

Sunday, 8 October 2023

Nodemailer

Nodemailer is the npm module that allows us to send the emails with the help of nodes. Sometimes we have requirements to send the automatic emails to all the users like welcome emails, promotions emails, OTP emails. So in this case we can use the Nodemailer.

Nodemailer is a popular Node.js library for sending emails. It simplifies the process of sending email messages through various email services and providers. Here are some key notes on using Nodemailer in Node.js

Nodemailer just act as a transporter that will send the emails by using your account only. It just act as transporter on the behalf of your account and send the mail to receipts.

Installation

You can install Nodemailer using npm (Node Package Manager):

bash

 Copy code

```
npm install nodemailer
```

Basic Usage

1. Import Nodemailer:

To use Nodemailer in your Node.js application, require the `'nodemailer'` module:

javascript

 Copy code

```
const nodemailer = require('nodemailer');
```

2. Create a Transporter:

You need to create a transporter object that defines the email service and credentials:

javascript

 Copy code

```
const transporter = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    user: 'your-email@gmail.com',
    pass: 'your-password',
  },
});
```

3. Compose and Send an Email:

To send an email, you can use the `sendMail` method of the transporter:

```
javascript Copy code

const mailOptions = {
  from: 'your-email@gmail.com',
  to: 'recipient@example.com',
  subject: 'Hello, Nodemailer!',
  text: 'This is a test email sent from 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.

Advanced Usage

1. HTML Emails:

You can send HTML-formatted emails by setting the `html` property in the `mailOptions` object:

```
javascript Copy code

const mailOptions = {
  from: 'your-email@gmail.com',
  to: 'recipient@example.com',
  subject: 'HTML Email Example',
  html: '<p>This is an HTML email sent from Nodemailer.</p>',
};
```

2. Attachments:

Nodemailer allows you to attach files to your email using the `attachments` property:


```
javascript Copy code

const mailOptions = {
  from: 'your-email@gmail.com',
  to: 'recipient@example.com',
  subject: 'Email with Attachment',
  text: 'Check out this attachment!',
  attachments: [
    {
      filename: 'example.txt',
      content: 'This is the content of the attachment.',
    },
  ],
};
```

3. Using Promises:

Nodemailer supports both callbacks and Promises. You can use Promises for cleaner and more readable code:

javascript

 Copy code

```
transporter.sendMail(mailOptions)
  .then((info) => {
    console.log('Email sent: ', info.response);
  })
  .catch((error) => {
    console.error('Error sending email: ', error);
  });
```

Email Services

Nodemailer supports various email services and transport options beyond Gmail. You can configure it to work with SMTP, SendGrid, Mailgun, and more.