



Jimmy Alvarez Mendoza

NodeJs

Andrey Alfaro

Nodemailer

Repositorio

<https://github.com/JgeovanniAm/ContactMe>

Introducción



Alguna vez te has preguntado, ¿Cómo enviar un Email con NodeJs?

Es muy importante aprender cómo poder hacerlo, ya que muchas de las páginas que usamos las usamos como medio de comunicación con nuestros usuarios. Y Cuando estás desarrollando el backend en un servicio web, una de las funcionalidades más útiles y recurrentes suele ser el envío automático de emails.

Hay muchas maneras de enviar correos y maneras más eficaces para ello.

Una manera es realizarlo con nodejs.

En este caso usaremos nodejs para enviar emails, aprovechando su rapidez y facilidad.

Hay muchos plugins o paquetes que nos ayudarán a cumplir con nuestra tarea, Por ejemplo uno de ellos es:



Mailgun:

La API Mailgun está construida en HTTP. Es una API RESTful. Utiliza URL predecibles orientadas a los recursos.

Utiliza capacidades HTTP incorporadas para pasar parámetros y autenticación.

Responde con códigos de respuesta HTTP estándar para indicar errores.

Se puede utilizar bibliotecas o bibliotecas HTTP / REST para su lenguaje de programación favorito, para realizar llamadas HTTP a Mailgun.

Nodemailer:



Nodemailer es un módulo para aplicaciones Node.js que permite el envío de correos electrónicos de manera fácil, hoy en día es la solución a la que la mayoría de los usuarios de Node.js recurren por defecto.

Nos permitirá añadir estas funcionalidades en nuestro servidor Node.js.

Para este proyecto usaremos nodemailer ya que nos da muchas ventajas como rapidez y es muy completa si hablamos de documentación y va muy bien con nodejs.

Instalación:

- Se hace una referencia al módulo nodemailer de nodejs.

```
npm install nodemailer
```

- Definimos un controlador *mail Ctrl.js* que exporte un método *sendEmail*.

```
const nodemailer = require('nodemailer');  
function exports.sendEmail = (req, res) => {  
  nodemailer stuff will go here }  
}
```

- Definimos un transportador de la siguiente manera, sustituyendo los datos por los de nuestra cuenta gmail.

```
const transporter = nodemailer.createTransport ( {  
  service: 'Gmail',  
  auth: {  
    user: 'geovanniam171078@gmail.com',  
    pass: 'patofeo'  
  }  
});
```

- Ahora que ya tenemos el transporter, definiremos el propio email:

```
const mailsend = {  
  from: 'tucorreo@gmail.com',  
  to: 'mi-amigo@yahoo.com',  
  subject: 'Asunto Del Correo',  
  text: 'mensaje'  
};
```

- Ahora le “decimos o comunicamos” el “transporter” con el propio email y enviamos el email.

```
transporter.sendMail(mailOptions, function(error, info){  
  if (error){  
    console.log(error);  
    res.send(500, err.message);  
  } else {  
    console.log("Email sent");  
    res.status(200).jsonp(req.body);  
  }  
});
```

Requerimientos.



Los únicos requerimientos son : Si puede ejecutar Node.js versión 6 o posterior, puede usar Nodemailer.

No hay requisitos específicos de plataforma o recurso. Todos los métodos públicos de Nodemailer admiten devoluciones de llamada y Promesas (si se omite la devolución de llamada). Debe tener al menos el Nodo v8.0.0 si desea usar asíncrono.

¿Porque usar Nodemailer?

Por qué deberíamos de usar Nodemailer para enviar correos automáticos en nuestra webs?

Todos podemos tener opiniones diferentes al respecto pero algunas de estas son :

- No usa dependencias es un único módulo:
A veces se vuelve engorroso tener muchas dependencias y que afecte un trabajo por eso mismo.
- Gran enfoque en la seguridad lo cual nos da mucha confianza.

- Compatibilidad con Windows:
Puede instalarlo con npm en Windows como cualquier otro módulo, no hay dependencias compiladas. Úselo sin problemas desde Azure o desde su caja de Windows.
- Utiliza contenido HTML:
Es una alternativa de uso de texto sin formato
- Se puede agregar archivos adjuntos a los mensajes.
- Entrega segura de correo electrónico utilizando TLS / STARTTLS.
- Tiene diferentes métodos de transporte además del soporte de SMTP incorporado.
- Soporte de plugin personalizados para manipular mensajes.
- Proxies para conexiones SMTP
- Cuentas de prueba de correo electrónico autogeneradas de Ethereum.email

Recursos.

Mailgun para Wordpress

<https://www.mailgun.com/>

Nodemailer

<https://nodemailer.com/usage/why-smtp/>

github

<https://github.com/nodemailer/nodemailer-amqp-example>

<https://github.com/nodemailer/nodemailer>

Conclusión.

Es importante darse cuenta de tecnologías como Nodemailer ya que existen para facilitarnos la vida y crecer como programadores en el ámbito laboral, conocer nuevas cosas y superarse a sí mismo día con día.

Muchas de estas herramientas también vienen a cubrir o brindar soluciones a las malas prácticas que aún se siguen dando, como por ejemplo;

Poner un `<p/>` o un `<h1/>` como texto del correo, exponiéndose en riesgos de privacidad o muchas de estos casos ser víctima de un spam hecha o producida por un Bot inteligente, hasta miedo me provoca pensarlo!

¡Siempre hay que dar un paso o una milla extra!

