Laravel - Guest User Gates

Topics : Laravel
Written on December 26, 2023

In Laravel, gates and policies are used to authorize actions within your application. Gates are callbacks that determine if a user is authorized to perform a certain action, while policies are classes that organize the authorization logic for a particular model or resource.

If you want to handle guest users (users who are not logged in) in gates, you can do so by checking if the user is authenticated or not. Laravel provides a convenient way to define gates using the `Gate` facade.

Here's an example of how you might define a gate that allows access only to authenticated users:

```
use Illuminate\Support\Facades\Gate;

Gate::define('access-guest-content', function ($user = null) {
    return auth()->check(); // Check if the user is authenticated
});
```

In this example, the gate is named `access-guest-content`, and the callback function receives a user as a parameter. The `auth()->check()` method is used to determine if the user is authenticated. If the user is authenticated, the gate will pass; otherwise, it will fail.

Now, you can use this gate in your controllers or views:

```
if (Gate::allows('access-guest-content')) {
    // User is allowed to access guest content
    return view('guest-content');
} else {
    // User is not allowed
    return redirect()->route('login');
}
```

Alternatively, you can use the `@can` directive in your Blade views:

```
@can('access-guest-content')
    {{-- User is allowed to access guest content --}}
    @include('guest-content')
@else
    {{-- User is not allowed --}}
```

```
    @include('login-link')
@endcan
```