
Lier et Afficher les Produits par Marque dans Laravel

Dans votre projet, vous avez deux entités principales : **Produits** (**Product**) et **Marques** (**Marque**). L'objectif est de lier ces deux entités pour pouvoir afficher les produits associés à une marque spécifique. Voici une explication détaillée de ce concept et de son implémentation dans votre projet.

1. Relation entre les Tables

Migration **products**

Dans la migration **create_products_table**, vous avez ajouté une colonne **marque_id** pour établir une relation entre les produits et les marques.

```
Schema::create('products', function (Blueprint $table) {
    $table->id();
    $table->string('title');
    $table->decimal('price', 10, 2)->default(50);
    $table->unsignedBigInteger('marque_id'); // Clé étrangère
    $table->timestamps();
});
```

- **marque_id** : Cette colonne est une clé étrangère qui référence l'ID de la table **marques**.

Migration **marques**

Dans la migration **create_marque_table**, vous avez défini la structure de la table **marques**.

```
Schema::create('marques', function (Blueprint $table) {
    $table->id();
    $table->string("name")->unique();
    $table->string("description")->nullable();
    $table->timestamps();
});
```

2. Relations dans les Modèles

Modèle **Product**

Dans le modèle **Product**, vous avez défini une relation **belongsToMany** pour indiquer qu'un produit appartient à une marque.

```
public function marque()  
{  
    return $this->belongsTo(Marque::class, 'marque_id');  
}
```

- **belongsTo** : Un produit appartient à une marque.

Modèle **Marque**

Dans le modèle **Marque**, vous avez défini une relation **hasMany** pour indiquer qu'une marque peut avoir plusieurs produits.

```
public function products()  
{  
    return $this->hasMany(Product::class, 'marque_id');  
}
```

- **hasMany** : Une marque peut avoir plusieurs produits.

3. Affichage des Produits par Marque

Route

Vous avez défini une route pour afficher les produits d'une marque spécifique.

```
Route::get('/products/marque/{id}', [ProductController::class,  
'productsByMarque'])->name('products.byMarque');
```

- **/products/marque/{id}** : Cette route prend l'ID d'une marque en paramètre et affiche ses produits.

Méthode **productsByMarque** dans le Contrôleur

Dans le contrôleur **ProductController**, vous avez implémenté la méthode **productsByMarque** pour récupérer les produits d'une marque spécifique.

```
public function productsByMarque($id)  
{  
    $marque = Marque::find($id);  
  
    if (!$marque) {  
        return redirect()->route('marque.index')->with('error', 'Marque non  
trouvée.');    }  
}
```

```

    $products = $marque->products()->paginate(10);

    return view('tp5.marque', compact('products', 'marque'));
}

```

- **\$marque->products()** : Récupère tous les produits associés à la marque.
- **paginate(10)** : Pagine les résultats pour afficher 10 produits par page.

Vue `marque.blade.php`

Dans la vue `marque.blade.php`, vous affichez les produits de la marque sélectionnée.

```

@extends('layouts.app')

@section('content')
    <div class="container">
        <h1>Produits de la Marque: {{ $marque->name }}</h1>

        <table class="table">
            <thead>
                <tr>
                    <th>Titre</th>
                    <th>Description</th>
                    <th>Prix</th>
                    <th>Actions</th>
                </tr>
            </thead>
            <tbody>
                @foreach ($products as $product)
                    <tr>
                        <td>{{ $product->title }}</td>
                        <td>{{ $product->description }}</td>
                        <td>{{ $product->newPrice }} dh</td>
                        <td>
                            <a href="{{ route('products.edit', $product->id) }}">Modifier</a>
                            <a href="{{ route('products.show', $product->id) }}">Détails</a>
                            <form action="{{ route('products.destroy', $product->id) }}" method="POST" style="display:inline;">
                                @csrf
                                @method('DELETE')
                                <button type="submit">Supprimer</button>
                            </form>
                        </td>
                    </tr>
                @endforeach
            </tbody>
        </table>
    </div>

```

```
<!-- Pagination -->
<div>
    {{ $products->links() }}
</div>
</div>
@endsection
```

- `@foreach ($products as $product)` : Boucle pour afficher chaque produit.
- `{{ $marque->name }}` : Affiche le nom de la marque.
- `{{ $products->links() }}` : Affiche la pagination.

4. Lien entre les Pages

Dans la vue `index.blade.php` des marques, vous avez ajouté un bouton pour afficher les produits d'une marque spécifique.

```
<a href="{{ url('/products/marque/' . $m->id) }}" class="btn btn-info btn-sm">
    <i class="fas fa-list"></i> Voir Produits
</a>
```

- `/products/marque/{id}` : Redirige vers la page des produits de la marque sélectionnée.

5. Gestion des Erreurs

Dans la méthode `productsByMarque`, vous avez ajouté une vérification pour s'assurer que la marque existe.

```
if (!$marque) {
    return redirect()->route('marque.index')->with('error', 'Marque non
trouvée.');
```

- Si la marque n'existe pas, l'utilisateur est redirigé vers la liste des marques avec un message d'erreur.

6. Pagination

La pagination est gérée avec la méthode `paginate(10)` dans le contrôleur et affichée dans la vue avec `{{ $products->links() }}`.

Résumé des Étapes

1. **Migration** : Ajoutez `marque_id` dans la table `products` pour établir la relation.
2. **Modèles** : Définissez les relations `belongsToMany` et `hasMany` dans les modèles `Product` et `Marque`.

3. **Route** : Créez une route pour afficher les produits d'une marque spécifique.
 4. **Contrôleur** : Implémentez la méthode `productsByMarque` pour récupérer les produits d'une marque.
 5. **Vue** : Affichez les produits dans une vue dédiée (`marque.blade.php`).
 6. **Lien** : Ajoutez un bouton dans la liste des marques pour accéder aux produits d'une marque.
 7. **Gestion des Erreurs** : Vérifiez que la marque existe avant d'afficher ses produits.
 8. **Pagination** : Paginez les résultats pour une meilleure expérience utilisateur.
-

Exemple Pratique

Cas d'Utilisation

- Vous avez une marque "Nike" avec l'ID `1`.
- Vous souhaitez afficher tous les produits de la marque "Nike".

Étapes

1. Cliquez sur le bouton "**Voir Produits**" dans la liste des marques.
 2. La route `/products/marque/1` est appelée.
 3. Le contrôleur récupère les produits de la marque "Nike".
 4. Les produits sont affichés dans la vue `marque.blade.php`.
-

Conclusion

En suivant ces étapes, vous avez correctement lié les produits et les marques dans Laravel. Vous pouvez maintenant afficher les produits d'une marque spécifique, gérer les erreurs, et paginer les résultats. Ce concept est essentiel pour créer des relations entre les tables et afficher des données connexes dans une application Laravel.
