

# Laravel Websockets

1 - Instalar no projeto as dependências do Laravel Websockets. Seguir documentação oficial.

<https://docs.beyondco.de/laravel-websockets/1.0/getting-started/installation.html>

2 - Configurar arquivo .env.

```
37
38  PUSHER_APP_ID=anyID
39  PUSHER_APP_KEY=anyKey
40  PUSHER_APP_SECRET=anySecret
41  PUSHER_APP_CLUSTER=mt1
42
43  MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
44  MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
```

Você pode usar qualquer ID, KEY e SECRET.

3 - Instalar o Pusher PHP SDK Oficial. Seguir documentação oficial.

<https://docs.beyondco.de/laravel-websockets/1.0/basic-usage/pusher.html#requirements>

4 - Instalar biblioteca JAVASCRIPT Laravel-Echo e Pusher-js.

4.1 - Instalar package.json | npm install.

4.2 - Instalar laravel-echo e pusher-js | npm install laravel-echo pusher-js

5 - Configurar Laravel Echo no bootstrap.js | resources/js/components.

```

bootstrap.js - Laravel-WebSockets-Chat - Visual Studio Code
File Edit Selection View Go Debug Terminal Help
EXPLORER
OPEN EDITORS
bootstrap.js resources\js
LARAVEL-WEBSOCKETS-CHAT
  DOCS\app
    config
    database
    node_modules
    public
  resources
    js
      components
        ChatsComponent.vue
        app.js
        bootstrap.js
    lang
    sass
    views
  routes
    api.php
    channels.php
    console.php
    web.php
  storage
  tests
  vendor
    editorconfig
    .env
    .env.example
  OUTLINE
  NPM SCRIPTS
bootstrap.js ×
resources > js > bootstrap.js > ...
40
41 /**
42 * Echo exposes an expressive API for subscribing to channels ...
43 * for events that are broadcast by Laravel. Echo and event bro...
44 * allows your team to easily build robust real-time web applic...
45 */
46
47 import Echo from 'laravel-echo'
48
49 window.Pusher = require('pusher-js');
50
51 window.Echo = new Echo({
52   broadcaster: 'pusher',
53   key: process.env.MIX_PUSHER_APP_KEY,
54   wsHost: window.location.hostname,
55   wsPort: 6001, You, 20 hours ago • inicial commit
56   disableStats: true,
57 });
58
59
60 // window.Echo.channel('DemoChannel')
61 // .listen('WebsocketDemoEvent', (e) => {
62 //   console.log(e);
63 // });
64
65 // });
66
67

```

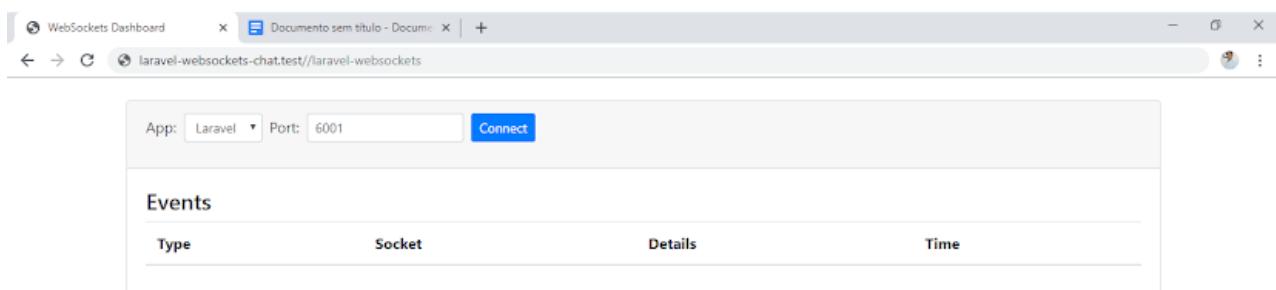
You, 20 hours ago Ln 55, Col 18 Spaces: 4 UTF-8

Descomentar as linhas de código e adicionar as linhas de código da documentação oficial.  
<https://docs.beyondco.de/laravel-websockets/1.0/basic-usage/pusher.html#usage-with-laravel-echo>

6 - Executar comando no terminal | npm run watch, para assistir as mudanças e compilar no app.js.

7 - Abra outro terminal, execute o comando | php artisan serve, para iniciar servidor php.

8 - Em outro terminal, execute o comando | php artisan websocket:serve, para iniciar o servidor websocket.



Clique no botão connect.

The screenshot shows the Laravel WebSockets Dashboard interface. At the top, there is a header with tabs for "WebSockets Dashboard" and "Documento sem título - Documento". Below the header, the URL "laravel-websockets-chat.test/laravel-websockets" is visible. The main content area is divided into two sections: "Realtime Statistics" and "Event Creator".

**Realtime Statistics:** This section contains a chart with a single data series. The Y-axis ranges from 0 to 4, and the X-axis ranges from -1 to 6. The chart shows a constant value of 0 across all points.

**Event Creator:** This section includes fields for "Channel" and "Event", and a large "Data" input area. A "Send event" button is located at the bottom right of this section.

The screenshot shows the "Events" table section of the Laravel WebSockets Dashboard. The table has columns for Type, Socket, Details, and Time. The data is as follows:

Type	Socket	Details	Time
subscribed	463646490.769770643	Channel: private-websockets-dashboard-statistics	14:33:47
occupied		Channel: private-websockets-dashboard-statistics	14:33:47
subscribed	463646490.769770643	Channel: private-websockets-dashboard-api-message	14:33:47
occupied		Channel: private-websockets-dashboard-api-message	14:33:47
subscribed	463646490.769770643	Channel: private-websockets-dashboard-client-message	14:33:47
occupied		Channel: private-websockets-dashboard-client-message	14:33:47
subscribed	463646490.769770643	Channel: private-websockets-dashboard-vacated	14:33:47
occupied		Channel: private-websockets-dashboard-vacated	14:33:47
subscribed	463646490.769770643	Channel: private-websockets-dashboard-occupied	14:33:47
occupied		Channel: private-websockets-dashboard-occupied	14:33:47
subscribed	463646490.769770643	Channel: private-websockets-dashboard-connection	14:33:47
subscribed	463646490.769770643	Channel: private-websockets-dashboard-subscribed	14:33:47

Aqui você pode ver os eventos e seus detalhes.

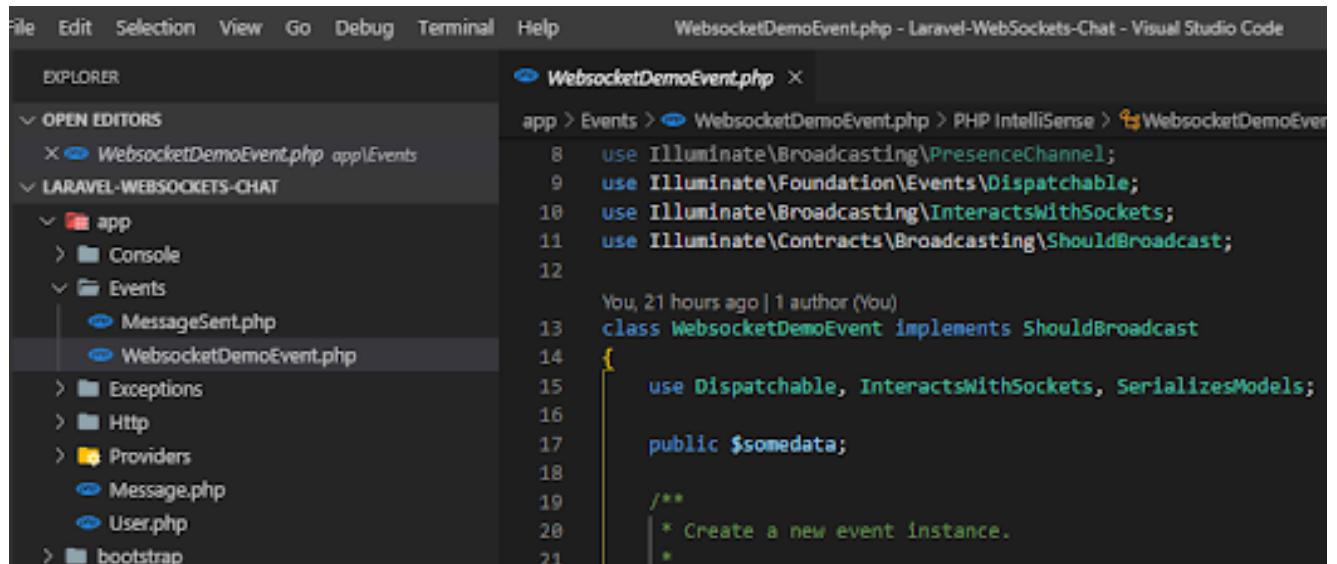
PRONTO SERVIDOR WEBSOCKET JÁ ESTÁ CONFIGURADO !!!

## Criar Evento de Demonstração

1 - Abra um novo terminal, execute | php artisan make:event WebsocketDemoEvent

## 2 - Configurar o arquivo criado em app/Events/WebsocketDemoEvent.php

### 2.1 - Implementar ShouldBroadcast.



The screenshot shows the Visual Studio Code interface. The left sidebar displays the project structure under 'OPEN EDITORS' and 'LARAVEL-WEBSOCKETS-CHAT'. The 'Events' folder contains 'MessageSent.php' and 'WebsocketDemoEvent.php', which is currently selected. The right pane shows the code for 'WebsocketDemoEvent.php':

```
You, 21 hours ago | 1 author (You)
8  use Illuminate\Broadcasting\PresenceChannel;
9  use Illuminate\Foundation\Events\Dispatchable;
10 use Illuminate\Broadcasting\InteractsWithSockets;
11 use Illuminate\Contracts\Broadcasting\ShouldBroadcast;
12
13 class WebsocketDemoEvent implements ShouldBroadcast
14 {
15     use Dispatchable, InteractsWithSockets, SerializesModels;
16
17     public $somedata;
18
19     /**
20      * Create a new event instance.
21      *
22      * @return void
23      */
24     public function __construct($somedata)
25     {
26         $this->somedata = $somedata;
27     }
28 }
```

### 2.2 - implementar método construtor.

```
You, 21 hours ago | 1 author (You)
13 class WebsocketDemoEvent implements ShouldBroadcast
14 {
15     use Dispatchable, InteractsWithSockets, SerializesModels;
16
17     public $somedata;
18
19     /**
20      * Create a new event instance.
21      *
22      * @return void
23      */
24     public function __construct($somedata)
25     {
26         $this->somedata = $somedata;
27     }
28 }
```

### 2.3 - Implementar a transmissão do Demo Channel.

```

28
29     /**
30      * Get the channels the event should broadcast on.
31      *
32      * @return \Illuminate\Broadcasting\Channel|array
33      */
34     public function broadcastOn()
35     {
36         return new Channel('DemoChannel');
37     }
38 }

```

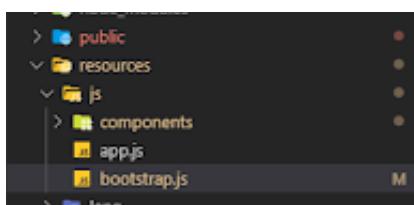
3 - Transmitir o evento na rota.

```

14
15
16 Route::get('/', function () {
17     // transmite o evento
18     broadcast(new WebsocketDemoEvent('some data'));
19     return view('welcome');
20 });
21

```

4 - Ouvir o websocketDemoEvent no arquivo bootstrap.js. | resources/js/components



```

58
59 // esse canal ouve o WebsocketDemoEvent
60 window.Echo.channel('DemoChannel')
61 .listen('WebsocketDemoEvent', (e) => {
62     console.log(e);
63 });
64
65

```

5 - Configurar a página html.

5.1 - Inserir o link do arquivo app.js na view welcome.

```

96         |     </div>
97         |   </div>
98
99         <script src="js/app.js"></script>
100
101     </body>
102 </html>
103

```

5.2 - Ainda na mesma view inserir o token meta tag.

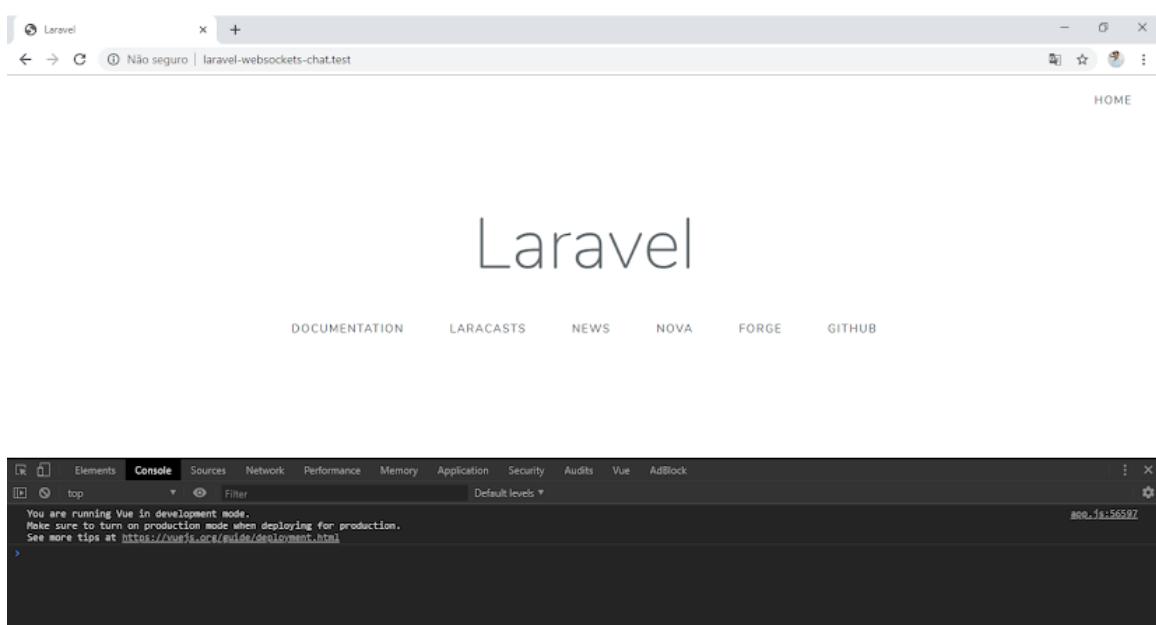
```
6
7 |     {{-- token meta tag --}}
8 |     <meta name="csrf-token" content="{{ csrf_token() }}">
9 |
```

5.3 - Adicionar atributo id="app" na tag div logo abaixo da tag body.

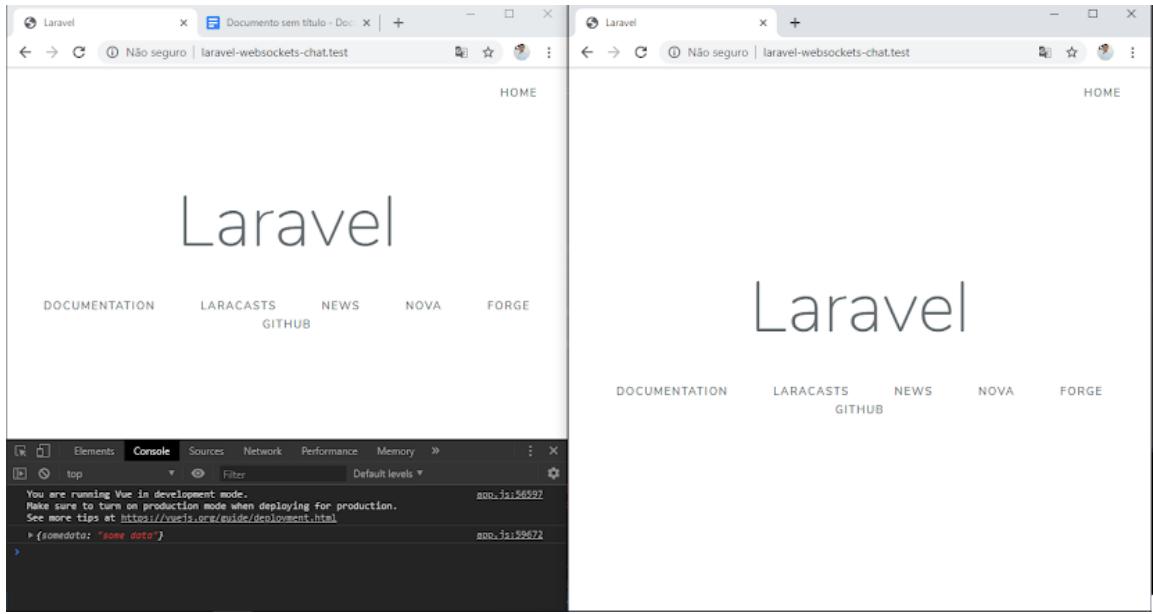
```
69 <body>
70     <div id="app" class="flex-center position-ref full-height">
71         @if (Route::has('login'))
72             <div class="top-right links">
73                 @auth
74                     <a href="{{ url('/home') }}>Home</a>
75                 @else
76                     <a href="{{ route('login') }}>Login</a>
77
78                     @if (Route::has('register'))
79                         <a href="{{ route('register') }}>Register</a>
80                     @endif
81                 @endauth
82             </div>
83         @endif

```

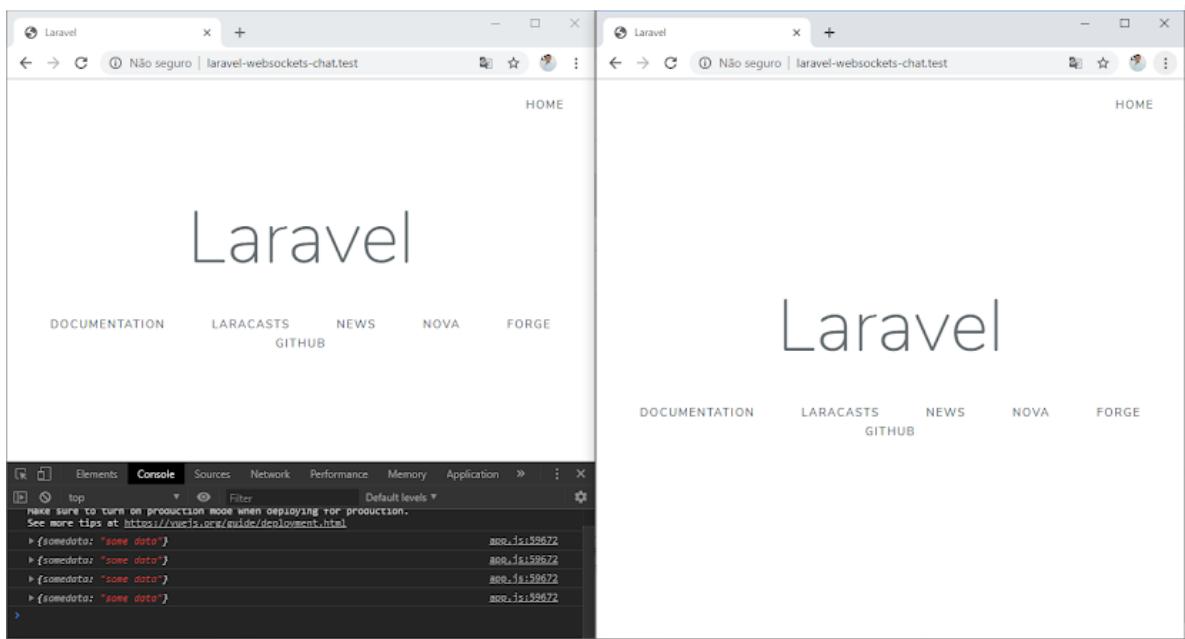
6 - Abra o laravel na página inicial e abra o console do navegador.



6.1 - Abra outra guia na página inicial do laravel.



Verifique que em tempo real apareceu no console os dados referentes ao evento adicionado. Atualiza a página aberta recentemente e veja novamente os dados sendo carregados.



7 - Você pode também adicionar eventos no painel do Laravel Websocket.

7.1 - Abra o painel do Laravel Websocket. Clique no botão “Connect”.

The screenshot shows a browser window titled "WebSockets Dashboard". At the top, there are buttons for "App: Laravel" (with a dropdown arrow), "Port: 6001", and "Disconnect". Below this, a message says "Channels current state is connected". A chart titled "Realtime Statistics" displays a horizontal line graph with values 0, 2, and 4 on the y-axis and 0, 2, 4, 6 on the x-axis. Under "Event Creator", there are three input fields: "Channel" (containing "DemoChannel"), "Event" (containing "App\\Events\\WebsocketDemoEvent"), and "Data" (containing "[{"qualquer": "qualquer dado"}]"). A blue "Send event" button is located at the bottom right of the form.

7.2 - Adicione um evento manual, passando dados no formato json, depois clique no botão “Send event”.

The screenshot shows the "Event Creator" form from the previous dashboard. The "Channel" field contains "DemoChannel", the "Event" field contains "App\\Events\\WebsocketDemoEvent", and the "Data" field contains "[{"qualquer": "qualquer dado"}]". A blue "Send event" button is visible at the bottom right.

8 - Se tudo de certo aparecerá na outra página aberta os dados enviados.

The screenshot shows a web browser window with the title "Laravel". The address bar indicates the site is "Não seguro | laravel-websockets-chat.test". The page content displays the large "Laravel" logo. Below the logo, there is a navigation bar with links: DOCUMENTATION, LARACASTS, NEWS, NOVA, FORGE, and GITHUB. At the bottom of the page, there is a footer with the text "© 2016 - 2017 Laravel, Inc." and a link to "Laravel.com".

The developer tools console is open, showing the following JavaScript object:

```
top
See more tips at https://edu-5.org/guide/debug.html
[{"->": {"0: {qualquer: "qualquer dado"}, length: 1}, __proto__: Array(0)}
```