



Desenvolvimento Web II

Framework Laravel 5

Mailables / Importação Dados “.txt”

<https://laravel.com/docs/5.6/mail#generating-mailables>

Gil Eduardo de Andrade





Mailable – Enviando E-mails

Introdução:

- O *Laravel* fornece uma API simples e transparente baseada na biblioteca *SwiftMailer* – que fornece *drivers* para SMTP, Mailgun, SparkPost, entre outros;
- Ela permite ao desenvolvedor, enviar e-mails através de serviços locais ou baseados em nuvem, de acordo com a sua necessidade;





Mailable – Enviando E-mails

Introdução:

- Todos os drivers da API exigem a biblioteca *HTTP Guzzle*, que pode ser instalada por meio do gerenciador de pacotes do *Composer*:
 - ***composer require guzzlehttp/guzzle***
- Normalmente, ela já encontra-se disponível quando criamos um projeto Laravel, como no caso desta aula;





Mailable – Enviando E-mails

Introdução:

- Todos os drivers da API exigem a biblioteca *HTTP Guzzle*, que pode ser instalada por meio do gerenciador de pacotes do *Composer*:
 - ***composer require guzzlehttp/guzzle***
- Normalmente, ela já encontra-se disponível quando criamos um projeto Laravel, como no caso desta aula;





Mailable – Enviando E-mails

Gerando E-mails:

- No Laravel, os e-mails que serão enviados pela aplicação são representados por uma classe “*mailable*”, armazenada em `app/Mail`;
- Este diretório não é gerado por padrão quando criamos uma nova aplicação Laravel, mas sim quando criamos a primeira classe “*mailable*”, via comando *artisan*: **`php artisan make:mail NomeClasse`**;





Mailable na Prática

Framework Laravel

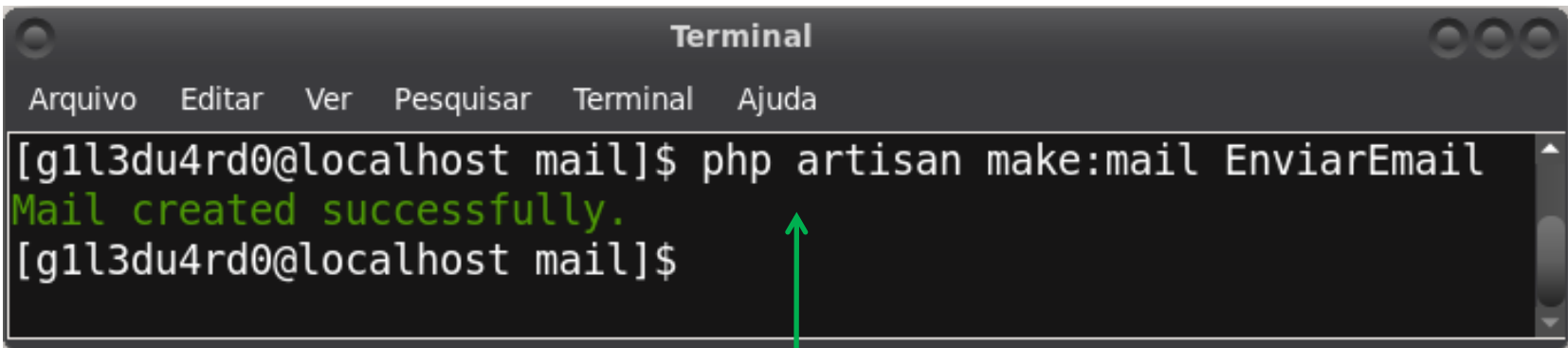
http://www.gileduardo.com.br/ifpr/dwii/downloads/ex_doc11.zip



Mailable – Enviando E-mails

Exemplo:

(Criando classe Mailable: *EnviarEmail.php*)



```
Terminal
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
[g1l3du4rd0@localhost mail]$ php artisan make:mail EnviarEmail
Mail created successfully.
[g1l3du4rd0@localhost mail]$
```

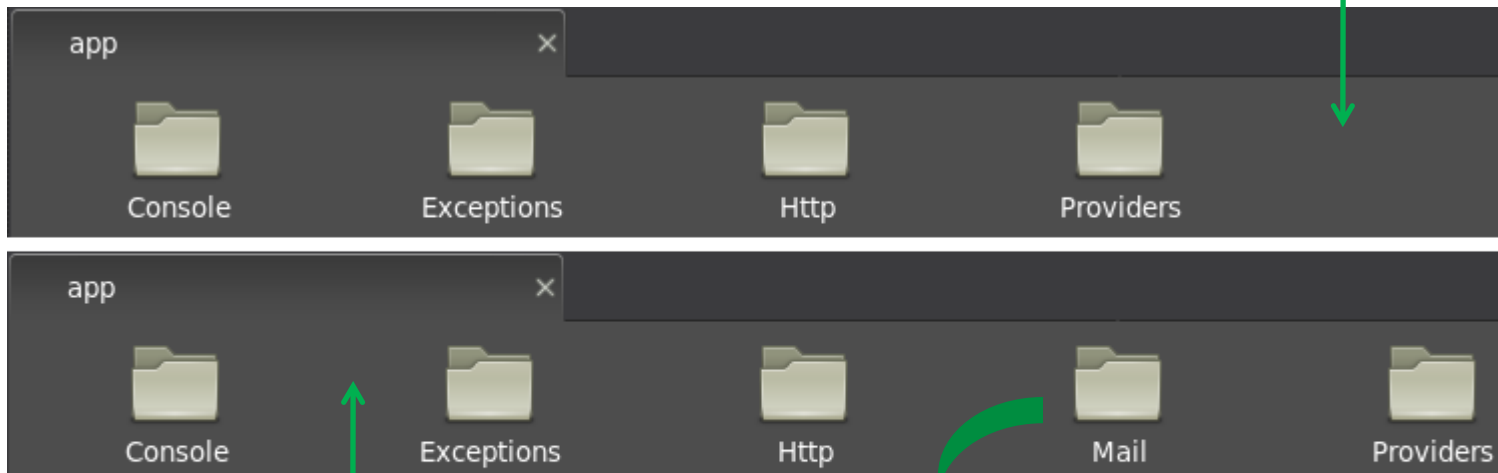
Dentro do diretório da aplicação Laravel criada, via terminal, utilizamos a ferramenta de linha de comando **artisan** – assim como fizemos anteriormente para criar classes de Controle e Modelo.

Mailable – Enviando E-mails

Exemplo:

(Criando classe Mailable: *EnviarEmail.php*)

Diretório “app” *antes* da execução do *comando artisan*.



Diretório “app” *após* a execução do *comando artisan*. A classe Mailable *EnviarEmail.php* é criada dentro do diretório *Mail*.

Mailable – Enviando E-mails

Exemplo:

(Estrutura da classe Mailable: EnviarEmail.php)

```
class EnviarEmail extends Mailable {  
  
    use Queueable, SerializesModels;  
    public $view;  
    public $dados;  
    public $titulo;  
  
    public function __construct($view, $dados, $titulo) {  
        $this->view = $view;  
        $this->dados = $dados;  
        $this->titulo = $titulo;  
    }  
  
    public function build() {  
        return $this->view($this->view)  
            ->from("gilgea@gmail.com", "SIG")  
            ->subject($this->titulo)  
            ->with('dados', $this->dados);  
    }  
}
```

Herda as características da **Mailable**, permitindo utilizar a API de envio de e-mails disponível no Laravel.

Atributos da nova classe construída, não são criados por padrão, foram definidos por mim, servem para indicar a **view** utilizada para envio da mensagem, os **dados** que serão exibidos pela **view** e o **título do e-mail**. São recebidos pelo construtor da classe.

Constrói o e-mail retornando a **view** que define a sua estrutura visual, configura as principais informações do e-mail.

Mailable – Enviando E-mails

Exemplo:

(Estrutura da view: *maillImportar.blade.php*)

```
<!DOCTYPE html>
<html>
  <head>
    <title>SIG - Homens mais Inteligentes da História</title>
  </head>
  <body>
    <h3>Seja Bem-vindo, <br/>{{ $dados['nome'] }} </h3>
    <br/>
    <b>Sua senha é: {{ $dados['senha'] }}</b>
    <br/>
    <h4>{{ $dados['conteudo'] }}</h4>
    <br/>
    
  </body>
</html>
```

A estrutura visual do e-mail é definida dentro de uma *view*, utilizando o formato HTML. Repare que os dados (variáveis php) passado para a *view* são utilizados para montar o e-mail, como, por exemplo, o nome do destinatário e o conteúdo da mensagem enviada. Observe também que uma imagem (hospedada na web) é utilizada como rodapé.

Mailable – Enviando E-mails

Exemplo:

(Formato do E-mail após Recebimento)

Gmail ▾

← 📎 ! 🗑️ Mover para a Caixa de Entrada

ESCREVER

Aula Mail Laravel

Entrada (1.128)
Com estrela
Importante
Bate-papos
Enviados
Rascunhos (5)
Todos os e-mails

 **SIG** <gilgea@gmail.com>
para pc ▾

Nome do destinatário

Seja Bem-vindo, ISAAC NEWTON

Senha Gerada e Enviada

Conteúdo da Mensagem

Sua senha é: 28981542

Temos imenso prazer em tê-lo como mais novo integrante!

Imagem de Rodapé

 **Sistema de Identificação de Gênios**
www.genios.org
contato@genios.com.br
+55 (41) 0000-0000

SIG 

Mailable – Enviando E-mails

Configurações – SMTP:

(Arquivo: “.env” – considerando *mailtrap* e *gmail*)

```
.env
26 MAIL_DRIVER=smtp
27 MAIL_HOST=smtp.mailtrap.io
28 MAIL_PORT=2525
29 MAIL_USERNAME=ce1c97397b85f5
30 MAIL_PASSWORD=
31 MAIL_ENCRYPTION=null
```

```
.env
26 MAIL_DRIVER=smtp
27 MAIL_HOST=smtp.gmail.com
28 MAIL_PORT=587
29 MAIL_USERNAME=gilgea@gmail.com
30 MAIL_PASSWORD=
31 MAIL_ENCRYPTION=tls
```

O **mailtrap** (<https://mailtrap.io/>) é uma ferramenta que permite testar se o envio de e-mails de uma aplicação está funcionando corretamente. Nela o desenvolvedor pode mandar uma determinada quantidade de e-mails de teste, é útil porque não congestiona a caixa de entrada da conta do e-mail do desenvolvedor.

Quando em produção, o desenvolvedor pode optar por utilizar uma conta de e-mail da empresa para a qual o sistema está sendo construído, ou ainda, utilizar uma conta de e-mail de serviços como o **google**, como no exemplo acima.

Mailable – Enviando E-mails

Exemplo:

(Arquivo: “.MailController.php > concluir()” – Código de Envio)

```
// Envia e-mail com a senha para os gênios importados do .txt
$dados_mail = array();
$dados_mail['nome'] = mb_strtoupper($dados[0], 'UTF-8');
$dados_mail['senha'] = $senha;
$dados_mail['conteudo'] = $conteudo;
$email = mb_strtolower($dados[2], 'UTF-8');
\Mail::to($email)->send( new EnviarEmail("mailImportar", $dados_mail, $titulo) );
```



Primeiro cria o array ***\$dados_mail*** para armazenar as informações que serão passadas para classe ***Mailable*** “***EnviarMail***”, criada via linha de comando com ***artisan***. Os dados nome e e-mail foram carregados (importados) de um arquivo texto, que será explicado a seguir. Os dados título e conteúdo foram especificados pelo usuário dentro da aplicação criada. O envio do e-mail acontece na linha ***\Mail***, onde especificamos o destinatário: ***to(\$email)***, e invocamos o método ***send()*** passando como parâmetro uma instância da classe ***EnviarEmail***, explicada anteriormente.



Importação de Dados “.txt”

Exemplo:

(Arquivo: “.MailController.php > concluir()” – Código de Importação)

```
// Arquivo Selecionado
$arquivo = Request::file('arq_alunos');
// Nenhum Arquivo Selecionado
if (empty($arquivo)) {
}
// Efetua o Upload do Arquivo
$path = $arquivo->store('uploads');

// Efetua a Leitura do Arquivo
$fp = fopen("../storage/app/". $path, "r");
```

Obtém o arquivo “.txt” selecionado pelo usuário.

Verifica se o usuário selecionou algum arquivo.

Efetua o upload do arquivo para o diretório padrão do Laravel: **“storage/app/uploads”**. Como o arquivo é renomeado, o método *store()* retorna a path **/uploads** juntamente com o novo nome dado para arquivo importado.

Abre o arquivo “.txt” para leitura e importação.



Aplicação Exemplo

Mailable / Importação

http://www.gileduardo.com.br/ifpr/dwii/downloads/ex_doc11.zip



Aplicação Exemplo

Tela principal:

(Arquivo: “.main.blade.php”)



Importar / Enviar E-mail

Componente HTML que permite navegar nos diretórios locais e selecionar um arquivo.

Título:

Título do e-mail a ser enviado.

Arquivo de e-mails:

Abrir Navegador

Nenhum arquivo selec

Conteúdo/Mensagem:

Conteúdo do e-mail a ser enviado.

Importar / Enviar



Aplicação Exemplo

Arquivo Texto Importado:

(Arquivo: “dados_aluno.txt”)

```
Gil Eduardo de Andrade#18/11/1983#contato@gileduardo.com.br#  
Stephen Hawking#08/01/1942#lp@gileduardo.com.br#  
Albert Einstein#17/03/1879#web@gileduardo.com.br#  
Nikola Tesla#10/07/1856#oo@gileduardo.com.br#  
Isaac Newton#04/01/1643#pc@gileduardo.com.br#  
Leonardo da Vinci#15/04/1452#andrade@gileduardo.com.br#
```

São importado: “nome”, “data de nascimento” e “e-mail”
O término de cada informação é dado pelo caractere “#”





Aplicação Exemplo

Efetuando Importação e Envio do E-mail:

Título:

Aula Mail Laravel

Arquivo de e-mails:

[Abrir Navegador](#)

dados_mail.txt

Conteúdo/Mensagem:

Temos imenso prazer em tê-lo como mais novo integrante!

Importar / Enviar

IMPORTAÇÃO DE DADOS

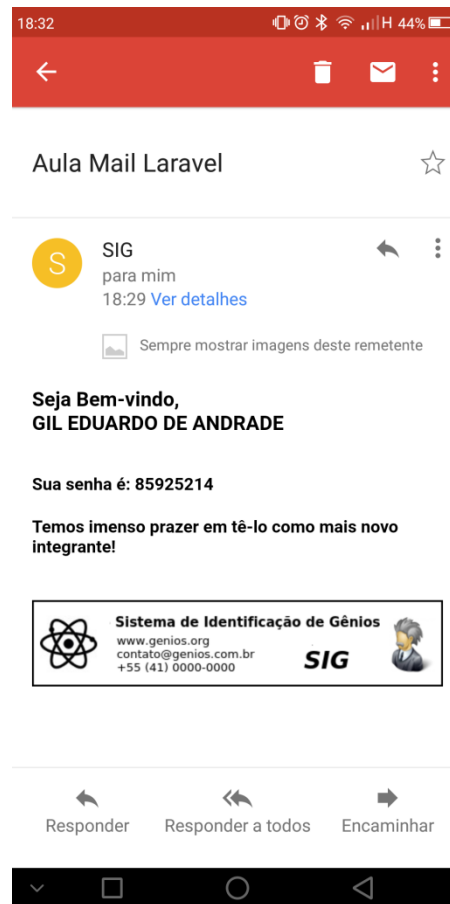
Um total de '6' gênios(s) foi importado com sucesso!

OK



Aplicação Exemplo

E-mail Recebido – Caixa de Entrada:





Mailable – Enviando E-mails

Exemplos Utilizados no Documento

http://www.gileduardo.com.br/ifpr/dwii/downloads/dwii_exdoc11.zip

Exercício sobre o Conteúdo

http://www.gileduardo.com.br/ifpr/dwii/downloads/dwii_pratica11.pdf

