Laravel site:

By default, broadcasting is not enabled in new Laravel applications. You may enable broadcasting using the install:broadcasting Artisan command:

Broadcasting Installation

```
php artisan install:broadcasting
```

Set Up Pusher in Laravel

Install the required package for Pusher in your Laravel project:

```
composer require pusher/pusher-php-server
```

Next, you should configure your Pusher Channels credentials in the config/broadcasting.php configuration file. An example Pusher Channels configuration is already included in this file, allowing you to quickly specify your key, secret, and application ID. Typically, you should configure your Pusher Channels credentials in your application's .env file:

```
PUSHER_APP_ID="your-pusher-app-id"
PUSHER_APP_KEY="your-pusher-key"
PUSHER_APP_SECRET="your-pusher-secret"
PUSHER_HOST=
PUSHER_PORT=443
PUSHER_SCHEME="https"
PUSHER_APP_CLUSTER="mt1"
```

The config/broadcasting.php file's pusher configuration also allows you to specify additional options that are supported by Channels, such as the cluster.

Then, set the BROADCAST_CONNECTION environment variable to pusher in your application's .env file:

```
BROADCAST_CONNECTION=pusher
```

Next : Defining Broadcast Events

Run the following command:

```
php artisan make:event MyCustomEvent
```

Edit the event class in app/Events/MyCustomEvent.php to pass data to the frontend:

```php
<?php

namespace App\Events;

use Illuminate\Broadcasting\Channel;
use Illuminate\Broadcasting\InteractsWithSockets;
use Illuminate\Broadcasting\PrivateChannel;
use Illuminate\Contracts\Broadcasting\ShouldBroadcast;
use Illuminate\Contracts\Broadcasting\ShouldBroadcastNow;
use Illuminate\Foundation\Events\Dispatchable;
use Illuminate\Queue\SerializesModels;

class BookingStatusUpdateNotification implements ShouldBroadcastNow
{
    use Dispatchable, InteractsWithSockets, SerializesModels;
    public $member_id;
    /**
     * Create a new event instance.
     */
    public function __construct($member_id)
    {
        $this->member_id = $member_id;
    }

    public function broadcastOn()
    {
        return [
            new Channel('booking-channel'),
        ];
    }

    public function broadcastAs(): string
    {
        return 'booking-updated';
    }
    public function broadcastWith(): array
    {
        return ['bookingData' => 'Booking Data has been modified'];
    }
}
```

Frontend Site:

<u>Set Up Laravel Echo in Your Vue Project</u>

```
npm install --save laravel-echo pusher-js
```

Configure main.js:

```
import Echo from 'laravel-echo';

import Pusher from 'pusher-js';
window.Pusher = Pusher;

window.Echo = new Echo({
    broadcaster: 'pusher',
    key: 'd2336503bae69be21669',
    cluster: "ap2",
    forceTLS: true,
});
```

Component config:

```
mounted() {
   window.Echo.channel('my-channel')
       .listen('my-event', (e) => {
           console.log("Received event:", e.message);
           // Handle the event data in your frontend as needed
       });
}
```