PUSHER

Getting Started

laravel-chat

STEP 1 Subscribe to events on the client

The snippet below connects to Channels and subscribes to a channel called `my-channel`, listening for an event called `my-event`.

Vue.js

index.html Download Copy

```
<!DOCTYPE html>
<head>
  <title>Pusher Test</title>
</head>
<body>
  <h1>Pusher Test</h1>
  <p>
    Publish an event to channel <code>my-channel</code>
    with event name <code>my-event</code>; it will appear below:
  </p>
  <div id="app">
    <ul>
      <li v-for="message in messages">
        {{message}}
      </li>
    </ul>
  </div>

  <script src="https://js.pusher.com/7.0/pusher.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></sc
  <script>
    // Enable pusher logging - don't include this in production
    Pusher.logToConsole = true;
    var pusher = new Pusher('da677ff93854d3c2ca06', {
      cluster: 'mt1'
    });

    var channel = pusher.subscribe('my-channel');
    channel.bind('my-event', function(data) {
      app.messages.push(JSON.stringify(data));
    });

    // Vue application
    const app = new Vue({
      el: '#app',
      data: {
        messages: [],
      },
    });
  </script>
</body>
```

View pusher-js on Github

Note: This page also includes the Javascript snippet and so is subscribed to `my-channel` too.

○ Connection status: Connected

STEP 2

Publish events from the server

Publish an event to your subscribed clients using the server code below and it will be received by any (and all) of the clients you have open, including this page.

Laravel ⌄

Console

Copy

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

Console Copy

```
PUSHER_APP_ID=1103853
PUSHER_APP_KEY=da677ff93854d3c2ca06
PUSHER_APP_SECRET=6e567c0c9908c7abdad0
```

config/broadcasting.php Download Copy

```
'options' => [
  'cluster' => 'mt1',
  'useTLS' => true
],
```

index.php Download Copy

```
<?php

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

class MyEvent implements ShouldBroadcast
{
  use Dispatchable, InteractsWithSockets, SerializesModels;

  public $message;

  public function __construct($message)
  {
      $this->message = $message;
  }

  public function broadcastOn()
  {
      return ['my-channel'];
  }

  public function broadcastAs()
  {
      return 'my-event';
  }
}
```

index.php Download Copy

```
event(new MyEvent('hello world'));
```

Refer to the official Laravel docs for more information.

View pusher-http-php on Github

Note: You can also send yourself a message from the debug console in another tab.

Keep learning with our docs

Overview

Explore core product concepts

Client API

Subscribe to published events

Server API

Publish events from your server

Support | Status