Les opérations CRUD avec Eloquent Laravel 2-3

1. Objectifs

- Être capable de générer une application CRUD très simple dès le départ dans Laravel 8?

2. Étapes de réalisation

- **Étape - 1**: Créer un nouveau projet Laravel 8
- **Étape - 2**: Configurer la base de données
- **Étape - 3**: Créer une migration
- **Étape - 4**: Générer le fichier de seeds
- **Étape - 5**: Ajouter une route de ressources
- **Étape - 6**: Ajouter un contrôleur et un modèle
- **Étape - 7**: Ajouter des fichiers de Blades
- **Étape - 8**: Exécuter l'application

1. Étape - 1: Créer un nouveau projet Laravel 8

- Dans la toute première étape, nous devons créer l'application **laravel8** dans votre système local en utilisant la commande composer suivante dans votre terminal.
- `composer create-project --prefer-dist laravel/laravel laravelApplicationCRUD`

2. Étape - 2: Configurer la base de données

- Soit la base de données **BANQUE** contenant les tables suivantes :
  - **agences** (*Num_Agence, Nom, Ville, Actif)
  - **clients** (*Num_Client, Nom, Prenom, Ville)
  - **comptes** (*Num_Compte, Num_Agence#, Num_Client#, Solde)
  - **emprunts** (*Num_Emprunt, Num_Agence#, Num_Client#, Montant)
- Les clefs primaires sont précédées d'une étoile (*) et les clefs étrangères sont suivies d'un astérisque (#).
- Ouvrir simplement le fichier `.env` de l'application **laravel** et effectuer les modifications suivantes liées à la base de données.

apcpedagogie.com

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=banque
DB_USERNAME=root
DB_PASSWORD=
```

3. **Étape - 3: Créer une migration**

    1. **Créer les fichiers de migration**

        1. **agences**

            - Maintenant, vous devez créer une migration à l'aide de la commande artisan suivante dans le répertoire racine de votre projet.
            - Créez une migration pour la table "agences" en utilisant la commande Laravel: `php artisan make:migration create_agences_table --create=agences`
            - 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 "**agences**".

        2. **clients**

            - Créez une migration pour la table "clients" en utilisant la commande Laravel: `php artisan make:migration create_clients_table --create=clients`

        3. **comptes**

            - Créez une migration pour la table "comptes" en utilisant la commande Laravel: `php artisan make:migration create_comptes_table --create=comptes`

apcpedagogie.com

4. **emprunts**

    - Créez une migration pour la table "emprunts" en utilisant la commande Laravel: `php artisan make:migration create_emprunts_table --create=emprunts`

2. **Changer le type de base de données**

> *Pour changer le type **MyISAM**, définir **InnoDB** comme moteur de stockage par défaut de l'application via **config/database.php** afin que de nouvelles tables soient créées dans **InnoDB**.* `'connections' => ['mysql' => ['engine' => 'InnoDB',`

3. **Exécuter la migration**

    - Si la tache précédente est réalisée convenablement le fichier "**/database/migrations/**" sera modifié comme suit:

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

    - Vous devez ainsi vous retrouver avec six tables dans votre base ainsi que la table migrations :

    - Vous devez aussi vous retrouver avec quatre tables liées entre eux :

4. **Étape - 4: Générer le fichier de seeds**

1. **agences**

    - `php artisan make:seeder AgencesTableSeeder`
    - Le fichier ainsi généré sera placé dans le dossier **database\seeds**, sous le dossier du projet, et contiendra le code suivant :

apcpedagogie.com

```
<?php
namespace Database\Seeders;
use Illuminate\Database\Seeder;
class AgencesTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        //
    }
}
```

2. Créer le fichier seeder avec la commande suivante : `php artisan make:seeder AgencesTableSeeder`
3. Éditez ce fichier pour y ajouter les données désirées à l'aide de \DB::table().

```
public function run()
    {
        \DB::table('agences')->insert([
            [
                'Num_Agence' => 'A12APCPEDAGOGIE',
                'Nom_Agence' => 'A12APCPEDAGOGIE Siège',
                'Ville_Agence' => 'Nabeul Centre',
                'Actif_Agence'=> '1245.320',
                'created_at' => now(),
                'updated_at' => now(),
            ],

            [
                'Num_Agence' => 'A13APCPEDAGOGIE',
                'Nom_Agence' => 'A13APCPEDAGOGIE Filiale',
                'Ville_Agence' => 'Darchaabene elfehri',
                'Actif_Agence'=> '1245.320',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'Num_Agence' => 'A14APCPEDAGOGIE',
```

apcpedagogie.com

```
                'Nom_Agence' => 'A14APCPEDAGOGIE Filiale',
                'Ville_Agence' => 'Hammamet',
                'Actif_Agence'=> '1245.320',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'Num_Agence' => 'A15APCPEDAGOGIE',
                'Nom_Agence' => 'A15APCPEDAGOGIE Filiale',
                'Ville_Agence' => 'Bni khiar',
                'Actif_Agence'=> '1245.320',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'Num_Agence' => 'A16APCPEDAGOGIE',
                'Nom_Agence' => 'A16APCPEDAGOGIE Filiale',
                'Ville_Agence' => 'Korba',
                'Actif_Agence'=> '1245.320',
                'created_at' => now(),
                'updated_at' => now(),
            ],

        ]);
    }
}
```

4. Puis exécuter la commande suivante: **`php artisan db:seed --class=AgencesTableSeeder`**

5. **clients**

   - Créer le fichier seeder avec la commande suivante : `php artisan make:seeder ClientsTableSeeder`
   - Éditez ce fichier pour y ajouter les données désirées à l'aide de \DB::table().

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;
```

apcpedagogie.com

```
class ClientsTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        \DB::table('clients')->insert([
            [
                'Num_Client' => 'C4010001',
                'Nom_Client' => 'Mohamed Salah',
                'Prenom_Client' => 'FOURATI',
                'Ville_Client'=> 'Nabeul',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'Num_Client' => 'C4010002',
                'Nom_Client' => 'Nadia',
                'Prenom_Client' => 'BEN ARFA',
                'Ville_Client'=> 'Nabeul',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'Num_Client' => 'C4010003',
                'Nom_Client' => 'Sadok',
                'Prenom_Client' => 'HARRATH',
                'Ville_Client'=> 'Nabeul',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
                'Num_Client' => 'C4010004',
                'Nom_Client' => 'Moufida',
                'Prenom_Client' => 'SFAR',
                'Ville_Client'=> 'Nabeul',
                'created_at' => now(),
                'updated_at' => now(),
            ],
            [
```

apcpedagogie.com

```
                'Num_Client' => 'C4010005',
                'Nom_Client' => 'Montassar',
                'Prenom_Client' => 'JMAA',
                'Ville_Client'=> 'Nabeul',
                'created_at' => now(),
                'updated_at' => now(),
            ],

        ]);
    }
    }
```

6. Puis exécuter la commande suivante: **`php artisan db:seed --class=ClientsTableSeeder`**

7. **comptes**

   - Créer le fichier seeder avec la commande suivante : `php artisan make:seeder ComptesTableSeeder`
   - Éditez ce fichier pour y ajouter les données désirées à l'aide de \DB::table().

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

class ComptesTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        \DB::table('comptes')->insert([
            [

                'Num_Compte' => 'Compte000001',
                'Solde_Compte' => '1245.320',
                'created_at' => now(),
```

apcpedagogie.com

```
                'updated_at' => now(),
                'Num_Client' => 'C4010001',
                'Num_Agence' => 'A12APCPEDAGOGIE',
            ],
            [
                'Num_Compte' => 'Compte000002',
                'Solde_Compte' => '3245.000',
                'created_at' => now(),
                'updated_at' => now(),
                'Num_Client' => 'C4010002',
                'Num_Agence' => 'A13APCPEDAGOGIE',
            ],
            [
                'Num_Compte' => 'Compte000003',
                'Solde_Compte' => '1245.320',
                'created_at' => now(),
                'updated_at' => now(),
                'Num_Client' => 'C4010002',
                'Num_Agence' => 'A12APCPEDAGOGIE',
            ],
            [
                'Num_Compte' => 'Compte000004',
                'Solde_Compte' => '4214.000',
                'created_at' => now(),
                'updated_at' => now(),
                'Num_Client' => 'C4010001',
                'Num_Agence' => 'A15APCPEDAGOGIE',
            ],
            [
                'Num_Compte' => 'Compte000005',
                'Solde_Compte' => '1245.320',
                'created_at' => now(),
                'updated_at' => now(),
                'Num_Client' => 'C4010005',
                'Num_Agence' => 'A15APCPEDAGOGIE',
            ],

        ]);
    }
}
```

8. Puis exécuter la commande suivante: **`php artisan make:seeder ComptesTableSeeder`**

apcpedagogie.com

9. **emprunts**

- Créer le fichier seeder avec la commande suivante : `php artisan make:seeder EmpruntsTableSeeder`

```
<?php

namespace Database\Seeders;

use Illuminate\Database\Seeder;

class EmpruntsTableSeeder extends Seeder
{
    /**
     * Run the database seeds.
     *
     * @return void
     */
    public function run()
    {
        \DB::table('Emprunts')->insert([
            [

                'Num_Emprunt' => 'Emprunt000001',
                'Montant_Emprunt' => '1245.320',
                'created_at' => now(),
                'updated_at' => now(),
                'Num_Client' => 'C4010001',
                'Num_Agence' => 'A12APCPEDAGOGIE',
            ],
            [
                'Num_Emprunt' => 'Emprunt000002',
                'Montant_Emprunt' => '3245.000',
                'created_at' => now(),
                'updated_at' => now(),
                'Num_Client' => 'C4010002',
                'Num_Agence' => 'A13APCPEDAGOGIE',
            ],
            [
                'Num_Emprunt' => 'Emprunt000003',
                'Montant_Emprunt' => '1245.320',
                'created_at' => now(),
```

apcpedagogie.com

```
                'updated_at' => now(),
                'Num_Client' => 'C4010002',
                'Num_Agence' => 'A12APCPEDAGOGIE',
            ],
            [
                'Num_Emprunt' => 'Emprunt000004',
                'Montant_Emprunt' => '4214.000',
                'created_at' => now(),
                'updated_at' => now(),
                'Num_Client' => 'C4010001',
                'Num_Agence' => 'A15APCPEDAGOGIE',
            ],
            [
                'Num_Emprunt' => 'Emprunt000005',
                'Montant_Emprunt' => '1245.320',
                'created_at' => now(),
                'updated_at' => now(),
                'Num_Client' => 'C4010005',
                'Num_Agence' => 'A15APCPEDAGOGIE',
            ],

        ]);
    }
}
```

10. Puis exécuter la commande suivante: **`php artisan make:seeder EmpruntsTableSeeder`**

11. **Étape - 5: Ajouter une route de ressources**

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

```
Route::resource('agences', App\Http\Controllers\AgenceController::class);
Route::resource('clients', App\Http\Controllers\ClientController::class);
Route::resource('comptes', App\Http\Controllers\CompteController::class);
Route::resource('emprunts', App\Http\Controllers\EmpruntController::class);
```

12. **Étape - 6: Ajouter un contrôleur et un modèle**

    - Dans cette étape, nous allons créer nos modèles "**TableController**" et "**Table**" en utilisant les commandes artisan suivantes dans le terminal.
    - `php artisan make:controller AgenceController --resource --model=Agence`

apcpedagogie.com

- `php artisan make:controller ClientController --resource --model=Client`
- `php artisan make:controller CompteController --resource --model=Compte`
- `php artisan make:controller EmpruntController --resource --model=Emprunt`

- Après l'exécution des commandes ci-dessous, vous trouverez deux nouveaux fichiers pour chaque classe:
- 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()

- **Fichier `AgenceController`**

```
<?php

namespace App\Http\Controllers;

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

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

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

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
```

apcpedagogie.com

```
     */
    public function create()
    {
        return view('agences.create');
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $request->validate([
            'Num_Agence' => 'required',
            'Nom_Agence' => 'required',
            'Ville_Agence' => 'required',
            'Actif_Agence' => 'required',
            'created_at' => 'required',
        ]);

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

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

    /**
     * Display the specified resource.
     *
     * @param  \App\Models\Agence  $agence
     * @return \Illuminate\Http\Response
     */
    public function show(Agence $agence)
    {
        return view('agences.show',compact('agence'));
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param  \App\Models\Agence  $agence
```

apcpedagogie.com

```
     * @return \Illuminate\Http\Response
     */
    public function edit(Agence $agence)
    {
        return view('agences.edit',compact('agence'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \App\Models\Agence  $agence
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, Agence $agence)
    {
        $request->validate([
            'Num_Agence' => 'required',
            'Nom_Agence' => 'required',
            'Ville_Agence' => 'required',
            'Actif_Agence' => 'required',
            'updated_at' => 'required',
        ]);

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

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

    /**
     * Remove the specified resource from storage.
     *
     * @param  \App\Models\Agence  $agence
     * @return \Illuminate\Http\Response
     */
    public function destroy(Agence $agence)
    {
        $agence->delete();

        return redirect()->route('agences.index')
                        ->with('success','Agence supprimé avec succès');
```

apcpedagogie.com

```
    }
}
```

13. **Fichier Model `Agence.php`**

```
<?php

namespace App\Models;

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

class Agence extends Model
{
    use HasFactory;
    public $primaryKey = 'Num_Agence';
    public $incrementing = false;
    protected $fillable = [
        'Num_Agence', 'Nom_Agence', 'Ville_Agence', 'Actif_Agence',
'created_at'
    ];
}
```

14. **Étape - 7: Ajouter des fichiers de Blades**

- Il faut commencer par la création d'une structure de vues
- Avant de commencer la création des views, créer leur structure Views en créant les dossiers et les fichiers dans l'exemple suivant:

– resources — views — **layouts** ——- default.blade.php — **elements** ——- head.blade.php ——- header.blade.php ——- footer.blade.php — **pages** ——- home.blade.php

— **Agence** ——- index.blade.php ——- edit.blade.php ——- show.blade.php — **client** ——- index.blade.php ——- edit.blade.php ——- show.blade.php — **emprunt** ——- index.blade.php ——- edit.blade.php ——- show.blade.php — **compte** ——- index.blade.php ——- edit.blade.php ——- show.blade.php

- Fichier index.blade.php de l'agence

apcpedagogie.com

```
@extends('layouts.default')
@section('content')
    <div class="row">
        <div class="col-lg-12">
            <div class="float-start">
                <h2>Gestion des agences</h2>
            </div>
            <div class="float-end mb-2" style="margin-right: -30px">
                <a class="btn btn-success" href="{{
route('agences.create') }}"> Ajouter une nouvelle agence</a>
            </div>
        </div>
    </div>

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

    <table class="table table-bordered">
        <tr>
            <th>Num</th>
            <th>Nom</th>
            <th>Ville</th>
            <th>Actif</th>
            <th width="240px">Action</th>
        </tr>
        @foreach ($agences as $agence)
        <tr>
            <td>{{ $agence->Num_Agence }}</td>
            <td>{{ $agence->Nom_Agence }}</td>
            <td>{{ $agence->Ville_Agence }}</td>
            <td>{{ $agence->Actif_Agence}}</td>

            <td>
                <form action="{{ route('agences.destroy',$agence-
>Num_Agence) }}" method="POST">

                    <a class="btn btn-info" href="{{
route('agences.show',$agence->Num_Agence) }}">Show</a>
```

apcpedagogie.com

```
                    <a class="btn btn-primary" href="{{
route('agences.edit',$agence->Num_Agence) }}">Edit</a>

                    @csrf
                    @method('DELETE')

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

15. **Étape - 8: Exécuter l'application**

Sommaire du cours Laravel

Source : https://apcpedagogie.com/les-operations-crud-avec-eloquent-laravel-2-3 · téléchargé le 08/11/2026 · © APC Pédagogie · apcpedagogie.com