

[<--- Volver](#)

Comments

Escriba el margen para comentarios en los post / Write the markup for a Post Comment

Vamos al archivo show en la carpeta posts y modificamos para agregar una nueva seccion para los comentarios

```
<article class="flex bg-gray-100 border border-gray-200 p-6 rounded-xl space-x-4">
  <div class="flex-shrink-0">
    
  </div>
  <div>
    <header>
      <h3 class="font-bold">Jonh Doe</h3>
      <p class="text-xs">
        Posted
        <time>8 months ago</time>
      </p>
    </header>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
      eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
      veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
      consequat.
    </p>
  </div>
</article>
```

Este seria solo la base para crear la seccion de comentarios, este codigo lo vamos a mover a un componentesque llamaremos `post-comment`

Asi se veria los comentarios de los posts dentro del post



Jonh Doe

Posted 8 months ago

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.



Jonh Doe

Posted 8 months ago

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.



Jonh Doe

Posted 8 months ago

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Consistencia de tablas y restricciones de clave externa / Table Consistency and Foreign Key Constraints

Desde la terminal de la VM webserver vamos a crear un nuevo modelo y le vamos a agregar -mfc que significa

- m = migration
- f = factory
- c = controller

```
vagrant@webserver:/vagrant/sites/lfts.isw811.xyz$ php artisan make:model Comment -mfc
Model created successfully.
Factory created successfully.
Created Migration: 2023_10_27_185059_create_comments_table
Controller created successfully.
vagrant@webserver:/vagrant/sites/lfts.isw811.xyz$
```

Ahora nos dirigimos a la migracion de Comment para modificarla con las columnas que queremos que lleva la tabla

```
Schema::create('comments', function (Blueprint $table) {
    $table->id();
    $table->foreignId('post_id')->constrained()->cascadeOnDelete();
    $table->foreignId('user_id')->constrained()->cascadeOnDelete();
    $table->unsignedBigInteger('user_id');
    $table->text('body');
    $table->timestamps();
});
```

Nos movemos a la terminal de la VM webserver y corremos el comando para migrar la tabla a la base de datos

```
php artisan migrate
```

```
vagrant@webserver:/vagrant/sites/lfts.isw811.xyz$ php artisan migrate
Migrating: 2023_10_27_185059_create_comments_table
Migrated: 2023_10_27_185059_create_comments_table (153.13ms)
```

Hacer que los comentarios sean dinamicos / Make the Comments Section Dynamic

Editamos la fabrica de comments para que se creen los datos de la forma que queremos dentro de la base de datos

```
public function definition()
{
    return [
        'name' => $this->faker->word(),
        'slug' => $this->faker->slug()
    ];
}
```

Nos movemos a la terminal de la Vm webserver para crear un comment con los comandos

```
php artisan tinker
App\Models\Comment::factory()->create();
```

```
> App\Models\Comment::factory()->create();
= App\Models\Comment {#6260
  post_id: 24,
  user_id: 25,
  body: "Minus in rerum magni a quia unde iste. Maxime quas voluptas pariatum rerum. Repudiandae dolore qui ea ex voluptates animi dolores eum.",
  updated_at: "2023-10-27 19:41:03",
  created_at: "2023-10-27 19:41:03",
  id: 2,
}
```

Ahora despues de crear el comment vamos a el modelo Post para agregar la relacion con el comment

```
public function comments() {
    return $this->hasMany(Comment::class);
}
```

Ahora vamos al modelo Comment y creamos las relaciones con Post y Author

```
public function post() {
    return $this->belongsTo(Post::class);
}

public function author() {
    return $this->belongsTo(User::class, 'user_id');
}
```

Y vemos como tiene relacion con el Post

```
> $c = App\Models\Comment::first();
App\Models\Comment {#6862
  id: 2,
  post_id: 24,
  user_id: 25,
  body: "Natus in rerum magni a quia unde iste. Maxime quas voluptas pariatur rerum. Repudiandae dolore qui ea ex voluptates animi dolores eum.",
  created_at: "2023-10-27 19:41:03",
  updated_at: "2023-10-27 19:41:03",
}

> $c->post;
App\Models\Post {#6167
  id: 24,
  user_id: 28,
  category_id: 14,
  slug: "doloremque-laudantium-et-cumque-dolorem-velit",
  title: "Ipsam voluptatem quis velit nihil libero est et.",
  excerpt: "Quis sint reiciendis suscipit et quidem dolores natus fuga. Non voluptatem magnam minima perferendis. Eligendi distinctio id suscipit in optio repudiandae omnis.",
  body: "Quis autem optio soluta. Cum magni vel vel perspiciatis nulla. Velit possimus quisquam eius repudiandae modi voluptate cum.",
  created_at: "2023-10-27 19:41:03",
  updated_at: "2023-10-27 19:41:03",
  published_at: null,
  category: App\Models\Category {#7128
    id: 14,
    name: "Quid",
    slug: "debitis-quidem-ratione-quia-odit-aut-totam",
    created_at: "2023-10-27 19:41:03",
    updated_at: "2023-10-27 19:41:03",
  },
  author: App\Models\User {#7129
    id: 28,
    username: "lea.kertzmann",
    name: "Miss Lura Labadie BW",
    email: "bart.schinner@example.com",
    email_verified_at: "2023-10-27 19:41:02",
    password: "$2y$10$1G5uM.DFNZnp/ahly6q.b06HDe9uAvrhG75SFy36mfG9QRs2vin",
    remember_token: "Z1tjoU1F80",
    created_at: "2023-10-27 19:41:03",
    updated_at: "2023-10-27 19:41:03",
  },
}
```

Ahora nos dirigimos a la vista show para emepezar con los comentarios dinamicos, el primer cambio que haremos sera este

```
@foreach($post->comments as $comment)
    <x-post-comment :comment="$comment" />
@endforeach
```

Ahora nos vamos al componente post-comment

```
@props()
<article class="flex bg-gray-100 border border-gray-200 p-6 rounded-xl space-x-4">
    <div class="flex-shrink-0">
        
    </div>
```

```

<div>
  <header>
    <h3 class="font-bold">{{ $comment->author->username }}</h3>
    <p class="text-xs">
      Posted
      <time>{{ $comment->created_at }}</time>
    </p>
  </header>
  <p> {{ $comment->body }} </p>
</div>
</article>

```

Y vemos como dentro de la web tenemos un comentario dinamico



LuisGomez

Posted 2023-10-27 19:41:03

Minus in rerum magna quia unde iste. Maxime quas voluptas pariatur rerum. Repudiandae dolore qui ea ex voluptates animi dolores eum.

Disenar el formulario para comentarios / Design the Comment Form

Lo primero sera ir a la vista show y crear un form ahi que sera donde se ingresen los comentarios

```

<form action="POST" action="#" class="border border-gray-200
p-6 rounded-xl">
  @csrf
  <header class="flex items-center">
    
    <h2 class="ml-4">Want to participate?</h2>
  </header>
  <div class="mt-6">
    <textarea name="body" class="w-full text-sm
focus:outline-none focus:ring" rows="5" placeholder="Quick, thing of something to
say"/>
  </div>
  <div class="flex justify-end mt-6 pt-6 border-t border-
gray-200">
    <button type="submit" class="bg-blue-500 text-white
uppercase font-semibold text-xs py-2 px-10 rounded-2xl hover:bg-blue-600"/>
  </div>
</form>

```

Ahora creamos un componente al que llamaremos panel y le ponemos este codigo

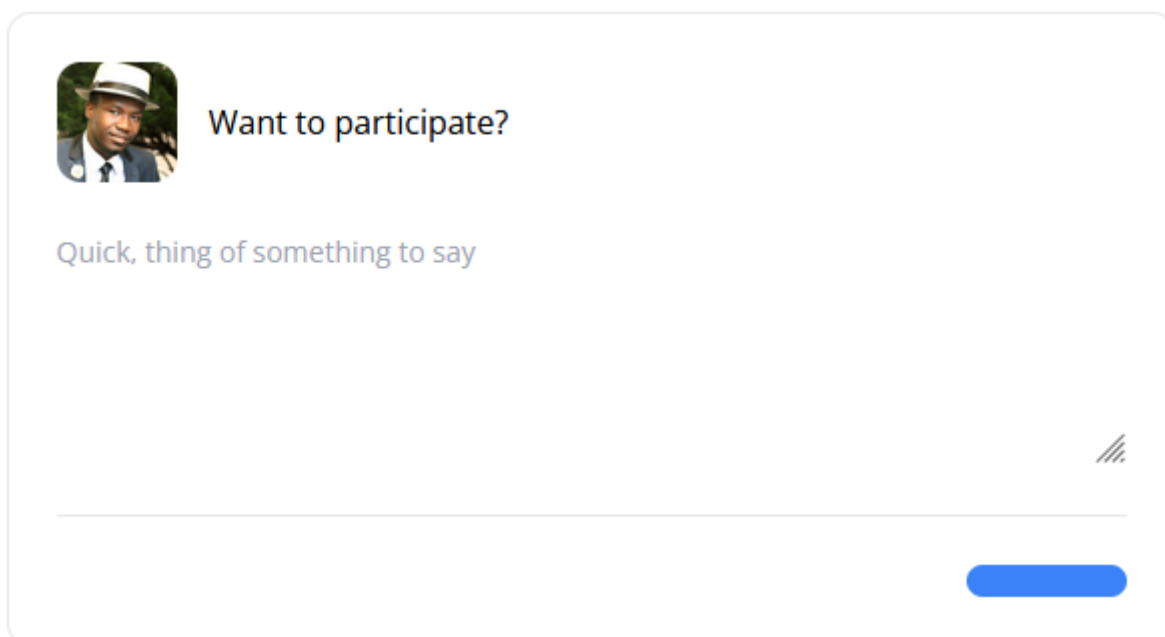
```
<div class="border border-gray-200 p-6 rounded-xl">
  {{ $slot }}
</div>
```

Y al código anterior lo metemos entre las etiquetas `<x-panel>`

El código de post-comment también lo metemos entre esas mismas etiquetas

```
<x-panel class="bg-gray-50">
  <article class="flex bg-gray-100 border border-gray-200 p-6 rounded-xl space-
x-4">
    <div class="flex-shrink-0">
      
    </div>
    <div>
      <header>
        <h3 class="font-bold">{{ $comment->author->username }}</h3>
        <p class="text-xs">
          Posted
          <time>{{ $comment->created_at }}</time>
        </p>
      </header>
      <p> {{ $comment->body }} </p>
    </div>
  </article>
</x-panel>
```

Por último, así quedaría el form estilizado.



The image shows a stylized comment form. At the top left is a circular profile picture of a man wearing a hat. To its right is the text "Want to participate?". Below this is a text input area with the placeholder text "Quick, thing of something to say". At the bottom right of the form is a blue button. There is also a small icon of three diagonal lines in the bottom right corner of the form area.

Activar el formulario de comentarios / Activate the Comment Form

Añadimos una nueva endpoint en `web.php` para poder agregar los comentarios

```
Route::post('posts/{post:slug}/comments', [PostCommentController::class, 'store']);
```

Vamos a crear un nuevo controller con el nombre que le dimos en el endpoint

```
php artisan make:controller PostCommentsController
```

```
vagrant@webserver:/vagrant/sites/lfts.isw811.xyz$ php artisan make:controller PostCommentsController
Controller created successfully.
```

En la vista show, actualizamos esta línea

```
<form action="POST" action="/posts/{{ $post->slug }}/comments">
```

Ahora dentro de este nuevo controller creamos lo necesario para que el form funcione

```
public function store(Post $post) {
    request()->validate([
        'body' => 'required'
    ]);
    $post->comments()->create([
        'user_id' => request()->user()->id,
        'body' => request('body')
    ]);
    return back();
}
```

Esto nos dará un error y para resolverlo lo que tendremos que hacer es añadir esta línea de código al Model de Comment

```
protected $guarded = [];
```

También se puede hacer de la siguiente forma, vamos a la ruta `/app/Providers/` y al archivo `AppServiceProvider` y en la función `boot()` agregamos

```
Model::unguard();
```

Y así se verán los comentarios de los post

**LuisGomez**

Posted October 30, 2023, 6:50 pm

Hello

Ahora que haremos sera una condicional para evitar que personas que no se encuentran logueadas puedan escribir comentarios, ya que si lo hacen la pagina se cae. Para esto lo que haremos sera mete toda la logica de la vista del form de comentarios dentro de un `@auth` y agregar un link para loguearse en la seccion de comentarios.

```
<a href="/register" class="hover:underline">Register</a> or <a href="/login" class="hover:underline">log in </a> to leave a comment.
```

Un poco de limpieza del capítulo / Some Light Chapter Clean Up

Agregamos mensaje de error en caso de que sea un comentario vacio

```
@error('body')
    <span class="text-xs text-red-500">{{ $message }}</span>
@enderror
```

Extraemos el codigo del form de comentarios y lo annadimos a una nueva vista que crearemos para este, el cual se llamara `add-comment-form`, lo llamaremos dentro de la vista show con `@include`

**LuisGomez**

Posted October 30, 2023, 7:45 pm

Still work

Extraemos el boton de add-comment, para nada mas llamarlo la proxima vez que necesitemos un boton, por lo que crearemos un nuevo componente llamado `submit-button.blade.php`