

Guide is related to Laravel version 11

Step 1: Creating the project,

laravel new realtime-laravel-basics

```
Which Breeze stack would you like to install? [Blade with Alpine]:
[blade] Blade with Alpine
[livewire] Livewire (Volt Class API) with Alpine
[livewire-functional] Livewire (Volt Functional API) with Alpine
[react] React with Inertia
[vue] Vue with Inertia
[api] API only
> blade
blade

Would you like dark mode support? (yes/no) [no]:
> no

Which testing framework do you prefer? [Pest]:
[0] Pest
[1] PHPUnit
> 1
1

Would you like to initialize a Git repository? (yes/no) [no]:
> no
```

Step 2: Installing broadcasting,

php artisan install:broadcasting

After a successful install, resources->js er under e ekta echo.js naam e ekta file create hobe and .env teo change ashbe,

```
REVERB_APP_ID=116444
REVERB_APP_KEY=ucw6h0e0s99jlbhljgca
REVERB_APP_SECRET=cdllrog1x4a4cwoskkch
REVERB_HOST="realtime-laravel-basics.test"
REVERB_PORT=8080
REVERB_SCHEME=http
```

```
▼ js
  JS app.js
  JS bootstrap.js
  JS echo.js
```

Step 3: starting server,

php artisan reverb:start

Joining Channels and events

Step 4: npm run dev

Step 5: **php artisan reverb:start --debug** [eta step 3 ke close kore run korte hobe]

Step 6: Ekta event create kora holo,

php artisan make:event Example

***** Step 4 and 5 on thaka lagbe always *****

Ekhn **events->Example.php** er moddhe ekta channel create kora holo,

```
public function broadcastOn(): array
{
    return [
        new Channel('chat'),
    ];
}
```

Full Example.php code, adding “**ShouldBroadcastNow**”

```
<?php
namespace App\Events;
use Illuminate\Broadcasting\Channel;
use Illuminate\Broadcasting\InteractsWithSockets;
use Illuminate\Contracts\Broadcasting\ShouldBroadcastNow;
use Illuminate\Foundation\Events\Dispatchable;
use Illuminate\Queue\SerializesModels;

class Example implements ShouldBroadcastNow
{
    use Dispatchable, InteractsWithSockets, SerializesModels;

    public function __construct()
    {
        //
    }

    public function broadcastOn(): array
    {
        return [
            new Channel('chat'),
        ];
    }
}
```

Ekhn **dashboard.php** te **x-init** add kora holo, **x-init** er kaj hocche simply js code lekha jay code er moddhe,

```
<div class="p-6 text-gray-900"
  x-init="
    Echo.channel('chat').listen('Example', (event)=>{
      console.log(event)
    })
  "
>
```

- Ei example e Echo use kore channel ke listen kora holo
- Ekhn <http://127.0.0.1:8000/broadcast> ei route jeta te broadcast er event chilo, ei link e click korle trigger hobe publicly, cause channel ta hocche public route er.

Adding data to events

Ekhn data ke jevabe broadcast korte hoy,

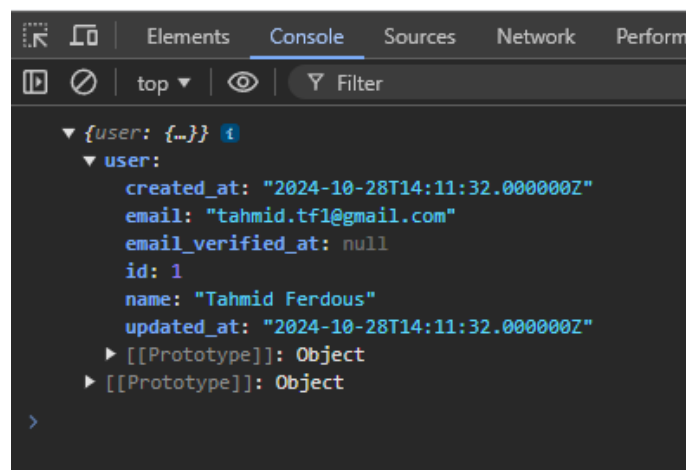
From **web.php**,

```
Route::get('broadcast', function () {
    broadcast(new \App\Events\Example(\App\Models\User::find(1)));
});
```

From **Example.php**, just etuk ei change kora,

```
public function __construct(public User $user)
{
}
}
```

- Ekhn <http://127.0.0.1:8000/broadcast> ke hit korle console e emn output paoa jabe,



Customizing broadcast event data

broadcastWith() er kaj ta hocche broadcast kora message ta, but direct __construct method er through te hobe na,

```
public function broadcastWith(): array{
    return [
        'id' => $this->user->id,
        'name' => $this->user->name,
    ];
}
```

Full Example.php snippet,

```
<?php
namespace App\Events;
use App\Models\User;
use Illuminate\Broadcasting\Channel;
use Illuminate\Broadcasting\InteractsWithSockets;
use Illuminate\Contracts\Broadcasting\ShouldBroadcastNow;
use Illuminate\Foundation\Events\Dispatchable;
use Illuminate\Queue\SerializesModels;

class Example implements ShouldBroadcastNow
{
    use Dispatchable, InteractsWithSockets, SerializesModels;

    public function __construct(protected User $user)
    {

    }

    public function broadcastWith(): array{
        return [
            'id' => $this->user->id,
            'name' => $this->user->name,
        ];
    }

    public function broadcastOn(): array
    {
        return [
            new Channel('chat'),
        ];
    }
}
```

- __construct() method er data ekhane fully protected state e ase, publicly accessible na.

Now adding a new model named **Message**, for project purpose,

php artisan make:model Message -m

Ekhon web.php route e 2nd parameter post kora holo and contructor and broadcastWith() method er majhe pathano holo data gula ke,

web.php code,

```
Route::get('broadcast', function () {
    broadcast(new \App\Events\Example(User::find(1), Message::find(1)));
});
```

Full Example.php code,

```
<?php
namespace App\Events;
use App\Models\Message;
use App\Models\User;
use Illuminate\Broadcasting\Channel;
use Illuminate\Broadcasting\InteractsWithSockets;
use Illuminate\Contracts\Broadcasting\ShouldBroadcastNow;
use Illuminate\Foundation\Events\Dispatchable;
use Illuminate\Queue\SerializesModels;

class Example implements ShouldBroadcastNow
{
    use Dispatchable, InteractsWithSockets, SerializesModels;

    public function __construct(protected User $user, protected Message $message)
    {
    }

    public function broadcastWith(): array{
        return [
            'user' => [
                'id' => $this->user->id,
                'name' => $this->user->name,
            ],
            'message' => [
                'message' => $this->message->id
            ]
        ];
    }

    public function broadcastOn(): array
    {
        return [
            new Channel('chat'),
        ];
    }
}
```

- Notice broadcastWith() er moddhe data gula ke array kore pathano hoise, output snippet,

```
▼ {user: {...}, message: {...}} ⓘ
  ► message: {message: 1}
  ► user: {id: 1, name: 'Tahmid Ferdous'}
  ► [[Prototype]]: Object
```

Namespace Events

Suppose directory base e event create holo folder er moddhe like controllers. Tahole Echo diye kaj korar shomoy full path value ta evabe bole dite hobe,

```
x-init="

Echo.channel('chat')

    .listen('.App\\Events\\Chat\\ExampleTwo', (event) => {

        console.log(event)

    })
```

##Queued Events

By default e queue setup korai thake Laravel 11 veresion e, just run the command. Eta hocche jokhon amra **ShouldBroadcast** mode e rakhbo

```
Usage
class Example implements ShouldBroadcast
```

php artisan queue:work