

FIC 2 – Módulo II

Aula 12 – Enviando e-mail com o Nodemailer

Criação de páginas
customizadas com Node.js

Leonardo Dalvi Pio

Enviando e-mail com o Nodemailer

- Enviando e-mail com o módulo *Nodemailer*.
- Utilização do *Mailtrap* (<https://mailtrap.io>) para testes.
- No *Replit*, se o pacote/recurso (***pagkage.json***, ***express***, ***nodemailer***) não estiver disponível, ele baixará automaticamente.

Enviando e-mail com o Nodemailer

- No *Replit*, crie uma pasta nomeada "Aula 12".
- Dentro da pasta, crie um novo arquivo (New Repl). Selecione o Template 'Node.js' e atribua o nome do arquivo como: "email.js" (arquivo do tipo .js).

Enviando e-mail com o Nodemailer

- Iniciaremos criando as constantes e importando os módulos. Criaremos o módulo 'express' e 'nodemailer', utilizando a porta padrão = (443) com o comando 'require()'.

```
const express = require('express')
const app = express()
const nodemailer = require('nodemailer')
const porta = 443
```

- Utilizando o comando 'get', na raiz ('/'), criaremos uma *Arrow Functions* informando uma resposta com: "Enviando e-mail com o Nodemailer!"

```
app.get('/', (req, res) => {
  res.send('Enviando e-mail com o Nodemailer!')
})
```


Enviando e-mail com o Nodemailer

- Criaremos uma rota para enviar o e-mail utilizando o código com *Arrow Functions*:

```
app.get("/sendemail", async (req, res) => {  
  
});
```

- Utilizaremos o comando 'get' para conseguir o acesso via *Browser*.
- Acessando o URL '/sendemail' enviaremos o e-mail depois de configurado.



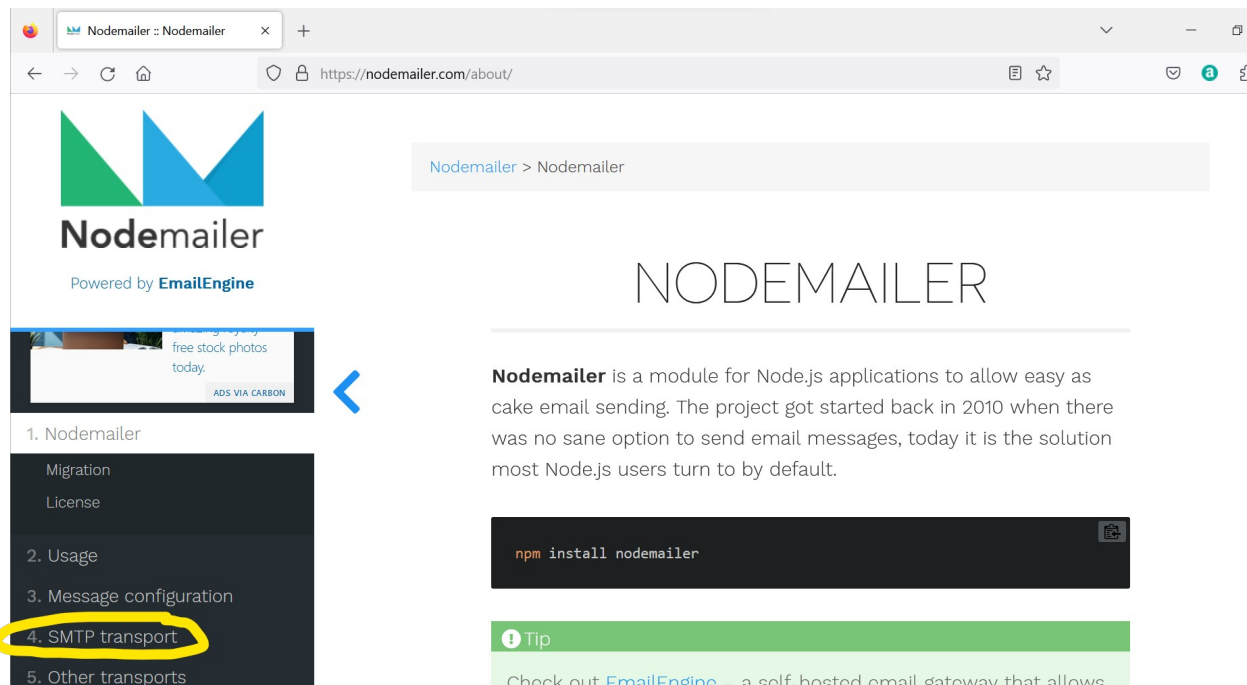
Enviando e-mail com o Nodemailer

- Organizaremos nosso código em:
 - Declaração de constantes
 - Criação da rota (envio do e-mail)
 - Transporte - *transport*
 - Mensagem (corpo)
 - Mensagem (envio ou erro) - `'sendMail'`



Criação da rota - Transporte

- Na criação da rota para envio do e-mail, começaremos com a criação do Transporte. Utilizaremos o módulo 'nodemailer'.
- Acessaremos o link: <https://nodemailer.com/>, menu esquerdo em '4. SMTP transport'.



Criação da rota - Transporte

- Utilizaremos o exemplo da página apresentada, conforme código abaixo:

Examples

- Utilizaremos a função
- 'createTransport' do
- módulo 'nodemailer'.

1. Single connection

This example would connect to a SMTP server separately for every single message

- Observe que teremos
- o host, a porta, as
- credenciais do usuário.

```
nodemailer.createTransport({  
  host: "smtp.example.com",  
  port: 587,  
  secure: false, // upgrade later with STARTTLS  
  auth: {  
    user: "username",  
    pass: "password",  
  },  
});
```


Criação da rota - Transporte

- Copiaremos o código do exemplo da página:
- *nodemailer.createTransport({*
- *host: "smtp.example.com",*
- *port: 587,*
- *secure: false, // upgrade later with STARTTLS*
- *auth: {*
- *user: "username",*
- *pass: "password",*
- *},*
- *});*



Criação da rota - Transporte

- Iremos colar no nosso código em:

```
app.get("/sendemail", async (req, res) => {  
  
  nodemailer.createTransport({  
    host: "smtp.example.com",  
    port: 587,  
    secure: false, // upgrade later with STARTTLS  
    auth: {  
      user: "username",  
      pass: "password",  
    },  
  });  
});
```

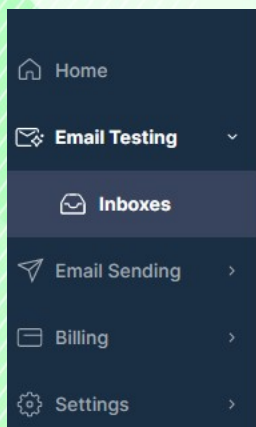


Criação da rota - Transporte




- Como é um código copiado é um modelo/padrão, teremos que testar com uma alternativa que funcione. Utilizaremos o *mailtrap*, disponível em: <https://mailtrap.io/>, para criar uma conta de e-mail “fake” para testar o envio.
- Acesse o site e realize o seu cadastro (e/ou faça login). Você terá que acessar seu e-mail (informado no cadastro) para confirmar o acesso ao *mailtrap*.
- Após a confirmação, acesse o link: <https://mailtrap.io/home>.
- No menu (lado esquerdo) teremos a opção: ‘Email Testing’. Selecione-a. Por padrão, teremos um projeto com uma caixa de e-mail, ‘*My Inbox*’.
- Delete essa caixa de e-mail e crie uma nova (‘*Add Inbox*’), nomeando com: “Minha Caixa”.

Criação da rota - Transporte

- Com a nova caixa de e-mail criada. Vá em 'engrenagens':

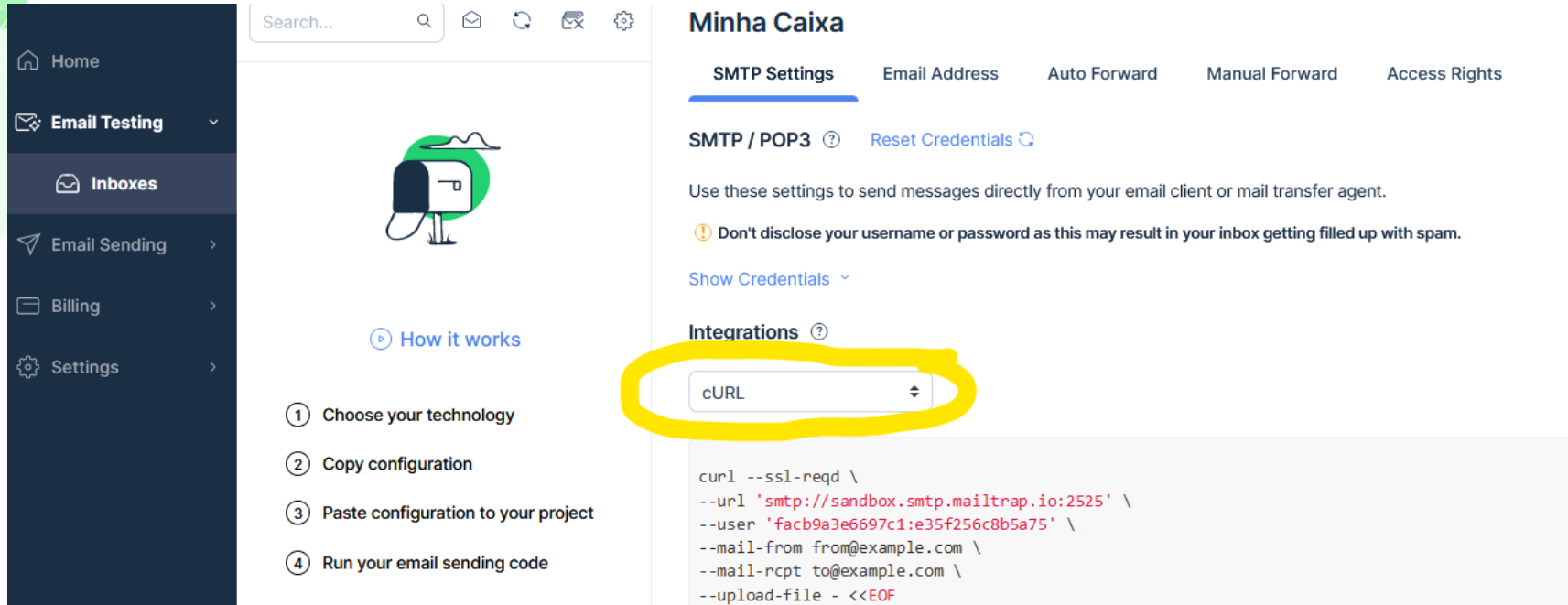


Projects

My Project :						Add Project
Inboxes	Total Sent [?]	Messages	Max size	Last message	Action	Add Inbox
Minha Caixa	0	✉ 0 / 0	50	Empty	  	

Criação da rota - Transporte

- Selecione o campo: *'Integrations'*.



The screenshot displays the Mailtrap web interface. On the left is a dark sidebar with navigation links: Home, Email Testing, Inboxes, Email Sending, Billing, and Settings. The main content area is titled 'Minha Caixa' and includes tabs for SMTP Settings, Email Address, Auto Forward, Manual Forward, and Access Rights. The 'SMTP Settings' tab is active, showing 'SMTP / POP3' configuration. Below this, there's a warning about not disclosing credentials. The 'Integrations' section is highlighted with a yellow circle, showing a dropdown menu with 'cURL' selected. To the left of the Integrations section, a 'How it works' section lists four steps: 1. Choose your technology, 2. Copy configuration, 3. Paste configuration to your project, and 4. Run your email sending code. At the bottom right, a code block shows a cURL command for sending an email via Mailtrap's sandbox.

Search...

Home

Email Testing

Inboxes

Email Sending

Billing

Settings

How it works

- 1 Choose your technology
- 2 Copy configuration
- 3 Paste configuration to your project
- 4 Run your email sending code

Minha Caixa

SMTP Settings | Email Address | Auto Forward | Manual Forward | Access Rights

SMTP / POP3 ? [Reset Credentials](#)

Use these settings to send messages directly from your email client or mail transfer agent.

⚠ Don't disclose your username or password as this may result in your inbox getting filled up with spam.

[Show Credentials](#)

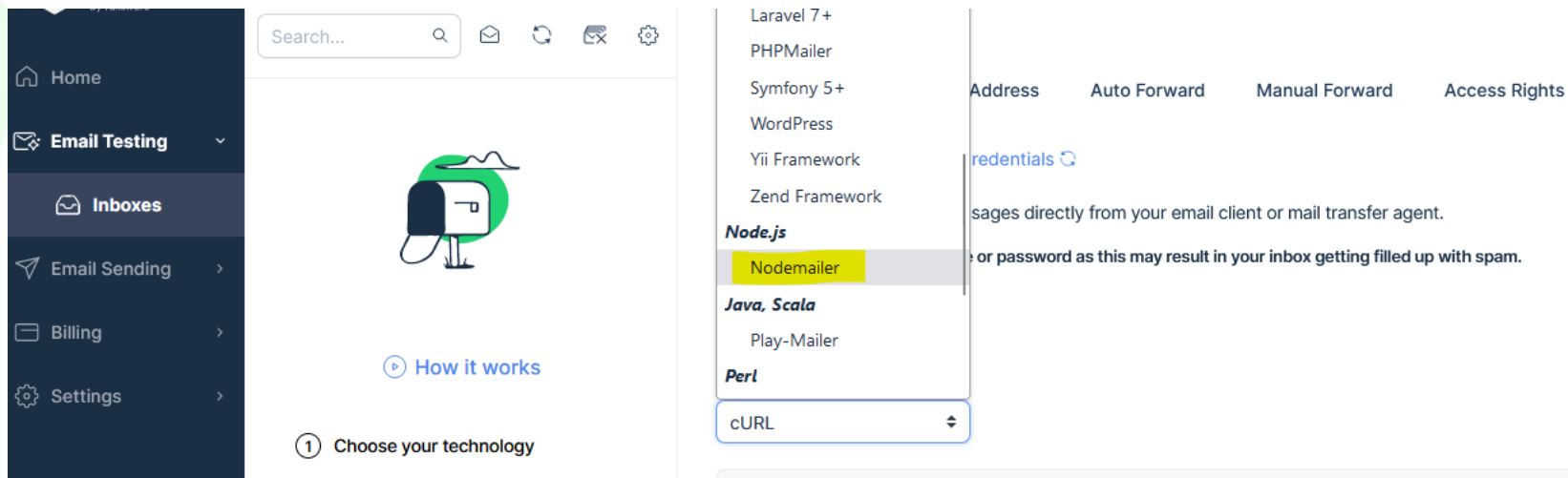
Integrations ?

cURL

```
curl --ssl-reqd \  
--url 'smtp://sandbox.smtp.mailtrap.io:2525' \  
--user 'facb9a3e6697c1:e35f256c8b5a75' \  
--mail-from from@example.com \  
--mail-rcpt to@example.com \  
--upload-file - <<EOF
```


Criação da rota - Transporte

- Selecione: '*Nodemailer*'.



Criação da rota - Transporte

- Copie o código gerado.

Nodemailer



Nodemailer is an easy to use module to send e-mails with Node.JS:

```
var transport = nodemailer.createTransport({  
  host: "sandbox.smtp.mailtrap.io",  
  port: 2525,  
  auth: {  
    user: "[REDACTED]",  
    pass: "[REDACTED]"  
  }  
});
```

- Ocultei o usuário e senha pois esse usuário é o meu. Se você utilizar o meu, o e-mail chegará na minha caixa postal.

Criação da rota - Transporte

- Cole no nosso código, substituindo o código padrão. Agora com o código da nossa caixa postal do *Nodemailer*, com o host, porta e dados do usuário (seus dados).

```
app.get("/sendemail", async (req, res) => {  
  
  var transport = nodemailer.createTransport({  
    host: "sandbox.smtp.mailtrap.io",  
    port: 2525,  
    auth: {  
      user: "██████████",  
      pass: "██████████"  
    }  
  });  
  
});
```



Criação da rota - Mensagem

- O próximo passo será a criação da Mensagem do e-mail.
- Retornando na documentação do *Nodemailer*, acessando o link: <https://nodemailer.com/>, menu esquerdo em '*3. Message configuration*'. Teremos todas as informações sobre o conteúdo do e-mail a ser enviado.
- Utilizaremos o exemplo padrão:

```
var message = {  
  from: "sender@server.com",  
  to: "receiver@sender.com",  
  subject: "Message title",  
  text: "Plaintext version of the message",  
  html: "<p>HTML version of the message</p>"  
};
```

Criação da rota - Mensagem

- Iremos copiar o exemplo abaixo e colar no nosso código (abaixo da codificação de transporte).
- *var message = {*
- *from: "sender@server.com",*
- *to: "receiver@sender.com",*
- *subject: "Message title",*
- *text: "Plaintext version of the message",*
- *html: "<p>HTML version of the message</p>"*
- *};*



Criação da rota - Mensagem

- Vamos alterar o corpo da mensagem ('message').
- Alterei com o domínio @reprograme-se.com, com: origem ('from'); destino ('to'); assunto ('subject') e mensagem em texto e *HTML*, conforme abaixo:

```
var message = {  
  from: "noreplay@reprograme-se.com",  
  to: "reprograme-se@reprograme-se.com",  
  subject: "Instrução para recuperar a senha",  
  text: "Leonardo. \n\n Você solicitou alteração de senha. \n\nEquipe Reprograme-se",  
  html: "Prezado Leonardo, <br><br> Você solicitou alteração de senha.<br><br>Equipe Reprograme-se<br>"  
};
```



Criação da rota – Envio da mensagem

- Para enviar nossa mensagem, acessaremos a documentação do Nodemailer, acessando o link: <https://nodemailer.com/>, menu esquerdo em '2. Usage'. Teremos todas as informações sobre o conteúdo do e-mail a ser enviado.
- Utilizaremos o exemplo padrão:

```
transporter.sendMail(data[, callback])
```

- Adaptando ao nosso código, teremos:

```
transport.sendMail(message)
```

- Já que enviaremos a nossa mensagem ('message') via nosso transporte ('transport').



Criação da rota – Envio da mensagem

- Poderemos ainda tratar erros dentro da função `'sendMail'`. Caso aconteça um erro, enviaremos uma mensagem informando o erro (HTTP Cat). Criaremos uma função ('function') para receber via parâmetro o erro ('err'), conforme código abaixo:

```
transport.sendMail(message, function(err) {  
  if (err)  
    return res.status(400).json({  
      erro: true,  
      mensagem: "Erro: E-mail não enviado!"  
    });  
  else  
    return res.json({  
      erro: false,  
      mensagem: "E-mail enviado com sucesso!"  
    });  
});
```

Enviando e-mail com o Nodemailer

- Se acontecer um erro (se (err) for igual a 'true'), enviaremos um retorno do erro (*true*) e a mensagem: "Erro: E-mail não enviado!", caso aconteça algum erro. Se não, enviaremos um retorno do erro (*false*) e uma mensagem de: "E-mail enviado com sucesso!"
- Para finalizar, incluiremos o comando '.listen()' e a porta de comunicação (porta 443) com o Servidor, apresentando a mensagem: 'Servidor rodando' (com o comando: console.log).

```
app.listen(porta, () => { console.log('Servidor rodando') })
```

- Pronto! Já temos uma *API* que envia e-mails via *Nodemailer*!

Enviando e-mail com o Nodemailer

Codificação final (1/3):

```
1  const express = require('express')
2  const app = express()
3  const nodemailer = require('nodemailer')
4  const porta = 443
5
6  app.get('/', (req, res) => {
7    res.send('Enviando e-mail com o Nodemailer!')
8  })
9
10 app.get("/sendemail", async (req, res) => {
11
12   var transport = nodemailer.createTransport({
13     host: "sandbox.smtp.mailtrap.io",
14     port: 2525,
15     auth: {
16       user: "7af58ef8974bbf",
17       pass: "c736319eaf9efc"
18     }
19   });
```


Enviando e-mail com o Nodemailer

Codificação final (2/3):

```
21 var message = {
22   from: "noreplay@reprograme-se.com",
23   to: "reprograme-se@reprograme-se.com",
24   subject: "Instrução para recuperar a senha",
25   text: "Leonardo. \n\n Você solicitou alteração de senha. \n\n Equipe Reprograme-se",
26   html: "Prezado Leonardo, <br><br> Você solicitou alteração de senha.<br><br>Equipe Reprograme-se<br>"
27 };
28
29 transport.sendMail(message, function(err) {
30   if (err)
31     return res.status(400).json({
32       erro: true,
33       mensagem: "Erro: E-mail não enviado!"
34     });
35   else
36     return res.json({
37       erro: false,
38       mensagem: "E-mail enviado com sucesso!"
39     });
40 });
41 });
```

Enviando e-mail com o Nodemailer

Codificação final (3/3):

```
43 ✓ app.listen(porta, () => {  
44   console.log("Servidor rodando")  
45 });
```

Testes!

- Para testar o código, inicie com o botão 'Run' do *Replit*.
- Na aba 'Console' é possível observar a mensagem: "Servidor rodando"!
- Na aba 'Webview' é possível observar a mensagem: "Enviando e-mail com o Nodemailer!"".
- Abra uma nova aba do código no browser. (ex: <https://APIdeRotas.usuario.repl.co/>)
- Informe o comando para enviar o e-mail: "sendemail"
- <https://APIdeRotas.usuario.repl.co:/sendemail>

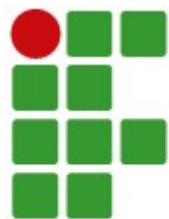
Testes!

- Acesse sua caixa postal do *MailTrap* (<https://mailtrap.io/inboxes>).
- Verifique se recebeu um e-mail (*Messages*).

Messages

✉ 1 / 1

- Entre na sua caixa postal e visualize o e-mail recebido (criado e enviado por você).



**INSTITUTO
FEDERAL**
Espírito Santo