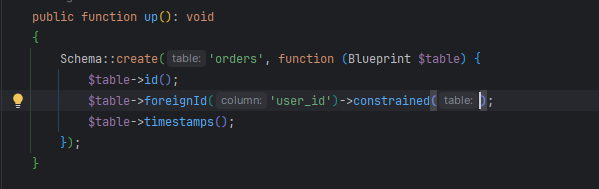
#Follow private channel docs #1 first

Step 1: Making an Order model migration,



Step 2: Editing the Order dispatch \_\_construct() function,

<?php  
  
namespace App\Events;  
  
use App\Models\Order;  
use Illuminate\Broadcasting\Channel;  
use Illuminate\Broadcasting\InteractsWithSockets;  
use Illuminate\Broadcasting\PresenceChannel;  
use Illuminate\Broadcasting\PrivateChannel;  
use Illuminate\Contracts\Broadcasting\ShouldBroadcast;  
use Illuminate\Contracts\Broadcasting\ShouldBroadcastNow;  
use Illuminate\Foundation\Events\Dispatchable;  
use Illuminate\Queue\SerializesModels;  
use App\Models\User;  
  
class OrderDispatched implements ShouldBroadcastNow  
{  
 use Dispatchable, InteractsWithSockets, SerializesModels;  
  
 public function \_\_construct(public User $user, public Order $order)  
 {  
 //  
 }  
  
 public function broadcastOn(): array  
 {  
 return [  
 new PrivateChannel('users.'.$this->user->id),  
 ];  
 }  
}

Step 3: add Order data on “/broadcast” route,

Route::get('/broadcast', function () {  
 broadcast(new \App\Events\OrderDispatched(User::*find*(1), \App\Models\Order::*find*(1)));  
});

Step 4: order data ke alpine store er moddhe Rakha hoise, and **x-text** use kore data ke show kora hoise,

**x-text** is also an **alpine js** code, Full **dashboard.php** code,

<x-app-layout>  
 <x-slot name="header">  
 <h2 class="font-semibold text-xl text-gray-800 leading-tight">  
 {{ \_\_('Dashboard') }}  
 </h2>  
 </x-slot>  
  
  
 <div class="py-12">  
 <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">  
 <div class="bg-white overflow-hidden shadow-sm sm:rounded-lg">  
 <div class="p-6 text-gray-900" x-init="privateClass()">  
  
 <template x-if="$store.dispatchedStore.dispatched">  
 <div>  
 Order (# <span x-text="$store.dispatchedStore.order.id"></span>) has been dispatched  
 </div>  
 </template>  
 </div>  
 </div>  
 </div>  
 </div>  
  
  
 <script>  
 function privateClass() {  
 Echo.private('users.{{ auth()->id() }}').listen('OrderDispatched', (event) => {  
  
 Alpine.store('dispatchedStore').dispatched = true;  
 Alpine.store('dispatchedStore').order = event.order  
 })  
 }  
 </script>  
  
 {{-- Alpine js store script --}}  
  
 <script>  
 *document*.addEventListener('alpine:init', () => {  
 Alpine.store('dispatchedStore', {dispatched: false, order: null});  
 });  
 </script>  
  
  
</x-app-layout>