

LFTS - Filtering

39. Advanced Eloquent Query Constraints

En este capítulo nos enseñan otra manera de hacer el filtrado de categorías de una manera mas directa y limpia utilizando las propiedades que nos brinda "Eloquent"

```
return view('posts', [
    'posts' => Post::latest()->filter(request(['search', 'category']))->get(),
    'categories' => Category::all(),
    'currentCategory' => Category::firstWhere('slug', request('category'))
]);
```

```
<x-dropdown-item href="/" class="{{request()->routeIs('home') && !request('category') ? 'bg-blue-500 text-white' : ''}}" >All Categories</x-dropdown-item>
@foreach ($categories as $category)
<x-dropdown-item href="/?category={{$category->slug}}"
class="{{isset($currentCategory) && $currentCategory->id === $category->id ? 'bg-blue-500 text-white' : ''}}">
    {{ucwords($category->name)}}
</x-dropdown-item>
```

```
public function scopeFilter($query, array $filters)
{
    $query->when($filters['search'] ?? false, function ($query, $search) {
        $query->where('title', 'like', '%' . $search . '%')
        ->orWhere('excerpt', 'like', '%' . $search . '%')
        ->orWhere('body', 'like', '%' . $search . '%');
    });

    $query->when($filters['category'] ?? false, function ($query, $category) {
        $query->whereHas('category', fn ($query) =>
            $query->where('slug', $category)
        );
    });
}
```

Al hacer la busqueda por categorias de esta manera nos ahorramos bastante codigo ya que ya no ocupamos la ruta que nos llevaba a la vista de la categoria que estabamos buscando

```
1 <?php
2
3 use App\Models\Category;
4 use App\Models\User;
5 use Illuminate\Support\Facades\Route;
6 use App\Http\Controllers\PostController;
7
8 Route::get('/', [PostController::class, 'index'])->name('home');
9
10 Route::get('/posts/{post:slug}', [PostController::class, 'show'])->name('post');
11
12 Route::get('authors/{author:username}', function (User $author) {
13     return view('posts', [
14         'posts' => $author->posts,
15         'categories' => Category::all()
16     ]);
17 }->name('author');
18
```

40. Extract a Category Dropdown Blade Component

Ahora vamos a cambiar el dropdown de categorias para que utilice blade

```
return view('posts.index', [
    'posts' => Post::latest()->filter(request(['search', 'category']))->get(),
]);
}

public function show(Post $post)
{
    return view('posts.show', [
        'post' => $post,
    ]);
}
```

```

1  <?php
2
3  namespace App\View\Components;
4
5  use App\Models\Category;
6  use Illuminate\View\Component;
7
8  class CategoryDropdown extends Component
9  {
10     /**
11      * Create a new component instance.
12      *
13      * @return void
14      */
15
16     /**
17      * Get the view / contents that represent the component.
18      */
19     public function render()
20     {
21         return view('components.category-dropdown', [
22             'categories' => Category::all(),
23             'currentCategory' => Category::firstWhere('slug', request('category'))
24         ]);
25     }
26 }

```

```

<x-dropdown>
  <x-slot name="trigger">
    <button class="py-2 pl-3 pr-9 text-sm font-semibold w-full lg:w-32 text-left flex lg:inline-flex">
      {{isset($currentCategory) ? ucwords($currentCategory->name) : 'Categories'}}
      <x-icon name="down-arrow" class="absolute pointer-events-none" />
    </button>
  </x-slot>
  <x-dropdown-item href="/" class="{{request()->routeIs('home') && !request('category') ? 'bg-blue-500 text-white' : ''}}">
    @foreach ($categories as $category)
      <x-dropdown-item href="/?category={{$category->slug}}"
        class="{{isset($currentCategory) && $currentCategory->id === $category->id ? 'bg-blue-500 text-white' : ''}}
        {{ucwords($category->name)}}
      </x-dropdown-item>
    @endforeach
  </x-dropdown>

```

```

Route::get('authors/{author:username}', function (User $author) {
    return view('posts', [
        'posts' => $author->posts
    ]);
});

```

```

<x-layout>
  @include('posts._header')

```

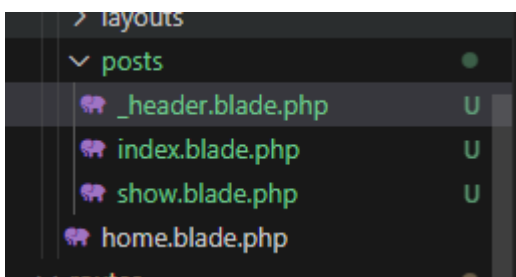
```
<div class="relative lg:inline-flex bg-gray-100 rounded-xl">
  <x-category-dropdown />
</div>
```

Tambien le cambiamos el nombre a los ciertos archivos por mero orden y los metemos en una carpeta que vamos a llamar posts

_posts-header.blade.php → _header.blade.php

posts.blade.php → index.blade.php

post.blade.php → show.blade.php



41. Author Filtering

En este episodio como indica el titulo vamos a hacer la filtracion por autor, asi cada vez que clickeamos en el nombre de un autor vamos a ser capaces de ver todos los posts que tiene ese autor

```
return view('posts.index', [
    'posts' => Post::latest()->filter(request(['search', 'category', 'author']))->get(),
]);
```

```
$query->when($filters['author'] ?? false, function ($query, $author) {
    $query->whereHas('author', fn ($query) =>
        $query->where('username', $author)
    );
});
```

```
<div class="ml-3">
  <h5 class="font-bold">
    <a href="/?author={{$post->author->username}}">{{$post->author->name}}</a>
  </h5>
</div>
```

```
<h5 class="font-bold">
  <a href="/?author={{$post->author->username}}">{{$post->author->name}}</a>
</h5>
```

```


<h5 class="font-bold">
    <a href="/?author={{$post->author->username}}">{{$post->author->name}}</a>
  </h5>


```

Con los cambios que hemos venido haciendo, ya no es necesario usar las rutas para los filtros que hemos venido modificando (categoría, autor) ahora todos lo hacemos desde el controlador y el modelo de los post utilizando queries

```

routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\PostController;
5
6  Route::get('/', [PostController::class, 'index'])->name('home');
7
8  Route::get('/posts/{post:slug}', [PostController::class, 'show'])->name('post');
9
10

```

42. Merge Category and Search Queries

Lo que hacemos es modificar ligeramente el código para que nos permita hacer filtros por categorías y texto utilizando el textbox de "search"

```

<!-- Search -->
<div class="relative flex lg:inline-flex items-center bg-gray-100 rounded-xl px-3 py-2">
  <form method="GET" action="/">
    @if (request('category'))
      <input type="hidden" name="category" value="{{request('category')}}">
    @endif
    <input type="text" name="search" placeholder="Find something" class="bg-transparent place
  </form>
</div>
</div>
</header>

```

```

@foreach ($categories as $category)
  <x-dropdown-item
    href="/?category={{$category->slug}}&{{http_build_query(request()->except('category'))}}"
    class="{{isset($currentCategory) && $currentCategory->id === $category->id ? 'bg-blue-500 text-w
    {{ucwords($category->name)}}
  </x-dropdown-item>

```

43. Fix a Confusing Eloquent Query Bug

Arreglamos un bug que no habíamos notado en el anterior episodio, este error lo que causaba era que trágieramos trabajos no correspondiente a la categoría que buscamos

```
$query->when($filters['search'] ?? false, function ($query, $search) {  
    $query->where(fn($query) =>  
        $query->where('title', 'like', '%' . $search . '%')  
        ->orWhere('excerpt', 'like', '%' . $search . '%')  
        ->orWhere('body', 'like', '%' . $search . '%')  
    );  
});
```