Node y el gestor de paquetes

Protocolo SMTP y consulta HTTP

{desafío} latam_

Aplicar procedimiento de puesta en ejecución de una aplicación Node.js implementando mecanismos para la detección de errores durante la ejecución.

- Unidad 1:
  Introducción a Node
- Unidad 2:
  Node y el gestor de paquetes
- Unidad 3:
  Persistencia

Te encuentras aquí

{desafío} latam_

¿Qué aprenderás en esta sesión?

- *Reconocer el uso del protocolo SMTP para el envío de correos electrónicos.*
- *Construir un servidor que procese el envío de un correo electrónico a partir de una consulta HTTP*

¿Sabes lo que es el protocolo SMTP?

{desafío} latam_

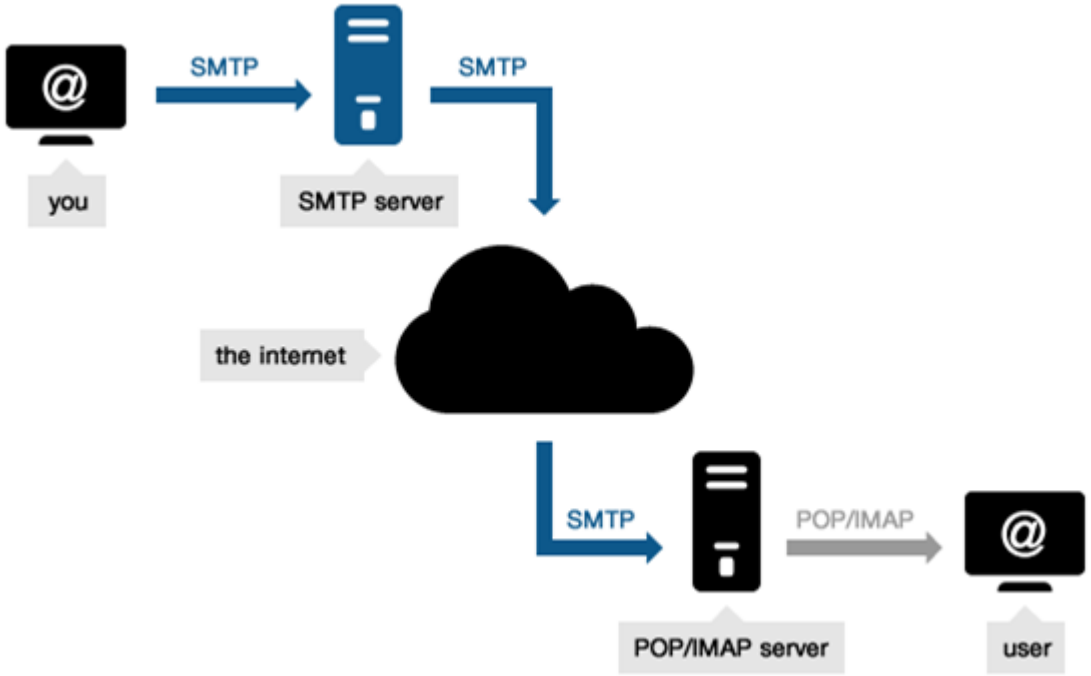
/* El protocolo SMTP */

El protocolo SMTP

Por sus siglas, Simple Mail Transfer Protocol (Protocolo simple de transferencia de correo) es un protocolo usado para el envío de correos electrónicos, bajo simples parámetros o propiedades que indican declarativamente la información expuesta en el correo, la información correspondiente al origen y destino del correo. Este protocolo por supuesto utiliza el internet como medio de transmisión, por lo que no podría ejecutarse si no se está conectado de forma online.

El protocolo SMTP

Es importante entender cómo funciona el protocolo SMTP y para esto te muestro el proceso diagramado en la siguiente imagen:



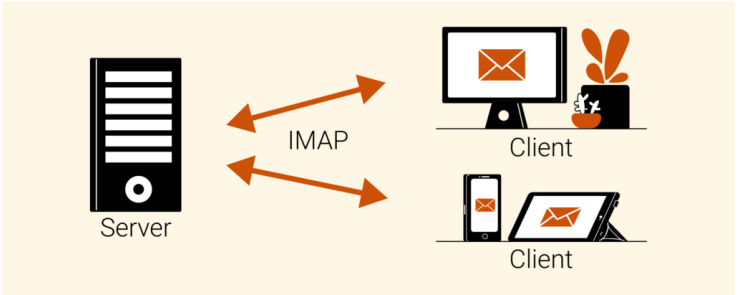
Fuente: www.serversmtp.com

/* IMAP , POP */

IMAP

- Almacena el correo en el servidor del cliente, por lo que se puede acceder a él desde cualquier dispositivo, ya sean teléfonos, ordenadores o tablets. Los correos electrónicos no se descargan si no haces clic en ellos y los archivos adjuntos no se descargan automáticamente, lo que te permite consultar tus mensajes rápidamente desde cualquier dispositivo.

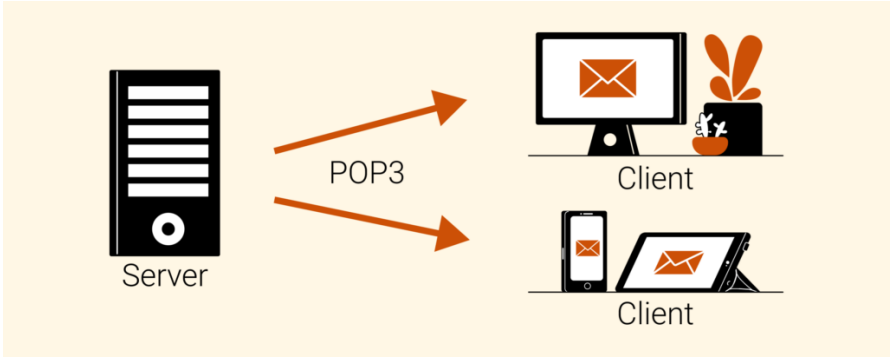
- Si accedes desde varios dispositivos y quieres que el correo electrónico y los archivos adjuntos se almacenen en el servidor del cliente, usa IMAP. (Fuente: G Suite)



Fuente: mailbord.com

POP

- Se conecta al servidor del cliente, descarga todos los mensajes nuevos en un solo dispositivo y luego elimina el correo del servidor. Si lees los correos electrónicos en un único dispositivo y quieres que se eliminen del servidor después de descargarlos en tu dispositivo, usa POP. (Fuente: G Suite)



Fuente: mailbord.com

/* Servidor para correos electrónicos */

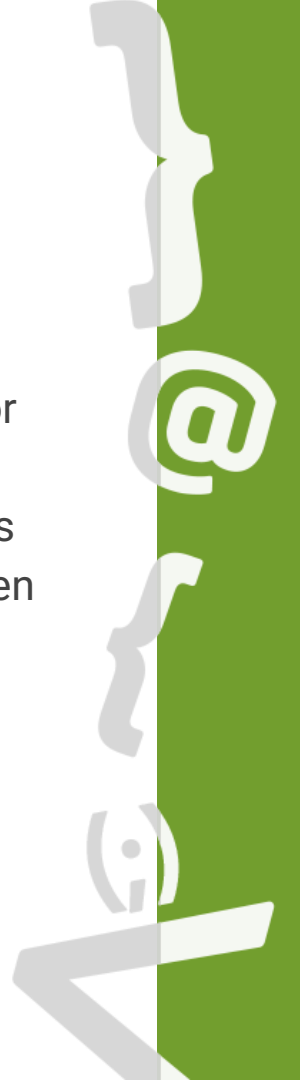
Demostración "Servidor para correos electrónicos"

{desafío} latam_

Ejercicio guiado

Servidor para correo electrónico

Crear un servidor que reciba la información que se enviará en un correo desde el cliente. Inicialmente por medio de una consulta que realizaremos en el navegador con una URL que incluya los parámetros que capturaremos en el servidor, esta consulta próximamente será emitida por un formulario HTML, el cual recordemos por defecto emite una consulta GET con los parámetros escritos por un usuario en los inputs.



Ejercicio guiado

Servidor para correo electrónico

Ya que tenemos la lógica para el envío de correos, pasa ese código a un archivo llamado “mailer.js” y sigue los siguientes pasos para hacerle las modificaciones pertinentes:

- **Paso 1**: Crear una función llamada “enviar” que reciba como parámetro los valores de “to”, “subject” y “text”.

- **Paso 2**: Exportar la función enviar como un módulo.

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

// Paso 1
function enviar(to, subject, text) {
 let transporter = nodemailer.createTransport({
   service: 'gmail',
   auth: {
     user: 'nodemailerADL@gmail.com',
     pass: 'vamoscontodo',
   },
 })

 let mailOptions = {
   from: 'nodemailerADL@gmail.com',
   to,
   subject,
   text,
 }

 transporter.sendMail(mailOptions, (err, data) => {
   if (err) console.log(err)
   if (data) console.log(data)
 })
}

// Paso 2
module.exports = enviar
```

{desafío} latam_

Ejercicio guiado

Servidor para correo electrónico

Teniendo entonces este archivo listo, sigue los siguientes pasos para la lógica del servidor en el archivo index.js

- **Paso 1:** Importar la función enviar del archivo mailer.js.
- **Paso 2:** Ya que la idea es recibir los valores desde el cliente en una consulta HTTP, importa el módulo url para poder extraer los parámetros de la query strings.
- **Paso 3:** Importar el módulo http para crear el servidor.

Ejercicio guiado

Servidor para correo electrónico

- **Paso 4:** Crear el servidor con el método createServer del módulo http.
- **Paso 5:** Guardar en variables los parámetros “para”, “asunto” y “contenido”.
- **Paso 6:** Crear la ruta raíz del servidor.
- **Paso 7:** Ejecutar la función enviar importada del archivo mailer.js pasándole como argumento los parámetros recibidos de la consulta

```
// Paso 1
const enviar = require('./mailer')
// Paso 2
const url = require('url')
// Paso 3
const http = require('http')
// Paso 4
http
 .createServer(function (req, res) {
   // Paso 5
   let { para, asunto, contenido } = url.parse(req.url,
true).query
   // Paso 6
   if (req.url.startsWith('/')) {
     // Paso 7
     enviar(para, asunto, contenido)
   }
 })
 .listen(3000)
```

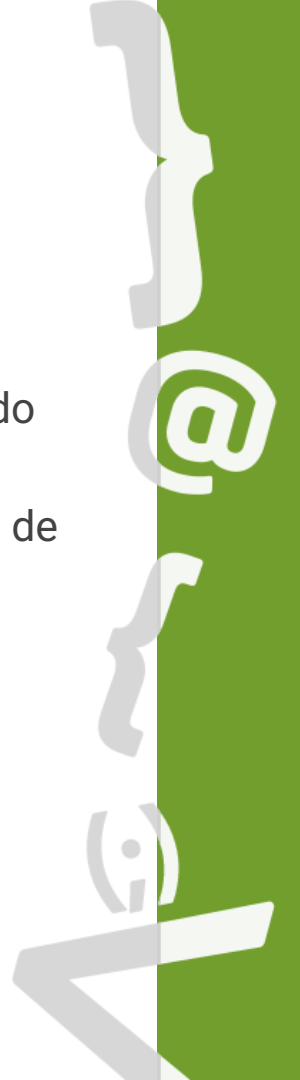
Ejercicio
“Practiquemos lo aprendido”

{desafío}
latam_

Ejercicio propuesto

Apliquemos lo aprendido

Basado en el ejercicio de envío de correo con una consulta HTTP, desarrolla un servidor que disponibilice una ruta “enviar” que envíe un correo electrónico basado en los parámetros recibidos en la consulta y le devuelva al cliente un mensaje diciendo “Se procedió a intentar mandar el correo electrónico… revise su bandeja de entrada para confirmar que se haya enviado.”



Ejercicio propuesto

Solución

```
const enviar = require('./mailer')
const url = require('url')
const http = require('http')
http
 .createServer(function (req, res) {
   let { para, asunto, contenido } =
url.parse(req.url, true).query

   if (req.url.startsWith('/enviar')) {
     enviar(para, asunto, contenido)
     res.end(
       'Se procedió a intentar mandar el correo
electrónico... revise su bandeja de entrada para
confirmar que se haya enviado.'
     )
   }
 })
 .listen(3000)
```

¿Cómo crees que la comprensión de estos conceptos puede contribuir a mejorar la experiencia del usuario en el uso de servicios de correo electrónico?

Próxima sesión...

- *Reconocer los parámetros básicos del paquete Yargs para crear una interfaz de línea de comando*
- *Implementar una interfaz de línea de comando con el paquete Yargs para levantar una aplicación Node*

{desafío}
latam_

Academia de
talentos digitales