**<https://medium.com/@y.mehnati_49486/how-to-send-an-email-from-your-gmail-account-with-nodemailer-837bf09a7628>**

**How to Send an Email from Your Gmail Account with Nodemailer**

Introduction

[[Yousef Mehnati](https://medium.com/@y.mehnati_49486?source=post_page-----837bf09a7628--------------------------------)](https://medium.com/@y.mehnati_49486?source=post_page-----837bf09a7628--------------------------------)

[Yousef Mehnati](https://medium.com/@y.mehnati_49486?source=post_page-----837bf09a7628--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2Fbf551fb0897d&operation=register&redirect=https%3A%2F%2Fmedium.com%2F%40y.mehnati_49486%2Fhow-to-send-an-email-from-your-gmail-account-with-nodemailer-837bf09a7628&user=Yousef+Mehnati&userId=bf551fb0897d&source=post_page-bf551fb0897d----837bf09a7628---------------------post_header-----------)

3 min read

·

Oct 5, 2023

43

Sending emails programmatically is a common task in web development, and Nodemailer is a popular Node.js library that simplifies this process. In this tutorial, we’ll walk through the steps to send an email from your Gmail account using Nodemailer. Whether you want to send notifications, newsletters, or automate email-related tasks, Nodemailer is a reliable choice.

**Prerequisites**

Before we begin, make sure you have the following prerequisites in place:

1. Node.js and npm (Node Package Manager) installed on your machine.
2. A Gmail account with two-step verification enabled.
3. An “App Password” generated for your Gmail account. This is required because Gmail does not allow the use of your regular password for third-party apps.

**Getting Started**

1. **Create a New Node.js Project**

Start by creating a new directory for your project and navigate to it in your terminal.

mkdir nodemailer-gmail  
cd nodemailer-gmail

Initialize a new Node.js project using the following command:

npm init -y

**2. Install Nodemailer**

Now, let’s install the Nodemailer package. Run the following command:

npm install nodemailer

**3. Import Nodemailer**

In your project directory, create a JavaScript file (e.g., sendEmail.js) and import Nodemailer:

const nodemailer = require("nodemailer");

**4. Configure Nodemailer**

Set up the transporter to use your Gmail account. Replace your\_email@gmail.com and your\_app\_password with your Gmail address and the generated App Password:

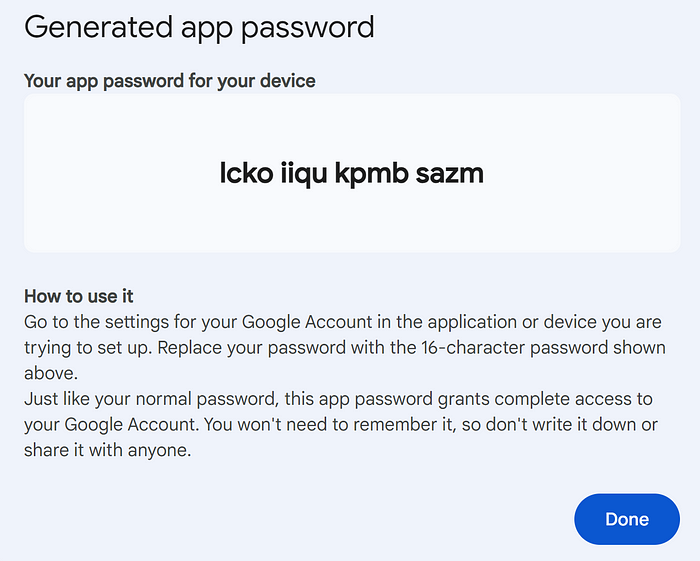
const transporter = nodemailer.createTransport({  
 service: "Gmail",  
 host: "smtp.gmail.com",  
 port: 465,  
 secure: true,  
 auth: {  
 user: "your\_email@gmail.com",  
 pass: "your\_app\_password",  
 },  
});

**5. Get the app password from your Gmail account**

To get an App Password for your Gmail account, follow these detailed steps:

Note: You’ll need to have **two-step verification** enabled on your Gmail account before generating an App Password. If you haven’t enabled it, do so first by going to your Google Account settings.

1. Access Your Google Account:
2. Start by visiting the Google Account management page. You can do this by navigating to <https://myaccount.google.com/>.
3. **Sign In**: Sign in to the Google Account associated with the Gmail address you want to use for sending emails programmatically.
4. **Security**: In the left sidebar, click on “Security.”
5. Scroll down to How you sign in to google and click on 2-step verificaiton.
6. **App Passwords:** Scroll down to “App passwords.” Click on “App passwords.” You may be prompted to re-enter your password for security purposes.
7. **App name**: Enter a custom name for this App Password. It helps you identify it later, so choose something related to the application or use case where you plan to use this App Password.
8. **Create**: Click the “Create” button. Google will create a unique 16-character App Password for your custom application/device.



Generated app password

Once generated, Google will display the App Password on the screen. **Important: This is the only time you’ll see this password. Make sure to save it securely because you won’t be able to view it again. You’ll use this App Password in your application code to authenticate with Gmail’s SMTP server.**

In your Node.js application (or whichever application you’re using to send emails), replace your regular Gmail account password with this App Password. This App Password will act as your Gmail password for the specific application or device you generated it for.

**6. Compose and Send the Email**

Now, you can compose and send your email. Here’s an example of sending a simple email.

const mailOptions = {  
 from: "your\_email@gmail.com",  
 to: "recipient@example.com",  
 subject: "Hello from Nodemailer",  
 text: "This is a test email sent using Nodemailer.",  
};

transporter.sendMail(mailOptions, (error, info) => {  
 if (error) {  
 console.error("Error sending email: ", error);  
 } else {  
 console.log("Email sent: ", info.response);  
 }  
});

Replace "recipient@example.com" with the recipient's email address.

**7. Run the Script**

Save your changes and run the script using the following command:

node sendEmail.js

If everything is set up correctly, you should see a success message indicating that the email has been sent.

**Conclusion**

In this tutorial, we’ve covered how to send an email from your Gmail account using Nodemailer. You’ve learned how to configure Nodemailer, create a transporter, compose an email, and send it programmatically. With this knowledge, you can automate various email-related tasks in your Node.js projects. Happy coding!