Générer un fichier PDF avec laravel dompdf

[sommaire]

1. ## Objectifs

- Apprendre les méthodes essentielles pour créer et montrer les exemples de tables de données yajra de laravel 8
- Découvrir comment créer un fichier PDF à partir d'un code HTML ou d'une vue (template Blade) avec le package laravel-dompdf dans un projet Laravel.

- ## Présentation

- Dans ce tutoriel, nous utilisons le package **laravel-dompdf** pour générer un fichier **pdf** dynamique et en utilisant ce package, nous pouvons également télécharger le fichier pdf sur notre ordinateur local.
- Le package **laravel-dompdf** est un **wrapper** pour **Laravel** de la librairie **Dompdf** qui permet de convertir le code HTML en PDF (Portable Document Format).
    - Il génère des fichiers PDF en prenant en charge les fonctionnalités suivantes :
    - La plupart des propriétés CSS 2.1 et CSS 3, les feuilles de style internes et externes
    - La plupart des attributs HTML 5
    - Les images (GIF, PNG, JPG, ...) et offre un support pour SVG
    - Les tableaux (lignes, colonnes, bordures de tableau, style de cellule, ...)
    - La police d’écriture (font)
    - Et autres..
- Dans ce tutoriel, nous developpons l'application suivante:

- ## Créez le projet Laravel 8

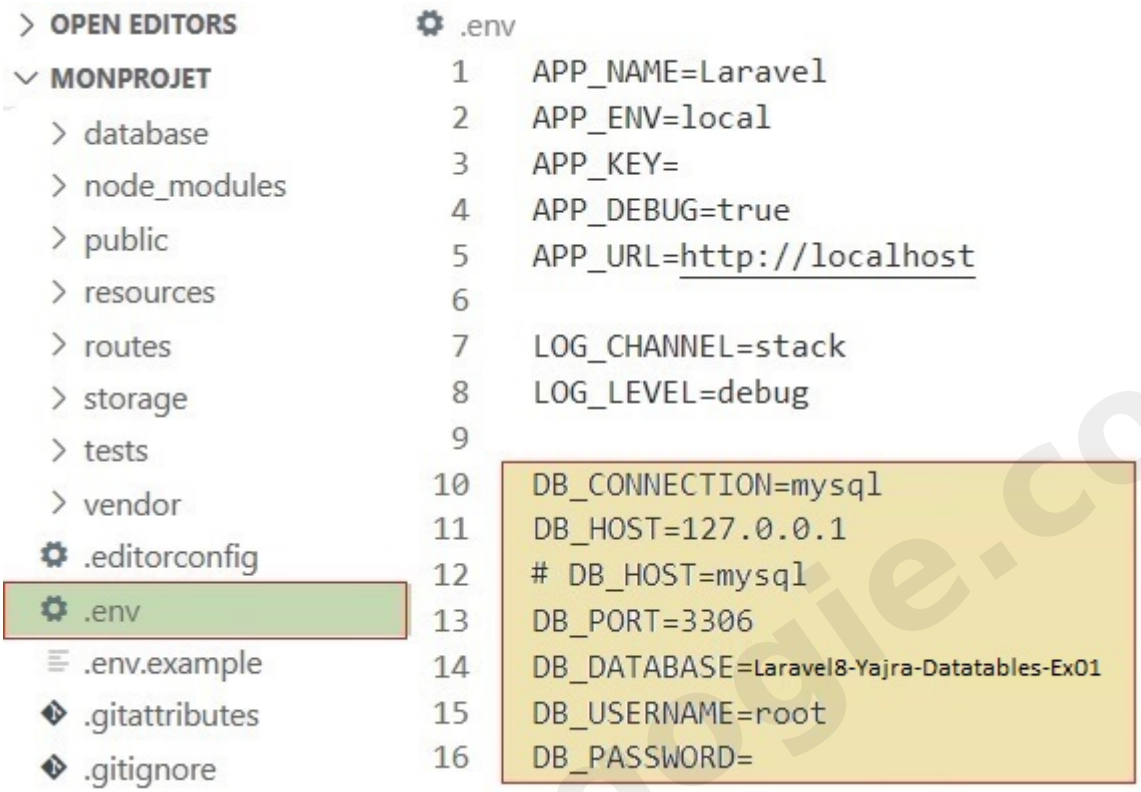
- Ouvrez votre terminal ou l'invite de commande et exécutez la commande ci-dessous: `Composer create-project laravel/laravel Laravel8-PDF`

- ## Configurez de la base de données

- Comme MySQL est spécifié comme base de données par défaut dans **config/database.php**, nous n'avons pas besoin de faire de modifications dans ce fichier et nous pouvons donc passer directement au fichier `.env`

apcpedagogie.com

- Maintenant, vous devrez trouver un fichier nommé `.env` , où vous devrez spécifier les détails du serveur MySQL , comme le nom de la base de données, le nom d'utilisateur, etc. Dans ce fichier, vous devrez rechercher des noms commençant par `DB_` .



- En cela, vous trouverez la ligne `DB_CONNECTION` = **mysql** . En dessous se trouvent tous les détails spécifiés pour la connexion à une base de données.
- Vous devrez spécifier le nom de la base de données **Laravel8-Yajra-Datatables-crud-modal** , que nous avons créée, après DB_DATABASE = et également spécifier le nom d'utilisateur et le mot de passe en fonction de vos besoins. Enregistrez ensuite le fichier.

Créez un Modèle et des migrations

- Exécutez la commande pour créer un modèle et elle contient le schéma de la table de base de données: `php artisan make:model Etudiant -m`
- Ouvrez le fichier **database/migrations/timestamp_create_etudiants_table.php** et ajoutez le code ci-dessous.

apcpedagogie.com

database > migrations > 2021_04_20_102555_create_etudiants_table.php

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateEtudiantsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('etudiants', function (Blueprint $table) {
            $table->integer('cin')->primary();
            $table->string('prenomNom');
            $table->string('email')->unique();
            $table->string('nomUtil');
            $table->string('telephone');
            $table->date('dteNaissance');
            $table->string('genre');
            $table->timestamps();
        });
    }
    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('etudiants');
    }
}
```

- Ouvrez **app/Models/Etudiant.php** et ajoutez le code ci-dessous.

apcpedagogie.com

app > Models > Etudiant.php >

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Etudiant extends Model
{
    use HasFactory;
    public $primaryKey = 'cin';
    public $incrementing = false;
    protected $fillable = [
        'cin',
        'prenomNom',
        'email',
        'nomUtil',
        'telephone',
        'dteNaissance',
        'genre'
    ];
}
```

- Exécutez la migration à l'aide de la commande suivante: `php artisan migrate`

- **Ajoutez des enregistrements dans la base de données**

  - Placez le code ci-dessous dans **database/seeds/DatabaseSeeder.php** pour ajouter les faux enregistrements dans la base de données:

apcpedagogie.com

database > seeders > DatabaseSeeder.php > DatabaseSeeder

```php
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
use Illuminate\Support\Facades\DB;
use Faker\Factory as Faker;
class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     *
     * @return void
     */
    public function run()
    {
        $faker = Faker::create('it_IT');
        //$faker = Faker\Factory::create('fr_FR');

        foreach (range(1,100) as $index) {
            $gender = $faker->randomElement(['Homme', 'Femme']);
            DB::table('etudiants')->insert([
                'cin'=>$faker->unique()->numerify('########'),
                'prenomNom' => $faker->name($gender),
                'email' => $faker->unique()->email,
                'nomUtil' => $faker->username,
                'telephone' => $faker->numerify('9# ### ###'),
                'dteNaissance' => $faker->dateTimeBetween(
                    '-22 years', '-20 years')->format('Y-m-d'),
                'genre'=>$gender
            ]);
        }
    }
}
```

- Exécutez la commande suivante pour générer des données factices:`php artisan db:seed`
- Vous pouvez utiliser la commande suivante pour obtenir le même résultat: `php artisan db:seed --class=DatabaseSeeder`

Créez le fichier model

- Ouvrez votre terminal et exécutez la commande suivante ; `php artisan make:model Etudiant -m`
- Ok, donc après avoir exécuté la commande ci-dessous, vous trouverez **"app/Models/Etudiant.php"** et mettez le contenu ci-dessous dans le fichier **Etudiant.php**:

Installer laravel-dompdf

- Pour créer un fichier PDF dans Laravel, nous voulons d'abord télécharger le package **laravel-dompdf**.

apcpedagogie.com

- **Importation**
  - Pour importer **laravel-dompdf** avec ses dépendances, ouvrez la console (invite de commandes ou sur terminal) à la racine du projet, exécutez la commande composer suivante :`composer require barryvdh/laravel-dompdf`

- **Iintegration**
  - Maintenant, nous devons enregistrer ce package dans l'environnement de travail Laravel, pour cela, nous devons ouvrir le fichier **config/app.php** et chargeons-le dans l'application en ajoutant le service provider au tableau **$providers** et la façade (« PDF » pour faire court) au tableau **$aliases**.

```
// ...
'providers' => [
// ...,
// Service Provider DomPDF
 Barryvdh\DomPDF\ServiceProvider::class
],
'aliases' => [
// ...,
// Façade DomPDF
"PDF" => Barryvdh\DomPDF\Facade::class
],
// ...
```

- **Configuration**
  - Pour créer le fichier de configuration **config/dompdf.php** où vous pouvez modifier les options par défaut de **dompdf**, exécutez la commande artisan suivante : **`php artisan vendor:publish --provider="Barryvdh\DomPDF\ServiceProvider"`**

- **Utilisation**
  - Maintenant que **laravel-dompdf** est installé et configuré, nous pouvons créer une nouvelle instance **Barryvdh\DomPDF\PDF** en indiquant le contenu du fichier PDF par :

1. **Une vue (template Blade) avec ses données :**

apcpedagogie.com

```
use App\Etudiant;
use PDF;
// ...
public function getEtudiantPdf (Etudiant $etudiant)
{
// L'instance PDF avec une vue : resources/views/etudiants/liste.blade.php
 $pdf = PDF::loadView('etudiants.liste', compact('etudiant'));
}
// ...
```

- **Le chemin vers un fichier :**

```
$pdf = PDF::loadFile(public_path("documents/fichier.html"));
```

- **Une chaîne de caractères HTML :**

```
$pdf = PDF::loadHTML("Mon contenu HTML ici");
```

- **De cet instance PDF, nous pouvons :**

    1. **Forcer le navigateur à lancer le téléchargement du fichier PDF généré :**

```
use App\Etudiant;
use PDF;
// ...
public function getEtudiantPdf (Etudiant $etudiant)
{
// L'instance PDF avec la vue resources/views/etudiants/liste.blade.php
 $pdf = PDF::loadView('etudiants.liste', compact('etudiant'));
// Lancement du téléchargement du fichier PDF
return $pdf->download(\Str::slug($etudiant->title).".pdf");
}
// ...
```

- **Ou l’afficher dans le navigateur :**

```
return $pdf->stream();
```

- **Ou encore l'enregistrer :**

apcpedagogie.com

```
$pdf->save(public_path("storage/documents/fichier.pdf"));
```

- ## Créer un contrôleur

  - Après cela, nous devons créer un contrôleur dans notre application Laravel en utilisant la commande **composer**. Ce contrôleur gérera **http request** pour convertir le code html en fichier PDF: `php artisan make:controller EtudiantPDFController`
  - Cette commande créera le contrôleur dans le dossier **app/Http/Controllers** avec un nom tel que **EtudiantPDFController.php** .
  - Dans ce contrôleur, nous devons d'abord créer la méthode `index()` pour récupérer les données de la base de données et ici, pour récupérer les données, nous devons utiliser la méthode **get_etudiant_data()** et après cela, dans la méthode **index()**, nous avons envoyé des données pour afficher le fichier.
  - Pour convertir des données HTML en PDF, nous avons créé une autre méthode **pdf()** en utilisant cette méthode, elle convertira le code html en fichier PDF.
  - Pour les données html, nous utilisons la méthode **convert_etudiant_data_to_html()**.
  - Vous trouverez ci-dessous le code source complet du contrôleur.

- ## Créer un fichier Blade

  - Après cela, nous avons créé un fichier de vue dans **resources/view/liste_etudiant_pdf.blade.php**.
  - Ce fichier a reçu des données de la méthode `index()` du contrôleur et affiche les données dynamiques au format de table html.
  - Après cela, nous avons créé un bouton pour convertir ces données html en fichier pdf.
  - Lorsque nous avons cliqué sur ce bouton, il a envoyé une demande à la méthode `pdf()` du contrôleur qui convertira le html en fichier PDF en utilisant le **package dompdf** dans **laravel** que nous pouvons voir sur la page Web.

- ## Ajouter une route

```
Route::get('/liste_etudiant_pdf', 'App\Http\Controllers\EtudiantPDFController@index');

Route::get('/liste_etudiant_pdf/pdf', 'App\Http\Controllers\EtudiantPDFController@pdf');
```

apcpedagogie.com

Sommaire du cours Laravel

Source : https://apcpedagogie.com/generer-un-fichier-pdf-avec-laravel-dompdf · téléchargé le 03/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com