

# 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
```

Você pode usar qualquer ID, KEY e SECRET.

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

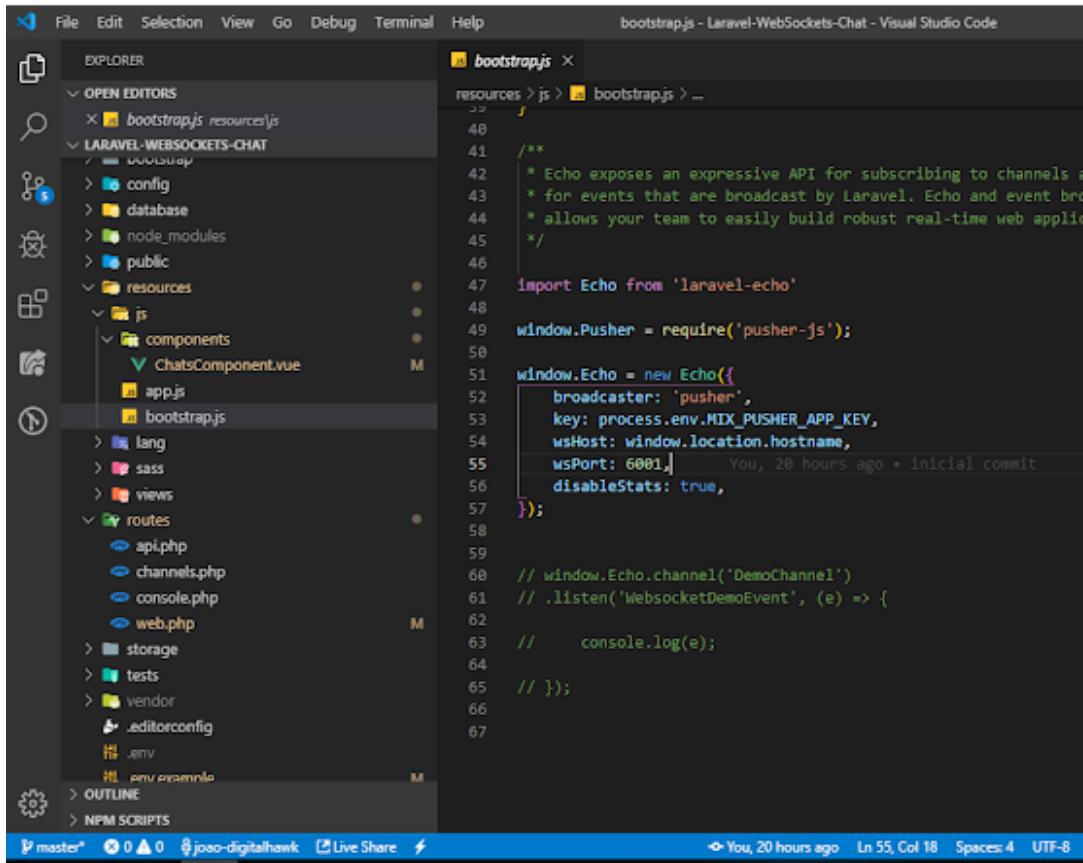
<https://docs.beyondco.de/laravel-websockets/1.0/basic-usage/pusher.html#requirements>

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.



The screenshot shows the Visual Studio Code interface with the following details:

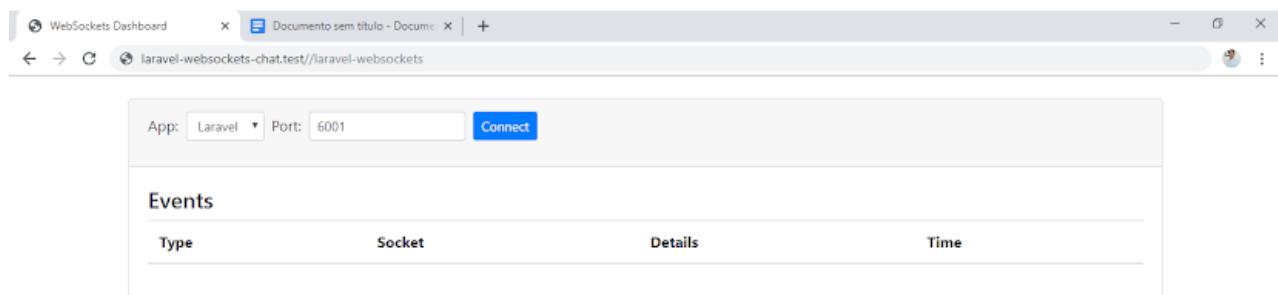
- File Explorer:** Shows the project structure:
  - bootstrap.js
  - resources
  - LARAVEL-WEBSOCKETS-CHAT
    - config
    - database
    - node\_modules
    - public
    - resources
      - js
        - components
          - ChatsComponent.vue
          - app.js
          - bootstrap.js
    - routes
      - api.php
      - channels.php
      - console.php
      - web.php
    - storage
    - tests
    - vendor
      - .editorconfig
      - .gitignore
- Editor:** The bootstrap.js file is open, showing its code. The code imports Echo from 'laravel-echo' and defines a new instance of Echo with various configuration options like broadcaster, key, wsHost, wsPort, and disableStats.
- Bottom Status Bar:** Shows the file path as bootstrap.js - Laravel-WebSockets-Chat - Visual Studio Code, and other status information like 'You, 20 hours ago'.

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>

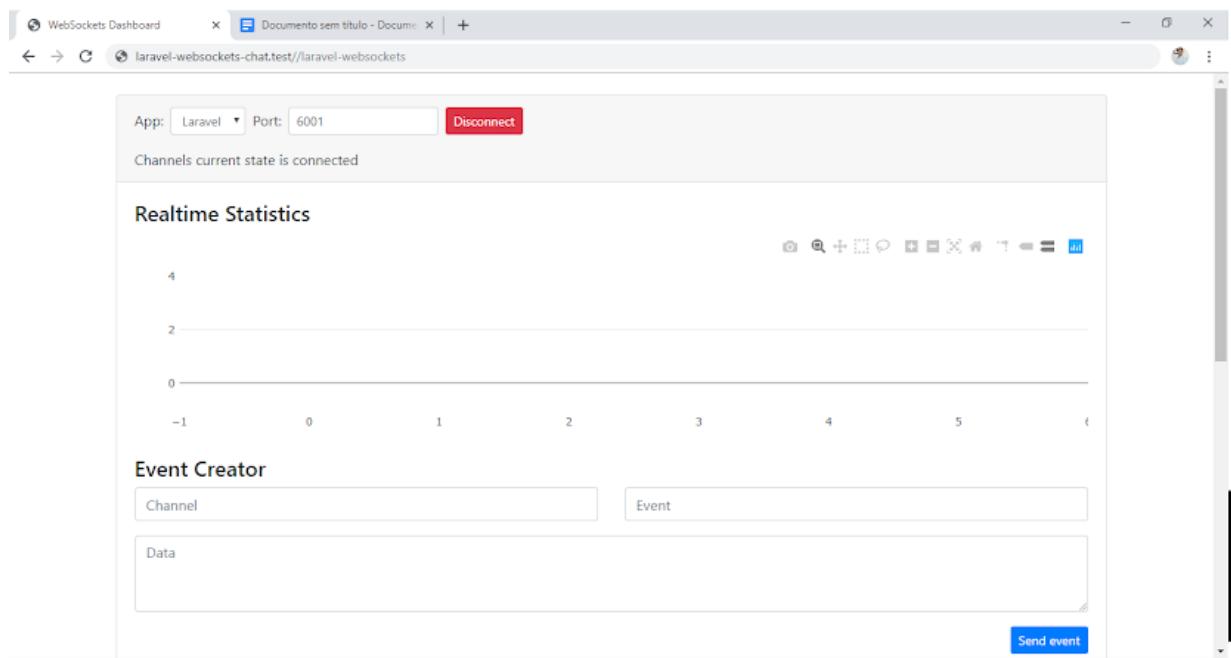
6 - Executar comando no terminal | npm run watch, para assistir as mudanças e compilar no app.js.

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.



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.

```

File Edit Selection View Go Debug Terminal Help WebsocketDemoEvent.php - Laravel-WebSockets-Chat - Visual Studio Code

EXPLORER
OPEN EDITORS
  WebsocketDemoEvent.php app\Events
LARAVEL-WEBSOCKETS-CHAT
  app
    Console
    Events
      MessageSent.php
      WebsocketDemoEvent.php
    Exceptions
    Http
    Providers
      Message.php
      User.php
    bootstrap
app > Events > WebsocketDemoEvent.php > PHP Intellisense > WebsocketDemoEvent.php
8   use Illuminate\Broadcasting\PresenceChannel;
9   use Illuminate\Foundation\Events\Dispatchable;
10  use Illuminate\Broadcasting\InteractsWithSockets;
11  use Illuminate\Contracts\Broadcasting\ShouldBroadcast;
12
13 You, 21 hours ago | 1 author (You)
14  class WebsocketDemoEvent implements ShouldBroadcast
15  {
16
17
18
19
20
21
  
```

## 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 }
```

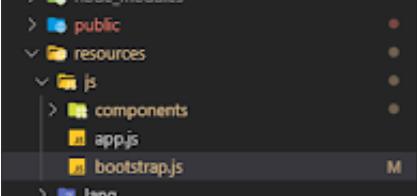
## 2.3 - Implementar a transmissão do Demo Channel.

```
28 /**
29  * Get the channels the event should broadcast on.
30  *
31  * @return \Illuminate\Broadcasting\Channel|array
32  */
33 public function broadcastOn()
34 {
35     return new Channel('DemoChannel');
36 }
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 // esse canal ouve o WebsocketDemoEvent
59 window.Echo.channel('DemoChannel')
60 .listen('WebsocketDemoEvent', (e) => {
61     console.log(e);
62 });
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 </body>
101 </html>
102
```

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

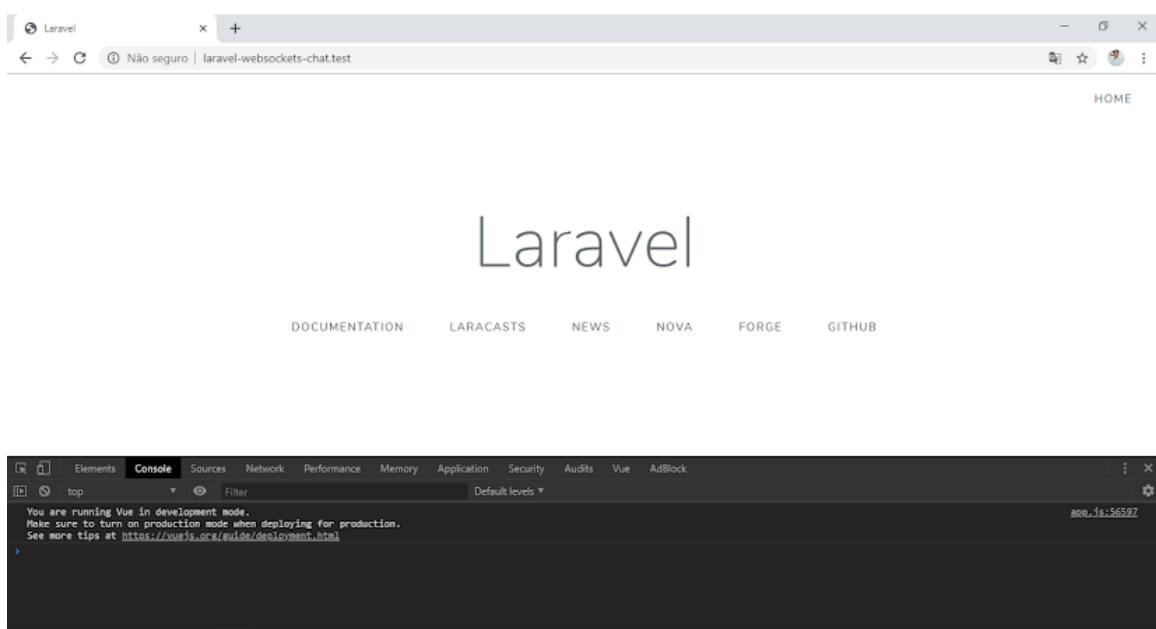
```
6
7     {{-- token meta tag --}}
8     <meta name="csrf-token" content="{{ csrf_token() }}>
9
```

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

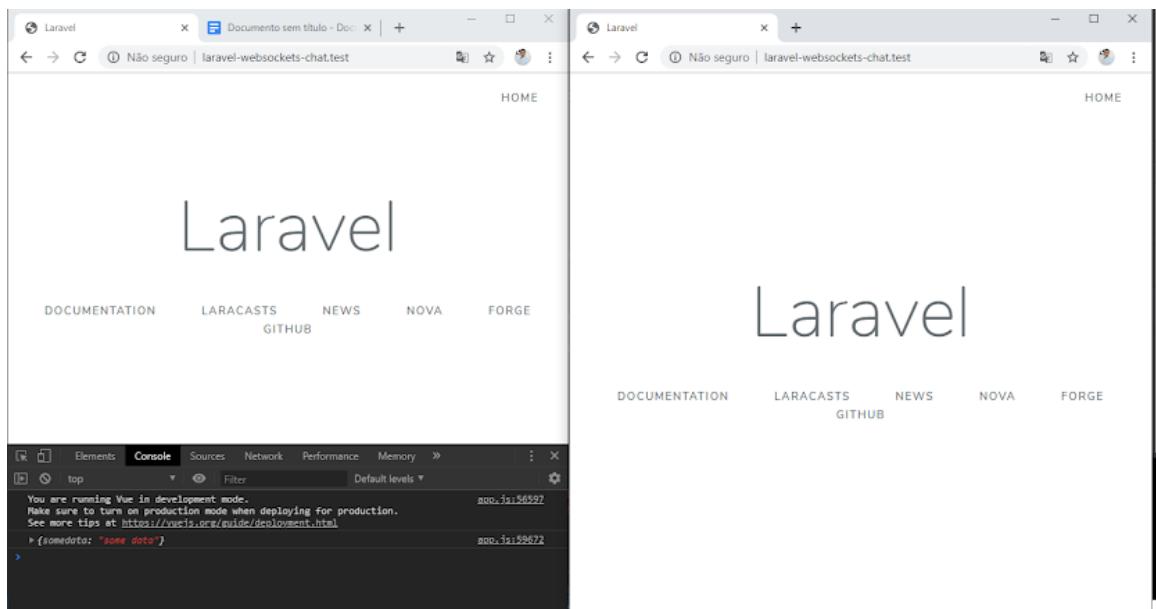
```
69 <body>
70     <div id="app" class="flex-center position-ref full-height">
71         @if (Route::has('login'))
72             <div class="top-right links">
73                 @auth
74                     <a href="{{ url('/home') }}>Home</a>
75                 @else
76                     <a href="{{ 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

```

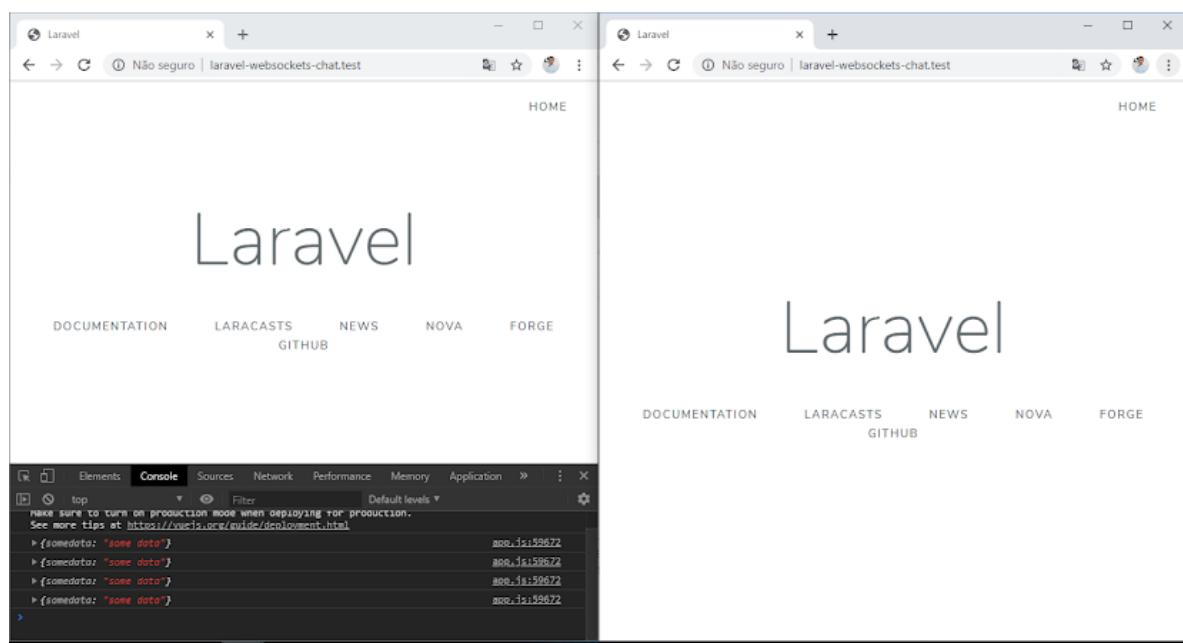
6 - Abra o laravel na página inicial e abra o console do navegador.



6.1 - Abra outra guia na página inicial do laravel.

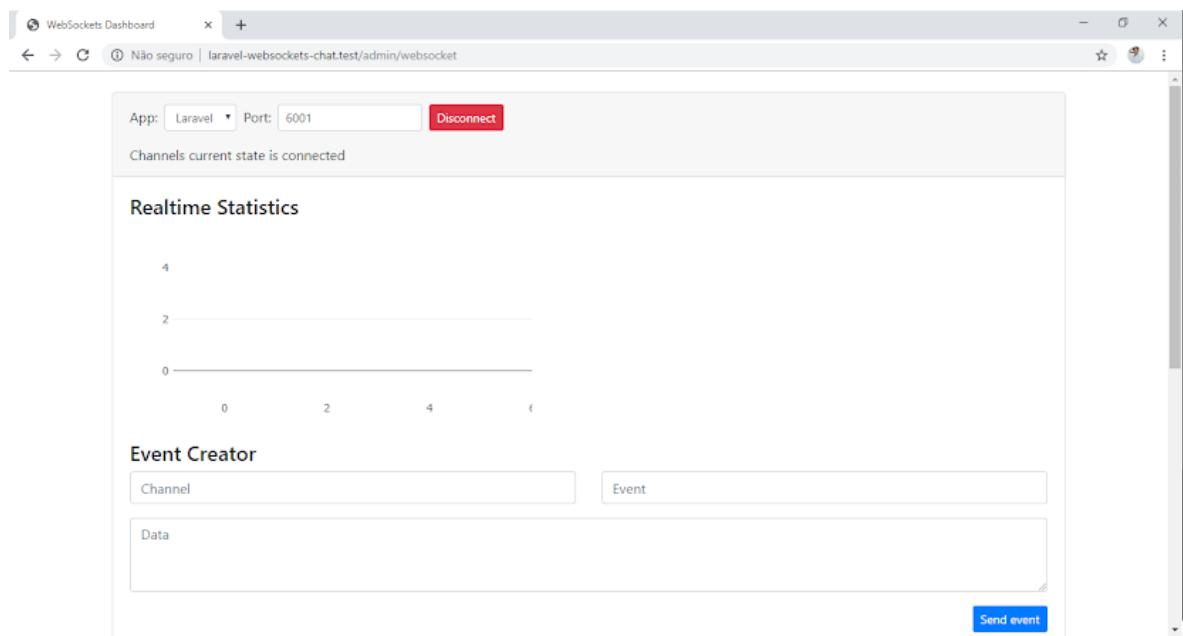


Verifique que em tempo real apareceu no console os dados referentes ao evento adicionado. Atualiza a página aberta recentemente e veja novamente os dados sendo carregados.



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

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



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

Event Creator

DemoChannel      App\Events\WebsocketDemoEvent

[{"qualquer": "qualquer dado"}]

Send event

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

Laravel

DOCUMENTATION LARACASTS NEWS NOVA FORGE GITHUB

```
Elements Console Sources Network Performance Memory Application Security Audits Vue AdBlock
See more tips at https://webtips.io/guide/development.html
> [{"qualquer": "qualquer dado"}]
= [{"qualquer": "qualquer dado"}]
  ↵ 0: {qualquer: "qualquer dado"}
  ↵ length: 1
  ↵ __proto__: Array(0)
>
```