Application CRUD avec Laravel 8

[sommaire]

1. **Objectifs**

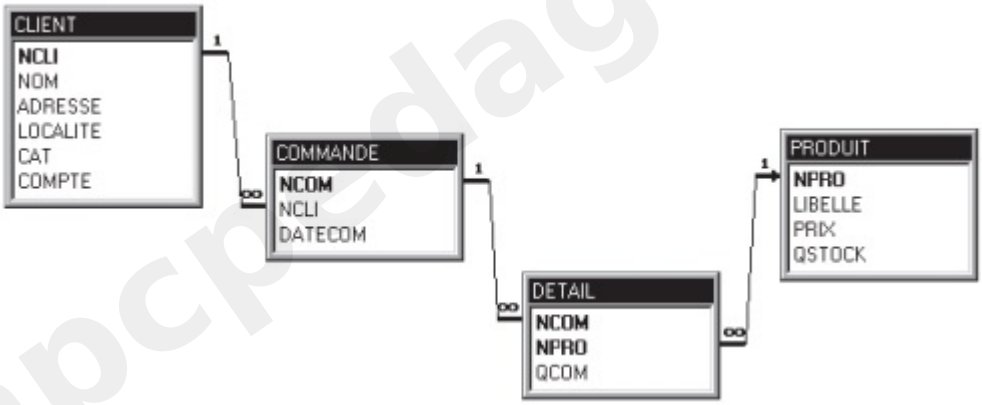
- Être capable de créer une application **CRUD** avec le Framework Laravel

- **Présentation**

- Dans ce tutoriel, nous commençons avec la présentation de la base de donnée qui servira comme exemple, puis nous allons implémenter une petite application **laravel 8** CRUD.

- **Base de données**

- La base de données (**gestionStock**) qui nous servira d'exemple dans ce tutoriel est structurée selon le schéma de la figure ci-dessous. On donnera une brève définition de la signification des tables et des colonnes



1. Table **CLIENT** : chaque ligne décrit un client; les colonnes décrivent successivement le numéro du client (**NCLI**), son nom (**NOM**), son adresse (**ADRESSE**), sa localité ou gouvernorat(**LOCALITE**), sa catégorie (**CAT**) et l'état de son compte (**COMPTE**). L'identifiant primaire est constitué de **NCLI**. La colonne **CAT** est facultative.
2. Table **PRODUIT** : chaque ligne décrit un produit; les colonnes décrivent successivement le numéro du produit (**NPRO**), son libellé (**LIBELLE**), son prix unitaire (**PRIX**) et la quantité restant en stock (**QSTOCK**). NPRO est l'identifiant primaire.
3. Table **COMMANDE** : chaque ligne décrit une commande passée par un client; les colonnes décrivent successivement le numéro de la commande (**NCOM**), le numéro du client qui a

apcpedagogie.com

passé la commande (**NCLI**) et la date de la commande (**DATECOM**). **NCOM** est l'identifiant primaire de la table. **NCLI** est une clé étrangère vers la table CLIENT.

4. Table **DETAIL** : chaque ligne représente un détail d'une commande; les colonnes décrivent successivement le numéro de la commande à laquelle le détail appartient (**NCOM**), le numéro du produit commandé (**NPRO**) et la quantité commandée (**QCOM**). L'identifiant primaire est constitué de **NCOM** et **NPRO**. **NCOM** et **NPRO** sont en outre chacune une clé étrangère respectivement vers les tables **COMMANDE** et **PRODUIT**.

- ## Étapes de réalisation

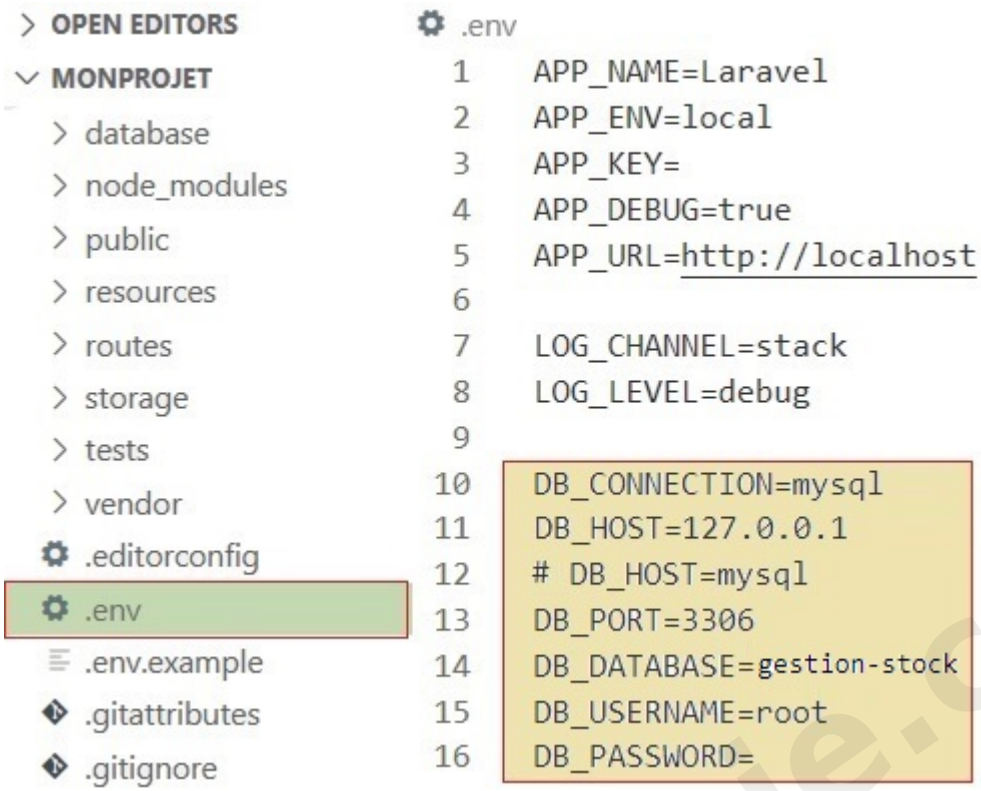
  - **Étape 1: Créez le projet Laravel 8**

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

  - **Étape 2: Configuration 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`
    - 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_` .

apcpedagogie.com



- 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 **gestion-stock** , 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.

- **Étape 3: Créer la migration**
  - Utilisez PhpMyAdmin pour créer une nouvelle base de données "gestion-stock"
  - Créez une migration pour la table "produits" en utilisant la commande Laravel: **`php artisan make:migration create_produits_table --create=produits`**
  - Après cette commande, vous trouverez un fichier dans le chemin suivant "**/database/migrations/**" et vous devez mettre le code ci-dessous dans votre fichier de migration pour créer la table des "produits".

```
<?php

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

class CreateProduitsTable extends Migration
```

```
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('produits', function (Blueprint $table) {
            $table->string('npro')->primary();
            $table->string('libelle');
            $table->float('prix', 12, 3);
            $table->float('qstock', 12, 3);
            $table->text('description');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('produits');
    }
}
```

- Vous devez maintenant exécuter cette migration en suivant la commande: **`php artisan migrate`**

```
PS C:\wamp64\www\laravel\blog> php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated:  2014_10_12_000000_create_users_table (107.50ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated:  2014_10_12_100000_create_password_resets_table (100.33ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated:  2019_08_19_000000_create_failed_jobs_table (120.06ms)
Migrating: 2021_01_28_134935_create_produits_table
Migrated:  2021_01_28_134935_create_produits_table (121.66ms)
PS C:\wamp64\www\laravel\blog>
```

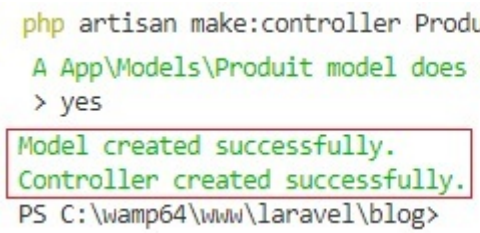
- **Étape 4: Ajouter une route de ressources**

- Pour ajouter une route ouvrez votre fichier "**routes/web.php**" et ajoutez la route suivante.

```
use App\Http\Controllers\ProduitController;
Route::resource('produits', ProduitController::class);
```

- **Étape 5: Ajouter un contrôleur et un modèle**

  - Pour créer un nouveau contrôleur en tant que **ProduitController** exécutez la commande suivante: **`php artisan make: controller ProduitController --resource --model=Produit`**



  - Après la commande ci-dessous, vous trouverez deux nouveaux fichiers:

1. **Premier fichier**

- Un nouveau fichier dans ce chemin "**app/Http/Controllers/ProduitController.php**" qui contient sept méthodes qui sont listées ci-dessous.
  1)index() **2)**create() **3)**store() **4)**show() **5)**edit() **6)**update() **7)**destroy()

```
<?php

namespace App\Http\Controllers;

use App\Models\Produit;
use Illuminate\Http\Request;

class ProduitController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $produits = Produit::latest()->paginate(4);
```

apcpedagogie.com

```
        return view('produits.index',compact('produits'))
            ->with('i', (request()->input('page', 1) - 1) * 4);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('produits.create');
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $request->validate([
            'npro' => 'required',
            'libelle' => 'required',
            'prix' => 'required',
            'qstock' => 'required',
            'description' => 'required',
        ]);

        Produit::create($request->all());

        return redirect()->route('produits.index')
                        ->with('success','Produit créé avec succès.');
    }

    /**
     * Display the specified resource.
     *
     * @param  \App\Models\Produit  $produit
     * @return \Illuminate\Http\Response
     */
    public function show(Produit $produit)
    {
```

apcpedagogie.com

```
        return view('produits.show',compact('produit'));
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param  \App\Models\Produit  $produit
     * @return \Illuminate\Http\Response
     */
    public function edit(Produit $produit)
    {
        return view('produits.edit',compact('produit'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \App\Models\Produit  $produit
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, Produit $produit)
    {
        $request->validate([
            'npro' => 'required',
            'libelle' => 'required',
            'prix' => 'required',
            'qstock' => 'required',
            'description' => 'required',
        ]);

        $produit->update($request->all());

        return redirect()->route('produits.index')
                        ->with('success','Produit mis à jour avec succès');
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  \App\Models\Produit  $produit
     * @return \Illuminate\Http\Response
     */
    public function destroy(Produit $produit)
```

apcpedagogie.com

```
    {
        $produit->delete();

        return redirect()->route('produits.index')
                        ->with('success','Produit supprimé avec succès');
    }
}
```

- **Deuxième fichier**
  - Notre deuxième fichier, vous trouverez dans "**app/Models/Product.php**"
  - Modifiez ce fichier (**Product.php**) comme suit:

```
<?php

namespace App\Models;

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

class Produit extends Model
{
    use HasFactory;
    public $primaryKey = 'npro';
    public $incrementing = false;
    protected $fillable = [
        'npro', 'libelle', 'prix', 'qstock', 'description'
    ];
}
```

- **Étape 6: Ajouter des fichiers Blade**
  - Dans cette étape, nous devons créer uniquement les fichiers **Blades**. Nous devons donc principalement créer un fichier de mise en page, puis créer un nouveau dossier "**produits**", puis créer des fichiers de Blade de l'application CRUD.
  - Donc, finalement, vous devez créer le fichier de Blade ci-dessous:

    1) layout.blade.php **2)** index.blade.php **3)** create.blade.php **4)** edit.blade.php **5)** show.blade.php
  - Alors créons simplement le fichier suivant et mettons le code ci-dessous

apcpedagogie.com

1. **layout.blade.php**

```
<!DOCTYPE html>
<html>
<head>
    <title>Exemple d'une application CRUD Laravel 8 - apcpedagogie.com</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css"
rel="stylesheet">
<style>
body{
    background-color: #FBFEFF;
}
</style>
</head>
<body>

<div class="container p-4 mt-5 border border-primary shadow-lg p-3 mb-5 rounded">
    @yield('content')
</div>

</body>
</html>
```

- **index.blade.php**

```
@extends('produits.layout')

@section('content')
    <div class="row">
        <div class="col-lg-12 margin-tb">
            <div class="float-start">
                <h2>Exemple d'une application CRUD Laravel 8 - apcpedagogie.com</h2>
            </div>
            <div class="float-end">
                <a class="btn btn-outline-success" href="{{ route('produits.create') }}"> Créer
un nouveau produit</a>
            </div>
        </div>
    </div>
```

apcpedagogie.com

```
    @if ($message = Session::get('success'))
        <div class="alert alert-success">
            <p>{{ $message }}</p>
        </div>
    @endif

    <table class="table table-bordered">
        <tr>
            <th>Numéro</th>
            <th>Libellé</th>
            <th>Prix</th>
            <th>Quantité</th>
            <th>Description</th>
            <th width="280px">Action</th>
        </tr>
        @foreach ($produits as $produit)
        <tr>

            <td>{{ $produit->npro }}</td>
            <td>{{ $produit->libelle }}</td>
            <td>{{ $produit->prix }}</td>
            <td>{{ $produit->qstock }}</td>
            <td>{{ $produit->description }}</td>
            <td>
                <form action="{{ route('produits.destroy',$produit->npro) }}" method="POST">

                    <a class="btn btn-outline-primary" href="{{ route('produits.show',$produit-
>npro) }}">Montrer</a>

                    <a class="btn btn-outline-success" href="{{ route('produits.edit',$produit-
>npro) }}">Éditer</a>

                    @csrf
                    @method('DELETE')

                    <button type="submit" class="btn btn-outline-danger">Supprimer</button>
                </form>
            </td>
        </tr>
        @endforeach
    </table>
    <div class="d-flex justify-content-center pagination-lg">
    {!! $produits->links('pagination::bootstrap-4') !!}
```

apcpedagogie.com

```
      </div>
@endsection
```

- **create.blade.php**

```
@extends('produits.layout')

@section('content')
<div class="row">
    <div class="col-lg-12 margin-tb">
        <div class="float-start">
            <h2>Ajouter un nouveau produit</h2>
        </div>
        <div class="float-end">
            <a class="btn btn-outline-primary" href="{{ route('produits.index') }}"> Retour</a>
        </div>
    </div>
</div>

@if ($errors->any())
    <div class="alert alert-danger">
        <strong>Oups! </strong> Il y a eu des problèmes avec votre entrée.<br><br>
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif

<form action="{{ route('produits.store') }}" method="POST">
    @csrf

     <div class="row">
        <div class="col-xs-6 col-sm-6 col-md-6">
            <div class="form-group">
                <strong>Numéro produit:</strong>
                <input type="text" name="npro" class="form-control" placeholder="Saisir un
numéro">
            </div>
        </div>
        <div class="col-xs-6 col-sm-6 col-md-6">
```

apcpedagogie.com

```
            <div class="form-group">
                <strong>Libellé:</strong>
                <input type="text" name="libelle" class="form-control" placeholder="Saisir un
libellé">
            </div>
        </div>
    </div>
    <div class="row">
        <div class="col-xs-6 col-sm-6 col-md-6">
            <div class="form-group">
                <strong>Prix:</strong>
                <input type="text" name="prix" class="form-control" placeholder="Saisir un
prix">
            </div>
        </div>
        <div class="col-xs-6 col-sm-6 col-md-6">
            <div class="form-group">
                <strong>Qté stock:</strong>
                <input type="text" name="qstock" class="form-control" placeholder="Saisir un
stock">
            </div>
        </div>
    </div>


    <div class="row">
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Detail:</strong>
                <textarea class="form-control" style="height:150px" name="description"
placeholder="Description"></textarea>
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12 text-center">
                <button type="submit" class="btn btn-primary">Soumettre</button>
        </div>
    </div>

</form>
@endsection
```

- **edit.blade.php**

apcpedagogie.com

```
@extends('produits.layout')

@section('content')
    <div class="row">
        <div class="col-lg-12 margin-tb">
            <div class="float-start">
                <h2>Modifier le produit</h2>
            </div>
            <div class="float-end">
                <a class="btn btn-outline-primary" href="{{ route('produits.index') }}">
Retour</a>
            </div>
        </div>
    </div>

    @if ($errors->any())
        <div class="alert alert-danger">
            <strong>Oups! </strong> Il y a eu des problèmes avec votre entrée.<br><br>
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif

    <form action="{{ route('produits.update',$produit->npro) }}" method="POST">
        @csrf
        @method('PUT')
        <div class="row">
        <div class="col-xs-6 col-sm-6 col-md-6">
            <div class="form-group">
                <strong>Numéro produit:</strong>
                <input type="text" name="npro" value="{{ $produit->npro }}"class="form-control"
placeholder="Saisir un numéro">
            </div>
        </div>
        <div class="col-xs-6 col-sm-6 col-md-6">
            <div class="form-group">
                <strong>Libellé:</strong>
                <input type="text" name="libelle" value="{{ $produit->libelle }}" class="form-
control" placeholder="Saisir un libellé">
            </div>
        </div>
```

apcpedagogie.com

```
        </div>
        <div class="row">
        <div class="col-xs-6 col-sm-6 col-md-6">
            <div class="form-group">
                <strong>Prix:</strong>
                <input type="text" name="prix" value="{{ $produit->prix }}" class="form-control"
placeholder="Saisir un prix">
            </div>
        </div>
        <div class="col-xs-6 col-sm-6 col-md-6">
            <div class="form-group">
                <strong>Qté stock:</strong>
                <input type="text" name="qstock" value="{{ $produit->qstock }}" class="form-
control" placeholder="Saisir un stock">
            </div>
        </div>
    </div>
         <div class="row">
            <div class="col-xs-12 col-sm-12 col-md-12">
                <div class="form-group">
                    <strong>Détail:</strong>
                    <textarea class="form-control" style="height:150px" name="description"
placeholder="Détail">{{ $produit->description }}</textarea>
                </div>
            </div>
            <div class="col-xs-12 col-sm-12 col-md-12 text-center pt-4">
              <button type="submit" class="btn btn-primary">Soumettre</button>
            </div>
        </div>

    </form>
@endsection
```

- **show.blade.php**

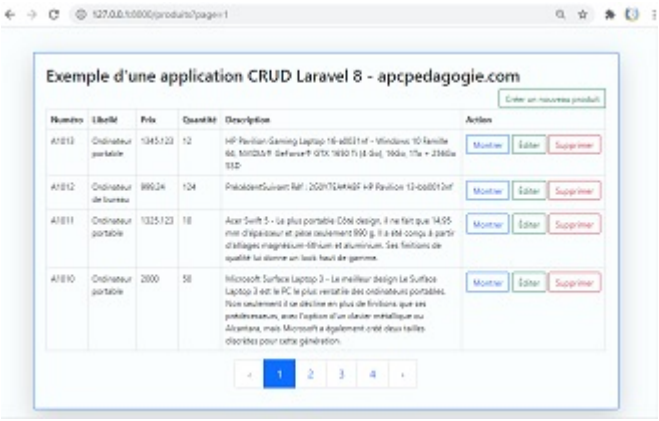
```
@extends('produits.layout')
@section('content')
    <div class="row">
        <div class="col-lg-12 margin-tb">
            <div class="float-start">
                <h2> Afficher le produit</h2>
            </div>
```

```
            <div class="float-end">
                <a class="btn btn-outline-primary" href="{{ route('produits.index') }}">
Retour</a>
            </div>
        </div>
    </div>

    <div class="row">
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Numéro:</strong>
                {{ $produit->npro }}
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Libellé:</strong>
                {{ $produit->libelle }}
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Prix:</strong>
                {{ $produit->prix }}
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Quantité en stock:</strong>
                {{ $produit->qstock }}
            </div>
        </div>
        <div class="col-xs-12 col-sm-12 col-md-12">
            <div class="form-group">
                <strong>Description:</strong>
                {{ $produit->description }}
            </div>
        </div>
    </div>
@endsection
```

- **Exemple d'écran**(Cliquez sur l'image pour agrandir)

apcpedagogie.com



Cours Laravel

Source : https://apcpedagogie.com/les-operations-crud-avec-eloquent-laravel-3-3 · téléchargé le 03/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com