**.

Welcome, Bassit!

To get up and running quickly, please follow the steps outlined below.

Send your first emails with Twilio SendGrid

Create a sender identity

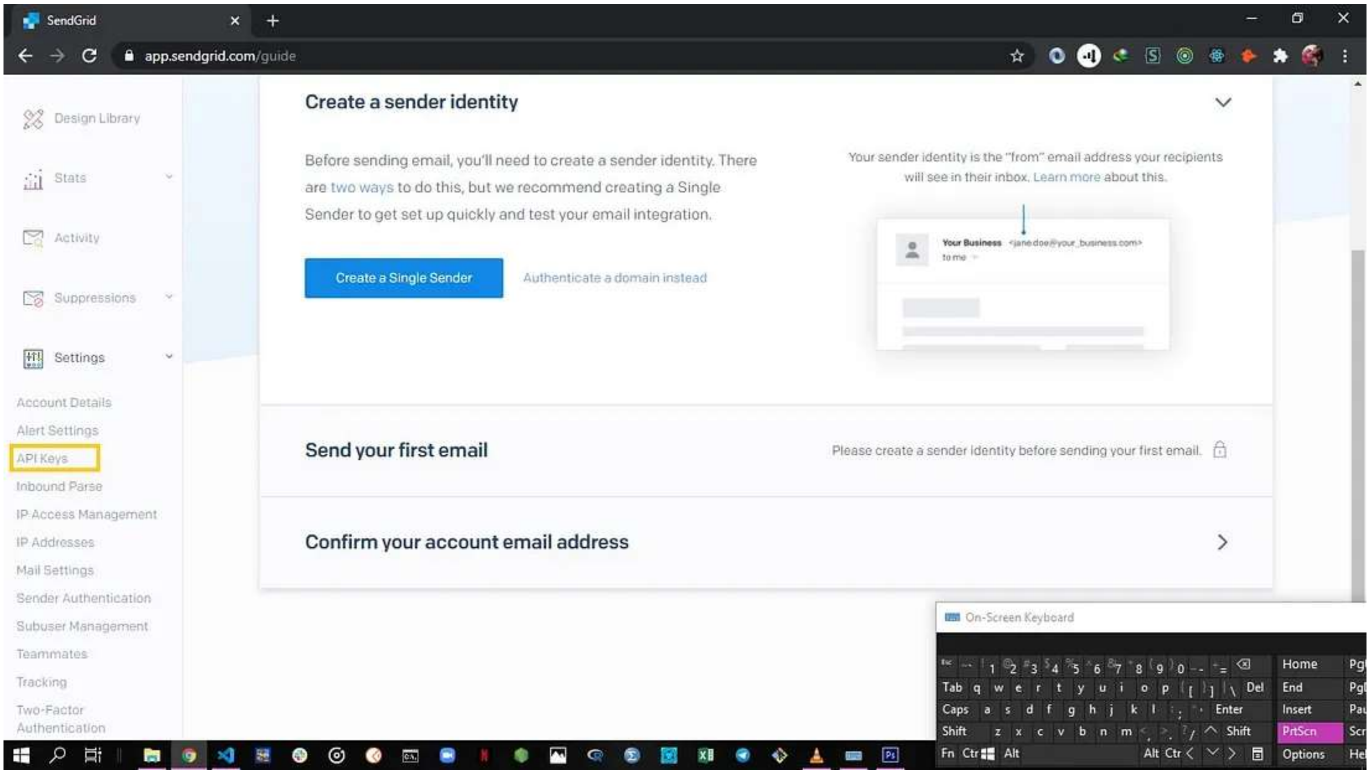
Before sending email, you'll need to create a sender identity. There are two ways to do this, but we recommend creating a Single Sender to get set up quickly and test your email integration.

Your sender identity is the "from" email address your recipients will see in their inbox. Learn more about this.

Send your first email

Please create a s

The settings tab will be expanded, click on API keys



Create a sender identity

Before sending email, you'll need to create a sender identity. There are two ways to do this, but we recommend creating a Single Sender to get set up quickly and test your email integration.

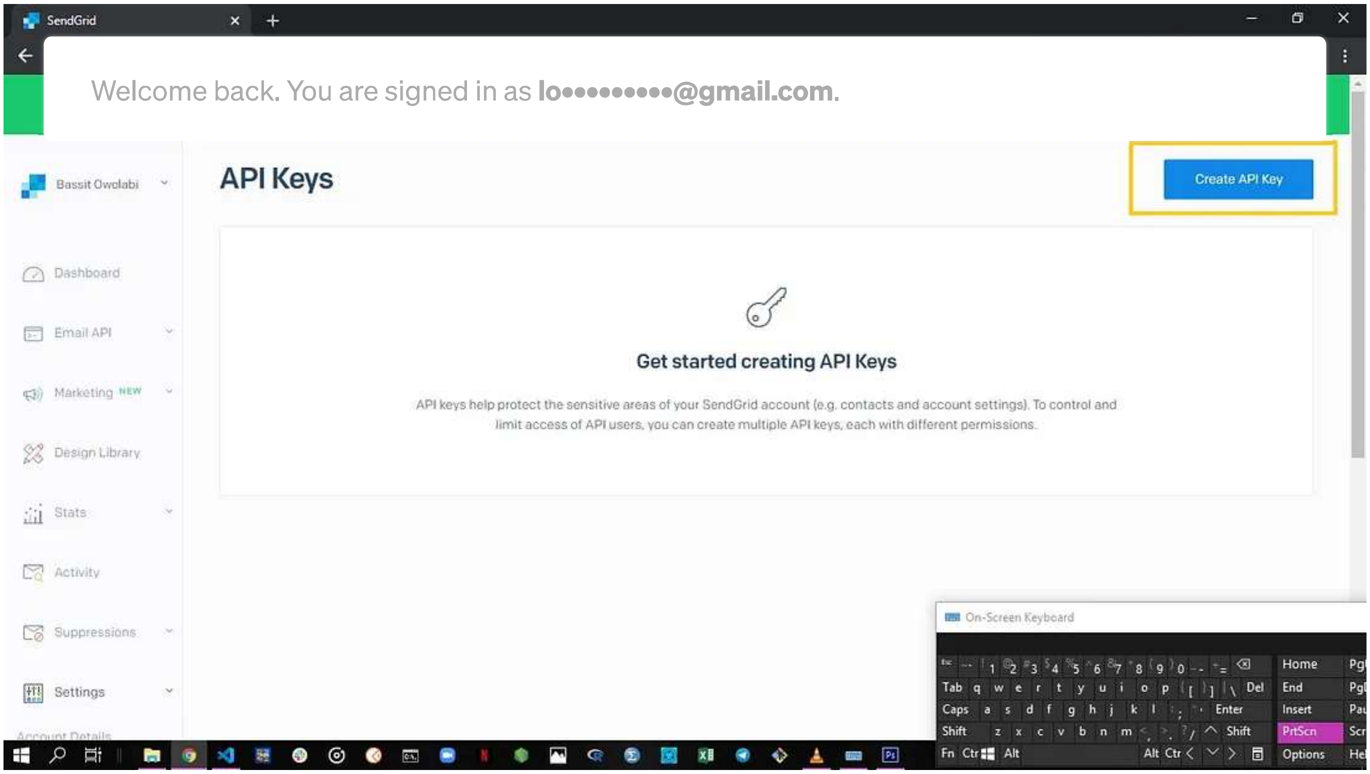
Your sender identity is the "from" email address your recipients will see in their inbox. Learn more about this.

Send your first email

Please create a sender identity before sending your first email.

Confirm your account email address

If you don’t have any, proceed to create a new API key



Welcome back. You are signed in as lo••••••••••@gmail.com.

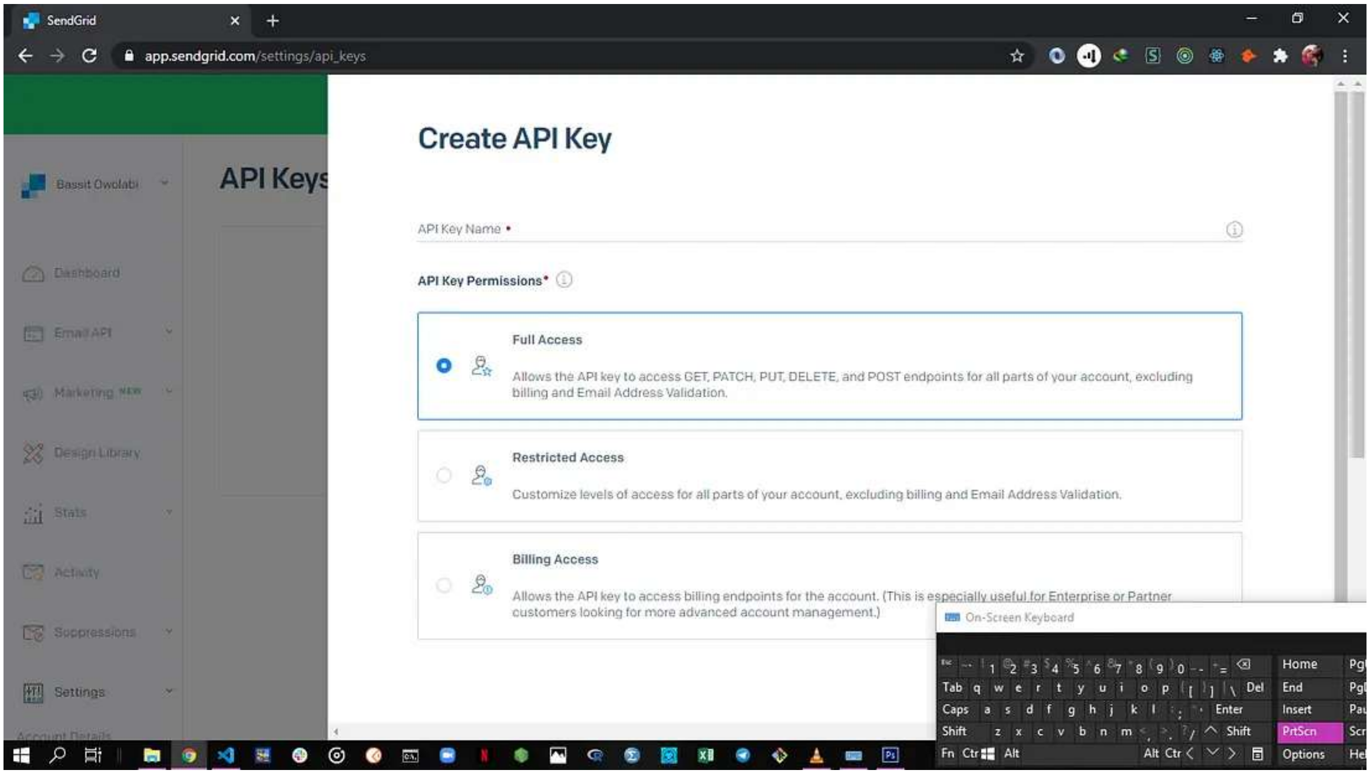
API Keys

Create API Key

Get started creating API Keys

API keys help protect the sensitive areas of your SendGrid account (e.g. contacts and account settings). To control and limit access of API users, you can create multiple API keys, each with different permissions.

Give it any name you desire and choose full access for API key permissions



app.sendgrid.com/settings/api_keys

Create API Key

API Key Name *

API Key Permissions*

● Full Access

Allows the API key to access GET, PATCH, PUT, DELETE, and POST endpoints for all parts of your account, excluding billing and Email Address Validation.

○ Restricted Access

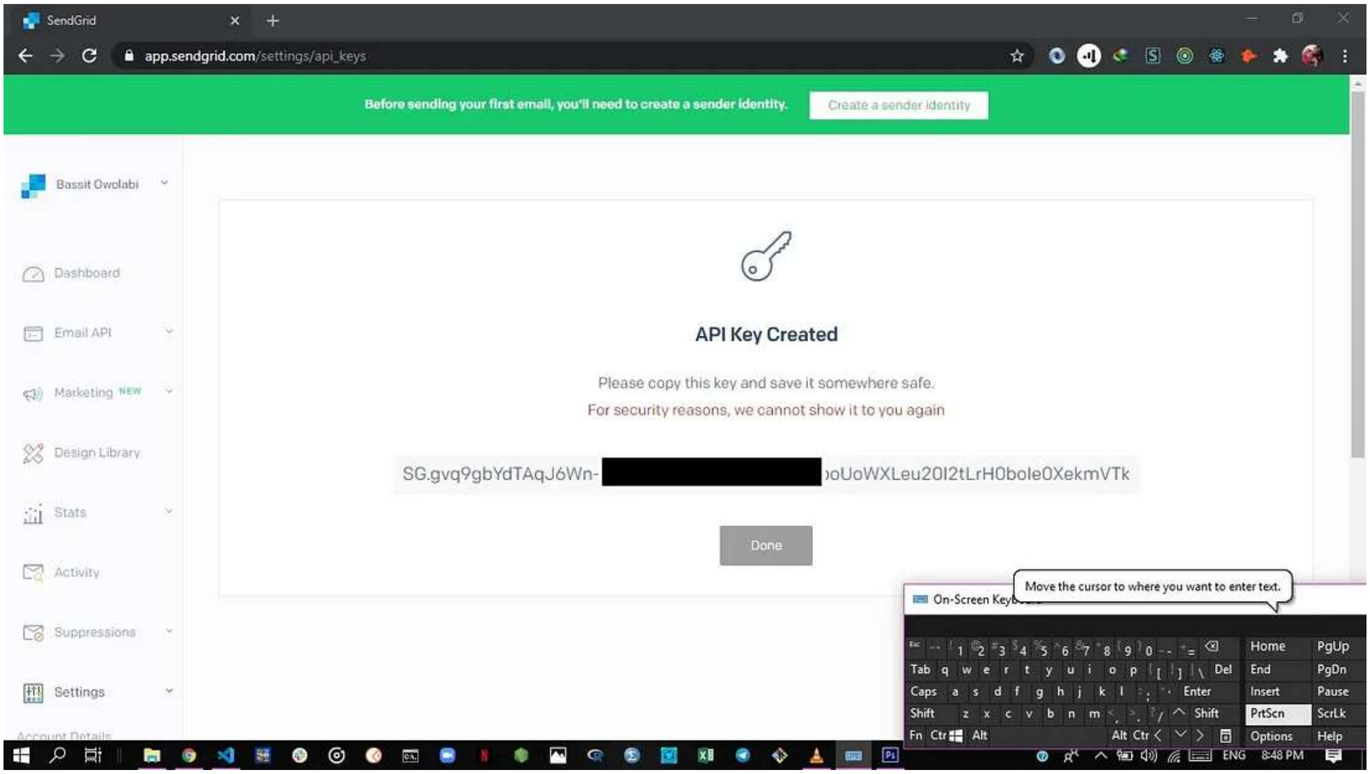
Customize levels of access for all parts of your account, excluding billing and Email Address Validation.

○ Billing Access

Allows the API key to access billing endpoints for the account. (This is especially useful for Enterprise or Partner customers looking for more advanced account management.)

Your API key will be generated. Copy and keep somewhere safe as you can

on



Before sending your first email, you'll need to create a sender identity.

API Key Created

Please copy this key and save it somewhere safe.
For security reasons, we cannot show it to you again

SG.gvq9gbYdTAqJ6Wn-[redacted]oUoWXLeu20I2tLrH0bole0XekmVTk

In the server side create a folder with the name "config", in this folder create a file with the name "keys.js". In this file paste this code

```
module.exports={
SENDGRID_API: 'YOUR API KEY'
}
```

Replace "YOUR API KEY" with your actual key. Don't forget to add this folder to your gitignore file. I would have used dotenv for improved security but I don't want this tutorial getting out of context.

2. Set-up client-side

Now let’s get to the coding part. The next step involves setting up the front-

en

isn’t a CSS tutorial I won’t be going over the styling aspect.

Create a new component and paste the following code in it.

```
import React,{useState} from ‘react’
const Message = ()=>{
const [name,setName] = useState(“”)
const [email,setEmail] = useState(“”)
const [message,setMessage] = useState(“”)
const [subject,setSubject] = useState(“”)
const PostData = ()=>{
fetch(“/send”,{
method:”post”,
headers:{
“Content-Type”:”application/json”
},
body:JSON.stringify({
name,
email,
subject,
message
})
}).then(res=>res.json())
.then(data=>{
alert(data.message)
setMessage(‘’)
setName(‘’)
```

```
setSubject(‘’)
```

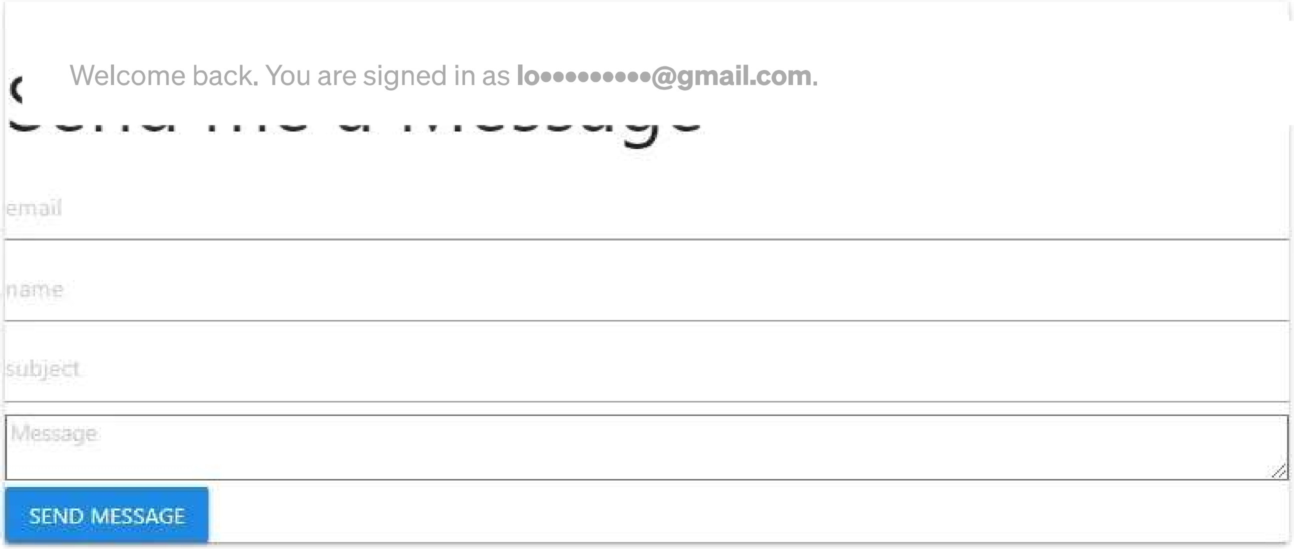
```
}).catch(err=>{

console.log(err)

})

}

return (

<div style={{maxWidth: ‘60%’, display:’flex’, flexDirection:
‘column’, alignItems:’center’}} className=”mycard”>

<div className=”card auth-card input-field”>

<h2>Send me a Message</h2>

<input

type=”text”

placeholder=”email”

value={email}

onChange={(e)=>setEmail(e.target.value)}

/>

<input

type=”text”

placeholder=”name”

value={name}

onChange={(e)=>setName(e.target.value)}

/>

<input

type=”text”

placeholder=”subject”

value={subject}

onChange={(e)=>setSubject(e.target.value)}

/>

<textarea

type=”text”
```

```
placeholder=”Message”
onChange={(e)=>setMessage(e.target.value)}
/>
<button className=”btn waves-effect waves-light #64b5f6 blue darken-1"
onClick={()=>PostData()}
>
Send Message
</button>
</div>
</div>
)
}
export default Message
```

Quick breakdown.

After the user fills the form, the onChange handler in each input tag updates the state with the value the user supplies. Once the Send message button is clicked, the ‘postData’ function is fired.

In the postData function we’re making a post request to the “/send” route on the server which we will be creating shortly. In this request, we’re sending all the data provided by the user to the server. After the data is successfully sent, we get back a response, alert the user with any information in the message object and clear all inputs.



output should look like this

3. **Configure Nodemailer**

Now that we are done with the client-side, we'll head over to our server side.

Open your terminal, navigate to the server folder and run these two commands to install nodemailer and sendgrid transport

```
npm install nodemailer nodemailer-sendgrid-transport
```

3.2 Import both packages and your api key into app.js file

```
const nodemailer = require(‘nodemailer’)

const sendGridTransport = require(‘nodemailer-sendgrid-transport’);
const {SENDGRID-API} = require('./config/keys');
```

3.3 Create a transporter for sendgrid

This is where we’ll need the API key.

```
const transporter = nodemailer.createTransport(sendGridTransport({
auth:{
api_key:SENDGRID_API
}
}))
```

3.3 Create post route

This will handle the request made from the client-side.

replace ‘YOUR EMAIL’ with the mail you want the message delivered to

```
app.post(‘/send’, (req, res) => {
const { name, email, message, subject } = req.body
transporter.sendMail({
to:’YOUR EMAIL’,
from: email,
subject:subject,
html:`<h3>${name}</h3>
<p>${message}</p>`
}).then(resp => {
res.json({resp})
})
.catch(err => {
```

```
console.log(err)
```

```
})
```

Your app.js file should look like this.

```
const express = require('express')
const app = express()
const nodemailer = require('nodemailer')
const sendGridTransport = require('nodemailer-sendgrid-transport');
const {SENDGRID_API} = require('./config/keys')
const PORT = process.env.PORT || 5000
app.use(express.json())
const transporter = nodemailer.createTransport(sendGridTransport({
auth:{
api_key:SENDGRID_API
}
}))
app.post('/send', (req, res) => {
const { name, email, message, subject } = req.body
transporter.sendMail({
to:'YOUR EMAIL',
from: email,
subject:subject,
html:`<h3>${name}</h3>
<p>${message}</p>`
}).then(resp => {
res.json({resp})
})
.catch(err => {
})
})
app.listen(PORT,()=>{
console.log(“server is running on”,PORT)
})
```

4. **Send E-mail**

Start both the client server and the node server.

Fill all inputs and click “send message”, your message should be delivered instantly.

If you run into a sendgrid sender verification error, head over to your sendgrid account, go to settings => Sender Authentication. You can then verify a single sender or verify your domain instead.

The source code for this tutorial can be found on my **Github profile**.

React Nodemailer Sendgrid Nodejs E Mail

Welcome back. You are signed in as **lo••••••••••@gmail.com**.

Written by Bassit Owolabi

Follow

156 Followers · Writer for The Startup

A NodeJs developer.

More from Bassit Owolabi and The Startup

Bassit Owolabi in Geek Culture

How I built an E-commerce API with NodeJs, Express and MongoDB

Part 1: Setting up the project

5 min read · Dec 13, 2021

155 1

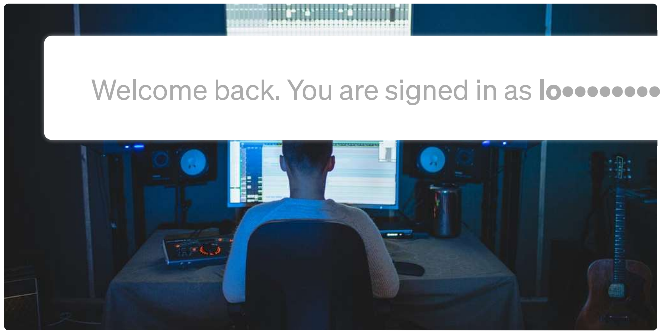
Maryam Merchant in The Startup

10 Habits That Are Damn Hard to Do, But Pay off Forever

Master it and you will find your way to your goals

✦ · 8 min read · Jan 4, 2024

4.6K 76



Welcome back. You are signed in as **lo••••••••••@gmail.com**.

Martina D. in The Startup

11 Companies Making $1M+ That Are [Practically] Bedroom Businesses

Time to feed the ideation part of your brain with some real treats.

✦ · 7 min read · Dec 31, 2023

3.2K 46

Bassit Owolabi in Geek Culture

How I built an E-commerce API with NodeJs, Express and MongoDB(Par...

Part 4: Completing routes

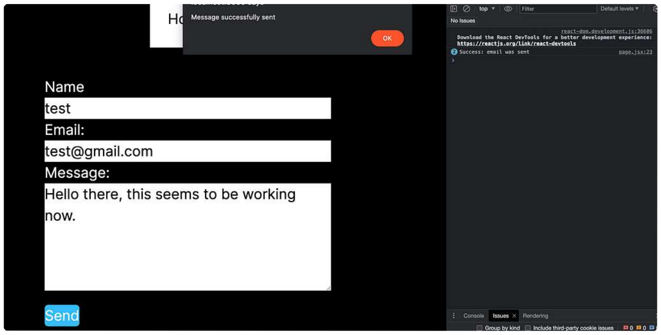
5 min read · Dec 13, 2021

129 6

See all from Bassit Owolabi

See all from The Startup

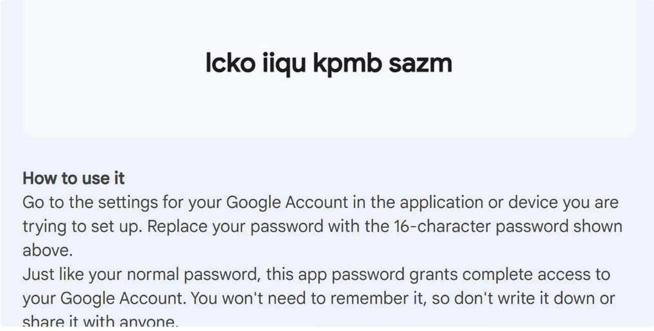
Recommended from Medium



Message successfully sent

Name
test
Email:
test@gmail.com
Message:
Hello there, this seems to be working now.

```
react-dom.development.js:36606
Download the React DevTools for a better development experience: https://reactjs.org/link/react-devtools
2 Success: email was sent                                  page.jsx:23
```



Icko iiqu kpmb sazm

How to use it

Go to the settings for your Google Account in the application or device you are trying to set up. Replace your password with the 16-character password shown above.

Just like your normal password, this app password grants complete access to your Google Account. You won't need to remember it, so don't write it down or share it with anyone.

How to send email forms using Next.js 13 with nodemailer

This article is to help anyone unsure of how to implement an email form using Next.js 13…

6 min read · Sep 18, 2023

66 1

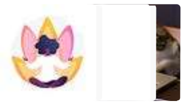
How to Send an Email from Your Gmail Account with Nodemailer

Introduction

3 min read · Oct 5, 2023

65 1

Lists

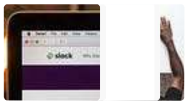


Stories to Help You Grow as a Software Developer

19 stories · 809 saves

General Coding Knowledge

20 stories · 916 saves

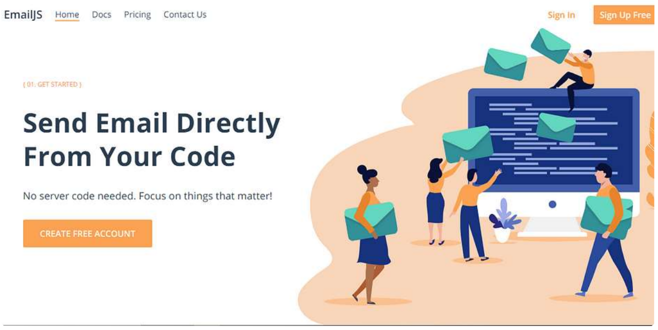


Productivity

237 stories · 323 saves

data science and AI

40 stories · 71 saves



[01. GET STARTED]

Send Email Directly From Your Code

No server code needed. Focus on things that matter!

CREATE FREE ACCOUNT

Thomas Augot

Create a React Contact Form with Email.js

In this tutorial, I'll walk you through creating a contact form in a React application and…

2 min read · Nov 1, 2023

Dushan Senadheera

Send emails using NodeJS

In this tutorial let's see how to send a text or html template email with file attachments…

3 min read · Sep 5, 2023

Welcome back. You are signed in as **lo••••••••••@gmail.com**.

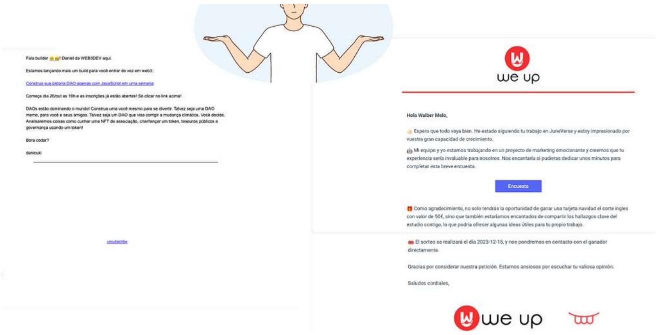
Asitha Muthumala

Forgot And Reset Password Using NodeJS (With Email OTP)

The “Forgot Password” function in an application is a crucial feature that allows…

6 min read · Sep 25, 2023

118 1



Walber Melo

Building Efficient Email templates with MJML and Node.JS — Send it t…

Introduction

10 min read · Dec 7, 2023

4

See more recommendations