PUSHER WORKING PROCEDURE

- STEP 1: Open account in pusher
  - https://pusher.com/
- STEP 2: Run command to install pusher:
  - `composer require pusher/pusher-php-server`
- STEP 3: After registration -> have to change in **.env** file
  - `BROADCAST_DRIVER=pusher`
  - `PUSHER_APP_ID=_____`
  - `PUSHER_APP_KEY=_____`
  - `PUSHER_APP_SECRET=_____`
  - `PUSHER_APP_CLUSTER=____`
- STEP 4: Comment out this line from **config/app.php**
  - `App\Providers\BroadcastServiceProvider::class,`
- STEP 5: Create Folder and file in **app folder**
  - `App\Events\NewNotification.php`
  - `App\Listeners\NewNotificationListener.php`
- STEP 6: Add **use** in Controller
  - `use App\Events\NewNotification;`
- STEP 7: Add code in function before return

```
$notificationData = array(
    'message_array' => 'Hello', //pass prepared array if needed
    'channel' => 'my-activity'
);
event(new NewNotification($notificationData));
```

- STEP 8: Add script in view file where have to change

```
<script src="https://js.pusher.com/5.0/pusher.min.js"></script>
<script>
        //Enable pusher logging - don't include this in production
        Pusher.logToConsole = true;
        var pusher = new Pusher('pusher_app_key', {
            cluster: 'ap2',
            forceTLS: true
        });
        var channel = pusher.subscribe('activity-comment');
        channel.bind('App\\Events\\NewNotification', function(data) {
            //alert(JSON.stringify(data));
        });

</script>
```

NewNotification.php

```
<?php
namespace App\Events;

use Illuminate\Broadcasting\Channel;
use Illuminate\Queue\SerializesModels;
use Illuminate\Broadcasting\PrivateChannel;
use Illuminate\Broadcasting\PresenceChannel;
use Illuminate\Foundation\Events\Dispatchable;
use Illuminate\Broadcasting\InteractsWithSockets;
use Illuminate\Contracts\Broadcasting\ShouldBroadcast;
use Session;
class NewNotification implements ShouldBroadcast
{
    use Dispatchable, InteractsWithSockets, SerializesModels;
    public $message;
    /**
     * Create a new event instance.
     *
     * @return void
     */
    public function __construct($message){
        $this->message = $message;
    }

    /**
     * Get the channels the event should broadcast on.
     *
     * @return \Illuminate\Broadcasting\Channel|array
     */
    public function broadcastOn(){
        return new Channel($this->message['channel']);
    }

    public function broadcastWith(){
        return $this->message;
    }
}
```

NewNotificationListener.php

```
<?php
namespace App\Listeners;

use App\Events\NewNotification;
use Illuminate\Queue\InteractsWithQueue;
use Illuminate\Contracts\Queue\ShouldQueue;
class NewNotificationListener{

    /**
     * Create the event listener.
     *
     * @return void
     */

    public function __construct(){

    }

    /**
     * Handle the event.
     *
     * @param  NewNotification $event
     * @return void
     */

    public function handle(NewNotification $event){
      //  return $event;
    }
}
```