

INF440

Programmation MVC
(sous le Framework Laravel)

MVC (LARAVEL)

- Laravel est un framework de développement de type MVC
- (Modele Vue Controleur)
- doc Laravel:
 - <http://docs.laravel.fr/4.1/introduction>



LARAVEL

Symfony

MVC

Blade

Eloquent

Artisan

View

Route

Model

Controller

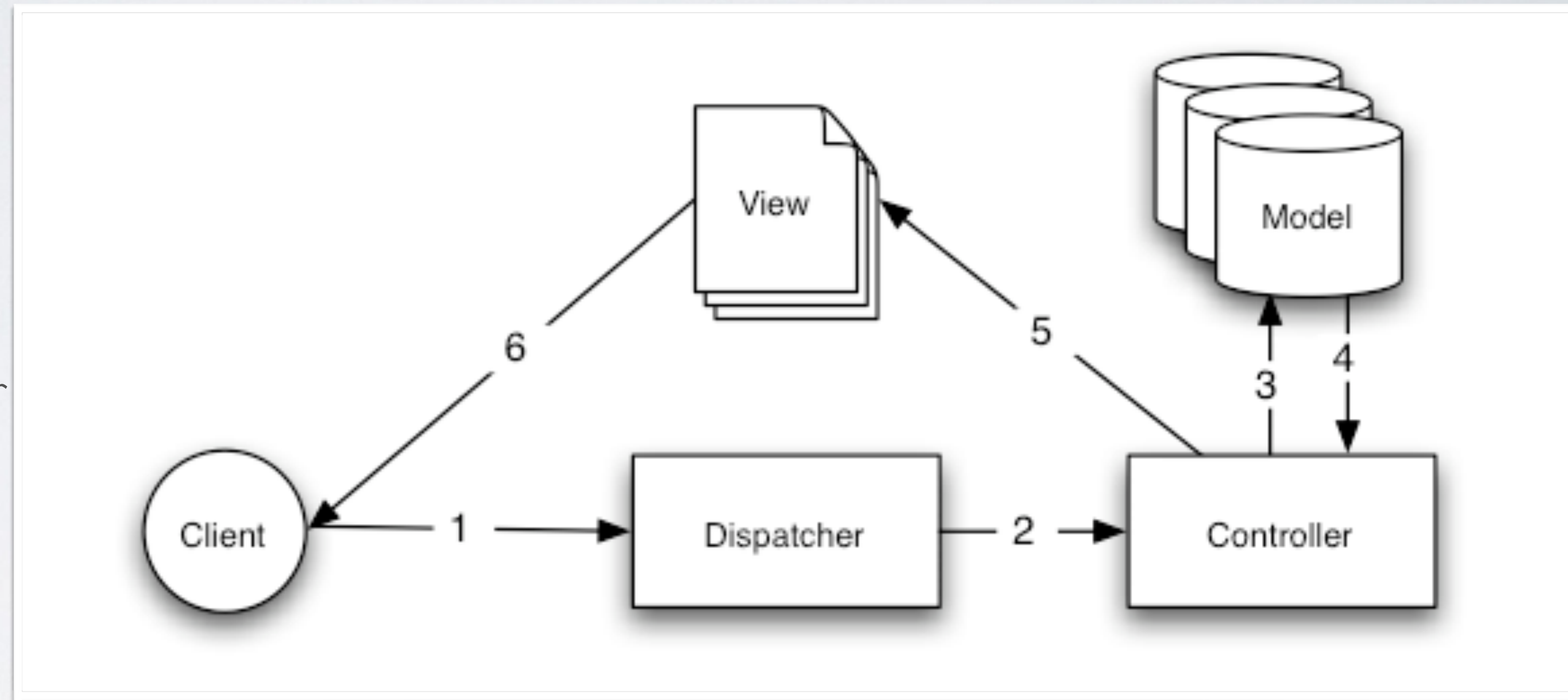
... (Fuelphp, cakephp, Yii, Synfony, Silex, Zen ...)

LARAVEL

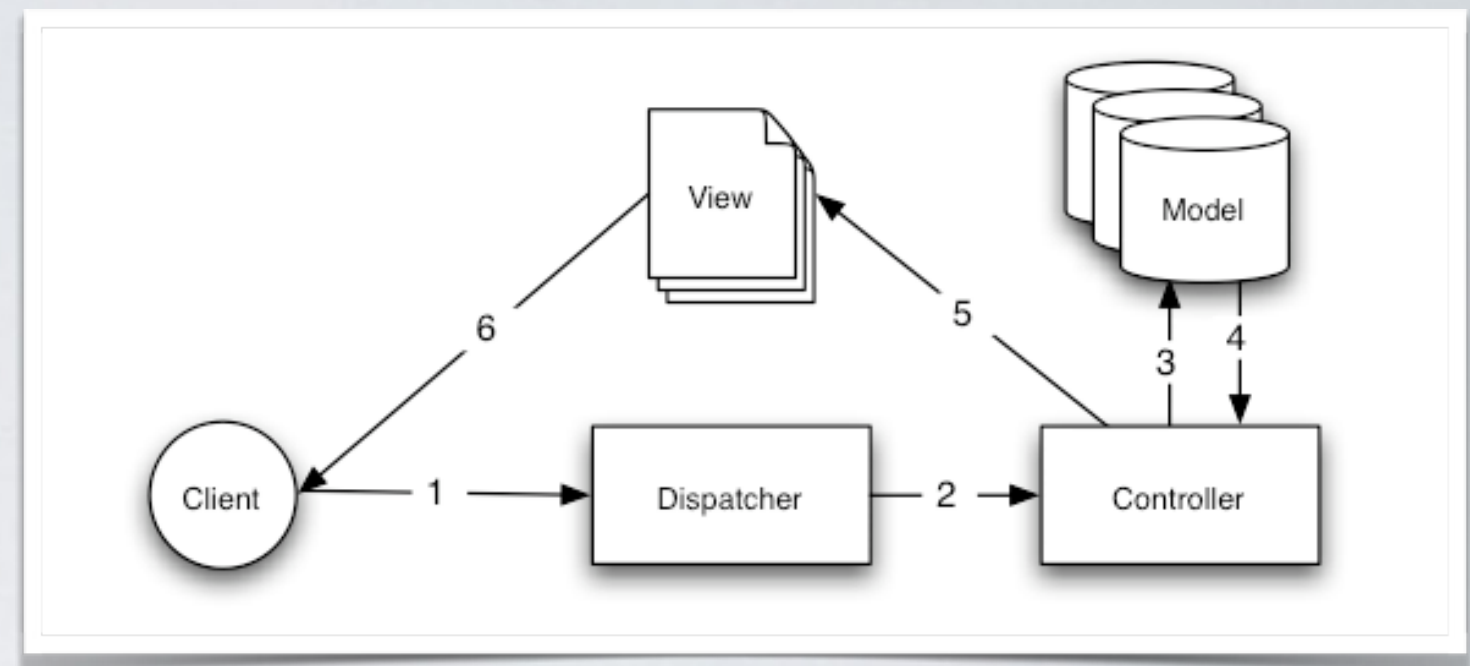
- installation laravel avec un nouveau projet:
 - utiliser composer (l'installer avant sur votre ordinateur!):
 - `composer create-project laravel/laravel monsite`
- (<http://laravel.com/docs/quick>)
- <http://docs.laravel.fr/4.1>
- attention! il faut php5.4 minimum
- mise en production sur un serveur distant: transfert des fichiers (ftp)
- utilisation de l'outils artisan (migration, transfert..)

MVC

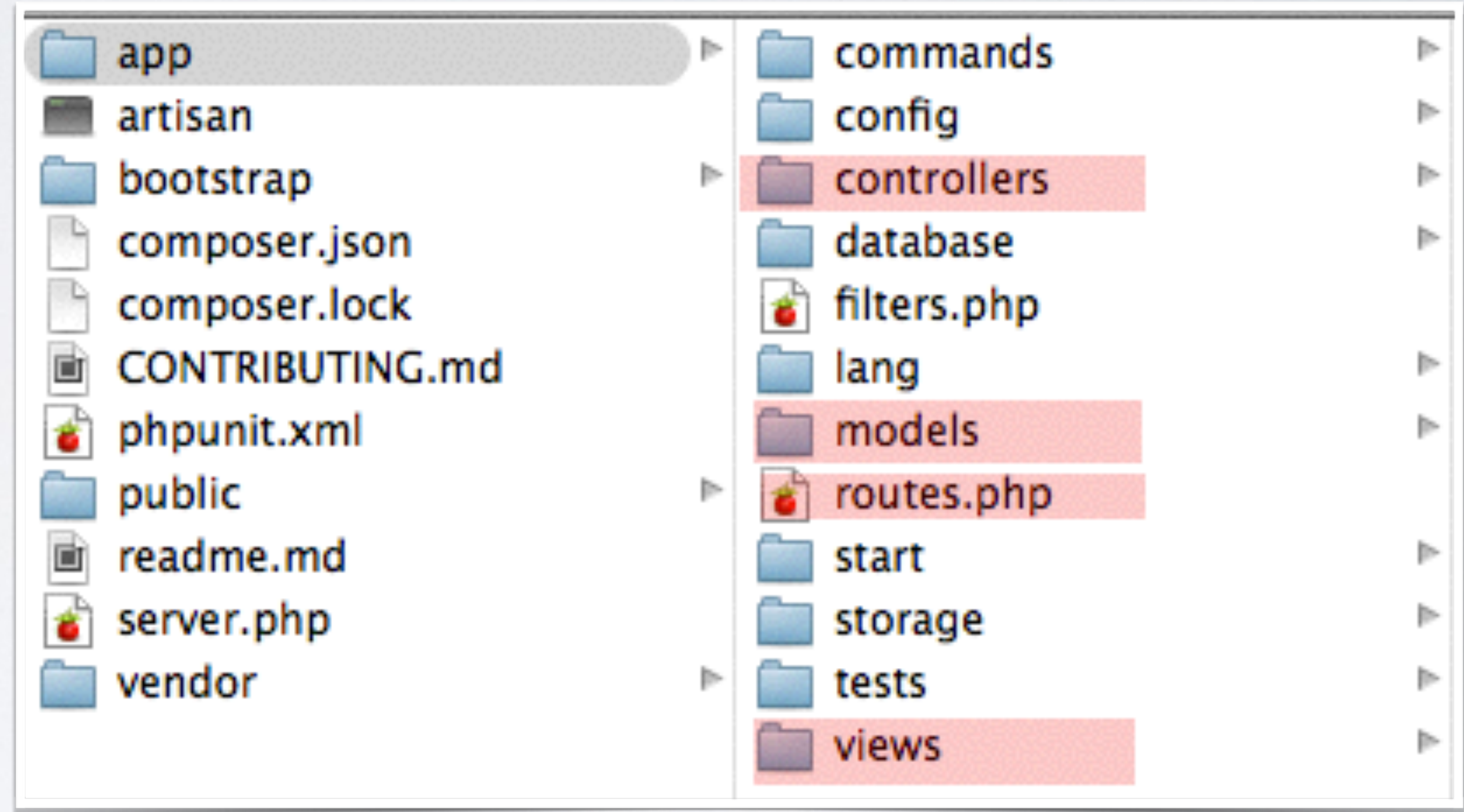
- la structure du MVC —->
- organisation de vos fichiers:
 - 1 fichier global pour le Dispatcher (la route « route.php »)
 - 1 fichier par vue « mavue.php »
 - 1 fichier par contrôleur « moncontroleController.php »
 - 1 fichier par model « Article.php »



MVC

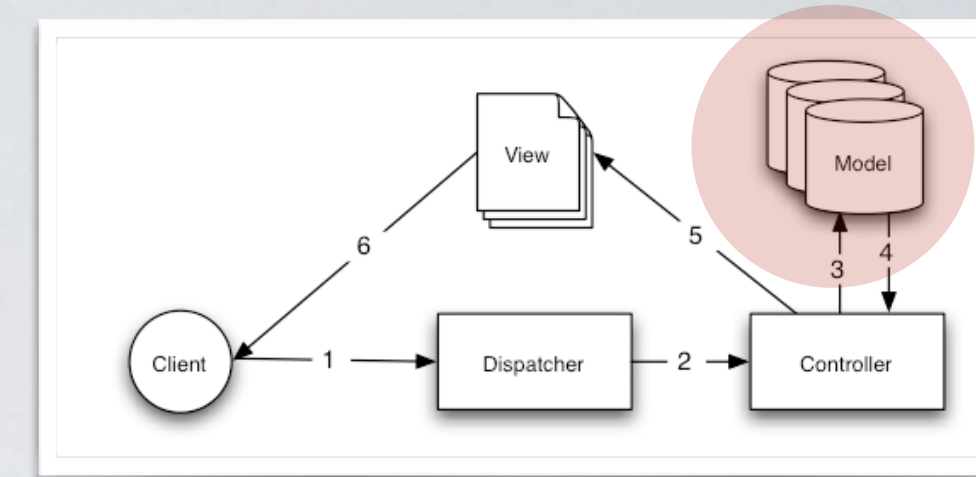


- emplacement de ces différents fichiers dans Laravel



LARAVEL

- MVC:



- le fichier Modele,

- il permet de décrire la méthode d'accès aux données de la base. tous cela à travers un objet.
- attention aux règles de nommage! (Laravel se sert de ces règles pour trouver les tables et les clés. exemple: une clé étrangère dans une table s'écrit « nom de la table »_id , la clé primaire d'une table doit impérativement s'appeler « id »)

| localhost » bdlaravel » articles | | | | | | | | |
|--|---|----------------|-------------|-------------------|-----------|------|--------|----------------|
| Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations | | | | | | | | |
| | # | Nom | Type | Interclassement | Attributs | Null | Défaut | Extra |
| <input type="checkbox"/> | 1 | id | int(11) | | | Non | Aucune | AUTO_INCREMENT |
| <input type="checkbox"/> | 2 | designation | varchar(50) | latin1_swedish_ci | | Non | Aucune | |
| <input type="checkbox"/> | 3 | prix | int(11) | | | Non | Aucune | |
| <input type="checkbox"/> | 4 | fournisseur_id | int(11) | | | Non | Aucune | |

| localhost » bdlaravel » fournisseurs | | | | | | | | |
|---|---|------------|-------------|-----------------|-----------|------|---------------------|----------------|
| Afficher Structure SQL Rechercher Insérer Exporter Importer Opérations Dé | | | | | | | | |
| | # | Nom | Type | Interclassement | Attributs | Null | Défaut | Extra |
| <input type="checkbox"/> | 1 | id | int(10) | | UNSIGNED | Non | Aucune | AUTO_INCREMENT |
| <input type="checkbox"/> | 2 | nom | varchar(50) | utf8_unicode_ci | | Non | Aucune | |
| <input type="checkbox"/> | 3 | codepostal | int(11) | | | Non | Aucune | |
| <input type="checkbox"/> | 4 | created_at | timestamp | | | Non | 0000-00-00 00:00:00 | |
| <input type="checkbox"/> | 5 | updated_at | timestamp | | | Non | 0000-00-00 00:00:00 | |

- ORM Eloquent

- Object-Relational Mapping, permet de voir notre base de données relationnelle comme un objet

```
<?php

class Article extends Eloquent {

    public function fournisseur(){
        return $this->belongsTo('Fournisseur');
    }

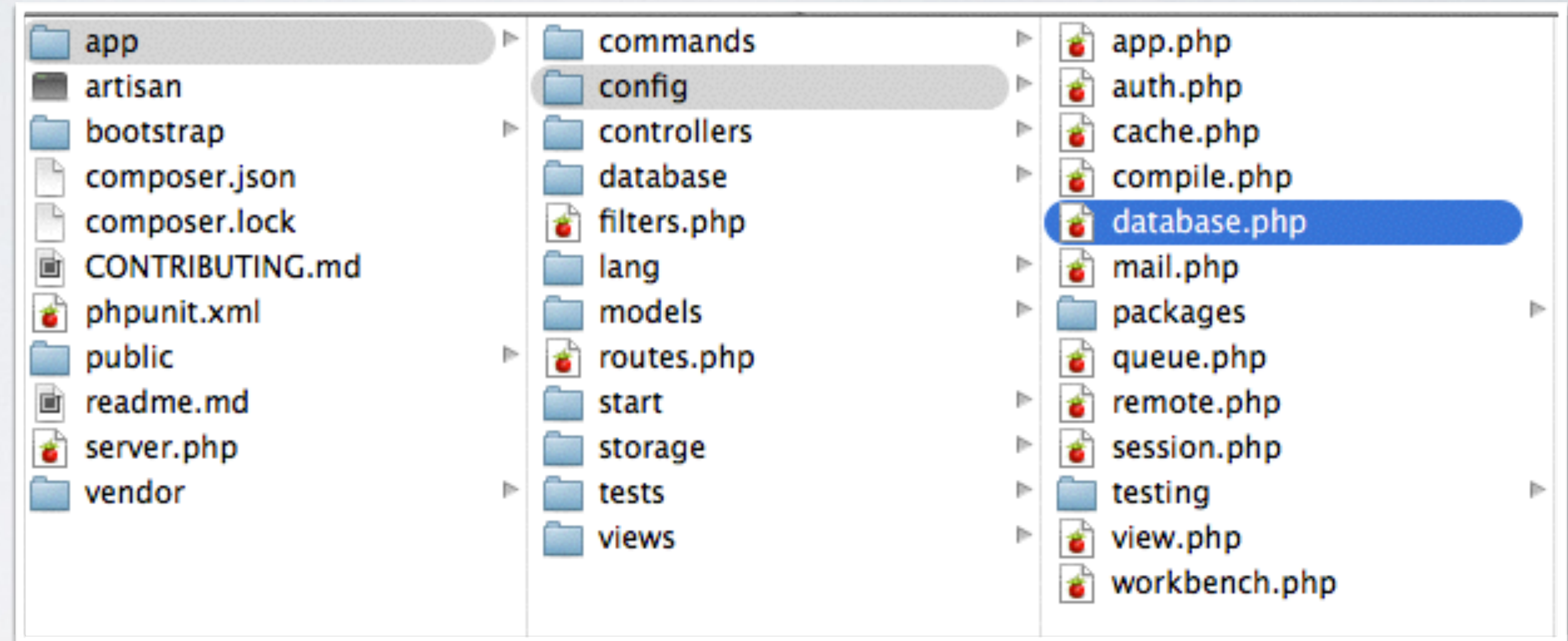
}
```

- <http://docs.laravel.fr/4.1/eloquent>

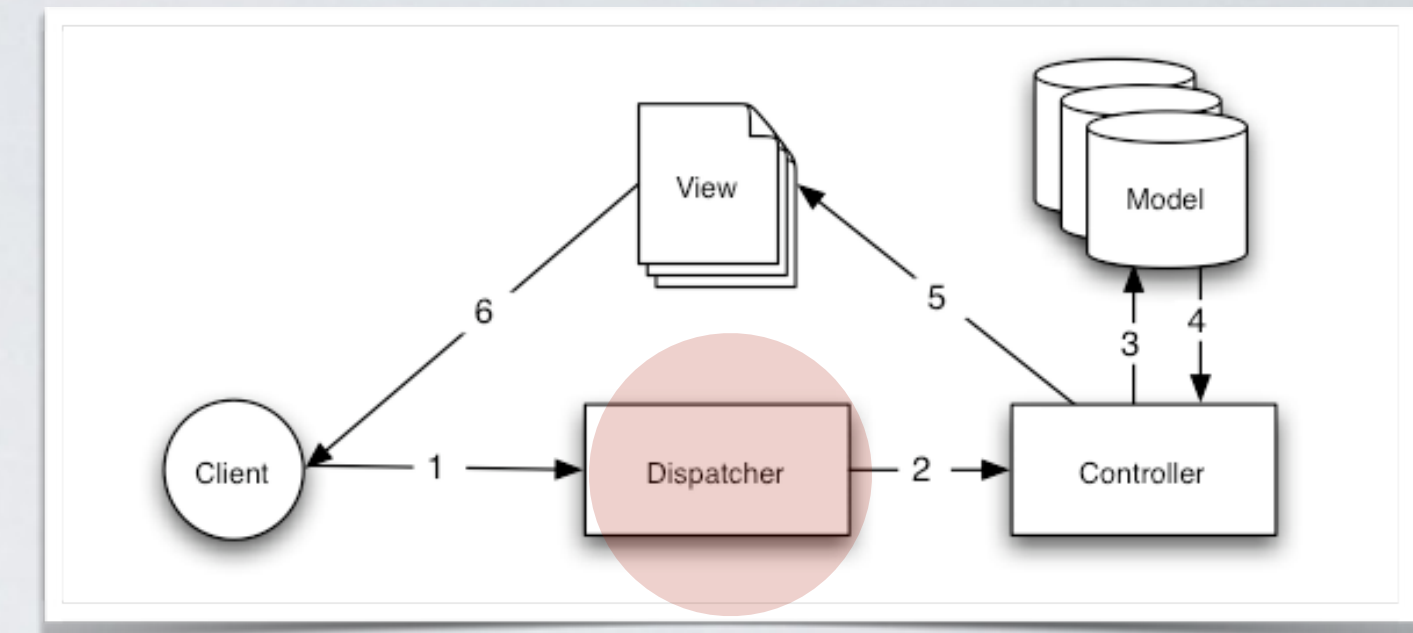
LARAVEL

- pour pouvoir accéder à votre base de données il faut paramétrer le fichier de config

```
database.php
51      'database' => __DIR__.'/../database/prod
52      'prefix'   => '',
53  ),
54
55      'mysql' => array(
56          'driver'   => 'mysql',
57          'host'     => 'localhost',
58          'database' => 'bdlaravel',
59          'username' => 'root',
60          'password' => 'root',
61          'charset'  => 'utf8',
62          'collation' => 'utf8_unicode_ci',
63          'prefix'   => '',
64      ),
65
66      'pgsql' => array(
67          'driver'   => 'pgsql',
```

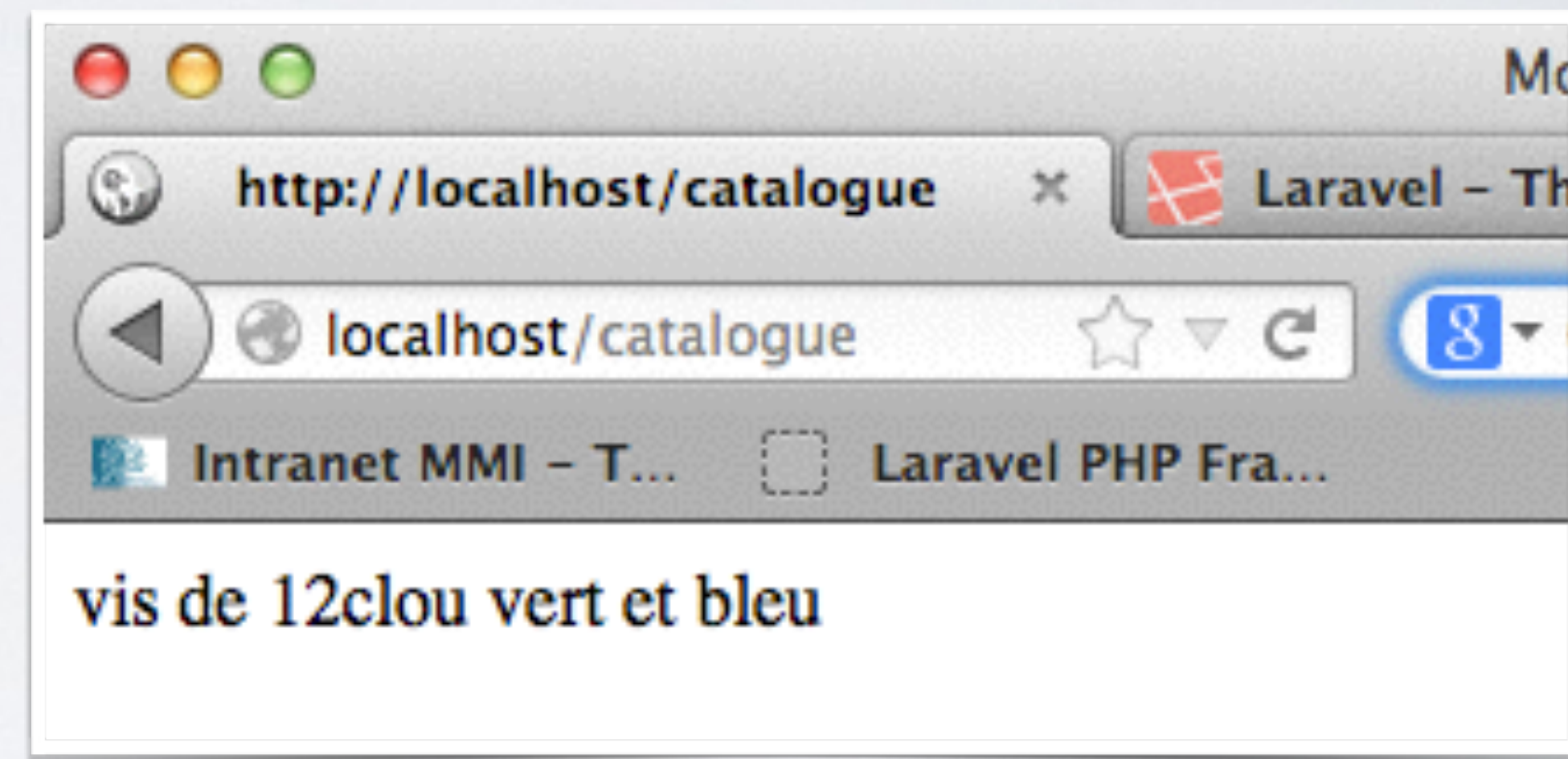


LARAVEL



- la route:

- elle permet de décrire la relation entre l'url tapée par le client et le contrôleur à lancer
- ici le client demande « catalogue ».
- dans la route « catalogue » lancera la méthode « index » du contrôleur « CarticlesController »

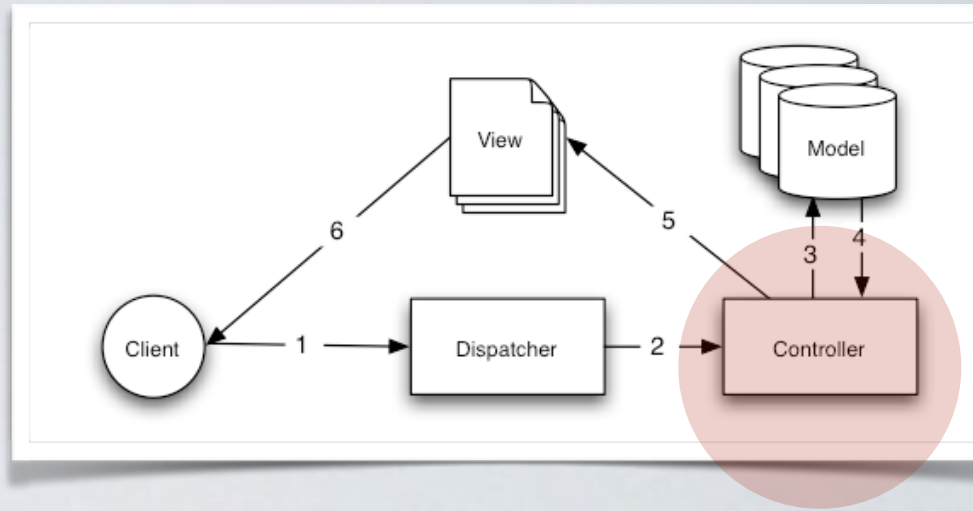


```
<?php

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

Route::get('/catalogue', "CarticlesController@index");
```


LARAVEL



- le contrôleur
- il permet de récupérer les informations du modèle et de l'envoyer vers la vue pour la mise en forme.
- « Article::all() » récupère toutes les données du modèle « Article »
- make(« Varticle ») appelle la vue « Varticle »
- ->with('articles',\$articles) passe toutes les données à la vue

```
<?php

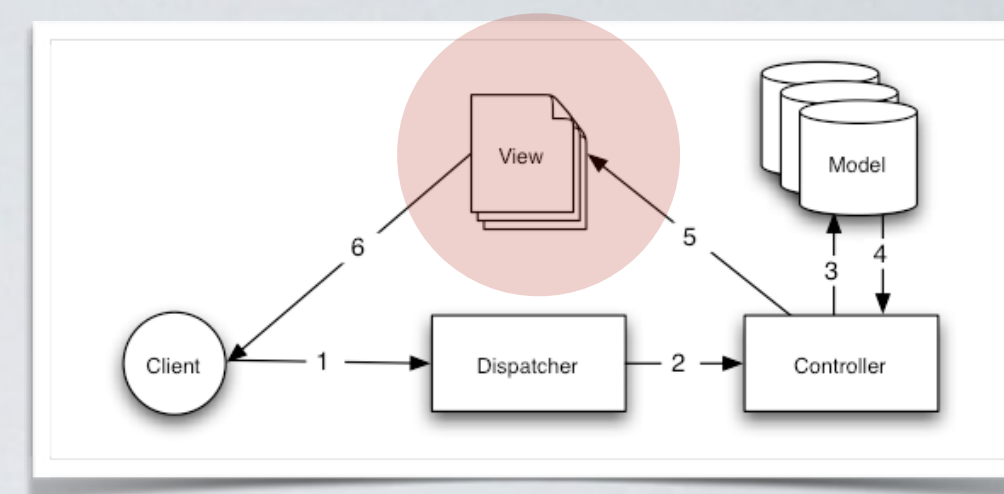
class ArticlesController extends \BaseController {

    public function index()
    {
        $articles = Article::all();

        return View::make('Varticle')->with('articles',$articles);
    }

}
```


LARAVEL



- la vue (ici une structure minimaliste avec boucle en php) « Varticle.php ». ici du php dans la vue.... à éviter, nous verrons plus loin l'utilisation de « Blade » pour restructurer cela.
- \$articles est la variable envoyée depuis le controleur

```
<?php  
  
foreach ($articles as $un ) {  
    |  
    echo $un->designation;  
}  
?>
```

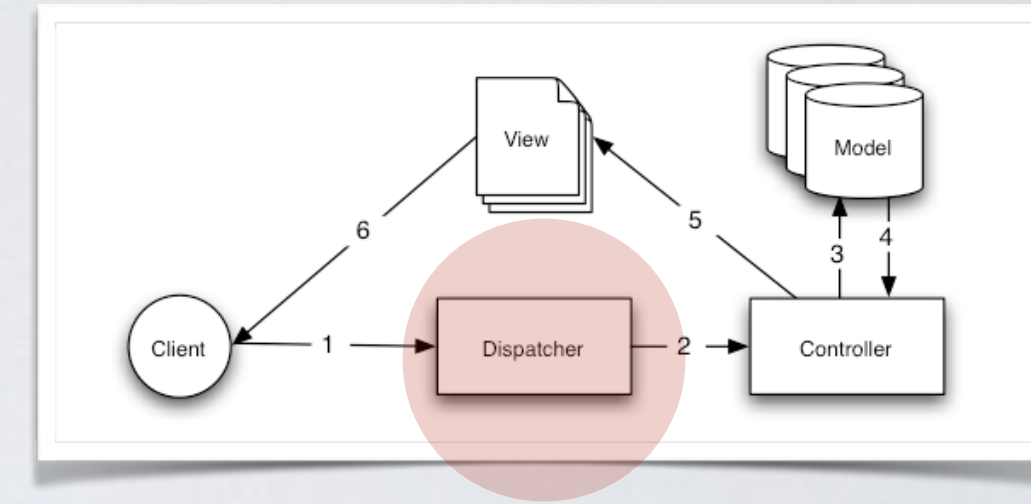

LARAVEL

- Récupération de paramètre d'URL (ex résultat d'un formulaire renvoyant ses infos via l'URL)
- declaration des paramètres d'url dans la route
- exemple « Dutaf »

```
<?php

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

Route::get('/catalogue', "CarticlesController@index");
Route::get('budget/{maxi}', "BudgetController@index" );
```



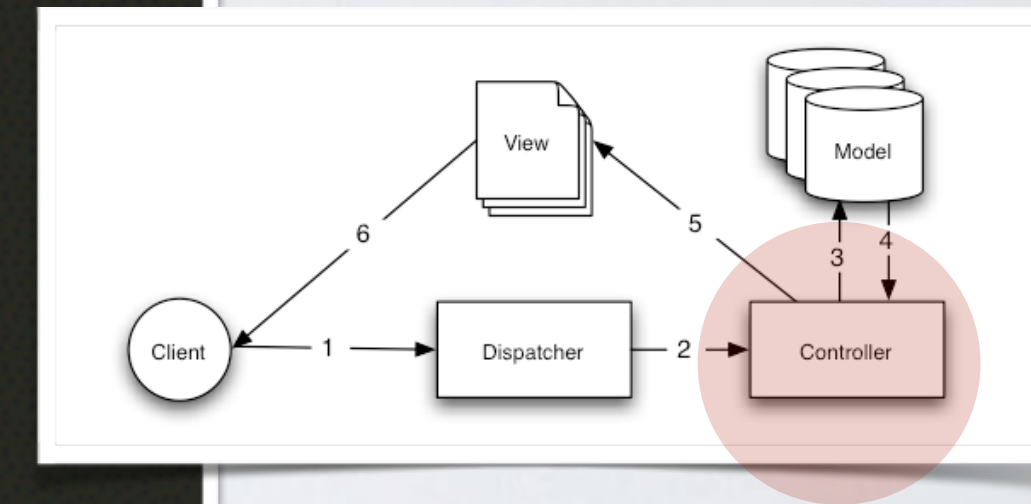
```
<?php

class BudgetController extends BaseController {

    public function index($maxi)
    {
        $articles = Article::where('prix', '<' , $maxi )->get();

        return View::make('Varticle')->with('articles',$articles);
    }

}
```



LARAVEL

- Moteur de template « Blade »:
- création d'un modele de template du site pour toutes les pages (« template.blade.php »)
- création de la vue « Vcatalogue.blade.php » qui fera appel au template global

```
<html>
<body>
    <h1>@yield('titre')</h1>

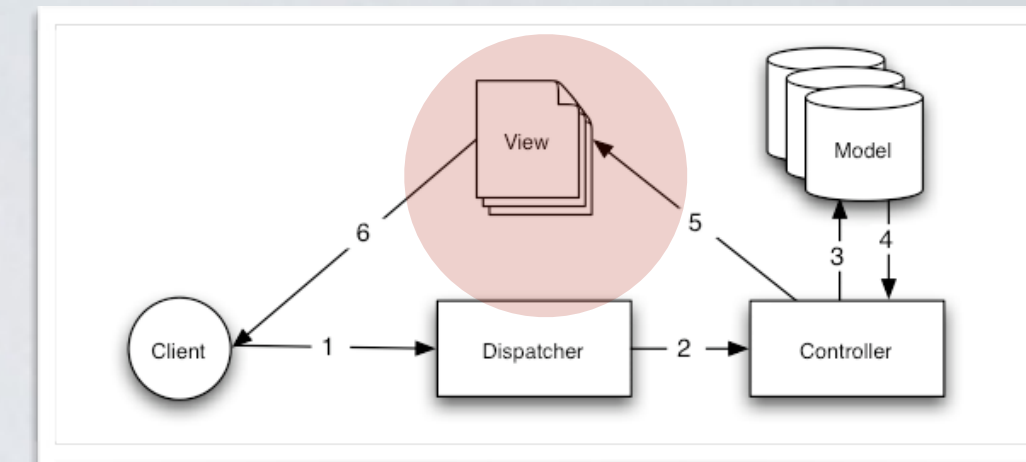
    @section('contenu')
        <p>Contenu principal.</p>
    @show
</body>
</html>
```

```
@extends('template')

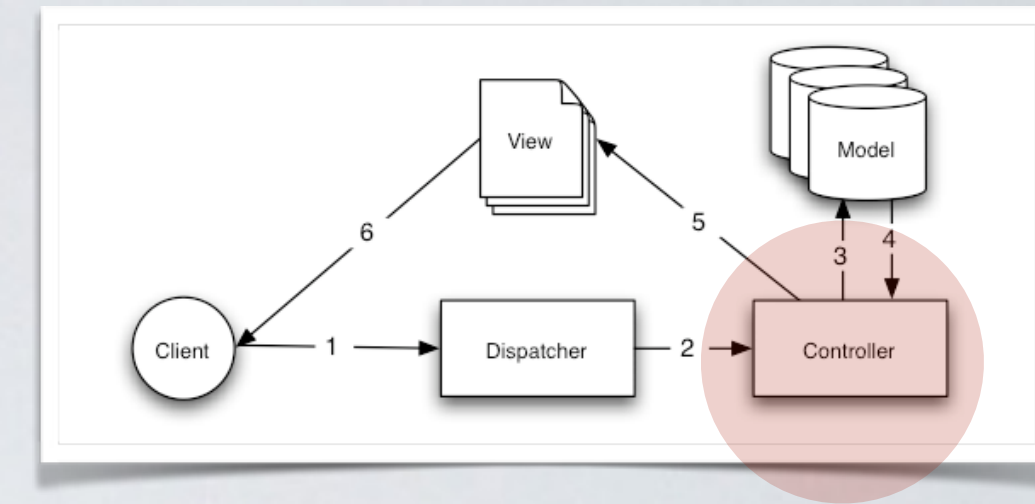
@section('titre')
    Voici mon catalogue
@stop

@section('contenu')
    @parent
    <p>Contenu de la page</p>
    @foreach ($articles as $un)
        {{ $un->designation }}<br>
    @endforeach

@stop
```



LARAVEL



```
<?php  
  
class ArticlesController extends \BaseController {  
  
    public function index()  
    {  
        $articles = Article::all();  
  
        return View::make('Vcatalogue')->with('articles',$articles);  
    }  
}
```

- appel de cette nouvelle vue dans le contrôleur:

LARAVEL

- Artisan:
- <http://docs.laravel.fr/4.1/artisan>