TP N°3

Base de données : Query Builder

ISTA TINGHIR — Module: M205

Realisée par: Groupe 3 — le: 21 mai 2023 — Déevelopper en Back-End

Introduction

Le Query Builder est un outil utilisé pour créer et exécuter des requêtes SQL. Il permet de créer des requêtes de sélection, d'insertion, de mise à jour et de suppression. Les conditions peuvent être utilisées pour filtrer les résultats. Le déverrouillage permet de protéger les données pendant une transaction pour éviter toute modification non désirée et le débogage est utilisé pour identifier et corriger les erreurs dans les requêtes ou le code SQL.

1 . Les condition en Query Builer

En Laravel Query Builder, les conditions sont utilisées pour ajouter des filtres à une requête SQL. Elles permettent de spécifier des critères pour sélectionner les enregistrements d'une table selon des valeurs de colonnes spécifiques.

Les conditions sont ajoutées à une requête SQL en utilisant les méthodes de condition fournies par Laravel Query Builder, telles que "where", "orWhere", "whereIn", "whereBetween", etc. Chacune de ces méthodes ajoute une condition supplémentaire à la requête SQL.

Exemples d'utilisation des conditions en Laravel Query Builder :

```php
DB::table('users')->where('name', '=', 'said')->get();

DB::table('products')->where('price', '>', 100)->get();

DB::table('products')->where('price', '>', 100)->get();

DB::table('orders')->where('status', '<>', 'cancelled')->get();

DB::table('users')->where('name', 'like', '%Mohamed%')
->orWhere('name', 'like', '%Hakim%')->get();

DB::table('users')->where('name', '=', 'said')->where('age', '>', 25)->get();


$names = ['hakim', 'Mohamed'];
DB::table('users')->whereIn('name', $names)->get();

DB::table('orders')->whereBetween('total_amount', [100, 200])->get();
```

2 . méthode insert en Query Builer

En Laravel Query Builder, l'insertion de données dans une table est effectuée en utilisant la méthode insert. Cette méthode prend un tableau associatif de valeurs à insérer, où les clés représentent les noms des colonnes et les valeurs sont les données à insérer.

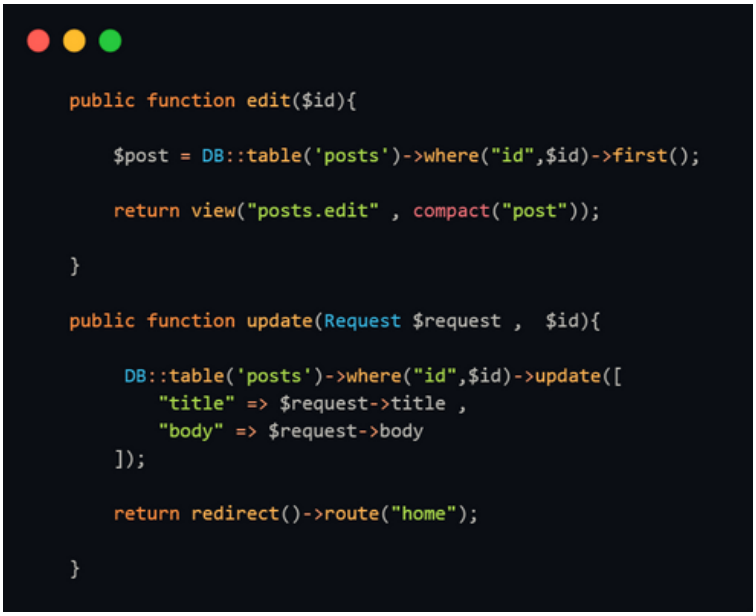
```php
public function insert(Request $request)
    {
        DB::table("posts")->insert([

            "title" => $request->title ,
            "body"=>$request->body

        ]);
        return redirect()->route("home");
    }
```

3 . méthode Update en Query Builer

la mise à jour de données dans une table est effectuée en utilisant la méthode update. Cette méthode prend un tableau associatif de valeurs à mettre à jour, où les clés représentent les noms des colonnes à mettre à jour et les valeurs sont les nouvelles données.



```php
public function edit($id){

    $post = DB::table('posts')->where("id",$id)->first();

    return view("posts.edit" , compact("post"));

}

public function update(Request $request ,  $id){

     DB::table('posts')->where("id",$id)->update([
        "title" => $request->title ,
        "body" => $request->body
    ]);

    return redirect()->route("home");

}
```

4 . méthode delete en Query Builer

La méthode de suppression du générateur de requêtes peut être utilisée pour supprimer des enregistrements de la table. Vous pouvez restreindre les instructions "delete" en ajoutant des clauses "where" avant d'appeler la méthode delete.

```php
public function delete($id){

    DB::table('posts')->where("id",$id)->delete();

    return redirect()->route("home");

}
```

Si vous souhaitez tronquer une table entière, ce qui supprimera tous les enregistrements de la table et réinitialisera l'ID d'auto-incrémentation à zéro, vous pouvez utiliser la méthode truncate :

```php
public function deleteAll(){

    DB::table('posts')->truncate();

    return redirect()->route("home");

}
```

5. Déverrouillage.

Le générateur de requêtes inclut également quelques fonctions pour vous aider à obtenir un "verrouillage pessimiste" lors de l'exécution de vos instructions de sélection. Pour exécuter une instruction avec un "shared lock", vous pouvez appeler la méthode sharedLock. Un verrou partagé empêche la modification des lignes sélectionnées tant que votre transaction n'est pas validée :

Vous pouvez également utiliser la méthode lockForUpdate. Un verrou "pour mise à jour" empêche les enregistrements sélectionnés d'être modifiés ou d'être sélectionnés avec un autre verrou partagé :

6. Débogage.

Vous pouvez utiliser les méthodes dd et dump lors de la création d'une requête pour vider les liaisons de requête actuelles et SQL. La méthode dd affichera les informations de débogage puis arrêtera l'exécution de la requête. La méthode dump affichera les informations de débogage mais permettra à la requête de continuer à s'exécuter :

Laravel Debugbar est un package qui fournit une barre de débogage pour les applications Laravel. Il comprend un onglet de requête qui affiche toutes les requêtes exécutées ainsi que leur temps d'exécution. Pour installer le package, exécutez la commande suivante