In Laravel, the best way to pass different types of flash messages in the session

I'm making my first app in Laravel and am trying to get my head around the session flash messages. As far as I'm aware in my controller action I can set a flash message either by going

```
Redirect::to('users/login')->with('message', 'Thanks for registering!'); //is this actually OK?
```

For the case of redirecting to another route, or

```
Session::flash('message', 'This is a message!');
```

In my master blade template I'd then have:

```
@if(Session::has('message'))
<p class="alert alert-info">{{ Session::get('message') }}
</p>
@endif
```

As you may have noticed I'm using Bootstrap 3 in my app and would like to make use of the different message classes: `alert-info`, `alert-warning`, `alert-danger` etc.

Assuming that in my controller I know what type of message I'm setting, what's the best way to pass and display it in the view? Should I set a separate message in the session for each type (e.g. `Session::flash('message_danger', 'This is a nasty message! Something's wrong.');`)?
Then I'd need a separate if statement for each message in my blade template.

Any advice appreciated.

asked Jan 8 '14 at 19:02

harryg

4,394 9 66 120

10 Answers 10

up vote 70 down vote accepted

One solution would be to flash two variables into the session:

1. The message itself
2. The "class" of your alert

for example:

```
Session::flash('message', 'This is a message!');
Session::flash('alert-class', 'alert-danger');
```

Then in your view:

```
@if(Session::has('message'))
<p class="alert {{ Session::get('alert-class', 'alert-info') }}">{{
Session::get('message') }}</p>
@endif
```

Note I've put a default value into the `Session::get()`. that way you only need to override it if the warning should be something other than the `alert-info` class.

(that is a quick example, and untested :))

up vote 17 down vote

My way is to always Redirect::back() or Redirect::to():

```
Redirect::back()->with('message', 'error|There was an
error...');

Redirect::back()->with('message', 'message|Record updated.');

Redirect::to('/')->with('message', 'success|Record updated.');
```

I have a helper function to make it work for me, usually this is in a separate service:

```
function displayAlert()
{
      if (Session::has('message'))

          list($type, $message) = explode('|', Session::get('message'));

          $type = $type == 'error' : 'danger';
          $type = $type == 'message' : 'info';

          return sprintf('<div class="alert alert-%s">%s</div>', $type,
message);
      }

      return '';
}
```

And in my view or layout I just do

```
{{ displayAlert() }}
```

up vote 16 down vote

In your view:

```
<div class="flash-message">
  @foreach (['danger', 'warning', 'success', 'info'] as $msg)
    @if(Session::has('alert-' . $msg))
    <p class="alert alert-{{ $msg }}">{{ Session::get('alert-' . $msg) }}
</p>
    @endif
  @endforeach
</div>
```

Then set a flash message in the controller:

```
Session::flash('alert-danger', 'danger');
Session::flash('alert-warning',
'warning');
Session::flash('alert-success',
'success');
Session::flash('alert-info', 'info');
```

answered Sep 4 '14 at 3:57

danelips

30624

up vote 5 down vote

Simply return with the 'flag' that you want to be treated without using any additional user function. The Controller:

```
return \Redirect::back()->withSuccess( 'Message you want show in View'
);
```

Notice that I used the 'Success' flag.

The View:

```
@if( Session::has( 'success' ))
     {{ Session::get( 'success' ) }}
@elseif( Session::has( 'warning' ))
   {{ Session::get( 'warning' ) }} <!-- here to 'withWarning()' -->
@endif
```

Yes, it really works!

up vote 4 down vote

Another solution would be to create a helper class How to Create helper classes here

```
class Helper{
    public static function format_message($message,$type)
    {
          return '<p class="alert alert-
'.$type.'">'.$message.'</p>'
    }
}
```

Then you can do this.

```
Redirect::to('users/login')->with('message', Helper::format_message('A bla blah
occured','error'));
```

or

```
Redirect::to('users/login')->with('message', Helper::format_message('Thanks for
registering!','info'));
```

and in your view

```
@if(Session::has('message'))

{{Session::get('message')}}
@endif
```

up vote 2 down vote

You can make a multiple messages and with different types. Follow these steps below:

1. Create a file: "`app/Components/FlashMessages.php`"

```
namespace App\Components;

trait FlashMessages
{
    protected static function message($level = 'info', $message = null)
    {
        if (session()->has('messages')) {
            $messages = session()->pull('messages');
        }

        $messages[] = $message = ['level' => $level, 'message' =>
$message];

        session()->flash('messages', $messages);

        return $message;
    }

    protected static function messages()
    {
        return self::hasMessages() ? session()->pull('messages') : [];
    }

    protected static function hasMessages()
    {
        return session()->has('messages');
    }

    protected static function success($message)
    {
        return self::message('success', $message);
    }

    protected static function info($message)
    {
        return self::message('info', $message);
    }

    protected static function warning($message)
    {
        return self::message('warning', $message);
    }

    protected static function danger($message)
    {
        return self::message('danger', $message);
    }
}
```

1. On your base controller "app/Http/Controllers/Controller.php".

```
namespace App\Http\Controllers;

use Illuminate\Foundation\Bus\DispatchesJobs;
use Illuminate\Routing\Controller as BaseController;
use Illuminate\Foundation\Validation\ValidatesRequests;
use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
use Illuminate\Foundation\Auth\Access\AuthorizesResources;

use App\Components\FlashMessages;

class Controller extends BaseController
{
    use AuthorizesRequests, AuthorizesResources, DispatchesJobs,
ValidatesRequests;

    use FlashMessages;
}
```

This will make the `FlashMessages` trait available to all controllers that extending this class.

1. Create a blade template for our messages: "`views/partials/messages.blade.php`"

```
@if (count($messages))
<div class="row">
  <div class="col-md-12">
  @foreach ($messages as $message)
      <div class="alert alert-{{ $message['level'] }}">{!!
$message['message']!!}</div>
  @endforeach
  </div>
</div>
@endif
```

1. On "`boot()`" method of "`app/Providers/AppServiceProvider.php`":

```
namespace App\Providers;

use Illuminate\Support\ServiceProvider;
use App\Components\FlashMessages;

class AppServiceProvider extends ServiceProvider
{
  use FlashMessages;

    public function boot()
    {
        view()->composer('partials.messages', function ($view)
{

            $messages = self::messages();

            return $view->with('messages', $messages);
        });
    }

    ...
}
```

This will make the `$messages` variable available to "`views/partials/message.blade.php`" template whenever it is called.

1. On your template, include our messages template - "`views/partials/messages.blade.php`"

```
<div class="row">
  <p>Page title goes here</p>
</div>

@include
('partials.messages')

<div class="row">
  <div class="col-md-12">
      Page content goes here
  </div>
</div>
```

You only need to include the messages template wherever you want to display the messages on your page.

1. On your controller, you can simply do this to push flash messages:

```
use App\Components\FlashMessages;

class ProductsController {

  use FlashMessages;

  public function store(Request $request)
  {
      self::message('info', 'Just a plain message.');
      self::message('success', 'Item has been added.');
      self::message('warning', 'Service is currently under
maintenance.');
      self::message('danger', 'An unknown error occured.');

      //or

      self::info('Just a plain message.');
      self::success('Item has been added.');
      self::warning('Service is currently under maintenance.');
      self::danger('An unknown error occured.');
  }

  ...
```

Hope it'l help you.

up vote 1 down vote

For my application i made a helper function:

```
function message( $message , $status = 'success', $redirectPath = null )
{
     $redirectPath = $redirectPath == null ? back() : redirect( $redirectPath
);

     return $redirectPath->with([
         'message'    =>  $message,
         'status'     =>  $status,
    ]);
}
```

message layout, "main.layouts.message":

```
@if($status)
   <div class="center-block affix alert alert-{{$status}}">
     <i class="fa fa-{{ $status == 'success' ? 'check' : $status}}">
</i>
     <span>
        {{ $message }}
     </span>
   </div>
@endif
```

and import every where to show message:

```
@include('main.layouts.message', [
    'status'    =>  session('status'),
    'message'   =>
session('message'),
])
```

up vote 1 down vote

You could use Laravel Macros.

You can create `macros.php` in `app/helpers` and include it routes.php.

if you wish to put your macros in a class file instead, you can look at this tutorial:
http://chrishayes.ca/blog/code/laravel-4-object-oriented-form-html-macros-classes-service-provider

```
HTML::macro('alert', function($class='alert-danger', $value="",$show=false)
{

    $display = $show ? 'display:block' : 'display:none';

    return
        '<div class="alert '.$class.'" style="'.$display.'">
            <button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button>
            <strong><i class="fa fa-times"></i></strong>'.$value.'
        </div>';
});
```

In your controller:

```
Session::flash('message', 'This is so dangerous!');
Session::flash('alert', 'alert-danger');
```

In your View

```
@if(Session::has('message') && Session::has('alert') )
  {{HTML::alert($class=Session::get('alert'), $value=Session::get('message'),
$show=true)}}
@endif
```

up vote 0 down vote

I usually do this

in my store() function i put success alert once it saved properly.

```
\Session::flash('flash_message','Office successfully
updated.');
```

in my destroy() function, I wanted to color the alert red so to notify that its deleted

```
\Session::flash('flash_message_delete','Office successfully deleted.');
```

Notice, we create two alerts with different flash names.

And in my view, I will add condtion to when the right time the specific alert will be called

```
@if(Session::has('flash_message'))
    <div class="alert alert-success"><span class="glyphicon glyphicon-ok"></span>
<em> {!! session('flash_message') !!}</em></div>
@endif
@if(Session::has('flash_message_delete'))
    <div class="alert alert-danger"><span class="glyphicon glyphicon-ok"></span>
<em> {!! session('flash_message_delete') !!}</em></div>
@endif
```

Here you can find different flash message stlyes Flash Messages in Laravel 5

answered Nov 26 '15 at 13:07

never

69731436

Not the answer you're looking for? Browse other questions tagged session laravel laravel-4 or ask your own question.