Le mécanisme CRUD dans Laravel

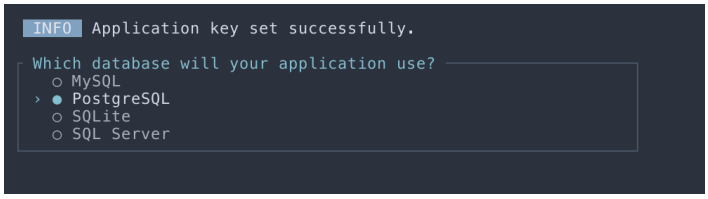
Laravel est un framework PHP populaire qui facilite la création d'applications web robustes et performantes. L'un des concepts fondamentaux de Laravel est la gestion des opérations CRUD (Create, Read, Update, Delete), qui permettent de manipuler les données dans une base de données de manière standardisée. Voici un aperçu du mécanisme CRUD dans Laravel :

Prérequis

- Assurez-vous que vous avez Laravel installé sur votre machine.
- Créez une nouvelle application Laravel si vous ne l'avez pas déjà fait :

  ```
  composer create-project --prefer-dist laravel/laravel <nom_de_projet>
  cd <nom_de_projet>
  ```

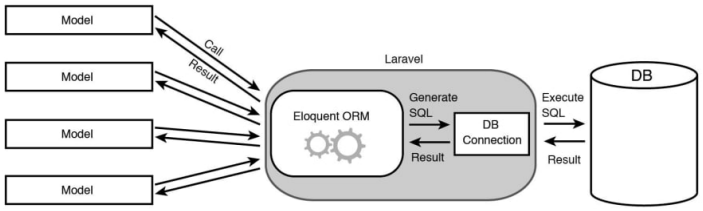
Étape 1: Configuration de la Base de Données



1. Configurez votre fichier `.env` pour la connexion à votre base de données :

   ```
   DB_CONNECTION=sqlite
   #   DB_HOST=127.0.0.1
   #   DB_PORT=3306
   #   DB_DATABASE=laravel
   #   DB_USERNAME=root
   #   DB_PASSWORD=
   ```

Étape 2: Création de la Migration et du Modèle



...

1. Créez une migration et un modèle pour les resources :

   ```
   php artisan make:model Resource -m
   ```

2. Éditez le fichier du modèle Resource et ajoutez:

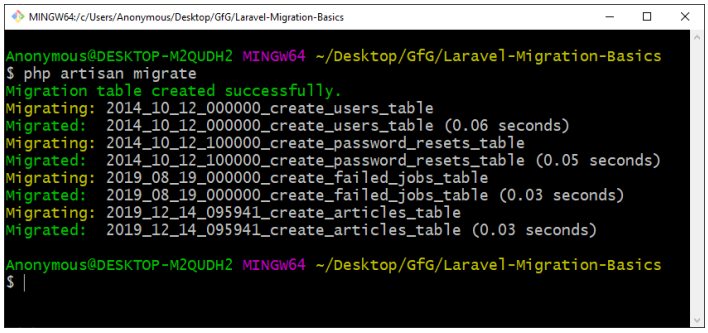
   ```
   /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'title',
        'body',
    ];
   ```

3. Ouvrez la migration générée dans `database/migrations/` et définissez les colonnes de la table `resources` :

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

   class CreateResourcesTable extends Migration
   {
       public function up()
       {
           Schema::create('resources', function (Blueprint $table) {
               $table->id();
               $table->string('title');
               $table->text('body');
               $table->timestamps();
           });
       }

       public function down()
       {
   ```

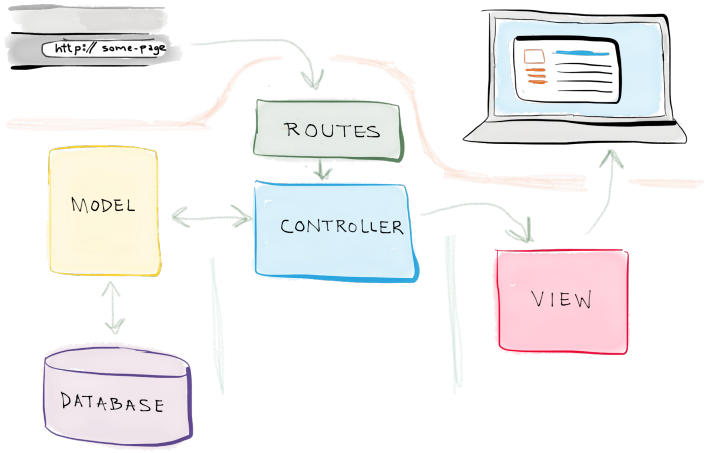
```
            Schema::dropIfExists('resources');
        }
    }
```

4. Exécutez la migration :



`php artisan migrate`

Étape 3: Création du Contrôleur et des Vues



1. Créez un contrôleur de type resource pour les resources :

   `php artisan make:controller ResourceController --resource`

2. Implémentez les méthodes du contrôleur dans `app/Http/Controllers/ResourceController.php` :

```
namespace App\Http\Controllers;

use App\Models\Resource;
use Illuminate\Http\Request;
```

```
class ResourceController extends Controller
{
```

2. **Lecture d'une ressource (Read)**

```
public function index()
{
    $resources = Resource::all();
    return view('resources.index', compact('resources'));
}

public function show(Resource $resource)
{
    return view('resources.show', compact('resource'));
}
```

1. **Création d'une nouvelle ressource (Create)**

```
public function create()
{
    return view('resources.create');
}

public function store(Request $request)
{
    $request->validate([
        'title' => 'required',
        'body' => 'required',
    ]);

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

    return redirect()->route('resources.index')
                     ->with('success', 'Resource created successfully.');
}
```

3. **Mise à jour d'une ressource (Update)**

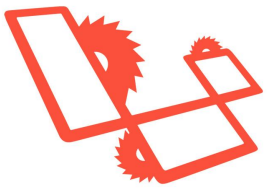
```
public function edit(Resource $resource)
{
    return view('resources.edit', compact('resource'));
}

public function update(Request $request, Resource $resource)
{
    $request->validate([
```

```
            'title' => 'required',
            'body' => 'required',
        ]);

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

        return redirect()->route('resources.index')
                         ->with('success', 'Resource updated successfully.');
    }
```

4. Suppression d'une ressource (Delete)

```
    public function destroy(Resource $resource)
    {
        $resource->delete();

        return redirect()->route('resources.index')
                         ->with('success', 'Resource deleted successfully.');
    }
}
```

Étape 4: Création des Vues



Créez les vues Blade pour afficher les resources. Créez un répertoire `resources` dans `resources/views` et ajoutez les fichiers suivants :

1. **Index (resources/views/resources/index.blade.php)** :

```
<!DOCTYPE html>
<html>
<head>
    <title>Resources</title>
</head>
<body>
    <h1>Resources</h1>
    <a href="{{ route('resources.create') }}">Create New Resource</a>
    @if ($message = Session::get('success'))
        <p>{{ $message }}</p>
```

```
    @endif
    <ul>
        @foreach ($resources as $resource)
            <li>
                <a href="{{ route('resources.show', $resource->id) }}">
                {{ $resource->title }}</a>
                <a href="{{ route('resources.edit', $resource->id) }}">Edit</a>
                <form action="{{ route('resources.destroy', $resource->id) }}"
                 method="POST" style="display:inline;">
                    @csrf
                    @method('DELETE')
                    <button type="submit">Delete</button>
                </form>
            </li>
        @endforeach
    </ul>
</body>
</html>
```

2. **Create (resources/views/resources/create.blade.php)** :

```
<!DOCTYPE html>
<html>
<head>
    <title>Create Resource</title>
</head>
<body>
    <h1>Create Resource</h1>
    @if ($errors->any())
        <div>
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif
    <form action="{{ route('resources.store') }}" method="POST">
        @csrf
        <div>
            <label>Title:</label>
            <input type="text" name="title" value="{{ old('title') }}">
        </div>
        <div>
            <label>Body:</label>
            <textarea name="body">{{ old('body') }}</textarea>
        </div>
```

```
        <div>
            <button type="submit">Submit</button>
        </div>
    </form>
    <a href="{{ route('resources.index') }}">Back to Resources</a>
</body>
</html>
```

3. **Show (resources/views/resources/show.blade.php)** :

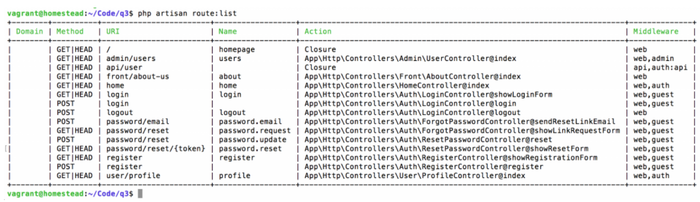
```
<!DOCTYPE html>
<html>
<head>
    <title>Show Resource</title>
</head>
<body>
    <h1>{{ $resource->title }}</h1>
    <p>{{ $resource->body }}</p>
    <a href="{{ route('resources.index') }}">Back to Resources</a>
</body>
</html>
```

4. **Edit (resources/views/resources/edit.blade.php)** :

```
<!DOCTYPE html>
<html>
<head>
    <title>Edit Resource</title>
</head>
<body>
    <h1>Edit Resource</h1>
    @if ($errors->any())
        <div>
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif
    <form action="{{ route('resources.update', $resource->id) }}" method="POST">
        @csrf
        @method('PUT')
        <div>
            <label>Title:</label>
            <input type="text" name="title" value="{{ $resource->title }}">
        </div>
        <div>
```

```
            <label>Body:</label>
            <textarea name="body">{{ $resource->body }}</textarea>
        </div>
        <div>
            <button type="submit">Submit</button>
        </div>
    </form>
    <a href="{{ route('resources.index') }}">Back to Resources</a>
</body>
</html>
```

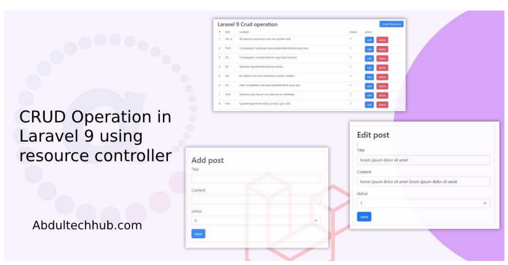
Étape 5: Définissez les routes dans `routes/web.php` :



```
vagrant@homestead:~/Code/q3$ php artisan route:list
+--------+----------+-------------------------+------------------+---------------------------------------------------------------------+---------------+
| Domain | Method   | URI                     | Name             | Action                                                              | Middleware    |
+--------+----------+-------------------------+------------------+---------------------------------------------------------------------+---------------+
|        | GET|HEAD | /                       | homepage         | Closure                                                             | web           |
|        | GET|HEAD | admin/users             | users            | App\Http\Controllers\Admin\UserController@index                     | web,admin     |
|        | GET|HEAD | api/user                |                  | Closure                                                             | api,auth:api  |
|        | GET|HEAD | front/about-us          | about            | App\Http\Controllers\Front\AboutController@index                    | web           |
|        | GET|HEAD | home                    | home             | App\Http\Controllers\HomeController@index                           | web,auth      |
|        | GET|HEAD | login                   | login            | App\Http\Controllers\Auth\LoginController@showLoginForm             | web,guest     |
|        | POST     | login                   |                  | App\Http\Controllers\Auth\LoginController@login                     | web,guest     |
|        | POST     | logout                  | logout           | App\Http\Controllers\Auth\LoginController@logout                    | web           |
|        | POST     | password/email          | password.email   | App\Http\Controllers\Auth\ForgotPasswordController@sendResetLinkEmail | web,guest   |
|        | GET|HEAD | password/reset          | password.request | App\Http\Controllers\Auth\ForgotPasswordController@showLinkRequestForm | web,guest  |
|        | POST     | password/reset          | password.update  | App\Http\Controllers\Auth\ResetPasswordController@reset             | web,guest     |
|        | GET|HEAD | password/reset/{token}  | password.reset   | App\Http\Controllers\Auth\ResetPasswordController@showResetForm     | web,guest     |
|        | GET|HEAD | register                | register         | App\Http\Controllers\Auth\RegisterController@showRegistrationForm   | web,guest     |
|        | POST     | register                |                  | App\Http\Controllers\Auth\RegisterController@register               | web,guest     |
|        | GET|HEAD | user/profile            | profile          | App\Http\Controllers\User\ProfileController@index                   | web,auth      |
+--------+----------+-------------------------+------------------+---------------------------------------------------------------------+---------------+
vagrant@homestead:~/Code/q3$
```

```php
use App\Http\Controllers\ResourceController;

Route::resource('resources', ResourceController::class);
```

Étape 6: Fonctionnalités CRUD de l'Application



- Lancez le serveur de développement :

  `php artisan serve`

- Accédez à votre application via `http://localhost:8000/resources`.
- Vous devriez pouvoir créer, lire, mettre à jour et supprimer des resources.

Conclusion

Le mécanisme CRUD de Laravel est conçu pour être intuitif et puissant, permettant aux développeurs de gérer les opérations de base de données avec un

minimum de code. En utilisant les fonctionnalités d'Eloquent ORM, des contrôleurs, des routes et des migrations, vous pouvez facilement créer, lire, mettre à jour et supprimer des enregistrements dans une application Laravel.