

Exercice CRUD Laravel

Voici un exercice complet pour créer un système CRUD (Create, Read, Update, Delete) en utilisant Laravel. Nous allons créer une application simple pour gérer des “Articles”.

Prérequis

- Créez une nouvelle application Laravel appelée “blog”.

Étape 1: Configuration de la Base de Données

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

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

2. Créez une base de données nommée `blog`.

Étape 2: Création de la Migration et du Modèle

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

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

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

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

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

    public function down()
}
```

```

    {
        Schema::dropIfExists('articles');
    }
}

```

3. Exécutez la migration :

```
php artisan migrate
```

Étape 3: Création du Contrôleur et des Vues

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

```
php artisan make:controller ArticleController --resource
```

2. Définissez les routes resource dans routes/web.php :

```
use App\Http\Controllers\ArticleController;

Route::resource('articles', ArticleController::class);
```

3. Implémentez les méthodes du contrôleur dans app/Http/Controllers/ArticleController.php :

```

namespace App\Http\Controllers;

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

class ArticleController extends Controller
{
    public function index()
    {
        $articles = Article::all();
        return view('articles.index', compact('articles'));
    }

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

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

```

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

return redirect()->route('articles.index')
    ->with('success', 'Article created successfully.');
}

public function show(Article $article)
{
    return view('articles.show', compact('article'));
}

public function edit(Article $article)
{
    return view('articles.edit', compact('article'));
}

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

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

    return redirect()->route('articles.index')
        ->with('success', 'Article updated successfully.');
}

public function destroy(Article $article)
{
    $article->delete();

    return redirect()->route('articles.index')
        ->with('success', 'Article deleted successfully.');
}

```

Étape 4: Création des Vues

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

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

```

<!DOCTYPE html>
<html>
```

```

<head>
    <title>Articles</title>
</head>
<body>
    <h1>Articles</h1>
    <a href="{{ route('articles.create') }}">Create New Article</a>
    @if ($message = Session::get('success'))
        <p>{{ $message }}</p>
    @endif
    <ul>
        @foreach ($articles as $article)
            <li>
                <a href="{{ route('articles.show', $article->id) }}">{{ $article->title }}</a>
                <a href="{{ route('articles.edit', $article->id) }}">Edit</a>
                <form action="{{ route('articles.destroy', $article->id) }}" method="POST"
                      @csrf
                      @method('DELETE')
                    <button type="submit">Delete</button>
                </form>
            </li>
        @endforeach
    </ul>
</body>
</html>

```

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

```

<!DOCTYPE html>
<html>
<head>
    <title>Create Article</title>
</head>
<body>
    <h1>Create Article</h1>
    @if ($errors->any())
        <div>
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif
    <form action="{{ route('articles.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('articles.index') }}">Back to Articles</a>
</body>
</html>
```

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

```

<!DOCTYPE html>
<html>
<head>
    <title>Show Article</title>
</head>
<body>
    <h1>{{ $article->title }}</h1>
    <p>{{ $article->body }}</p>
    <a href="{{ route('articles.index') }}">Back to Articles</a>
</body>
</html>
```

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

```

<!DOCTYPE html>
<html>
<head>
    <title>Edit Article</title>
</head>
<body>
    <h1>Edit Article</h1>
    @if ($errors->any())
        <div>
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif
    <form action="{{ route('articles.update', $article->id) }}" method="POST">
        @csrf
```

```

@method('PUT')
<div>
    <label>Title:</label>
    <input type="text" name="title" value="{{ $article->title }}">
</div>
<div>
    <label>Body:</label>
    <textarea name="body">{{ $article->body }}</textarea>
</div>
<div>
    <button type="submit">Submit</button>
</div>
</form>
<a href="{{ route('articles.index') }}">Back to Articles</a>
</body>
</html>

```

Étape 5: Test de l'Application

- Lancez le serveur de développement :

```
php artisan serve
```

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

Conclusion

Ce guide vous a montré comment créer une application CRUD simple avec Laravel. Vous pouvez maintenant étendre cette application en ajoutant des fonctionnalités supplémentaires telles que l'authentification, la validation avancée, et d'autres relations de base de données pour rendre l'application plus robuste et utile.