

Desarrollo de aplicaciones web con Laravel 5.1

Clase 9. Formularios, sesiones y email

Carlos Ruiz Ruso • @micromante



DEMO FORMULARIO DE CONTACTO

FORMULARIOS + REQUEST + BLADE + SESIONES + ERRORES

Formularios

Formularios

- Existe la posibilidad de integrar con blade a traves de:
 - `"illuminate/html": "~5.0"`
- Providers: `'Illuminate\Html\HtmlServiceProvider'`
- Facades:
 - `'Form' => 'Illuminate\Html\FormFacade',`
 - `'HTML' => 'Illuminate\Html\HtmlFacade',`
- Sintaxis sencilla y amigable
- Potencia e integración
- Seguridad con CSRF Protection
- Podemos usar también los formularios normales de html pero esto simplifica el código.

Crear un Formularios

{!! Form::open(\$config) !!} //Equivale a `<form class="" id="">`

{!! Form::close() !!} //Equivale a `</form>`

- `$config`, array de configuración
- `array('url' => 'ruta', 'method' => 'POST', 'files' => true)`
 - `method => Permite POST,GET`
 - `files => activamos el formdata para subida de ficheros.`
- `array('route' => 'route.name',)`
- `array('action' => 'ControllerName@metodo',$params_route)`
- `array('action' => 'ControllerName@metodo',$params_route)`
 - `$params_route` para acciones via URL

Añadir configuración a Formularios

- **Form::open(array(\$config), 'class' => 'form', 'id' => 'idname', ...)**
 - Podemos pasarle los parámetros que necesitamos y los añadirá.

Protección frente a SPAM y seguridad

- Dentro del formulario ejecutamos
 - `{{ Form::token() }}` esto genera un input type hidden con un número cifrado.
- En nuestra ruta del formulario añadimos el filtro
 - `Route::post('perfil', array('before' => 'csrf',))`
- Con esto evitamos
 - Evitamos peticiones Cross-site

Generación de labels

**Form::label('nameinput', 'Label Name', array
(**'class'** => '**...**', **'id'** => '**...**', ...))**

Input, Textarea, Password y Hidden

Form::text('nameinput', 'Label Name', array('class' => '...', 'id' => '...', ...))

Form::password('nameinput', 'Label Name', array('class' => '...', 'id' => '...', ...))

Form::email('nameinput', 'Label Name', array('class' => '...', 'id' => '...', ...))

Form::file('nameinput', \$attributes = array(), array('class' => '...', 'id' => '...', ...))

Checkbox, radio buttons y File Input

Form::**radio**('nameinput', value', array('class' => '...', 'id' => '...', ...))

Form::**checkbox**('nameinput', value', array('class' => '...', 'id' => '...', ...))

Form::**file**('nameinput', \$attributes = array(), array('class' => '...', 'id' => '...', ...))

Select boxes o desplegables

Form::select('nameinput', array('1' => 'Verdadero', '0' => 'Falso'),
array('class' => '...', 'id' => '...', ...))

Botones

```
{{ Form::submit('Aceptar envio', .....) }}
```

```
{{ Form::button('Aceptar envio', .....) }}
```

Documentación Laravel Collective

<http://laravelcollective.com/docs/5.1/html>

Sesiones

Sesiones

- **Manera de almacenar información entre peticiones**
- **API unificada**
- **Sintaxis sencilla**
- **Permite Memcache y redis**
- **Nombres reservados**
 - **“flash”, se utiliza para hacer mensajes de sesión**
- **<http://laravel.com/docs/5.1/session>**

Manejo de sesiones I

- **`$request->session()->put('llave','valor')`**
 - La llave podría ser un array y podemos usarlo con el '.' de modo que llave.subllave
- **`echo $request->session()->get('value')`**
 - Permite usar `Session::get('value','default value')`
- **`$request->session()->all()`**
 - Recuperamos todos los valores almacenados como array.
- **`$request->session()->has('llave')`**
 - Existencia de un valor o clave
- **`$request->session()->pull('key','value'), $request->session()->forget('key')`**
 - Borrado de valores

Manejo de sesiones II

- **`$request->session()->flush()`**
 - Borrado de todos los elementos en sesión
- **`$request->session()->regenerate()`**
 - Genera una nueva clave de sesión ID
- **Mensajes de avisos de un solo uso y luego autoborrado**
 - **`$request->session()->flash('message','value')`**
 - En la vista hacemos **`{{Session::get('message') }}`**
 - Automáticamente se borra en el salto siguiente
 - **`$request->session()->reflash()`** por si queremos regenerar el salto
- **`$request->session()->keep(array('key1','key2'))`**
 - Reflash solamente las claves indicadas

Guardado de sesiones en la base de datos

- **Por defecto se guarda en app/storage**
- **Podemos cambiarlo para que se guarde en la bd**
 - **automaticamente**
 - **php artisan session:table**
 - **composer dump-autload**
 - **php artisan migrate**
- **Fichero configuración**
 - **config/sessions.php**
 - **Drivers: native, cookie, database**

MAIL

<http://laravel.com/docs/5.1/mail>

MUCHAS GRACIAS A TODOS!

Podeis seguirme en la redes sociales como @micromante o Carlos Ruiz Ruso
www.micromante.com