

TP N° 8

Manipuler l'ORM Eloquent

ISTA TINGHIR

Module: M205

Réalisée par: Groupe 8

le: 19 mai 2023

Développeur en Back-End

Introduction

L'ORM (Object Relational Mapper) Eloquent de Laravel est une manière élégante, belle et efficace d'interagir et de gérer les bases de données. L'équipe Laravel a créé Eloquent pour faciliter l'interaction et la communication avec les bases de données et fournir un moyen facile à utiliser pour résoudre tous les problèmes liés aux bases de données et au développement. Laravel Eloquent offre la liberté d'écrire des codes personnalisables, bien formatés, efficaces, faciles à lire, bien documentés et durables en suivant les normes et les meilleures pratiques de l'industrie.

1. Aperçu de Laravel Eloquent (comment ça marche) ?

Laravel Eloquent vous donne la liberté de créer des schémas de base de données complexes et des interactions avec la base de données sans écrire et utiliser des requêtes SQL complexes ou longues, car les requêtes SQL sont fastidieuses et longues à apprendre et à coder dans une application complexe.

2. Qu'est-ce qu'un modèle Laravel ?

Laravel Model est un objet PHP unique qui représente un schéma de base de données dans une base de données. Le modèle vous aide à interagir et à communiquer avec un schéma de base de données spécifique. En utilisant le modèle Eloquent, vous pouvez exécuter de nombreuses fonctionnalités liées à la base de données, telles que la création, la mise à jour, la lecture et la suppression d'enregistrements.

3. MANIPULATION DES MODÈLES

3-1 Génération des classes de modèle

Pour commencer, nous allons créer un modèle Eloquent. Les modèles se trouvent généralement dans le répertoire **app\Models** et étendent la classe **Illuminate\Database\Eloquent\Model**. Vous pouvez utiliser la commande Artisan **make:model** pour générer un nouveau modèle :

php artisan make:model Book

3-2 Récupération des modèles

Une fois que vous avez créé un modèle et sa table de base de données associée, vous êtes prêt à commencer à récupérer des données dans votre base de données. La méthode `all` du modèle récupère tous les enregistrements de la table de base de données associée au modèle.

```
class PatientController extends Controller
{
    public function index(Request $request)
    {
        $patients = Patient::all();
        return view('patients.index', compact('patients'));
    }
}
```

3-3 Insertion de modèles

Pour insérer un nouvel enregistrement dans la base de données, vous devez instancier une nouvelle instance de modèle et définir des attributs sur le modèle. Ensuite, appelez la méthode d'enregistrement sur l'instance de modèle

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

public function store(Request $request)
{
    $patient = new Patient();
    $patient->nom = $request->input('nom');
    $patient->prénom = $request->input('prenom');
    $patient->age = $request->input('age');
    $patient->sexe = $request->input('sexe');
    $patient->téléphone = $request->input('telephone');
    $patient->adresse = $request->input('adresse');
    $patient->ville = $request->input('ville');
    $patient->antécédents = $request->input('antecedents');
    $patient->save();

    return redirect()->route('patients.index');
}
```

3-4 Mise à jour des modèles (update)

Pour mettre à jour un modèle, vous devez le récupérer et définir tous les attributs que vous souhaitez mettre à jour. Ensuite, vous devez appeler la méthode save du modèle.

```
public function edit($id)
{
    $patient = Patient::findOrFail($id);
    return view('patients.edit', compact('patient'));
}

public function update(Request $request, $id)
{
    $patient = Patient::findOrFail($id);
    $patient->nom = $request->input('nom');
    $patient->prénom = $request->input('prenom');
    $patient->age = $request->input('age');
    $patient->sexe = $request->input('sexe');
    $patient->téléphone = $request->input('telephone');
    $patient->adresse = $request->input('adresse');
    $patient->ville = $request->input('ville');
    $patient->antécédents = $request->input('antecedents');
    $patient->save();

    return redirect()->route('patients.index');
}
```

3-5 Suppression de modèles

Pour supprimer un modèle, vous pouvez appeler la méthode delete sur l'instance du modèle :

```
public function destroy($id)
{
    $patient = Patient::findOrFail($id);
    $patient->delete();

    return redirect()->route('patients.index');
}

}
```

