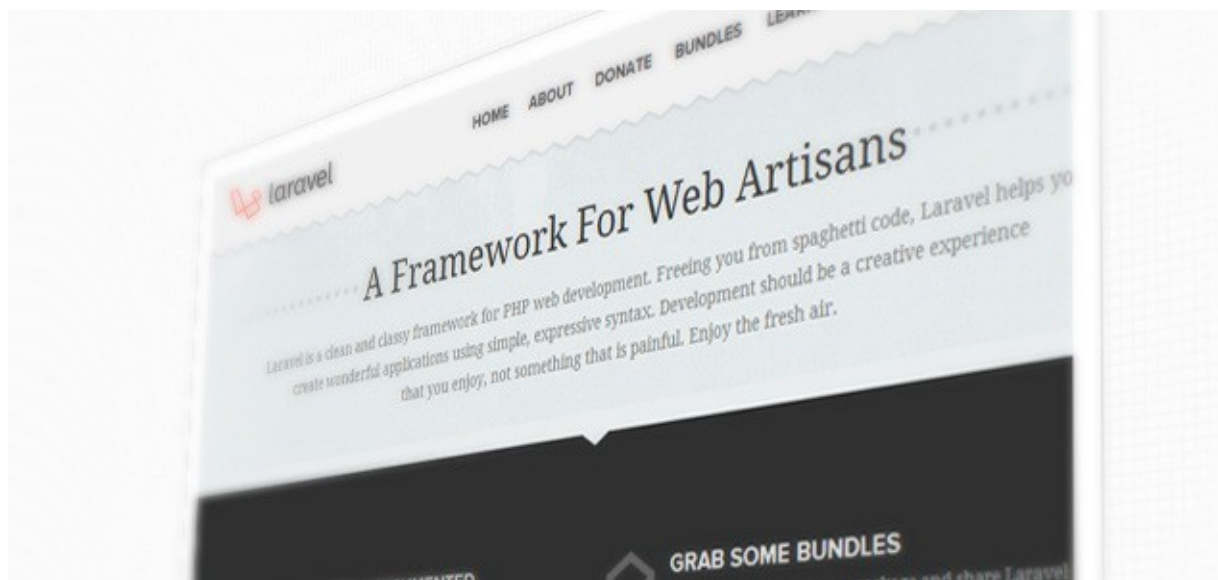


Objet : Utiliser LARAVEL 3

L A R A V E L 3



Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 1 / 102	

DOCUMENTATION SUR LARAVEL

URL officielles de Laravel :

Version 3.2	http://laravel.com/
Documentation français	http://laravel.fr/
Laravel News	http://laravelnews.tumblr.com/

Autres tutoriels

Tutoriels Tutsplus	http://net.tutsplus.com/tag/laravel/
Laravel pour débutant	http://antjanus.com/blog/web-development-tutorials/laravel-tips-tricks-for-beginners/
Tutoriels	http://maxoffsky.com/category/code-blog/
Tutoriels de la communauté Laravel	http://laravel.io/
Tutoriel de l'auteur de Code Happy	http://daylerees.com/category/laravel-tutorials/
Tutoriels	http://jasonlewis.me/laravel-tutorials
Création d'un système d'administration	https://gist.github.com/anchetaWern/4223764
Upload image	https://gist.github.com/sineld/3780572 http://jasny.github.com/bootstrap/javascript.html#fileupload
Upload image	http://forums.laravel.io/viewtopic.php?id=2022
Laravel 4	http://laravel.sl-creation.org/
URL Multi-langues	http://stackoverflow.com/questions/13763447/routing-urls-with-language-in-laravel

Livres .PDF

Par Dayles Rees	Laravel_Starter.pdf
Site Internet	http://heybigname.com/category/screencasts?page=1

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 2 / 102	Dernière impression le

Par Shawn McCool	Code Happy.pdf
Site Internet	http://codehappy.daylerees.com/

Pour Sublime2

Laravel Blade Syntax	https://github.com/Medalink/Laravel-Blade
Thème couleur	https://github.com/daylerees/colour-schemes

Laravel 4 : <http://laravel.sl-creation.org/laravel-4/>

AUTRES LIENS

Qualité code PHP	http://cs.sensiolabs.org/
Tester sous IE (Origine Microsoft)	http://www.modern.ie/fr-fr
Texte en Ascii	http://www.network-science.de/ascii/(ogre) http://patorjk.com/software/taag/#p=display&f=Small&t=Type%20Something%20
BootSnipp liste de ressources et création formulaires pour Bootstrap Twitter	http://bootsnipp.com/
Conception theme Bootstrap	http://pikock.github.com/bootstrap-magic/app/index.html#!/editor http://stylebootstrap.info/ http://bootswatch.com/
PAYANT - 16\$	https://wrapbootstrap.com/theme/crisp-responsive-fluid-business-template-WB0F7FH2P

TEST

Formulaire : <https://github.com/Anahkiasen/former>
<http://anahkiasen.github.com/former/>

Test du bundle Former : intéressant mais on s'éloigne trop de l'orthodoxie Laravel. Ne pas utiliser !!!

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 02/p2
Document : Laravel3_A_Savoir.odt	
Page 3 / 102	Dernière impression le

Administration : <https://github.com/Vespakoen/authority-laravel>

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 4 / 102	

FORMATIONS VIDEO

TUTORIELS LARAVEL 3 - SimpleDev

URL des vidéos en Français pour la création d'un système de commentaires avec authentification des utilisateurs avec LARAVEL 3. (PHP 5.3 minimum)

URL	Thématique
http://www.youtube.com/watch?feature=player_embedded&v=bC1Z5CerhJA# !	Installer LARAVEL 3 Déclaration des tables de l'application Utilisation du mode commande
http://www.youtube.com/watch?v=jONjXj9TJ-4	Élaboration des modèles
http://www.youtube.com/watch?v=HT7KaUDEk08	
http://www.youtube.com/watch?v=Lnmc7JWWNQk	
http://www.youtube.com/watch?v=mhQ2B87t-No	
	Système PAYPAL avec Laravel.
	Pagination avec Laravel et Bootstrap Twitter.

MODE DE COMMANDE

La commande **artisan** s'utilise en mode commande dans le dossier ou se trouve Laravel.

Pour connaître la liste des commandes possibles : (Laravel4)

```
>php artisan -h
```

Usage:

```
[options] command [arguments]
```

Options:

```
--help          -h Display this help message.
--quiet         -q Do not output any message.
--verbose       -v Increase verbosity of messages.
--version       -V Display this application version.
--ansi          Force ANSI output.
--no-ansi       Disable ANSI output.
--no-interaction -n Do not ask any interactive question.
--env           The environment the command should run under.
```

Available commands:

```
help          Displays help for a command
list          Lists commands
```

```
migrate       Run the database migrations
```

```
serve         Serve the application on the PHP development server
```

```
tinker        Interact with your application
```

```
workbench     Create a new package workbench asset
```

```
asset:publish Publish a package's assets to the public directory
```

```
auth:reminders Create a migration for the password reminders table
```

```
command:make  Create a new Artisan command config
```

```
config:publish Publish a package's configuration to the application controller
```

```
controller:make Create a new resourceful controller
```

```
db:seed       Seed the database with records
```

```
key:generate   Set the application key
```

```
migrate:install Create the migration repository
```

```
migrate:make   Create a new migration file
```

```
migrate:refresh Reset and re-run all migrations
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 6 / 102	

migrate:reset	Rollback all database migrations
migrate:rollback	Rollback the last database migration queue
queue:listen	Listen to a given queue
queue:work	Process the next job on a queue session
session:table	Create a migration for the session database table

Création de la clé du fichier de configuration app.php

```
>php artisan key:generate
```

Tester installation

Avec le navigateur se rendre à l'adresse : <http://127.0.0.1/laravel3/public/>

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 7 / 102	

FICHIER .HTACCESS

Le dossier `/laravel3/public/` contient un fichier `.htaccess` pour utiliser le mode URL REWRITING afin de masquer le mot `index.php` dans les URL. (voir aussi le fichier `/config/application.php`, paramètre : `'index' => ''`,)

Pré-requis :

Avec WAMP en local, le mode URL REWRITING doit être activé, pour cela voir le fichier `httpd.conf` et de dé commenter la ligne :

```
LoadModule rewrite_module modules/mod_rewrite.so
```

Relancer le serveur après la modification.

Contenu du fichier `.htaccess`

```
# Apache configuration file
# http://httpd.apache.org/docs/2.2/mod/quickreference.html

# Note: ".htaccess" files are an overhead for each request. This logic
should
# be placed in your Apache config whenever possible.
# http://httpd.apache.org/docs/2.2/howto/htaccess.html

# Turning on the rewrite engine is necessary for the following rules and
# features. "+FollowSymLinks" must be enabled for this to work
symbolically.

#<IfModule mod_rewrite.c>
#     Options +FollowSymLinks
#     RewriteEngine On
#</IfModule>

# For all files not found in the file system, reroute the request to the
# "index.php" front controller, keeping the query string intact

<IfModule mod_rewrite.c>
    RewriteCond %{REQUEST_FILENAME} !-f
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 8 / 102	Dernière impression le


```
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php/$1 [L]
</IfModule>
```

ATTENTION

Avec WAMP en local les instructions surlignées en jaune ont été commentées et remplacées par

```
Options +FollowSymLinks
RewriteEngine on

RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d

RewriteRule . index.php [L]
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 9 / 102	

GESTION ENVIRONNEMENT

Il est possible de gérer plusieurs environnements avec Laravel. Exemple un environnement de développement et un environnement de production.

<http://laravel.fr/docs/v3/doc/installation#Environnements>

Création d'un dossier : `/application/config/local`

Dans se nouveau dossier créer autant de scripts de configuration que nécessaire en reprenant le même nom que les scripts de Laravel dans son dossier

`/application/config.`

Exemple pour créer une connexion à une base de données locale :

`/application/config/local/database.php`

```
<?php
return array(
    'connections' => array(

        'mysql' => array(
            'driver'    => 'mysql',
            'host'      => '127.0.0.1',
            'database'  => 'mabase',
            'username'  => 'root',
            'password'  => '',
            'charset'   => 'utf8',
            'prefix'    => '',

        ),

    ),
);
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 10 / 102	Dernière impression le

Création Database

Créer une base de données et la déclarer dans le script de configuration de Laravel :

```
\app\config\database.php
```

Générer des tables en lignes de commande

Création de la table de migration

Cette commande permet de créer la table `migrations` dans la base de données. Elle ne doit être effectué qu'une seule fois.

```
>php artisan migrate:install
```

Création d'une table

Pour créer un script qui permettra la description d'une table dans le dossier

`\app\database\migrations` passer en mode commande et taper :

```
>php artisan migrate:make nom_de_la_table_au_pluriel
```

ou

```
>php artisan migrate:make nom_de_la_table_au_singulier --  
table=nom_de_la_table_au_pluriel --create
```

Le seconde syntaxe crée une structure complète de définition de la table dans le dossier des migrations..

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 11 / 102	

Ouvrir le script de description et définir les champs de la table à l'aide de la fonction `up()` (posts dans notre exemple)

```
public function up()
{
    Schema::create('posts', function($table){
        $table->increments('id');
        $table->string('title',100);
        $table->text('content');
        $table->integer('user_id');
        $table->timestamps();
    });
}
```

Dans la fonction `down()`, supprimer la table

```
public function down()
{
    Schema::drop('posts');
}
```

Pour créer la table dans la base de données, passer en mode commande et taper :

```
>php artisan migrate
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 12 / 102	

LES VUES : VIEWS

Elles permettent d'afficher les pages du sites.

Exemple pour afficher les données du fichier des posts :

Dans le dossier `\app\views` création d'un nouveau dossier : `\post` et créer un script `index.php` contenant les instruction suivantes :

```
<h1>Salut!</h1>
```

Pour afficher les données passées à la vue

```
<div class='container'>

    Bonjour {{$prenom}} {{$nom}}

</div>
```

Pour utiliser le moteur de TEMPLATE Blade de Laravel, les noms des scripts doivent être : `nom_du_script.blade.php`

Le script doit aussi respecter la syntaxe des templates Blade.

La première ligne indique le nom du template à utiliser (`template.main` dans notre exemple et correspond au script : `/application/views/template/main.blade.php`).

Une section nommée : `content` contient les données à afficher.

```
@layout('template.main')

@section('content')

Bonjour {{$prenom}} {{$nom}}

@endsection
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 13 / 102	

TEMPLATE

Un template est un modèle d'affichage utilisé tout au long de l'utilisation d'un site ou d'une portion de celui-ci.

Dans le dossier `/app/views/` créer un dossier `/_template`

Dans ce dossier créer un fichier `.php`, exemple : `main.blade.php`

Ce fichier doit contenir la structure HTML complète de la page. (HTML, CSS, JavaScript)

Pour injecter dans cette page les vues utiliser la syntaxe suivante :

```
<div class='container'>

    @yield('content')

</div>
```

L'instruction `@yield('Nom_de_la_Section')` permet d'injecter dans une classe quelconque le contenu de la section nommée : `content` de la vue.

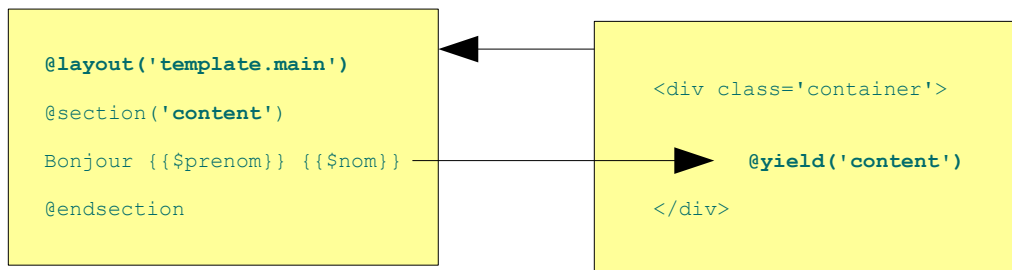
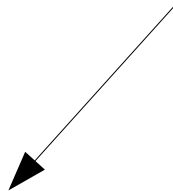
Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 14 / 102	

RELATIONS : CONTROLEUR / VUES / TEMPLATE

Dans cet exemple on définit le template dans le contrôleur.

/application/**controller**/nom_du_script_controller.php

```
<?php  
  
class Post extends BaseController {  
  
    protected $layout = 'template.main'; // Déclaration du Layout  
  
    public function action_index()  
    {  
        $data = array('nom'=>'Dupont', 'prenom'=>'Jean');  
        $this->layout->content = View::make('post.index')->with($data);  
    }  
  
}
```



/application/**views**/script_vue.blade.php

/application/views/**template**/main.blade.php

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 15 / 102	

INTEGRATION BOOTSTRAP TWITTER

Copier les dossiers du bootstrap : /js, /css et /img dans le dossier
/laravel3/public/ et remplacer les dossiers vides existants

Création du dossier : /laravel3/application/views/_template et du fichier :
main.blade.php Ce fichier est le layout principal de l'application.

Contenu du fichier main.blade.php

```
<?php
HTML::macro('deconnecter', function(){
    return '<a href="'.URL::base().'/user/logout.'"><i class="icon-off"></i>
Déconnexion</a></i>';
});?><!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="utf-8">
        <title>@yield('title')</title>
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta name="description" content="">
        <meta name="author" content="FLD">

        <!-- Le style pointe vers /public -->
        {{HTML::style('css/bootstrap.css')}}
        <style>
            body {
                padding-top: 60px; /* 60px to make the container go all the way to the bottom of the
topbar */
            }
        </style>
        {{HTML::style('css/bootstrap-responsive.css')}}

        <!-- HTML5 shim, for IE6-8 support of HTML5 elements -->
        <!--[if lt IE 9]>
            <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
        <![endif]-->

        <!-- Fav and touch icons -->
        <link rel="apple-touch-icon-precomposed" sizes="144x144"
href="http://twitter.github.com/bootstrap/assets/ico/apple-touch-icon-144-precomposed.png">
        <link rel="apple-touch-icon-precomposed" sizes="114x114"
href="http://twitter.github.com/bootstrap/assets/ico/apple-touch-icon-114-precomposed.png">
        <link rel="apple-touch-icon-precomposed" sizes="72x72"
href="http://twitter.github.com/bootstrap/assets/ico/apple-touch-icon-72-precomposed.png">
```

Rédacteur :	Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document :	Laravel3_A_Savoir.odt	
Page	16 / 102	Dernière impression le


```

    <link rel="apple-touch-icon-precomposed"
href="http://twitter.github.com/bootstrap/assets/ico/apple-touch-icon-57-precomposed.png">
    <link rel="shortcut icon"
href="http://twitter.github.com/bootstrap/assets/ico/favicon.png">
</head>

<body>

<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="navbar-inner">
        <div class="container">
            <a class="btn btn-navbar" data-toggle="collapse" data-target=".nav-collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </a>
            <a class="brand" href="/">FLD</a>

            <div class="nav-collapse collapse">

                <ul class="nav">
                    <li class="divider-vertical"></li>
                    <li><a href="#"><i class="icon-home icon-white"></i> Home</a></li>
                    <!--Test si l'utilisateur n'est pas logué-->
                    @if(Auth::guest())

                        <li>{{HTML::link('user/login', 'Login')}}</li>
                        <li>{{HTML::link('user/signup', 'Inscription')}}</li>

                    @else

                        <li>{{HTML::link('post/add', 'Ecrire')}}</li>
                        <li>{{HTML::link('pays', 'Pays')}}</li>

                    @endif

                    <li>{{HTML::link('contact', 'Contact')}}</li>

                </ul>

                @unless(Auth::guest())
                <div class="pull-right">
                    <ul class="nav pull-right">
                        <li class="dropdown"><a href="#" class="dropdown-toggle" data-
toggle="dropdown">Bienvenue, {{ Auth::user()->username }}<b class="caret"></b></a>
                            <ul class="dropdown-menu">
                                <li><a href="/user/preferences"><i class="icon-cog"></i>
Préférences</a></li>
                                <li><a href="/help/support"><i class="icon-envelope"></i>
Contacter Support</a></li>
                                <li class="divider"></li>

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 02/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 17 / 102	

```

        <li>{{HTML::deconnecter()}}</li>
    </ul>
</li>
</ul>
</div>
@endunless

</div>
</div>
</div>
</div>

<div class="container">

    @yield('content')

</div> <!-- /container -->

<!-- Le javascript
===== -->
<!-- Placed at the end of the document so the pages load faster -->
<script src="http://twitter.github.com/bootstrap/assets/js/jquery.js"></script>

{{HTML::script('js/bootstrap.min.js')}}

</body>
</html>

```

Technique des Asset

<http://laravel.fr/docs/v3/doc/vues/assets>

La classe Asset permet de gérer un container de ressources .css et .js contenu dans le dossier /public/

Création des dossiers

Si besoin créer des sous-dossiers et y placer les ressources appropriées.

/js : javascript (jquery)
/css : feuilles de styles (main.css)
/img : images (logo)
/assets : image sprites, icones (foundation zurb icons)
/libs : css frameworks (foundation, jquery ui)

Rédacteur :	Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document :	Laravel3_A_Savoir.odt	Dernière impression le
Page	18 / 102	

Déclaration des assets

Ouvrir le contrôleur de base d'ou sont dérivés tous les autres contrôleurs :

/application/controllers/base.php

Déclarer les Assets de l'application

```
public function __construct(){

//styles
Asset::add('main_style', 'css/main.css');
Asset::add('foundation_style', 'libs/foundation/stylesheets/foundation.min.css');
Asset::add('foundation_icons',
'assets/foundation_icons_general/stylesheets/general_foundicons.css');
Asset::add('jqueryui_style', 'libs/jquery-ui/css/smoothness/jquery-ui-
1.9.1.custom.min.css');
Asset::add('datatables_style', 'libs/datatables/media/css/jquery.dataTables.css');

//scripts
Asset::add('jquery', 'js/jquery.js');
Asset::add('foundation_script', 'libs/foundation/javascripts/foundation.min.js');
Asset::add('jqueryui_script', 'libs/jquery-ui/js/jquery-ui-1.9.1.custom.min.js');
Asset::add('datatables_script', 'libs/datatables/media/js/jquery.dataTables.js');
Asset::add('mustache_script', 'js/mustache.js');
Asset::add('main_script', 'js/main.js');

parent::__construct();

}
```

ou les placer dans le code d'une bibliothèque (voir helpers.php)

Utiliser les assets

Dans les templates remplacer :

```
{{HTML::style('css/bootstrap.css')}} , etc...
```

par :

```
{{Asset::styles()}}
```

Toutes les feuilles de styles déclarées seront appelées à l'aide d'une seule ligne de commande . Pratiquer de même avec le JavaScript `{{Asset::scripts()}}`

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 19 / 102	Dernière impression le

LES ROUTES

Dans le dossier **/application** se trouve le script : **routes.php**

Ouvrir ce script pour modifier la route proposée par défaut à l'installation du framework.

```
<?php

/*
|-----
| Application Routes
|-----
|
| Here is where you can register all of the routes for an application.
| It's a breeze. Simply tell Laravel the URIs it should respond to
| and give it the Closure to execute when that URI is requested.
|
*/

Route::get('/', function()
{
    return View::make('hello');
});
```

La bonne pratique est d'utiliser les contrôleurs pour déclarer les routes.

- Les contrôleurs ne sont pas chargés en mémoire avant d'être utilisés.
- Le script des routes est plus clair à lire
- Les contrôleurs sont dérivé d'une classe : `Base_Controller` se qui rend plus simple les modifications sur tous les contrôleurs qui en hérite de facto.

Exemple :

Le script : `civilites.php` dans le dossier `/application/controllers` contient toutes les méthodes pour travailler sur la table des civilités.

```
Route::controller('civilites');
```

Plutôt que d'écrire une route par contrôleur on peut passer un tableau

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 20 / 102	

```
Route::controller(array(
    'civilites',
    'post',
    'contact'
));
```

Pour renvoyer sur un contrôleur (exemple : post) si on est à la racine du site

```
Route::get('/', function()
{
    return Redirect::to('post');
});
```

Une ressource ne sera pas accessible tant qu'elle n'est pas déclarée dans les routes. Un contrôleur peut ensuite être soit constitué de méthodes précédées du mot `action_` (`action_index` par défaut, `action_creation`, etc ...) , soit ce conformer à la syntaxe RESTFull est dans ce cas les méthodes sont précédées du mot `get_` ou `post_`

Les actions seront alors déclenchées en fonction du verbe HTTP employé dans l'URL.

Interface REST

Verbe HTTP	URL - Path	Action - Méthode
GET	/resource	index
GET	/resource/create	create
POST	/resource	store
GET	/resource/{id}	show
GET	/resource/edit/{id}	edit
PUT/PATCH	/resource/{id}	update
DELETE	/resource/{id}	destroy

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 02/p2
Document : Laravel3_A_Savoir.odt	
Page 21 / 102	Dernière impression le

LES MODELES

Utiliser les modèles pour

- Placer les tests de validités
- Définir des messages personnalisés pour les règles de validité
- Établir les relations entre les tables
- Définir des Setter

Exemple modèle */models/post.php*

```
<?php
class Post extends Eloquent{

    // Relations entres les tables

    public function user(){
        // 1 post appartient à 1 Utilisateur
        return $this->belongs_to('user');
    }

    public function comments(){
        // 1 post peut avoir plusieurs commentaires
        return $this->has_many('comment');
    }

    // Exemple de Setter
    public function getDate(){
        // Renvoie la date de création du post format Français JJ/MM/AAAA
        return date('d/m/Y',strtotime($this->created_at));
    }

}
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 22 / 102	

Exemple modèle /models/civilite.php

```
<?php
class Civilite extends Eloquent{

    // Messages personnalisés : http://laravel.fr/docs/v3/doc/validation#custom-
    error-messages
    private static $messages = array(
        'lib_civilite_fr_required' => 'Le libellé de la civilité Française est
    requis.',
        'lib_civilite_en_required' => 'Le libellé de la civilité Anglaise est
    requis.',
        'lib_civilite_de_required' => 'Le libellé de la civilité Allemande est
    requis.'
    );

    // Règles de validités
    public static $rules = array(
        'lib_civilite_fr' => 'required|min:10|alpha_num',
        'lib_civilite_en' => 'required|min:10|alpha_num',
        'lib_civilite_de' => 'required|min:10|alpha_num'
    );

    /*
    / Test des règles
    / Utiliser : $validation = Civilite::validate(Input::all());
    /             if ( $validation->fails() ) { ... }
    */
    public static function validate($data) {
        return Validator::make($data, static::$rules, static::$messages);
    }
}
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 23 / 102	

CONTROLEURS

Exemple de contrôleur : posts.php

```
<?php
class Post_Controller extends Base_Controller {
    function __construct()
    {
        parent::__construct();
        $this->filter('before','auth')->only(array('comment','add'));
        $this->filter('before','csrf')->on('post');
    }
    /*
    /
    / Index Post : Liste de tous les Posts
    /
    */
    public function action_index()
    {
        $per_page = 3;

        // Sélection de tous les posts
        // $posts = Post::all();
        // Paginer
        // $posts = Post::paginate($per_page);

        // Paginer et trier
        $posts = Post::order_by('id','desc')->paginate($per_page);

        // Injecte une variable dans le template de la vue
        Section::inject('title','Liste des POSTS');

        // Afficher la vue avec les réponses
        return View::make('post.index')->with('posts', $posts);
    }
}
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 24 / 102	


```

/*
/
/ Ajouter Post :
/
*/
public function action_add()
{

    if(Request::method()=='POST'){
        $rules = array(
            'title'          =>'required|min:5|max:50',
            'content'        =>'required'
        );
        $validation = Validator::make(Input::all(),$rules);

        if ($validation->fails()){
            return Redirect::to('post/add')
                ->with_errors($validation)
                ->with_input();
        }
        else{
            $post = new Post(array(
                'title'=>Input::get('title'),
                'content'=>Input::get('content'),
                'user_id'=>Auth::user()->id
            ));
            if($post->save()){
                Session::flash('success','Post Ajouté');
                return Redirect::back();
            }
        }

    }

    Section::inject('title','Ajouter un Post');
    return View::make('post.add');

}

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 25 / 102	

```

/*
/
/ Lire le Post : Retrouve le Post passé dans l'URL
/ Modifie le titre de la vue /post/lire.blade.php
/ Appel la vue /post/lire avec les données de la table POST
/
*/
public function action_lire($slug,$id)
{

    $post = Post::find($id);

    Section::inject('title',$post->title.' | Article de '.$post->user-
>username);

    return View::make('post.lire')->with('post',$post);

}

/*
/
/ Lire les Posts d'un User
/
*/
public function action_user($slug,$id)
{

    $user = User::find($id);

    Section::inject('title','Les Posts de '.$user->username);

    return View::make('post.user')->with('user',$user);

}

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 26 / 102	

```

/*
/
/Enregistrement d'un commentaire sur un Post par un User
/
*/
public function action_comment($id)
{

    $post = Post::find($id);

    if (Request::method()=='POST'){
        $rules = array('comment'=>'required');
        $validation = Validator::make(Input::all(),$rules);
        if($validation->fails()){
            return Redirect::to('post/lire/'.Str::slug($post->title).'/'. $post->id)
                                ->with_errors($validation)
                                ->with_input();
        }
        else{
            $comment = new
Comment(array('content'=>Input::get('comment'),'user_id'=>Auth::user()->id));
            if($post->comments()->insert($comment)){
                Session::flash('success','Commentaire ajouté');
                return Redirect::back();
            }
        }
    }

}

}
}

```

Ce contrôleur est utilisés par les scripts :

```

/application/views/post/add.blade
/application/views/post/index.blade.php
/application/views/post/lire.blade.php
/application/views/post/user.blade.php

```

/application/views/post/index.blade.php

```
@layout('_template.main')
```

```
@section('content')
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 27 / 102	Dernière impression le

```

<h1>Liste des Posts</h1>

@if ($posts)

    @foreach($posts->results as $d)

        <!-- Lien vers site/public/post/lire/Le_Titre_Du_Post/Id_Du_Post
        Exemple : http://www.monsite.fr/public/post/lire/le-titre-du-post/1
        -->
        <h2><a href="{{URL::base()}.'/post/lire/'.Str::slug($d->title).'/'. $d-
>id}}">{{ $d->title}}</a></h2>

        <!--Affiche les 200 premières caractères du contenu du Post-->
        <p>{{Str::limit($d->content,200)}}</p>

        <div>Par <a href="{{URL::base()}.'/post/user/'.Str::slug($d->user-
>username).'/'. $d->user->id}}">{{ $d->user->username}}</a>. Le {{ $d-
>getDate()}}</div>

    @endforeach

    {{ $posts->links() }}

@endif

@endsection

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 28 / 102	

Exemple de contrôleur : *civilites.php* (type **RESTFUL**)

```
<?php

class Civilites_Controller extends Base_Controller
{
    public $restful = true;

    /*
    /
    / Il faut être Authentifié pour utiliser les méthodes : index, creation,
modifier et supprimer
    / les posts sont protégés contre les failles CSRF.
    /
    */
    public function __construct()
    {
        parent::construct();
        $this->filter('before', 'auth')->
only(array('index', 'create', 'edit', 'destroy'));
        $this->filter('before', 'csrf')->on('post');
    }

    /*
    /
    / Index Civilité : Liste des Civilités par Ordre Alphabétique de Libelles
Français
    /
    */
    public function get_index()
    {
        // Stock URL avec numéro de page
        Session::put('url_index', URL::full() );

        $per_page = 10;
        $civilite = Civilite::order_by('lib_civilite_fr', 'asc')->
paginate($per_page);

        Section::inject('title', 'Liste des Civilités');

        return View::make('civilite.index')->with('civilites', $civilite);
    }
}
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 29 / 102	Dernière impression le

```

/*
/
/ Afficher écran de saisie enregistrement : /civilites/create
/
*/
public function get_create()
{
    return View::make('civilite.create');
}
/*
/
/ Enregistrer nouvel enregistrement : submit du formulaire de saisie
/
*/
public function post_create()
{

    $validation = Civilete::validate(Input::all());

    if ( $validation->fails() ) {
        return Redirect::to('civilites/create')
            ->with('msg_pb', 'Vérifier votre saisie')
            ->with_errors( $validation )
            ->with_input();
    } else {
        try
        {
            $civilite = new Civilete;

            $civilite->lib_civilite_fr = e(Input::get('lib_civilite_fr'));
            $civilite->lib_civilite_en = e(Input::get('lib_civilite_en'));
            $civilite->lib_civilite_de = e(Input::get('lib_civilite_de'));

            $civilite->save();

            return Redirect::to(Session::get('url_index'));
        }
        catch( Exception $e )
        {
            return Redirect::to('civilites/create')
                ->with_input();
        }
    }
}
}

```

Rédacteur :	Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document :	Laravel3_A_Savoir.odt	Dernière impression le
Page	30 / 102	

```

/*
/
/ Modifier enregistrement : civilites/edit/#ID
/
*/
public function get_edit($id)
{

    $civilite = Civilite::find($id);

    if (is_null($civilite)) {
        return Redirect::to(Session::get('url_index'));
    }

    return View::make('civilite.update')->with('civilite',$civilite);
}
/*
/
/ Enregistrer modification enregistrement : submit du formulaire de modification
/
*/
public function post_edit($id)
{

    $civilite = Civilite::find($id);

    if (is_null($civilite)) {
        return Redirect::to(Session::get('url_index'));
    }

    $validation = Civilite::validate(Input::all());

    if ( $validation->fails() ) {
        return Redirect::to('civilites/edit/'.$id)
            ->with('msg_pb', 'Vérifier votre saisie')
            ->with_errors( $validation )
            ->with_input();
    } else {
        try
        {
            $civilite->lib_civilite_fr = e(Input::get('lib_civilite_fr'));
            $civilite->lib_civilite_en = e(Input::get('lib_civilite_en'));
            $civilite->lib_civilite_de = e(Input::get('lib_civilite_de'));

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 31 / 102	

```

        $civilite->save();

        return Redirect::to(Session::get('url_index'));
    }
    catch( Exception $e )
    {
        return Redirect::to('civilites/edit'.$id)
            ->with_input();
    }
}

}

/*
/
/ Supprimer un enregistrement : /civilites/destroy/ID
/
*/
public function get_destroy($id)
{

    $civilite = Civilite::find($id);

    if (!is_null($civilite)) {
        $civilite->delete();
    }

    return Redirect::to(Session::get('url_index'));

}

}

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 32 / 102	

Exemple de vue : *index.blade.php*

```
@layout('_template.main')

@section('content')

<h2>Liste des Civilités <a class="pull-right btn btn-primary"
href="civilites/create"><i class="icon-pencil icon-white"></i> Création</a></h2>

@if ($civilites)

    <table class="table table-bordered table-hover">
        <thead>
            <tr>
                <th>Français</th>
                <th>Anglais</th>
                <th>Allemand</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>

            @foreach($civilites->results as $civilite)

                <tr>
                    <td>{{e($civilite->lib_civilite_fr)}}</td>
                    <td>{{e($civilite->lib_civilite_en)}}</td>
                    <td>{{e($civilite->lib_civilite_de)}}</td>
                    <td>
                        <div class="pull-right">
                            {{ HTML::edit_enre($civilite->id,'civilites') }}
                            {{ HTML::modal_sup($civilite->id) }}
                        </div>
                    </td>
                </tr>

            @endforeach

        </tbody>
    </table>

    {{ $civilites->links() }}
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 33 / 102	Dernière impression le

```

        @foreach($civilites->results as $civilite)
            <div id="delete-{{ $civilite->id }}" class="modal hide fade" tabindex="-1"
role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
                <div class="modal-header">
                    <button type="button" class="close" data-dismiss="modal" aria-
hidden="true">x</button>
                    <h3 id="myModalLabel">Supprimer Enregistrement</h3>
                </div>
                <div class="modal-body">
                    <p>Confirmer la suppression de cet enregistrement</p>
                </div>
                <div class="modal-footer">
                    <button class="btn" data-dismiss="modal" aria-
hidden="true">Annuler</button>
                    {{ HTML::link('civilites/destroy/'.$civilite->id , 'Confirmer' ,
array('class' => 'btn btn-primary')) }}
                </div>
            </div>
        @endforeach

@else

    <div class="alert alert-block">
        <h4>Pas de données !!!</h4>
        Pour créer des enregistrements cliquer sur le bouton Création.
    </div>

@endif

@endsection

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 34 / 102	

Exemple de vue : *update.blade.php*

```
@layout('_template.main')

@section('content')

{{ Form::open() }}

<fieldset>

<div id="legend">
    <legend>Modifier Civilité</legend>
</div>

@if ($errors->has())
    <ul class="unstyled alert alert-error">
        {{ $errors->first('lib_civilite_fr') ? '<li>'. $errors->first('lib_civilite_fr'). '</li>' : '' }}
        {{ $errors->first('lib_civilite_en') ? '<li>'. $errors->first('lib_civilite_en'). '</li>' : '' }}
        {{ $errors->first('lib_civilite_en') ? '<li>'. $errors->first('lib_civilite_en'). '</li>' : '' }}
    </ul>
@endif

{{ Form::token() }}

<div class="control-group {{ $errors->has('lib_civilite_fr') ? 'error' : '' }}">
    {{ Form::label('français','Français *',array('class'=>'control-label')) }}
    <div class="controls">
        {{ Form::text('lib_civilite_fr',$civillite->lib_civilite_fr,array('class'=>'input-xlarge')) }}
        {{ $errors->first('lib_civilite_fr','<span class="help-inline">:message</span><br><br>') }}
        <span class="help-block">Nom civilité en français</span>
    </div>
</div>

<div class="control-group {{ $errors->has('lib_civilite_en') ? 'error' : '' }}">
    {{ Form::label('anglais','Anglais *',array('class'=>'control-label')) }}
    <div class="controls">
        {{ Form::text('lib_civilite_en',$civillite->lib_civilite_en,array('class'=>'input-xlarge')) }}
    </div>
</div>
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 35 / 102	

```

        {{ $errors->first('lib_civilite_en','<span class="help-
inline">:message</span><br><br>') }}
        <span class="help-block">Nom civilité en anglais</span>
    </div>
</div>
<div class="control-group {{ $errors->has('lib_civilite_de') ? 'error' : '' }}">
    {{ Form::label('allemand','Allemand *',array('class'=>'control-label')) }}
    <div class="controls">
        {{ Form::text('lib_civilite_de',$civilite-
>lib_civilite_de,array('class'=>'input-xlarge')) }}
        {{ $errors->first('lib_civilite_de','<span class="help-
inline">:message</span><br><br>') }}
        <span class="help-block">Nom civilité en allemand</span>
    </div>
</div>
<div class="control-group">
    <div class="form-actions">
        <a class="btn" href="{{ Session::get('url_index') }}">Annuler</a>
        {{ Form::submit('Valider',array('class'=>'btn btn-primary')) }}
        <a href="#delete-{{ $civilite->id }}" role="button" class="btn btn-inverse
pull-right" data-toggle="modal"><i class='icon-trash icon-white'></i> Supprimer</a>
    </div>
</div>
</fieldset>
{{ Form::close() }}

```

```

<div id="delete-{{ $civilite->id }}" class="modal hide fade" tabindex="-1"
role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
    <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-
hidden="true">×</button>
        <h3 id="myModalLabel">Supprimer Enregistrement</h3>
    </div>
    <div class="modal-body">
        <p>Confirmer la suppression de cet enregistrement</p>
    </div>
    <div class="modal-footer">
        <button class="btn" data-dismiss="modal" aria-hidden="true">Annuler</button>
        {{ HTML::link('civilites/destroy/'.$civilite->id , 'Confirmer' , array('class'
=> 'btn btn-primary')) }}
    </div>
</div>

```

@endsection

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 02/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 36 / 102	

Dans cette exemple ont utilise aussi une boite de dialogue pour confirmer la suppression de l'enregistrement, section en **gris foncé**.

Gestion Affichage des Erreurs

Pour afficher les **erreurs** ont peut choisir soit de les placer en haut du formulaire :

```
@if ($errors->has())
    <ul class="unstyled alert alert-error">
        {{ $errors->first('lib_civilite_fr') ? '<li>'. $errors->first('lib_civilite_fr'). '</li>' : '' }}
        {{ $errors->first('lib_civilite_en') ? '<li>'. $errors->first('lib_civilite_en'). '</li>' : '' }}
        {{ $errors->first('lib_civilite_de') ? '<li>'. $errors->first('lib_civilite_de'). '</li>' : '' }}
    </ul>
@endif
```

Soit en regard du champ en anomalie :

```
<div class="control-group {{ $errors->has('lib_civilite_fr') ? 'error' : '' }}">
    {{ Form::label('français','Français *',array('class'=>'control-label')) }}
    <div class="controls">
        {{ Form::text('lib_civilite_fr',$civilite->lib_civilite_fr,array('class'=>'input-xlarge')) }}
        {{ $errors->first('lib_civilite_fr','<span class="help-inline">:message</span><br><br>') }}
        <span class="help-block">Nom civilité en français</span>
    </div>
</div>
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 37 / 102	

Si on utilise la technique :

```
@if ($errors->has())  
    <ul class="unstyled alert alert-error">  
        @foreach ($errors->all() as $error)  
            <li>{{ $error }}</li>  
        @endforeach  
    </ul>  
@endif
```

On récupère tous les messages d'erreurs de chaque champ, pas seulement le premier.
(exemple : le champs ne doit contenir que des lettres et des chiffres, le champ doit faire 10 caractères minimum, etc,...)

Néanmoins, ceci peut convenir si il n'existe qu'une seule règle de validité par champ.

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 38 / 102	

Templates : errors et success

On peut aussi utiliser une autre technique consistant à charger une variable en session , `msg_pb`, ou `msg_ok` et à la retourner à une vue. Des templates sont alors chargés d'analyser le contenu des variables de session et d'afficher leur contenu.

Création d'une vue : `/views/template/errors.php`

```
<?php
$msg_pb = Session::get('msg_pb'); // passer avec ->with('msg_pb', 'Libellé
du message erreur personnalisé')
?>
@if ( (!empty($msg_pb)) || ($errors->messages) )
    <div class="alert alert-error alert-block">
        <button type="button" class="close" data-dismiss="alert">&times;</button>
        @if(!empty($msg_pb))
            {{ $msg_pb }}
        @endif
        @if($errors->messages)
            <ul>
                @foreach($errors->messages as $e)
                    <li> {{ $e[0] }} </li>
                @endforeach
            </ul>
        @endif
    </div>
@endif
```

Cette vue sera intégrée dans d'autres vues avec l'instruction :

@render('_template.errors')

La validation d'un formulaire pourra être effectué en passant une variable supplémentaire **msg_pb**, pour afficher un message d'erreur personnalisé.

```
if ( $validation->fails() ) {
    return Redirect::to('civilites/edit/'.$id)
        ->with('msg_pb', 'Vérifier votre saisie')
        ->with_errors( $validation )
        ->with_input();
}
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 39 / 102	Dernière impression le

De la même façon on peut créer une vue pour les messages de succès.

Création d'une vue : /views/template/**success.php**

```
<?php
$msg_ok = Session::get('msg_ok');

@if(!empty($msg_ok))
<div class="alert alert-success alert-block">
    <button type="button" class="close" data-dismiss="alert">&times;</button>
    {{ $msg_ok }}
</div>
@endif
```

Cette vue sera intégrée dans d'autres vues avec l'instruction :

```
@render('_template.success')
```

Le retour à 'un formulaire pourra être effectué en passant une variable supplémentaire **msg_ok**, pour afficher un message de succès personnalisé.

```
return Redirect::to(Session::get('url_index'))
    ->with('msg_ok', 'Nouvelle civilité enregistrée');
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 40 / 102	

Exemple : CONTACT

VIEW

```
@layout('_template.main')

@section('content')

{{URL::base()}}
<p>/views/contact/index.blade.php</p>

<h1>Contact</h1>

@if(Session::has('success'))
    <div class="alert alert-success">{{Session::Get('success')}}</div>
@endif

@if(Session::has('message'))
    <div class="{{Session::Get('class')}}">{{Session::Get('message')}}</div>
@endif

{{Form::open('contact', 'POST', array('class'=>'well'))}}

{{Form::token()}}

{{Form::text('nom', Input::old('nom'), array('class'=>'input-xlarge',
'placeholder'=>'Votre nom'))}}
{{ $errors->first('nom', '<span class="help-inline error">:message</span>')}}
<br>

{{Form::text('email', Input::old('email'), array('class'=>'input-xlarge',
'placeholder'=>'Votre e-mail'))}}
{{ $errors->first('email', '<span class="help-inline error">:message</span>')}}
<br>

{{Form::textarea('message', Input::old('message'), array('placeholder'=>'Votre
Post'))}}
{{ $errors->first('message', '<span class="help-inline error">:message</span>')}}
<br>

{{Form::submit('Valider', array('class'=>'btn btn-primary'))}}
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 41 / 102	

```
{{Form::close()}}
```

```
@endsection
```

RESULTAT

http://localhost/laravel3/public
/views/contact/index.blade.php

Contact

Votre nom

Votre e-mail

Votre Post

Valider

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 42 / 102	

CONTROLEUR : CONTACT.PHP

```
<?php
/*
    _ _ _ _ _ | | _ _ _ _ | | _
    / _ / _ \ | ' _ \ | _ / _ ` | / _ | _ |
    | ( _ | ( _ ) | | | | | | ( _ | | ( _ | | _
    \ _ _ \ _ _ / | _ | | _ \ _ \ _ , _ | \ _ _ | \ _ |

*/

class Contact_Controller extends Base_Controller {
    function __construct()
    {
        parent::construct();
        $this->filter('before','csrf')->on('post');
    }

    /*
    /
    / Saisie et contrôle d'un message
    / et envoie un mail
    / http://swiftmailer.org
    /
    // Give the message a subject
    ->setSubject('Your subject')

    // Set the From address with an associative array
    ->setFrom(array('john@doe.com' => 'John Doe'))

    // Set the To addresses with an associative array
    ->setTo(array('receiver@domain.org', 'other@domain.org' => 'A name'))

    // Give it a body
    ->setBody('Here is the message itself')

    // And optionally an alternative body
    ->addPart('<q>Here is the message itself</q>', 'text/html')

    // Optionally add any attachments
    ->attach(Swift_Attachment::fromPath('my-document.pdf'))
    ;

    */
    public function action index()
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2 Dernière impression le
Document : Laravel3_A_Savoir.odt	
Page 43 / 102	

```

{

    if(Request::method()=='POST'){

        $rules = array(

            'nom'            =>'required',

            'email'          =>'required|email',

            'message'        =>'required'

        );

        $validation = Validator::make(Input::all(),$rules);

        if ($validation->fails()){

            return Redirect::to('contact')

                ->with_errors($validation)

                ->with_input();

        }

        else{

            // Envoyer un Mail

            Laravel\IoC::register('mailer.transport', function()

            {

                return

Swift_SmtpTransport::newInstance('smtp.orange.fr',25);

            });

            $mailer = IoC::resolve('mailer');

            $message = Swift_Message::newInstance('Message de

Contact')

->setFrom(array(Input::get('email')=>Input::get('nom')))

                ->setTo(array('florge@msd-informatique.fr'=>'Mr  LORGE

Frédéric'))

                ->addPart(Input::get('message'),'text/plain')

                ->setBody('<h4>Vous avez reçu un message de

'.Input::get('nom').'</h4>

                <p>'.Input::get('message').'</p>','text/html');

            if ($mailer->send($message)){

                Session::flash('message','Message envoyé');

                Session::flash('class','alert alert-success');

            }

            else{

                Session::flash('message','Message non envoyé');

                Session::flash('class','alert alert-error');

            }

        }

    }

}

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 44 / 102	

```
        return Redirect::back();
    }
}

Section::inject('title','Nous Contacter');
return View::make('contact.index');
}
}
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 45 / 102	

LES LIENS

<http://laravel.fr/docs/v3/doc/vues/html>

<http://laravel.com/docs/views/html>

```
{{ HTML::link('civilites/update/'.$civilite->id, '', array('class' => 'icon-edit', 'alt'=>'Modifier', 'title'=>'Modifier')) }}
```

IDEM

```
<a href="{{ URL::base() . '/civilites/update/'.$civilite->id }}" alt="Modifier" title="Modifier"><i class="icon-edit"></i></a>
```

LINK_TO_ACTION

<http://laravel.io/topic/2/reverse-routing-with-controller-actions>

L'intérêt de `link_to_action` est de pouvoir changer le nom de la route sans avoir besoin d'utiliser les routes nommées, ni de reprendre tout le code si on change le nom d'une route.

Exemple :

Route	Adresse
<code>Route::any('register', 'users@register');</code>	http://www.mon.site.com/register
Lien dans le site	<pre>{{ HTML::link('register', 'Register Now!') }}</pre>
Nouvelle Route	Nouvelle Adresse
<code>Route::any('signup', 'users@register');</code>	http://www.mon.site.com/signup
Les liens dans le site ne fonctionnent plus !!!	
Pour les liens dans le site préférer la syntaxe	<pre>{{ HTML::link_to_action('users@register', 'Register Now!') }}</pre>
De la sorte même en modifiant la route de l'URL, les liens fonctionnent toujours dans le site	

MACRO POUR LES LIENS

Pour définir des attributs particuliers sur un lien il faut utiliser la technique de la macro.

Exemple dans une vue : index.blade.php

```
<?php
HTML::macro('deconnecter', function(){
    return '<a href="'.URL::base().'/user/logout'."><i class="icon-off"></i>
Déconnexion</a></i>';
});?>
```

Utiliser la macro

```
{{ HTML::deconnecter() }}
```

Donnera :

```
<a href="http://localhost/laravel3/public/user/logout"><i class="icon-off"></i> Déconnexion</a>
```

Enregistrer les macros

<http://www.youtube.com/watch?v=z-Wpoc-0o2Y>

Utiliser la technique de la conception de bibliothèque de code pour créer des macros de liens et les utiliser partout dans l'application

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 47 / 102	

BIBLIOTHEQUE DE CODE

Créer un script du nom de `helpers.php` dans le dossier `/application` et placer le code des macros, fonctions, constantes à utiliser dans les vues ou les contrôleurs.

Si on souhaite structurer encore plus, créer une bibliothèque par thème, exemple `lienmacro.php`, `bibliotheque.php`, `constante.php`, etc...

Déclarer la bibliothèque

Ouvrir le script : `/application/start.php` , se placer à la fin et rajouter instruction :

```
/*
|-----
| Bibliothèque de code disponible dans les vues, les contrôleurs et les modèles.
|-----
|
*/
require_once('helpers.php');
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 48 / 102	

EXEMPLE BIBLIOTHEQUE : HELPERS.PHP

```
<?php
/*
  _____
 | _ \ ( ) | | ( ) | | | |
 | | ) | | | _ | _ | | | | _ _ _ _ _
 | _ < | | ' \ | | / _ \ | ' \ / _ \ / _ \ | | | | / _ \
 | | ) | | | ) | | | ( ) | | | | | _ / ( | | | | _ /
 | _ _ / | | _ _ / | | | \ _ / \ _ | | | \ _ \ _ , | \ _ , | \ _ |
                                     | |
                                     | |
*/
/*
* Lien déconnexion utilisateur
*/
HTML::macro('deconnecter', function(){
    return '<a href="'.URL::base().'/user/logout.'"><i class="icon-off"></i>
Déconnexion</a></i>';
});

/*
* Lien suppression enregistrement
* @ $id      = Id enregistrement à supprimer
* Appel boîte de dialogue pour confirmation suppression enregistrement
*/
HTML::macro('modal_sup', function($id){
    return "<a data-toggle='modal' href='#delete-{$id}' alt='Supprimer'
title='Supprimer'><i class='icon-trash'></i></a>";
});

/*
* Lien modification enregistrement
* @ $id      = Id enregistrement à modifier
* @ $table   = Nom de la route (en général le nom de la table)
* Exemple    : {{ HTML::edit_enre($civilite->id,'civilites') }} URL :
http://localhost/laravel3/public/civilites/edit/12
*/
HTML::macro('edit_enre', function($id,$table){
    return "<a href='{ $table }/edit/{ $id }' alt='Modifier' title='Modifier'><i
class='icon-edit'></i></a>";
});
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 49 / 102	Dernière impression le

VALIDATIONS REGLES ET MESSAGES

Une bonne pratique consiste à coder le contrôle de validité des données dans les modèles puisque que c'est la que sont représentés les relations entre les tables.

Voir

<http://laravel.fr/docs/v3/doc/validation#custom-error-messages>

<http://www.youtube.com/watch?v=6Qbj7h2Zo00&feature=youtu.be>

<http://jasonlewis.me/article/laravel-using-forms-and-the-validator>

<http://jasonlewis.me/article/laravel-advanced-validation>

<http://jasonlewis.me/code/expressive-date/docs>

Exemple modèle : /application/models/civilite.php

```
<?php
class civilite extends Eloquent
{
    // Les règles
    public static $rules = array(
        'lib_civilite_fr' => 'required|min:10|alpha_num',
        'lib_civilite_en' => 'required',
        'lib_civilite_de' => 'required'
    );

    // Messages personnalisés
    private static $messages = array(
        'lib_civilite_fr_required' => 'Le libellé de la civilité Française
est requis.',
        'lib_civilite_en_required' => 'Le libellé de la civilité Anglaise
est requis.',
        'lib_civilite_de_required' => 'Le libellé de la civilité Allemande
est requis.'
    );
    /*
    / Test des règles
    / Utiliser : $validation = Nom_du_Model::validate(Input::all());
    /             if ( $validation->fails() ) { ... }
    */
    public static function validate($data)
    {
        return Validator::make($data, static::$rules, static::$messages);
    }
}
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 50 / 102	Dernière impression le

```
}
}
```

Pour la règle `required` des messages personnalisés sont utilisés. On les définit dans le tableau associatif : `$message`, la clé est constituée du nom du champ suivit de `_` et le nom du test .

Exemple : `lib_civilite_fr_required`

La valeur est définie avec le message à afficher.

Exemple d'utilisation dans le contrôleur : `/application/controllers/civilite.php`

```
/*
 /
 / Enregistrer nouvel enregistrement : submit du formulaire de saisie
 /
 */
public function post_create()
{

    $validation = Civilite::validate(Input::all());

    if ( $validation->fails() ) {
        return Redirect::to('civilites/create')
            ->with_errors( $validation )
            ->with_input();
    } else {
        $civilite = new Civilite;

        $civilite->lib_civilite_fr = e(Input::get('lib_civilite_fr'));
        $civilite->lib_civilite_en = e(Input::get('lib_civilite_en'));
        $civilite->lib_civilite_de = e(Input::get('lib_civilite_de'));

        $civilite->save();

        return Redirect::to(Session::get('url_index'));
    }

}
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 51 / 102	

Il peut aussi être intéressant de définir les libellés des contrôles pour les rendre disponibles dans toute l'application. Voir le chapitre : Multi-Langues.

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 52 / 102	

MULTI-LANGUES

<http://laravel.com/docs/localization>

Si on doit gérer un formulaire en plusieurs langues il faut utiliser une autre méthode. Dans cet exemple nous définissons à la fois une règle personnalisée et un affichage multi-langues des contrôles de validités.

Définir la règle

Ouvrir le fichier `/application/start.php` et enregistrer la règle : `commence_par` dans notre exemple. `commence_par` est le nom de la règle.

```
// File: application/start.php

Validator::register('commence_par', function($attribute, $value,
$parameters)
{
    return starts_with($value, $parameters[0]); // Retourne true ou false
});
```

Définir le libellé du contrôle de validité de la règle

Les Libellés des messages sont contenus dans les fichiers de configuration des langues de Laravel, le dossier : `/application/language`

Ce dossier contient une collection de sous dossiers représentant tous une langue. Chacun de ces sous dossiers contient deux scripts, un pour les libellés des paginations et un autre pour les textes des messages d'erreurs.

Il s'agit du fichier `/application/language/fr/validation.php`

Il contient un tableau associatif vide par défaut : `custom`

La clé doit être constituée par le nom du champ et de la règle, exemple :

`'siteinternet_commence_par'`

La valeur contient le libellé du message, exemple : `'Doit commencer par http://'`

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 53 / 102	Dernière impression le

Exemple : `nom du champ_nom de la règle`

Français

```
// File: application/language/fr/validation.php

'custom' => array(
    'siteinternet_commence_par' => 'Doit commencer par http://',
    'nationalite' => 'Vous devez confirmer votre nationalité'
)
```

Anglais

```
// File: application/language/en/validation.php

'custom' => array(
    'siteinternet_commence_par' => 'Must start with http://',
    'nationalite' => 'You must confirm your nationality'
)
```

Utiliser la règle

Dans le modèle, utiliser la règle comme les autres règles par défaut.

```
// Les règles
public static $rules = array(
    'url_site_internet' => 'required|commence_par:http://',
    'lib_civilite_fr' => 'required|min:10|alpha_num',
    'lib_civilite_de' => 'required'
);
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 54 / 102	

PAGINATION ET PARAMETRES

Afficher des données celons différents ordre de tri :

Faire passer le paramètre de tri dans l'URL, ?s=0 ou &s=0 exemple :

<http://localhost/laravel3/public/civilites?s=0>

Récupérer la valeur du paramètre dans le contrôleur.

```
// Récupère paramètres passé dans l'URL :  
// Ordre de Tri (0,1,2 = N° du champ dans le tableau - 1) :  
http://www.mon.site.fr?s=1 ou null si aucun  
$s = Input::get('s');  
if ( ($s==null) || ($s>2) ) {  
    $s=0;  
}  
$lib_tri = Array();  
$lib_tri[0]='lib_civilite_fr';  
$lib_tri[1]='lib_civilite_en';  
$lib_tri[2]='lib_civilite_de';  
  
// Enregistre ordre de tri dans tableau de session  
Session::put( 'url_tri', $s );  
  
$per_page = 10;  
$civilite = Civilite::order_by( $lib_tri[$s], 'asc' )->paginate($per_page);
```

Récupérer la valeur du paramètre dans la vue.

```
{{ $civilites->appends(array('s'=> Session::get('url_tri'))->links() ) }}
```

Générer des liens type : <http://localhost/laravel3/public/civilites?page=2&s=0>

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 55 / 102	

DERNIERE VERSION DU CONTROLEUR

Intégration Ordre de Tri et Recherche (Commence Par)

```
<?php

class Civilites_Controller extends Base_Controller
{
    public $restful = true;

    /*
    /
    / Il faut être Authentifié pour utiliser les méthodes : index,
creation, modifier et supprimer
    / les posts sont protégés contre les failles CSRF.
    /
    */
    public function __construct()
    {

        parent::__construct();

        $this->filter('before','auth')->
only(array('index','create','edit','destroy'));
        $this->filter('before','csrf')->on('post');

    }

    /*
    /
    / Analyse des Paramètres de l'URL pour Ordre de tri et Recherche ( ?
s=0?q=xxx )
    / @ Retourne un Tableau Associatif 's' => Nom du champ du champ à trier
/ 'q' => Contenu à rechercher
    /
    */
    private function analyse_url()
    {

        // Stock URL avec numéro de page dans tableau de session
['url_index'] = 'http://www.mon.site.fr/?s=1?q=libelle_recherche'
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 56 / 102	Dernière impression le


```

        Session::put( 'url_index',  URL::full() );

        // Ordre de Tri ( 0,1,... = N° du champ dans le tableau - 1 ) :
http://www.mon.site.fr?s=0 ou null si aucun
        $s = Input::get('s');
        if ( ($s==null) || ($s>2) ) {
            $s=0;
        }
        $lib_tri = Array();
        $lib_tri[0]='lib_civilite_fr';
        $lib_tri[1]='lib_civilite_en';
        $lib_tri[2]='lib_civilite_de';

        // Stock Libellé du champ pour Tri ['url_tri'] =
'Libelle_Du_Champ_Dans_la_Base_De_Donnée'
        Session::put('url_tri', $s );

        // Stock Libellé du champ pour Tri ['url_req'] =
'contenu_a_rechercher_dans_un_champ'
        $q = Input::get('q');
        Session::put('url_req', $q );

        // Retour analyse de l'URL
        $analyse_url = Array();
        $analyse_url['s']= $lib_tri[$s];
        $analyse_url['q']= $q;

        return $analyse_url;
    }

    /*
    /
    / Aquisition Paramètres Tri et Recherche via formulaire 'param' dans
index
    /
    */
    public function get_param()
    {
        return $this->get_index();
    }

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 57 / 102	

```

/*
/
/ Index Civilité : Liste des Civilités par Ordre Alphabétique de
Libelles Français
/
*/
public function get_index()
{
    $analyse_url = Array();
    $analyse_url = $this->analyse_url();

    $per_page    = 10;

    if ($analyse_url['q']==null) {
        $civillite    = Civillite::order_by($analyse_url['s'],'asc')->
>paginate($per_page);
    }
    else {
        $civillite    = Civillite::where('lib_civillite_fr', 'LIKE' ,
$analyse_url['q'].'%' )->order_by($analyse_url['s'],'asc')->
>paginate($per_page);
    }

    return View::make('civillite.index')
        ->with('titre','Liste des Civilités')
        ->with('civilites', $civillite);
}

/*
/
/ Afficher écran de saisie enregistrement : /civilites/create
/
*/
public function get_create()
{
    return View::make('civillite.create')->with('titre','Création
Civilité');
}

/*

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 58 / 102	Dernière impression le

```

/
/ Enregistrer nouvel enregistrement : submit du formulaire de saisie
/
*/
public function post_create()
{

    $validation = Civilite::validate(Input::all());

    if ( $validation->fails() ) {
        return Redirect::to('civilites/create')
            ->with('msg_pb', 'Vérifier votre saisie')
            ->with_errors( $validation )
            ->with_input();
    }
    else {
        try
        {
            $civilite = new Civilite;

            $civilite->lib_civilite_fr =
e(Input::get('lib_civilite_fr'));
            $civilite->lib_civilite_en =
e(Input::get('lib_civilite_en'));
            $civilite->lib_civilite_de =
e(Input::get('lib_civilite_de'));

            $civilite->save();

            return Redirect::to(Session::get('url_index'))
                ->with('msg_ok', 'Nouvelle civilité enregistrée');
        }
        catch( Exception $e )
        {
            return Redirect::to('civilites/create')
                ->with_input();
        }
    }

}

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 59 / 102	

```

/*
/
/ Modifier enregistrement : civilites/edit/#ID
/
*/
public function get_edit($id)
{

    $civilite = Civilite::find($id);

    if (is_null($civilite)) {
        return Redirect::to(Session::get('url_index'));
    }

    return View::make('civilite.update')
        ->with('titre','Modification Civilité')
        ->with('civilite',$civilite);
}
/*
/
/ Enregistrer modification enregistrement : submit du formulaire de
modification
/
*/
public function post_edit($id)
{

    $civilite = Civilite::find($id);

    if (is_null($civilite)) {
        return Redirect::to(Session::get('url_index'));
    }

    $validation = Civilite::validate(Input::all());

    if ( $validation->fails() ) {
        return Redirect::to('civilites/edit/'.$id)
            ->with('msg_pb', 'Vérifier votre saisie')
            ->with_errors( $validation )
            ->with_input();
    }
}

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 60 / 102	

```

    }
    else {
        try
        {
            $civilite->lib_civilite_fr =
e(Input::get('lib_civilite_fr'));
            $civilite->lib_civilite_en =
e(Input::get('lib_civilite_en'));
            $civilite->lib_civilite_de =
e(Input::get('lib_civilite_de'));

            $civilite->save();

            return Redirect::to(Session::get('url_index'));
        }
        catch( Exception $e )
        {
            return Redirect::to('civilites/edit'.$id)
                ->with_input();
        }
    }
}

/*
/
/ Supprimer un enregistrement : /civilites/destroy/ID
/
*/
public function get_destroy($id)
{
    $civilite = Civilite::find($id);

    if (!is_null($civilite)) {
        $civilite->delete();
    }

    return Redirect::to(Session::get('url_index'));
}

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 61 / 102	

}

Nouvelle civilité enregistrée

Liste des Civilités

Création



















Ordre de Tri Civilité Allemande



Recherche

Commence par...

Ok

Format	Français	Anglais	Allemand	Action
	Donneur d'ordre PHARMA	Pharma contractor	Pharma Auftraggeber	 
	Fournisseur Potentiel	Potential Supplier	Potentielle Lieferant	 
	qsdqsdqsdqsdqsdqsd	qsdqsd	qsdqsdqsd	 
	Rotomouleur	Rotamoulding company	Rotationformes Industrie.	 
	Donneur d'ordre ECRITURE	Writing Instrument contractor	Schreibung Auftraggeber	 
	sdfsdfsdfsdfsdf	sdfsdfsdfsdfsdf	sdfsdfsdfsdfsdf	 

« Précédent

1

2

Suivant »

ICONES OPA

Les icônes OPA de l'interface d'administration *Charisma* ont été intégrées.

Recopie du fichier : **opa-icons.css** dans le dossier **/public/css/**

Recopie des fichiers : **opa*.png** dans le dossier **/public/img/**

Déclaration de la feuille de style opa-icons.css dans les Assets :

```
Asset::add('css_opa' , 'css/opa-icons.css');
```

Exemple utilisation 1 :

```
<span class="icon32 icon-color icon-pdf"></span>
```

Exemple utilisation 2 :

```
<td><span class="icon icon-darkgray icon-mail-closed"></span><span class="icon icon-color icon-pdf"></span></td>
```

Format



Nouvelle civilité enregistrée

Liste des Civilités

Création



















Ordre de Tri

Civilité Allemande

Recherche

Commence par...

Ok

Format	Français	Anglais	Allemand	Action
	Donneur d'ordre PHARMA	Pharma contractor	Pharma Auftraggeber	 
	Fournisseur Potentiel	Potential Supplier	Potentielle Lieferant	 
	qsdqsdqsdqsdqsdqsd	qsdqsd	qsdqsdqsd	 
	Rotomouleur	Rotamoulding company	Rotationformes Industrie.	 
	Donneur d'ordre ECRITURE	Writing Instrument contractor	Schreibung Auftraggeber	 
	sdfsdfsdfsdf	sdfsdfsdfsdf	sdfsdfsdfsdf	 

DERNIERE VERSION DE LA VUE INDEX

```
@layout('_template.main')

@section('content')

@render('_template.success')

<div class="row-fluid">
    <div class="span10"><h2>{{ $titre }}</h2></div>
    <div class="span2"><a class="pull-right btn btn-warning btn-block"
href="create"><i class="icon-pencil icon-white"></i> Création</a></div>
</div>

@if ($civilites)

    {{ Form::open('civilites/param','GET',array('class'=>'form-
inline')) }}

    <fieldset>

        <div id="legend">
            <legend></legend>
        </div>

        {{ Form::label('s','Ordre de Tri') }}
        {{ Form::select('s', array('0' => 'Civilité Française', '1' =>
'Civilité Anglaise', '2' => 'Civilité Allemande'), Session::get('url_tri'))
}}

        {{ Form::label('q','Recherche') }}
        {{ Form::text('q',Session::get('url_req'),array('class'=>'input-
xlarge','placeholder'=>'Commence par...')) }}

        {{ Form::Submit('Ok',array('class'=>'btn btn-primary pull-right')) }}

    </fieldset>

{{ Form::close() }}
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 64 / 102	


```

<table class="table table-bordered table-hover">
  <thead>
    <tr>
      <th>Format</th>
      <th>Français</th>
      <th>Anglais</th>
      <th>Allemand</th>
      <th>Action</th>
    </tr>
  </thead>
  <tbody>

@foreach($civilites->results as $civilite)

    <tr>
      <td><span class="icon icon-darkgray icon-mail-closed"></span><span class="icon icon-color icon-pdf"></span></td>
      <td>{{e($civilite->lib_civilite_fr)}}</td>
      <td>{{e($civilite->lib_civilite_en)}}</td>
      <td>{{e($civilite->lib_civilite_de)}}</td>
      <td>
        <div class="pull-right">
          {{ HTML::edit_enre($civilite->id) }}
          {{ HTML::modal_sup($civilite->id) }}
        </div>
      </td>
    </tr>

  @endforeach

</tbody>
</table>

{{ $civilites->appends(array('s'=> Session::get('url_tri') ))->links() }}

@foreach($civilites->results as $civilite)
  <div id="delete-{{ $civilite->id }}" class="modal hide fade"
tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-
hidden="true">

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 02/p2
Document : Laravel3_A_Savoir.odt	
Page 65 / 102	Dernière impression le

```

        <div class="modal-header">
            <button type="button" class="close" data-dismiss="modal" aria-
hidden="true">×</button>
            <h3 id="myModalLabel">Supprimer Enregistrement</h3>
        </div>
        <div class="modal-body">
            <p>Confirmer la suppression de cet enregistrement</p>
        </div>
        <div class="modal-footer">
            <button class="btn" data-dismiss="modal" aria-
hidden="true">Annuler</button>
            {{ HTML::link('civilites/destroy/'.$civilite->id ,
'Confirmer' , array('class' => 'btn btn-primary')) }}
        </div>
    </div>
@endforeach

@else

    <div class="alert alert-block">
        <h4>Pas de données !!!</h4>
        Pour créer des enregistrements cliquer sur le bouton Création.
    </div>

@endif

@endsection

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 66 / 102	

CONTROLEUR MINI:Exemple Produit

ATTENTION : Problème avec les input::old

```
<?php

class Produit_Controller extends Base_Controller
{
    public $restful = true;

    /*
    /
    / Il faut être Authentifié pour utiliser les méthodes : index,
creation, modifier et supprimer
    /
    */
    public function __construct()
    {

        parent::__construct();

        $this->filter('before','auth')->only(array('index','edit','destroy'));

    }

    /*
    /
    / Index Produit : Liste des Produits : /produit
    /
    */
    public function get_index()
    {

        // Stock URL en cours avec numéro de page et tous les paramètres (?
page=xxx)
        Session::put('url_index', URL::full() );

        $per_page = 10;
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 67 / 102	

```

        $produits = Produit::order_by('Reference','asc')->paginate($per_page);

        return View::make('produit.index')
            ->with('titre','Liste des Produits')
            ->with('produits', $produits);
    }

    /*
    /
    / Création enregistrement : /produit/edit
    / Modifier enregistrement : /produit/edit/#ID
    /
    */
    public function get_edit($id=null)
    {

        if ($id!=null) {

            $produit = Produit::find($id);

            if (is_null($produit)) {
                return Redirect::to(Session::get('url_index'));
            }

            $titre='Modification Produit';
        }
        else
        {

            $produit = new Produit;

            $titre='Création Produit';
        }

        return View::make('produit.edit')
            ->with('titre',$titre)
            ->with('produit',$produit);
    }

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 68 / 102	

```

/*
/ Soumission du Formulaire
/ Enregistrement Création enregistrement      : /produit/edit
/ Enregistrement Modification enregistrement  : /produit/edit/#ID
/
*/
public function post_edit($id=null)
{

    if ($id!=null) {

        $produit = Produit::find($id);

        if (is_null($produit)) {
            return Redirect::to(Session::get('url_index'));
        }
    }

    $validation = Produit::validate(Input::all());

    if ( $validation->fails() ) {

        return Redirect::back()
            ->with('msg_pb', '<strong>Attention</strong> Vérifier
votre saisie')
            ->with_errors( $validation )
            ->with_input();
    }

    try
    {

        if ($id==null) {
            $produit = new Produit;
        }

        $produit->reference = e(Input::get('reference'));
        $produit->modele     = e(Input::get('modele'));

        $produit->save();
    }
}

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 69 / 102	Dernière impression le

```

    }
    catch( Exception $e )
    {
        return Redirect::back()
            ->with('msg_pb', 'Désolé, cet enregistrement n\'est pas
conforme (doublon,...).')
            ->with_input();
    }

    return Redirect::to(Session::get('url_index'))
        ->with('msg_ok', 'Enregistrement terminé');

}

/*
/
/ Supprimer un enregistrement : /produit/destroy/#ID
/
*/
public function get_destroy($id=null)
{

    if ($id!=null) {

        $produit = Produit::find($id);

        if (!is_null($produit)) {
            try
            {
                $produit->delete();
            }
            catch( Exception $e )
            {
                return Redirect::to('produit/edit'.$id)
                    ->with('msg_pb', 'Désolé, un problème technique
nous empêche de supprimer cet enregistrement.')
                    ->with_input();
            }
        }
    }
}

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 70 / 102	

```
}

return Redirect::to(Session::get('url_index'));

}

}
```

Rédacteur :	Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document :	Laravel3_A_Savoir.odt	Dernière impression le
Page	71 / 102	

DEBUGAGE

Utiliser le helper `dd($var_a_tester)` pour faire un `var_dump` et `die()`

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 72 / 102	

BONNES PRATIQUES

- Préférer les interfaces RESTful
- Utiliser les contrôleurs dans les routes
- Définir les contrôles de validités dans les modèles
- Utiliser 1 script pour la Création et un 1 pour la Gestion des données.
(create.blade.php et edit.blade.php)

Posts trouvés dans le forum **Best Practice** de Laravel

Gestion des vues

<http://forums.laravel.io/viewtopic.php?id=839>

<http://forums.laravel.io/viewtopic.php?id=5673>

Idem , si aucun paramètres à passer à la vue

<http://jasonlewis.me/article/laravel-controller-and-route-layouts>

Le modèle de Vue Principale

Entête et pied de pages sont définies dans un modèle principal nommé par exemple :

default.blade.php crée dans le dossier : /views/_template/

Exemple de template :

```
<html>

  <head>

    <title>{{ $title }}</title>

  </head>

  <body>

    {{ $content }}

  </body>

</html>
```

Les variables `$title` et `$content` sont gérées au niveau du contrôleur.

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 73 / 102	

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 74 / 102	

Le Contrôleur

Gère les valeurs des variables ; `$title` et `$content`

Cette technique permet aussi de gérer les sites multilingues.

Le layout est défini par défaut.

Les variables du layout sont assignées avec la technique :

```
$this->layout->title = "...";
```

La variable `$content` est chargée avec une vue à laquelle on passe des paramètres.

Exemple :

```
<?php
class Produit_Controller extends Controller
{

    public $layout = 'template.default';

    function __construct(){

        // Important si on utilise la méthode __construct
        parent::__construct();

        // autres définition (filtres, identification utilisateur,...)
    }

    public function get_index()
    {
        $per_page = 10;
        $produits = Produit::order_by('Reference','asc')->paginate($per_page);

        $this->layout->title = 'le titre dans la balise <title>';
        // Or I'll use languages for i18n.
        // $this->layout->title = __('whatever.page_title');

        $this->layout->content = View::make('test.test')
            ->with( 'titre', 'le titre vue dans la page')
            ->with( 'produits', $produits ) ;
    }
}
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 75 / 102	

```
}
```

La Vue à Afficher

La vue appelée par le contrôleur avec l'instruction :

```
$this->layout->content = View::make('test.test')
    ->with( 'titre', 'le titre vue dans la page')
    ->with( 'produits', $produits ) ;
```

```
<div class="row-fluid">
    <div class="span10"><h2>{{isset($titre) ? $titre : 'PAS DE TITRE' }}</h2></div>
    <div class="span2"><a class="pull-right btn btn-warning btn-block"
href="produit/edit"><i class="icon-pencil icon-white"></i> Création</a></div>
</div>
@if ($produits)
    <table class="table table-bordered table-hover">
        <thead>
            <tr>
                <th style="width:40%">Référence</th>
                <th style="width:40%">Modèle</th>
                <th style="width:20%">Action</th>
            </tr>
        </thead>
        <tbody>
            @foreach($produits->results as $produit)
                <tr>
                    <td>{{e($produit->reference)}}</td>
                    <td>{{e($produit->modele)}}</td>
                    <td>
                        <div class="pull-right">
                            {{ HTML::edit_enre($produit->id,'produit') }}
                        </div>
                    </td>
                </tr>
            @endforeach
        </tbody>
    </table>
    {{ $produits->links() }}
@else
    <div class="alert alert-block">
        <h4>Pas de données !!!</h4>
        <p>Pour créer des enregistrements cliquer sur le bouton Création.</p>
    </div>
@endif
```

Rédacteur :	Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document :	Laravel3_A_Savoir.odt	
Page	76 / 102	Dernière impression le

Vue : Formulaire Gestion CRUD

```
{{ Form::open() }}

<fieldset>

<div id="legend">
    <legend>{{ $titre }}</legend>
</div>

<div class="control-group">
    {{ Form::label('reference','Référence *',array('class'=>'control-label')) }}
    <div class="controls">
        {{ Form::text('reference',$produit->reference,array('class'=>'input-xlarge')) }}
    </div>
</div>

<div class="control-group">
    {{ Form::label('modele','Modèle *',array('class'=>'control-label')) }}
    <div class="controls">
        {{ Form::text('modele',$produit->modele,array('class'=>'input-xlarge')) }}
    </div>
</div>

<div class="control-group">
    <div class="form-actions">
        <a class="btn" href="{{ Session::get('url_index') }}">Annuler</a>
        {{ Form::submit('Valider',array('class'=>'btn btn-primary')) }}
        <a href="#delete-{{ $produit->id }}" role="button" class="btn btn-inverse pull-right" data-toggle="modal"><i class='icon-trash icon-white'></i> Supprimer</a>
    </div>
</div>

</fieldset>
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 77 / 102	Dernière impression le

```

{{ Form::close() }}

<div id="delete-{{ $produit->id }}" class="modal hide fade" tabindex="-1"
role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
  <div class="modal-header">
    <button type="button" class="close" data-dismiss="modal" aria-
hidden="true">×</button>
    <h3 id="myModalLabel">Supprimer Enregistrement</h3>
  </div>
  <div class="modal-body">
    <p>Confirmer la suppression de cet enregistrement</p>
  </div>
  <div class="modal-footer">
    <button class="btn" data-dismiss="modal" aria-
hidden="true">Annuler</button>
    {{ HTML::link('test/destroy/'. $produit->id , 'Confirmer' ,
array('class' => 'btn btn-primary')) }}
  </div>
</div>

```

Vue : Formulaire Création CRUD

```

{{ Form::open() }}

<fieldset>

<div id="legend">
  <legend>{{ $titre }}</legend>
</div>

<div class="control-group">
  {{ Form::label('reference','Référence *',array('class'=>'control-
label')) }}
  <div class="controls">

{{ Form::text('reference',Input::old('reference'),array('class'=>'input-
xlarge')) }}

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 78 / 102	

```

        </div>
    </div>
    <div class="control-group">
        {{ Form::label('modele', 'Modèle *', array('class'=>'control-label')) }}
        <div class="controls">
            {{ Form::text('modele', Input::old('modele'), array('class'=>'input-
xlarge')) }}
        </div>
    </div>

    <div class="control-group">
        <div class="form-actions">
            <a class="btn" href="{{ Session::get('url_index') }}">Annuler</a>
            {{ Form::submit('Valider', array('class'=>'btn btn-primary')) }}
        </div>
    </div>

</fieldset>

{{ Form::close() }}

```

Autre exemple

```

public function action_author( $user_id )
{

    $user = User::find( $user_id );

    if( is_null( $user ) ) return Response::error( '404' );

    $posts = Post::published_by_author( $user_id )->paginate();

    $this->layout->title = "Blog | Viewing posts by $user->name";
    $this->layout->content = View::make( 'blog::author' )->
    with( 'user', $user )->with( 'posts', $posts );

}

```

Rédacteur :	Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document :	Laravel3_A_Savoir.odt	
Page	79 / 102	Dernière impression le

LE CONTROLEUR COMPLET

```
<?php
class Test_Controller extends Controller
{

    public $restful = true;

    public $layout = '_template.default';

    function __construct(){

        parent::__construct();

        $this->filter( 'before', 'auth' )->only( array ( 'index', 'edit',
'create', 'destroy' ));
    }

    /*
    /
    / Index Produit : Liste des Produits : /produit
    /
    */
    public function get_index()
    {

        // Stock URL en cours avec numéro de page et tous les paramètres (?
page=xxx)
        Session::put( 'url_index',  URL::full() );

        $per_page = 10;
        $produits = Produit::order_by( 'Reference', 'asc' )->
paginate( $per_page );

        // Or I'll use languages for i18n.
        // $this->layout->title = __('whatever.page_title');

        $this->layout->title      = 'Produit | Liste';
        $this->layout->content    = View::make( 'test.index' )
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 80 / 102	


```

        ->with( 'titre', 'Liste des Produits' )
        ->with( 'produits', $produits );

    }

    /*
    /
    / Création enregistrement : /produit/edit
    / Modifier enregistrement : /produit/edit/#ID
    /
    */
    public function get_edit( $id=null )
    {

        if ( $id!=null ) {

            if( ! is_numeric($id) OR ( $id<0 ) OR ! $produit =
Produit::find($id) ) {
                return Redirect::to( Session::get( 'url_index' ) ); //
return Response::error( '404' );
            }

            $this->layout->title      = 'Produit | Modification';
            $this->layout->content    = View::make( 'test.edit' )
                ->with( 'titre', 'Modification Produit' )
                ->with( 'produit', $produit );
        }
        else
        {

            $produit = new Produit;

            $this->layout->title      = 'Produit | Création';
            $this->layout->content    = View::make( 'test.create' )
                ->with( 'titre', 'Création Produit' );

        }

    }

}

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 81 / 102	

```

/*
/ Soumission du Formulaire
/ Enregistrement Création enregistrement      : /produit/edit
/ Enregistrement Modification enregistrement  : /produit/edit/#ID
/
*/
public function post_edit( $id=null )
{

    if ( $id!=null ) {

        $produit = Produit::find( $id );

        if ( is_null( $produit ) ) {
            return Redirect::to( Session::get( 'url_index' ) );
        }
    }

    $validation = Produit::validate( Input::all() );

    if ( $validation->fails() ) {

        return Redirect::back()
            ->with('msg_pb', '<strong>Vérifier votre
saisie</strong>')
            ->with_errors( $validation )
            ->with_input();
    }

    try
    {

        if ( $id==null ) {
            $produit = new Produit;
        }

        $produit->reference = e( Input::get('reference') );
        $produit->modele     = e( Input::get('modele') );

        $produit->save();
    }
}

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 82 / 102	

```

    }
    catch( Exception $e )
    {
        return Redirect::back()
            ->with('msg_pb', '<strong>ATTENTION</strong>, cet
enregistrement n\'est pas conforme (doublon,...).')
            ->with_input();
    }

    return Redirect::to( Session::get( 'url_index' ) )
        ->with('msg_ok', 'Enregistrement terminé');

}

/*
/
/ Supprimer un enregistrement : /produit/destroy/#ID
/
*/
public function get_destroy( $id=null )
{

    if( is_numeric( $id ) && ( $id>0 ) && $id != null ) {

        $produit = Produit::find( $id );

        if (!is_null( $produit )) {
            try
            {
                $produit->delete();
            }
            catch( Exception $e )
            {
                return Redirect::to( 'produit/edit'.$id )
                    ->with('msg_pb', 'Désolé, un problème technique
nous empêche de supprimer cet enregistrement.')
                    ->with_input();
            }
        }
    }
}

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 83 / 102	

```
    }

    return Redirect::to( Session::get( 'url_index' ) );

}

}
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 84 / 102	

UPLOAD DE FICHIERS

En local avec WAMP vérifier que l'extension PHP **fileinfo** est montée. (Voir document Install_WAMP_2.2)

Concepts utilisés

- Les règles de validités sont dans le modèle
- Une méthode : `ctrl_val($input = null)` est utilisée pour tester la validité des données
- Les constantes sont définies dans le script :
`\application\config\constante.php`
- Une méthode `upl_img($NomChamp)` est chargé du upload et redimensionnement de l'image
- Le bundle : RESIZER n'est chargé que à la demande

URL Tutoriel :

<http://forums.laravel.io/viewtopic.php?id=1199&p=1>

<http://www.youtube.com/watch?v=6Qbj7h2Zo00&feature=youtu.be>

<http://jasonlewis.me/article/laravel-using-forms-and-the-validator>

<http://laravel.fr/docs/v3/doc/validation#custom-error-messages>

Installation du Bundle : RESIZER

URL : <http://bundles.laravel.com/bundle/resizer>

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 85 / 102	Dernière impression le

MODELE PRODUIT

```
<?php

class produit extends Eloquent
{
    // Règles de validités : http://www.youtube.com/watch?v=6Qbj7h2Zo00&feature=youtu.be
    // http://jasonlewis.me/article/laravel-using-forms-and-the-validator
    protected static $rules = array(
        'reference'          => 'required',
        'modele'             => 'required',
        'image'              => 'image|mimes:jpg,jpeg,jpe,gif,png'
    );
    // Messages personnalisés :
    http://laravel.fr/docs/v3/doc/validation#custom-error-messages
    protected static $messages = array(
        'reference_required' => 'La Référence du produit est obligatoire.',
        'modele_required'   => 'Le nom du modèle est requis.',
        'image_image'       => 'Le champ Image doit être une image.',
        'image_mimes'       => 'Le champ Image doit être une image de
type .jpg, .jpeg, .jpe, .gif ou .png'
    );

    public static $validation = false;

    public static $nom;          // Nom Aléatoire et Extension du
fichier après Upload
    private static $image;      // Acquisition fichier dans le input
    private static $rep;        // Dossier Cible pour déposer le Upload
    private static $ext;        // Extention du fichier sélectionné
    private static $upl;        // Résultat du Upload
    private static $rep_thumb;  // Dossier pour stocker les images
redimensionnées
    private static $width_thumb; // Largeur image miniature
    private static $height_thumb; // Hauteur image miniature
    private static $qual_thumb;  // Qualité image miniature
    private static $mode_thumb;  // Mode de redimensionnement
    private static $success;     // Résultat du Redimensionnement
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 86 / 102	

```

/*
/ Contrôle de validité
/ Appellé via  if ( !Produit::ctrl_val() )
*/
public static function ctrl_val( $input = null )
{

    if( is_null( $input ) )
        $input = Input::all();

    static::$validation = Validator::make( $input, static::$rules,
static::$messages );

    return static::$validation->passes();

}

/*
/ Fonction pour le Upload et Redim du champ $NomChamp
/ Upload du fichier et affecte l'URL au champ image de la base de
données.
/ Appellé via :  $produit->image      = Produit::upl_img('image');
/
*/
public static function upl_img($NomChamp)
{
    if( Input::has_file($NomChamp) ) {
        static::$image = Input::file($NomChamp);

        if( is_array( static::$image ) && isset( static::$
$image['error'] ) && static::$image['error'] == 0 )
        {

            static::$rep =
Laravel\Config::get('constante.produit_rep_upload_rel');

            static::$ext = File::extension(static::$image['name']);
            static::$nom = Str::random(32).'.'.static::$ext;
            static::$upl = Input::upload('image', static::$rep,
static::$nom);

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 87 / 102	

```

        if (static::$upl) {

            static::$rep_thumb      =
Laravel\Config::get('constante.produit_thumb_upload_rel');
            static::$width_thumb    =
Laravel\Config::get('constante.produit_thumb_width');
            static::$height_thumb   =
Laravel\Config::get('constante.produit_thumb_height');
            static::$mode_thumb     =
Laravel\Config::get('constante.produit_thumb_mode');
            static::$qual_thumb     =
Laravel\Config::get('constante.produit_thumb_qualite');

            Bundle::start( 'resizer' );

            static::$success = Resizer::open( static::$rep .
static::$nom )

                ->resize( static::$width_thumb , static::$
$height_thumb , static::$mode_thumb )
                ->save( static::$rep_thumb .static::$nom , static::$
$qual_thumb );

            if ( static::$success ) {
                return static::$nom;
            }
        }
    }
    return false;
}
//
Rien à Uploader ou Erreur de Upload ou Redim.
}
}

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 02/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 88 / 102	

SCRIPT CONTROLEUR

```
<?php

class Produit_Controller extends Base_Controller
{
    public $restful = true;

    public $layout = '_template.default';
    /*
    /
    / Il faut être Authentifié pour utiliser les méthodes : index,
creation, modifier et supprimer
    /
    */
    public function __construct()
    {

        parent::__construct();

        $this->filter('before','auth')->only(array('index','edit','destroy'));

    }

    /*
    /
    / Index Produit : Liste des Produits : /produit
    /
    */
    public function get_index()
    {

        // Stock URL en cours avec numéro de page et tous les paramètres (?
page=xxx)
        Session::put('url_index', URL::full() );

        $per_page = 5;
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 89 / 102	

```

        $produits = Produit::order_by('Reference','asc')->paginate($per_page);

        $rep_thumb =
Laravel\Config::get('constante.produit_thumb_upload');
        $thumb_default =
Laravel\Config::get('constante.produit_thumb_default');

        $this->layout->title = 'Produit | Liste';
        $this->layout->content = View::make( 'produit.index' )
            ->with('produits', $produits)
            ->with('rep_thumb',$rep_thumb)
            ->with('thumb_default',$thumb_default);
    }

    /*
    /
    / Création enregistrement : /produit/edit
    / Modifier enregistrement : /produit/edit/#ID
    /
    */
    public function get_edit($id=null)
    {

        if ($id!=null) {

            if( ! is_numeric($id) OR ! $produit = Produit::find($id) ) {
                return Redirect::to(Session::get('url_index')); // return
Response::error( '404' );
            }

            $titre = 'Modification Produit';

        }
        else
        {

            $produit = new Produit;

            $titre = 'Création Produit';

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 90 / 102	

```

    }

    $rep_img      =
Laravel\Config::get('constante.produit_rep_upload');
    $img_default  =
Laravel\Config::get('constante.produit_img_default');

    $this->layout->title      = $titre;
    $this->layout->content    = View::make('produit.edit')
        ->with('titre',$titre)
        ->with('produit',$produit)
        ->with('rep_img',$rep_img)
        ->with('img_default',$img_default);
    }

    /*
    / Soumission du Formulaire
    / Enregistrement Création enregistrement      : /produit/edit
    / Enregistrement Modification enregistrement  : /produit/edit/#ID
    /
    */
    public function post_edit($id=null)
    {

        if ($id!=null) {

            $produit = Produit::find($id);

            if (is_null($produit)) {
                return Redirect::to(Session::get('url_index'));
            }

        }

        if ( !Produit::ctrl_val() )
            return Redirect::back()
                ->with('msg_pb', '<strong>Vérifier votre saisie</strong>')
                ->with_errors( Produit::$validation )
                ->with_input();

        try

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 91 / 102	

```

{

    if ($id==null) {
        $produit = new Produit;
    }

    $produit->reference = e(Input::get('reference'));
    $produit->modele     = e(Input::get('modele'));

    // Gestion Upload et Redimenssion des Images
    $success = Produit::upl_img('image');
    if ( $success!=false ) {
        $produit->image = $success;
    }

    $produit->save();

}
catch( Exception $e )
{
    return Redirect::back()
        ->with('msg_pb', '<strong>ATTENTION</strong>, cet
enregistrement n\'est pas conforme (doublon,...)'.$e->getMessage() )
        ->with_input();
}

return Redirect::to(Session::get('url_index'))
    ->with('msg_ok', 'Enregistrement terminé');

}

/*
/
/ Supprimer un enregistrement : /produit/destroy/#ID
/
*/
public function get_destroy($id=null)
{

    if ($id!=null) {

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 92 / 102	Dernière impression le

```

        $produit = Produit::find($id);

        if (!is_null($produit)) {
            try
            {
                $produit->delete();
            }
            catch( Exception $e )
            {
                return Redirect::to('produit/edit'.'.$id)
                    ->with('msg_pb', 'Désolé, un problème technique
nous empêche de supprimer cet enregistrement.')
                    ->with_input();
            }
        }

    }

    return Redirect::to(Session::get('url_index'));

}

}

```

Rédacteur :	Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document :	Laravel3_A_Savoir.odt	Dernière impression le
Page	93 / 102	

VUE : INDEX

```
@if ($produits)

<table class="table table-bordered table-hover">
  <thead>
    <tr>
      <th style="width:40%">référence</th>
      <th style="width:40%">Modèle</th>
      <th style="width:15%">Image</th>
      <th style="width:5%">Action</th>
    </tr>
  </thead>
  <tbody>

    @foreach($produits->results as $produit)

      <tr>
        <td>{{e($produit->reference)}}</td>
        <td>{{e($produit->modele)}}</td>
        <td>
          
        </td>
        <td>
          <div class="pull-right">
            {{ HTML::edit_enre($produit->id,'produit') }}
          </div>
        </td>
      </tr>

    @endforeach

  </tbody>
</table>

{{ $produits->links() }}

@else
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2 Dernière impression le
Document : Laravel3_A_Savoir.odt	
Page 94 / 102	

```

<div class="alert alert-block">
    <h4>Pas de données !!!</h4>
    Pour créer des enregistrements cliquer sur le bouton Création.
</div>

@endif

```

VUE : EDIT

```

<div class="row-fluid">
    <div class="span6">

        {{ Form::open_for_files('#','POST',array('class'=>'form-
horizontal')) }}

        <fieldset>

            <div id="legend">
                <legend>{{ $titre }}</legend>
            </div>

            <div class="control-group">
                {{ Form::label('reference','Référence *',array('class'=>'control-
label')) }}
                <div class="controls">
                    {{ Form::text('reference',$produit-
>reference,array('class'=>'input-xlarge')) }}
                </div>
            </div>
            <div class="control-group">
                {{ Form::label('modele','Modèle *',array('class'=>'control-label'))
}}
                <div class="controls">
                    {{ Form::text('modele',$produit->modele,array('class'=>'input-
xlarge')) }}
                </div>
            </div>

            <div class="control-group">

```

Rédacteur :	Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document :	Laravel3_A_Savoir.odt	Dernière impression le
Page	95 / 102	

```

        {{ Form::label('image','Charger une image (.jpg, .jpeg, .gif,
.png)',array('class'=>'control-label')) }}

        <div class="controls">
            {{ Form::file( 'image' ) }}
        </div>
    </div>

    <div class="control-group">
        <div class="form-actions">
            <a class="btn" href="{{ Session::get('url_index') }}">Annuler</a>
            {{ Form::submit('Valider',array('class'=>'btn btn-primary')) }}
            <a href="#delete-{{ $produit->id }}" role="button" class="btn btn-
inverse pull-right" data-toggle="modal"><i class='icon-trash icon-
white'></i> Supprimer</a>
        </div>
    </div>

</fieldset>

    {{ Form::close() }}

</div>

<div class="span6">
    <h4>Image du Produit</h4>

</div>
</div>

<div id="delete-{{ $produit->id }}" class="modal hide fade" tabindex="-1"
role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
    <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-
hidden="true"><×</button>
        <h3 id="myModalLabel">Supprimer Enregistrement</h3>
    </div>
    <div class="modal-body">
        <p>Confirmer la suppression de cet enregistrement</p>

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 02/p2
Document : Laravel3_A_Savoir.odt	
Page 96 / 102	Dernière impression le


```

</div>
<div class="modal-footer">
    <button class="btn" data-dismiss="modal" aria-
hidden="true">Annuler</button>
    {{ HTML::link('produit/destroy/'.$produit->id , 'Confirmer' ,
array('class' => 'btn btn-primary')) }}
</div>
</div>

```

SCRIPT DES CONSTANTES

`\application\config\constante.php`

```

<?php
return array(

    /*
    |-----
    |  CONSTANTES APPLICATION
    |-----
    |
    |  Valeurs Constantes utilisées dans l'application
    |
    |  Utiliser : $rep_thumb = Laravel\Config::get('constante.produit_rep_upload');
    |
    */

    'produit_rep_upload'      => URL::to('uploads/users/'),      // pour
Affichage
    'produit_rep_upload_rel'  => path('public').'uploads/users/', // Pour le
Upload
    'produit_img_default'    =>
'http://www.placeholder.it/300x300/EFEFEF/AAAAAA&text=Aucune+image',
    'produit_thumb_upload'   => URL::to('uploads/users/thumb/'),
    'produit_thumb_upload_rel' => path('public').'uploads/users/thumb/',
    'produit_thumb_width'    => 75,
    'produit_thumb_height'   => 75,
    'produit_thumb_qualite'   => 90,
    'produit_thumb_mode'     => 'exact',      // mode= exact, portrait, landscape,
fit, auto , crop

```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 97 / 102	

```
'produit_thumb_default' =>
'http://www.placeholder.it/75x75/EEEEEF/AAAAAA&text=Aucune+image',
);
```

RESIZER

\application\bundles.php

Pour un appel automatique : 'resizer' => array('auto'=>true),

Resizer en mode appel manuel :

```
return array(
    'docs' => array('handles' => 'docs'),
    'swiftmailer' => array('auto'=>true),
    'resizer'
);
```

A TESTER

Générateur de formulaires

<http://forums.laravel.io/viewtopic.php?id=1894>

<https://github.com/JonoB/Flare-Form>

Issu de <http://bundles.laravel.com/bundle/bootstrapper>

<https://github.com/Pasvaz/bootstrapper>

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 98 / 102	Dernière impression le

ETENDRE LA CLASSE DE VALIDATION

ATTENTION POUR LARAVEL 2, pour LARAVEL 3 préférer la méthode de la page 51

<http://anchetawern.github.com/blog/2012/12/25/extending-laravels-validation-class/>

Conception d'une nouvelle librairie

/application/librairies/validator.php pour la validation d'un array ou d'une date.

```
<?php
class Validator extends Laravel\Validator {

}
?>
```

Supprimer l'autoloading de la classe validator

dans /application/config/application.php

```
'Task'      => 'Laravel\\CLI\\Tasks\\Task',
'URI'       => 'Laravel\\URI',
'Validator' => 'Laravel\\Validator', //remove this line
'View'      => 'Laravel\\View',
```

Création règle de validation pour un tableau

```
<form method="post">
  <label for="">Students</label>
  <input type="text" name="student_name[]" id="student_name">
  <input type="text" name="student_name[]" id="student_name">
  <input type="text" name="student_name[]" id="student_name">
  <input type="submit" value="Save">
</form>
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	
Page 99 / 102	Dernière impression le

Rédacteur :	Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document :	Laravel3_A_Savoir.odt	Dernière impression le
Page	100 / 102	

La règle

```
$rules = array('student_name' => 'required');
```

La fonction

```
/**
 * checks if an array input has no empty values
 */
public function validate_arrayfull($attribute, $value, $parameters){
    return in_array('', $value);
}
?>
```

Les Paramètres :

Attribute : Le nom (name) donné dans le input (Exemple : name ou age)

Value : Valeur dans le input (Exemple : Yael ou 27)

Parameters : Paramètres additionnels passés après la définition de la règle.

Exemple

```
<?php
$rules = array(
    'username' => unique:tbl_users,username
);
?>
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 101 / 102	

Validation Date

Tester si la date est complète et qu'il s'agit bien d'une date

```
<?php
public function validate_date($attribute, $value, $parameters){
    $e_str = explode("-", $value);
    if(count($e_str) === 3){

        //expected format for date input is: Y-m-d
        $year = $e_str[0];
        $month = $e_str[1];
        $day = $e_str[2];

        return checkdate($month, $day, $year); //expected format for checkdate
is: m-d-y
    }
    return false;
}
?>
```

Rédacteur : Frédéric LORGE	Créé le 05/02/2013 9:30 O2/p2
Document : Laravel3_A_Savoir.odt	Dernière impression le
Page 102 / 102	