



Desenvolvimento Web II

Framework Laravel 5

Gerando Documentos PDF – DOMPDF

<https://github.com/barryvdh/laravel-dompdf>

Gil Eduardo de Andrade





DOMPDF

Introdução:

- Durante o desenvolvimento de diversos tipos de aplicações web torna-se necessário gerar documentos (relatórios, certificados, recibos) em formato PDF;
- Nesse contexto, podemos utilizar, quando estamos desenvolvendo aplicações Laravel, o pacote ***DOMPDF*** ;





DOMPDF

Introdução:

- O pacote **DOMPDF** possui como vantagem a possibilidade de criarmos um documento **PDF** a partir de uma *View Laravel*;
- Essa característica permite ao desenvolvedor modelar/construir o formato do documento **PDF** via código HTML, Blade, etc.;





DOMPDF

Instalação (via composer):

\$ composer require barryvdh/laravel-dompdf

```
[g1l3du4rd0@localhost pdf]$ composer require barryvdh/laravel-dompdf
Using version ^0.8.2 for barryvdh/laravel-dompdf
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 5 installs, 0 updates, 0 removals
  - Installing sabberworm/php-css-parser (8.1.0): Loading from cache
  - Installing phenx/php-svg-lib (v0.3): Loading from cache
  - Installing phenx/php-font-lib (0.5.1): Loading from cache
  - Installing dompdf/dompdf (v0.8.2): Loading from cache
  - Installing barryvdh/laravel-dompdf (v0.8.2): Loading from cache
Writing lock file
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover
Discovered Package: fideloper/proxy
Discovered Package: laravel/tinker
Discovered Package: nunomaduro/collision
Discovered Package: barryvdh/laravel-dompdf
Package manifest generated successfully.
[g1l3du4rd0@localhost pdf]$
```

DOMPDF

Configuração – Provider:

(Arquivo: *config/app.php*)

```
'providers' => [  
    /*  
     * Laravel Framework Service Providers...  
     */  
    Illuminate\Auth\AuthServiceProvider::class,  
    Illuminate\Broadcasting\BroadcastServiceProvider::class,  
    Illuminate\Bus\BusServiceProvider::class,  
    Illuminate\Cache\CacheServiceProvider::class,  
    .  
    .  
    Barryvdh\DomPDF\ServiceProvider::class, ←  
    .  
    .  
],
```

Após a instalação do DOMPDF é preciso definir um provedor de serviço para o novo pacote instalado. Os provedores de serviço Laravel são responsáveis por inicializar/carregar todos os principais serviços da aplicação quando ela é executada.



DOMPDF

Configuração – Alias:

(Arquivo: *config/app.php*)

```
'aliases' => [  
    'App' => Illuminate\Support\Facades\App::class,  
    'Artisan' => Illuminate\Support\Facades\Artisan::class,  
    :  
    :  
    'PDF' => Barryvdh\DomPDF\Facade::class,  
    :  
    'View' => Illuminate\Support\Facades\View::class,  
],
```

Após configurar o provedor de serviço para o DOMPDF, vamos definir um Alias (apelido) para que possamos invocar os métodos do pacote DOM de modo transparente, permitindo simplificar a codificação necessária para gerar os documentos PDF.



DOMPDF

Aplicação Exemplo

http://www.gileduardo.com.br/ifpr/dwii/downloads/dwii_exdoc13.zip





DOMPDF

Aplicação Adaptada

SETA – Menu e Rotas Principais:

(Adaptação: Exemplo Primeira Aula Laravel - DOMPDF)

SETA - Sistema de Entrega de Trabalhos e Atividades [Home](#)

 Menu Principal

 Curso	 Turma	 Aluno	 Disciplina
 Conceito	 Relatório	 Importar	 Exportar



DOMPDF

Aplicação Adaptada

SETA – Models / Migrations – Aluno:

(Arquivos: `create_aluno_models_table.php`)

```
Schema::create('aluno_models', function (Blueprint $table) {  
    $table->increments('id');  
    $table->string('nome');  
    $table->string('curso');  
    $table->string('turma');  
    $table->timestamps();  
});
```

Uma classe de modelo e uma *migration* foram criadas para armazenar as informações dos alunos, que serão utilizadas para geração de relatório e certificado. Para o exemplo proposto consideramos que os alunos possuem um identificador, um nome, um curso e uma turma.

DOMPDF

Aplicação Adaptada

SETA – View Aluno:

(Arquivos: *aluno.blade.php*)

SETA - Sistema de Entrega de Trabalhos e Atividades [Home](#)

 **Alunos Cadastrados**

Relatório e Certificado Coletivo **Relatório e Certificado Individual**

[Cadastrar Novo Aluno](#)

[Relatório dos Alunos \(PDF\)](#) [Certificado dos Alunos \(PDF\)](#)

ID	ALUNO	CURSO	TURMA	EVENTOS
3	GIL EDUARDO DE ANDRADE	EMI - TÉCNICO EM INFORMÁTICA	INFO 17	   
4	LUIZ CARLOS DE OLIVEIRA BASTOS	TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO	TADS 17	   
5	MARIANA ALVES SOUZA	EMI - TÉCNICO EM INFORMÁTICA	INFO 15	   
6	DOUGLAS ROBERTO COSTA	TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO	TADS 16	   

DOMPDF

Aplicação Adaptada

SETA – View Relatório PDF:

(Arquivos: *alunoRelatorio.blade.php*)

```
<table>
  <thead>
    <tr>
      <th style="width : 60px">ID</th>
      <th style="width : 450px">ALUNO</th>
      <th style="width : 80px">TURMA</th>
    </tr>
  </thead>
  <tbody>
    @foreach ($alunos as $dados)
      <tr>
        <td>{{ $dados->id }}</td>
        <td>{{ $dados->nome }}</td>
        <td>{{ $dados->turma }}</td>
      </tr>
    @endforeach
  </tbody>
</table>
```

Uma nova *View* foi criada para gerar o relatório dos alunos em formato PDF. Por se tratar de uma *View*, a linguagem HTML é utilizada para definir o formato ou estrutura desse relatório.

Assim como visto nas primeiras aulas é possível enviar dados para uma *View* e manipulá-los e exibí-los via Blade.



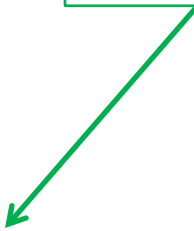
DOMPDF

Aplicação Adaptada

SETA – View Relatório PDF:

(Arquivo: *alunoRelatorio.blade.php*)

Relatório PDF Gerado a partir da View *alunoRelatório.blade.php*



Relatório de Alunos

ID	ALUNO	TURMA
3	GIL EDUARDO DE ANDRADE	INFO 17
4	LUIZ CARLOS DE OLIVEIRA BASTOS	TADS 17
5	MARIANA ALVES SOUZA	INFO 15
6	DOUGLAS ROBERTO COSTA	TADS 16
7	MARCOS AULÉRIO TROTTA	INFO 15
8	HELOÍSA MATIE DE ANDRADE	TADS 18
9	ANA PAULA FERREIRA	INFO16
10	RODRIGO ALBERTO SILVA	TADS 16
11	MARCOS AULÉRIO TROTTA	INFO 15
12	HELOÍSA MATIE DE ANDRADE	TADS 18
13	ANA PAULA FERREIRA	INFO16



DOMPDF

Aplicação Adaptada

SETA – View Certificado PDF:

(Arquivos: *alunoCertificado.blade.php*)

```
<style>
  div.page_break + div.page_break {
    page-break-before: always;
  }

  p.linha {
    border-top: 2px solid #000;
  }
</style>
:
@foreach ($alunos as $dados)
  <!-- Nova Página: Documento PDF -->
  <div class="page_break"></div>
```

Para geração dos certificados, é necessário gerar uma nova página dentro do documento PDF para cada um dos certificados dos alunos. Como os certificados são modelados por uma View, foi definida uma <div> contendo um page-break-before.

Foi definida, também, uma linha sólida para sempre que um novo parágrafo (tag <p>) é criado.

A <div> “page_break” é invocada sempre que um novo certificado vai ser gerado.

DOMPDF

Aplicação Adaptada

SETA – View Certificado PDF:

(Arquivos: *alunoCertificado.blade.php*)

```
<?php
    $img_logo = public_path()."/img/logo.png";
    $img_aluno = public_path()."/img/alunop_ico.png";
?>
:
:
<td width='30%' align='center'>
    
</td>
```

Para obter o caminho das imagens, de modo que elas possam ser carregadas pelo pacote DOM, é necessário utilizar o método *public_path()* via abertura da tag "*<?php*".

Para adicionar as imagens para o certificado também utilizamos a tag "*<?php*", passando para tag ** as variáveis php criadas anteriormente com os caminhos previamente definidos.

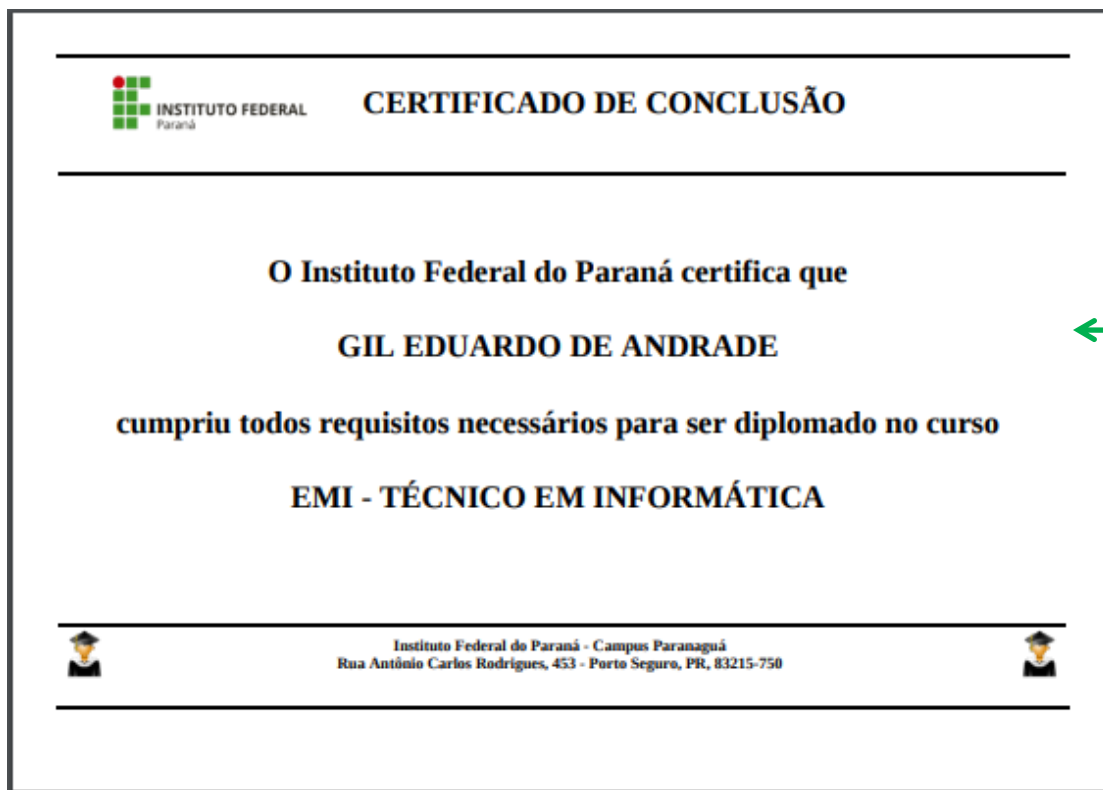
PS. o restante do código não é detalhado porque trata-se de HTML puro, que não faz parte do escopo da disciplina.

DOMPDF

Aplicação Adaptada

SETA – View Certificado PDF:

(Arquivo: *alunoCertificado.blade.php*)



Certificado PDF Gerado a
partir da *View*
alunoCertificado.blade.php

DOMPDF

Aplicação Adaptada

SETA – Controller Aluno:

(Arquivos: *AlunoController.php* – Gerar PDF)

```
public function gerarRelatorio($id) {
    // Todos os Alunos
    if($id == 0) { $alunos = AlunoModel::all(); }
    // Um Aluno Específico
    else { $alunos = AlunoModel::where('id', '=', $id)->get(); }

    return \PDF::loadView('alunoRelatorio', compact('alunos'))
        ->setPaper('A4', 'portrait')
        ->stream('relatorio_alunos.pdf');
    // ->download('relatorio_alunos.pdf');
}
```

Método `loadView()` é invocado, ele recebe como parâmetros principais: o nome da *View* (***alunoRelatorio***) que servirá de referência para geração do documento PDF e os dados (***compact('alunos')***) que serão utilizados pela *View*.

Os parâmetros complementares: ***setPaper('A4', 'portrait')*** e ***stream('.')*** permitem configurar, respectivamente, o tamanho / orientação das páginas do documento e a forma como o documento deve ser apresentado (*stream* indica que deve ser aberto no navegador, já *download* indica que deve ser baixado para máquina local).



DOMPDF

Exemplos Utilizados no Documento

http://www.gileduardo.com.br/ifpr/dwii/downloads/dwii_exdoc13zip

Exercício sobre o Conteúdo

http://www.gileduardo.com.br/ifpr/dwii/downloads/dwii_pratica13.pdf

