

내가 아는 Laravel Framework를 소개합니다.

(아는데 까지만...)

안정수
@findstar



발표자 소개



안정수 - Seafuture Inc. 해운 솔루션

전세계 날씨, 태풍 분석
Backend 개발자,
Linux, PHP, Java
아카펠라, 사진
남들보다 느려도 여유있기를 좋아하는.
@findstar

주의



주의사항?

내용은 많은데 아는게 없어서

이 문서는 수박에 겉 훑기 수준입니다.

미리 양해를 구합니다.

실은 시간이 없어서 미몽사몽 날림으로 만들어서 부실합니다.
진지합니다. 궁서체 입니다.

목차 - |

I. 배경

2. Laravel?

누구냐 넌? - 한국어 표현, 그것이 알고 싶다.

미녀는 석류를 좋아해 - laravel 과 composer

app run phase - 기본 동작 설명

Eloquent ORM

Restful, Route - controller - View(template)

Mail, queue

IoC

artisan cli : CLI 를 알려드립니다. 우리에겐 command 가 있다.

목차 - 2

3. 실제적인 문제들

마이그레이션.

유용한 Github 프로젝트들..

IDE 지원

4. 아직 못다한 이야기들..

facades, events, cache, session, migration & seeding .. 기타 등등.

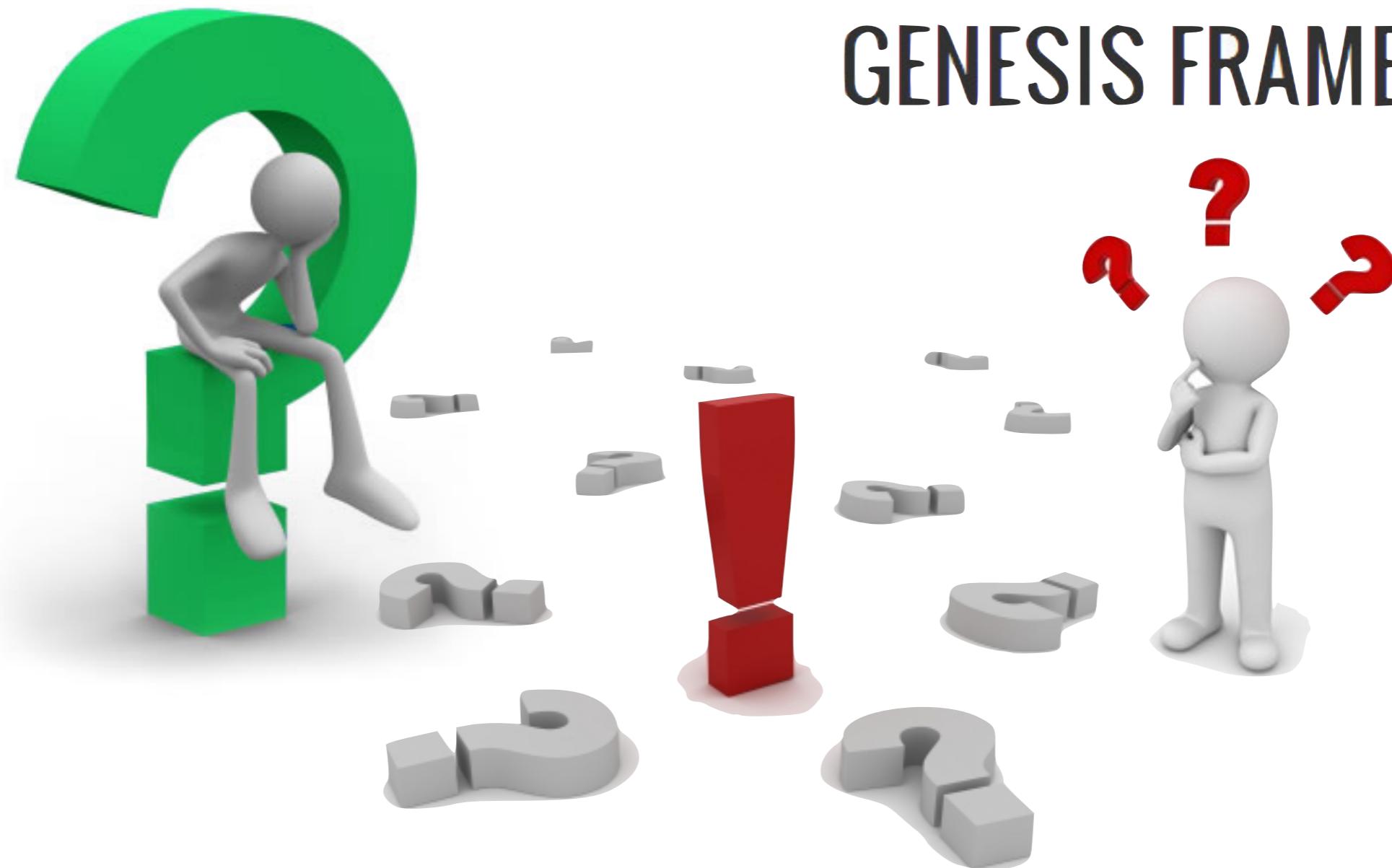
5. 참고

오늘의 주제 > 배경 I



전 여전히 티코 뜯어서 아우디 만드는 중입니다.

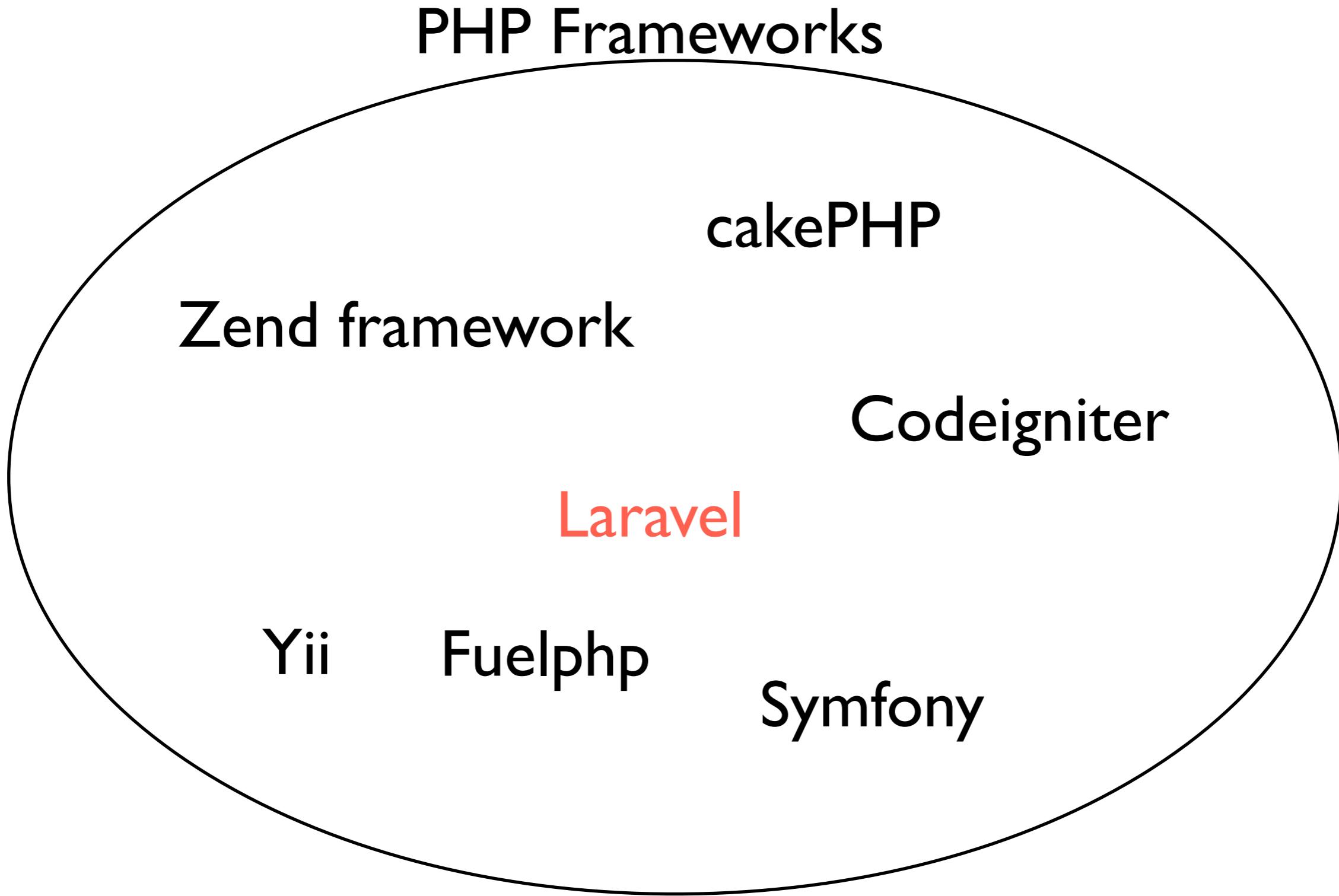
오늘의 주제 > 배경2



GENESIS FRAMEWORK

멋진 프레임워크를 찾아 보자!

오늘의 주제 > a lot of framework



오늘의 주제 > Why Laravel?

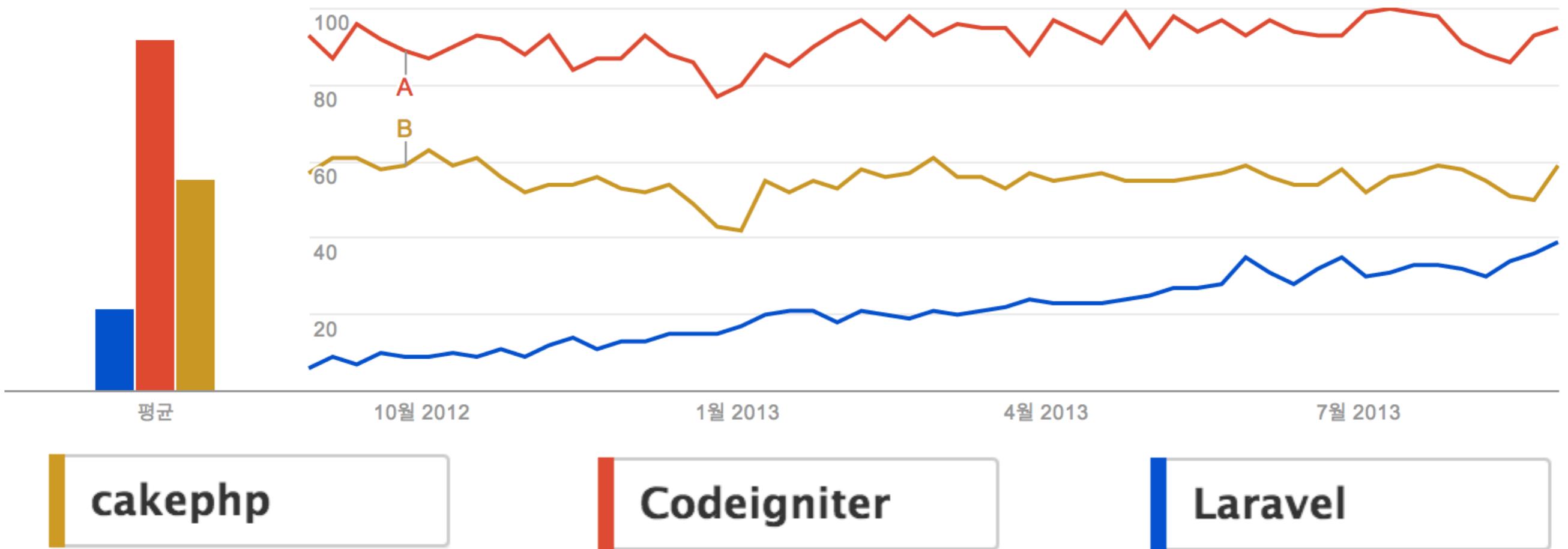
1. MVC with ORM
2. RESTFUL
3. Template
4. DI
5. TESTABLE



제일 HOT 한 걸로!

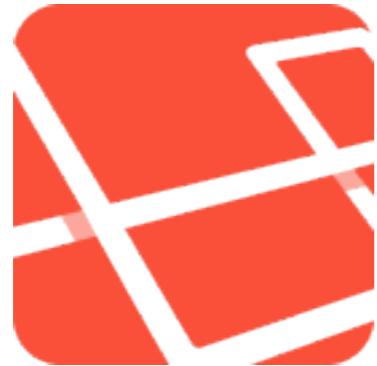
오늘의 주제 > Laravel is hot?

구글 트랜드 (지난 12개월간 PHP 프레임워크 검색)



2014 하반기에는 얼마나 트랜드가 올라갈까요?

오늘의 주제 > 시작합니다.



Laravel Framework를 알려드립니다.

팍팍!... 은 아니고 아는만큼만;;;;



Laravel > 이름이 뭐예요?



라라벨? 라르벨? 라러벨?
한국어 표기가 뭔가요?
무슨 뜻인가요?

Laravel > 이름이 뭐예요? |



Jason Lewis : Major 컴퓨터



Taylor Otwell : 메인 개발자

IRC에서 묻는 사람들 질문에 답변하기를...

메인 개발자인 Taylor Otwell 가 말하길 (공식적으로는)

“**Taylor Otwell** When trying to think of names, I thought about the geography of Tolkien's Middle Earth and C.S. Lewis' Narnia. In Narnia, Cair Paravel is the name of the castle where the kings and queens of Narnia live. Laravel rhymes with Paravel. I thought the name had a classy and sophisticated ring to it.”

“영화 나니아 연대기에 나오는 성이름이 Cair Paravel 인데 이 발음이 멋있어 보여서 이와

비슷한 발음이 나는 Laravel로 지었다” (from jason 블로그)

이말대로라면

한국식으로는 케어 페러~(르와 러 중간 발음) 벨에서 따온

라르벨/라러벨 정도가 될듯?

Laravel > 이름이 뭐예요? 2



Dayle Rees : Major 컨트리뷰터



Taylor Otwell : 메인 개발자

비공식적으로 :

“**Dayle Rees** Well you see, the name came to me, when during a Skiing holiday with Lara Croft, an unfortunate tumble led to me becoming Enveloped within her ample bosom. I had become Laraveled.. which clearly led to the name Laravel.”

“Lara Croft enveloped ”

Lara Croft + enveloped

=> Laravel

이말대로 라면

한국식으로는

“라라벨” (라라 크래프트 + 엔벨롭)



Laravel > 이름이 뭐예요? 3

그냥 편하게 ‘라라벨’로 부르겠습니다.

Laravel > Composer

Laravel 은 composer 와 함께 동작합니다.



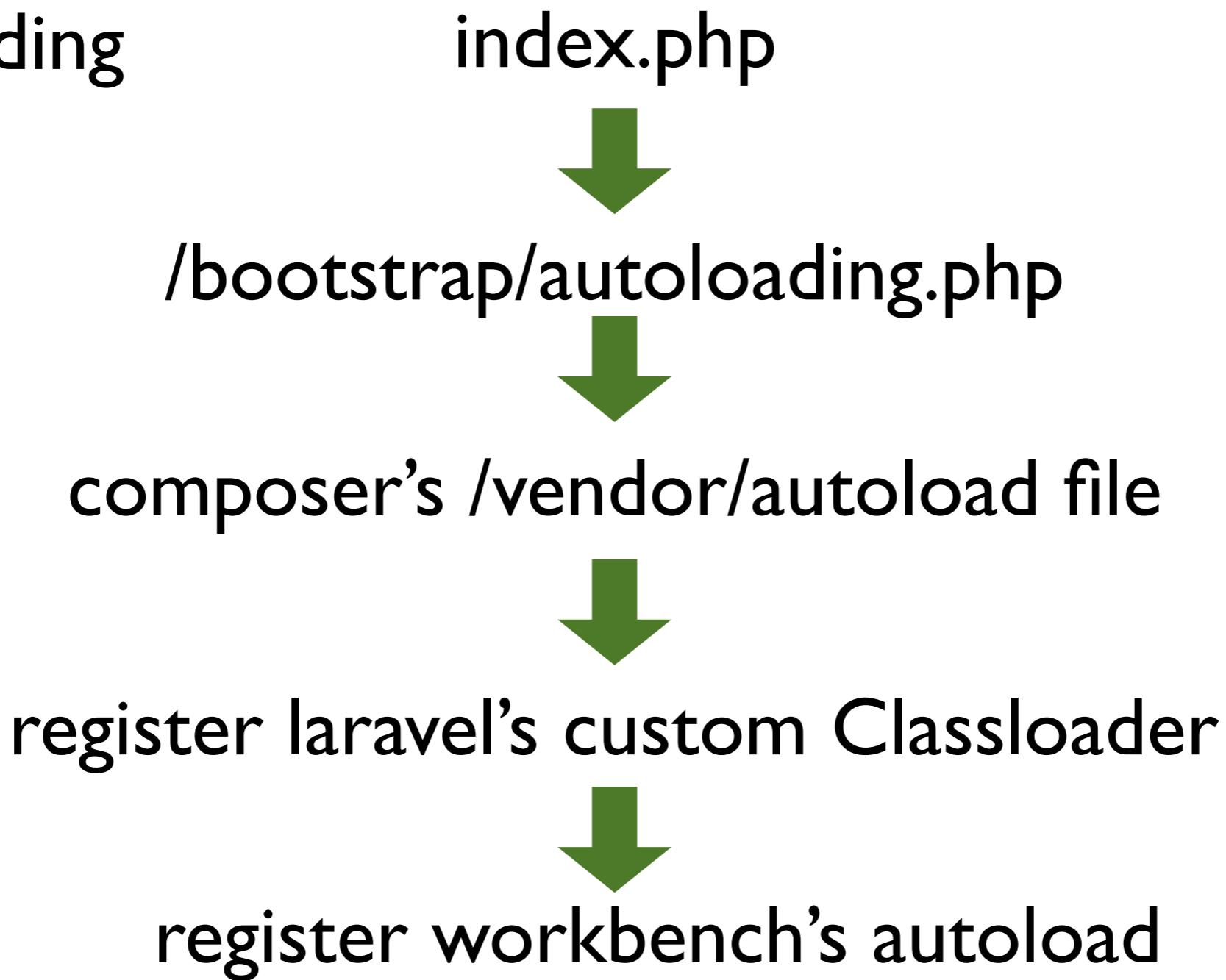
Laravel > Composer

```
{  
    "require": {  
        "illuminate/foundation": ">=1.0.0",  
        "mockery/mockery": "dev-master"  
    },  
    "autoload": {  
        "classmap": [  
            "app/controllers",  
            "app/models",  
            "app/database/migrations",  
            "app/tests\TestCase.php"  
        ]  
    }  
}
```

autoload 오예~

Laravel > Laravel app loading #1

Phase I.
Autoloading



Laravel > Laravel app loading #2

Phase 2.
Application

/bootstrap/start.php



Create new instance of illuminate application



register 3 serviceproviders



bind application filepaths application object



require foundation start.php

Laravel > Laravel app loading #3

Phase 3.
bootstrap

set \$env var



set the facade



load config files



/app/start/global.php & \$env.php



/app/route.php

Laravel > Laravel app loading #4

Phase 4.

run

run the app



set session



find the route

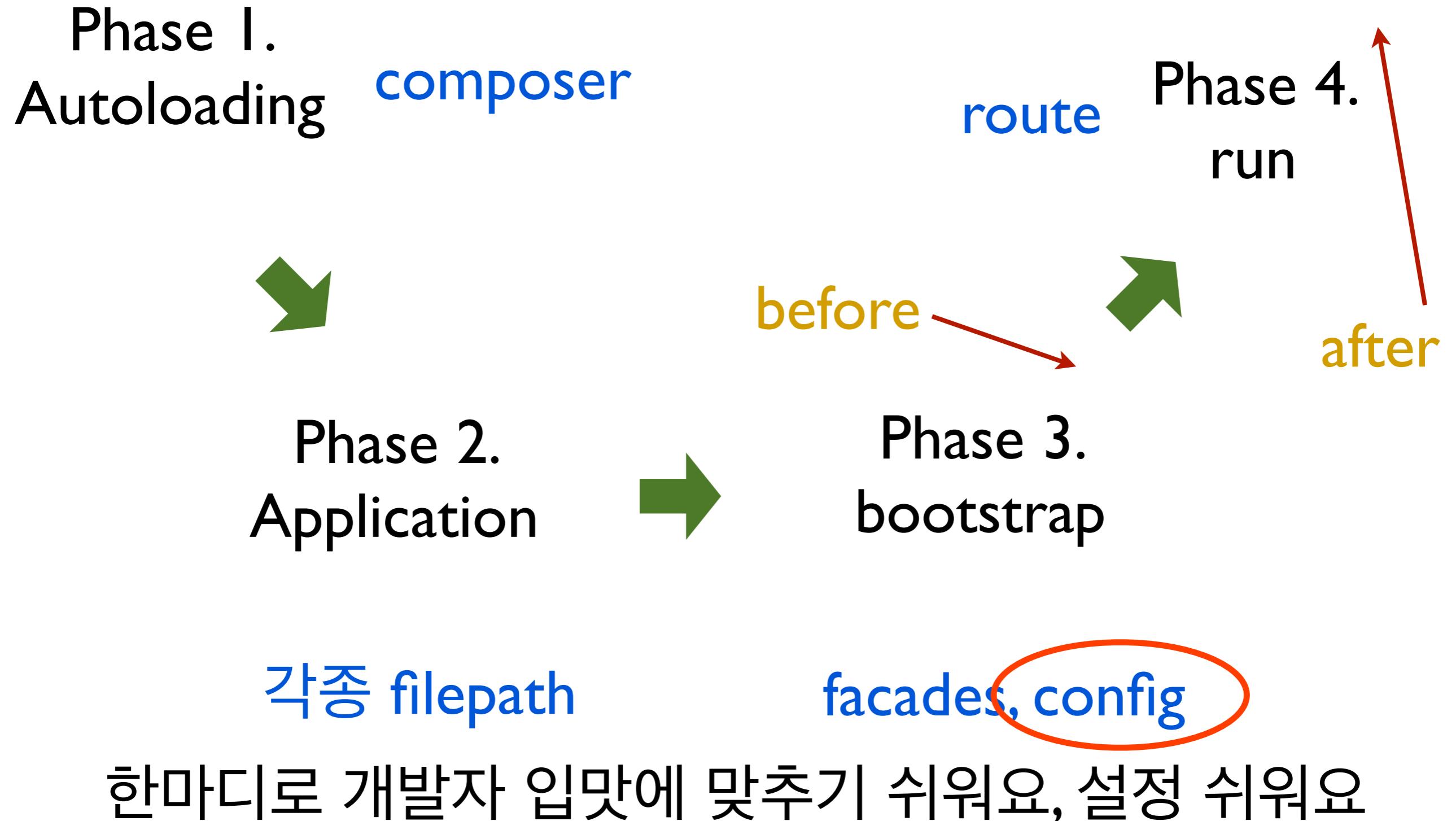


run the route code



response object

Laravel > Laravel app loading



Laravel > Laravel app loading

Phase 1.		
Autoloading	composer autoloading	no more ‘require’
Phase 2.		
Application	filepath setting	add my own lib
Phase 3.		
bootstrap	facades, config	easy config
Phase 4.		
run	resource - route handle	easy response
	쉽고 간편하게!	

Laravel > Eloquent ORM

```
// Fetch all tasks
$tasks = Task::all();

// Fetch the task with an id of 1
$task = Task::find(1);

// Update a task
$task = Task::find(1);
$task->title = 'Put that cookie down!';
$task->save();

// Create a new task
Task::create([
    'title' => 'Write article'
]);

// Delete a task
Task::find(1)->delete();
```

activerecord ORM 과 비슷.

Laravel > Eloquent ORM

```
// Get all tasks by the author with an id of 1
$tasks = User::find(1)->tasks;

// Get the author of a task
$author = Task::find(5)->user()->username;

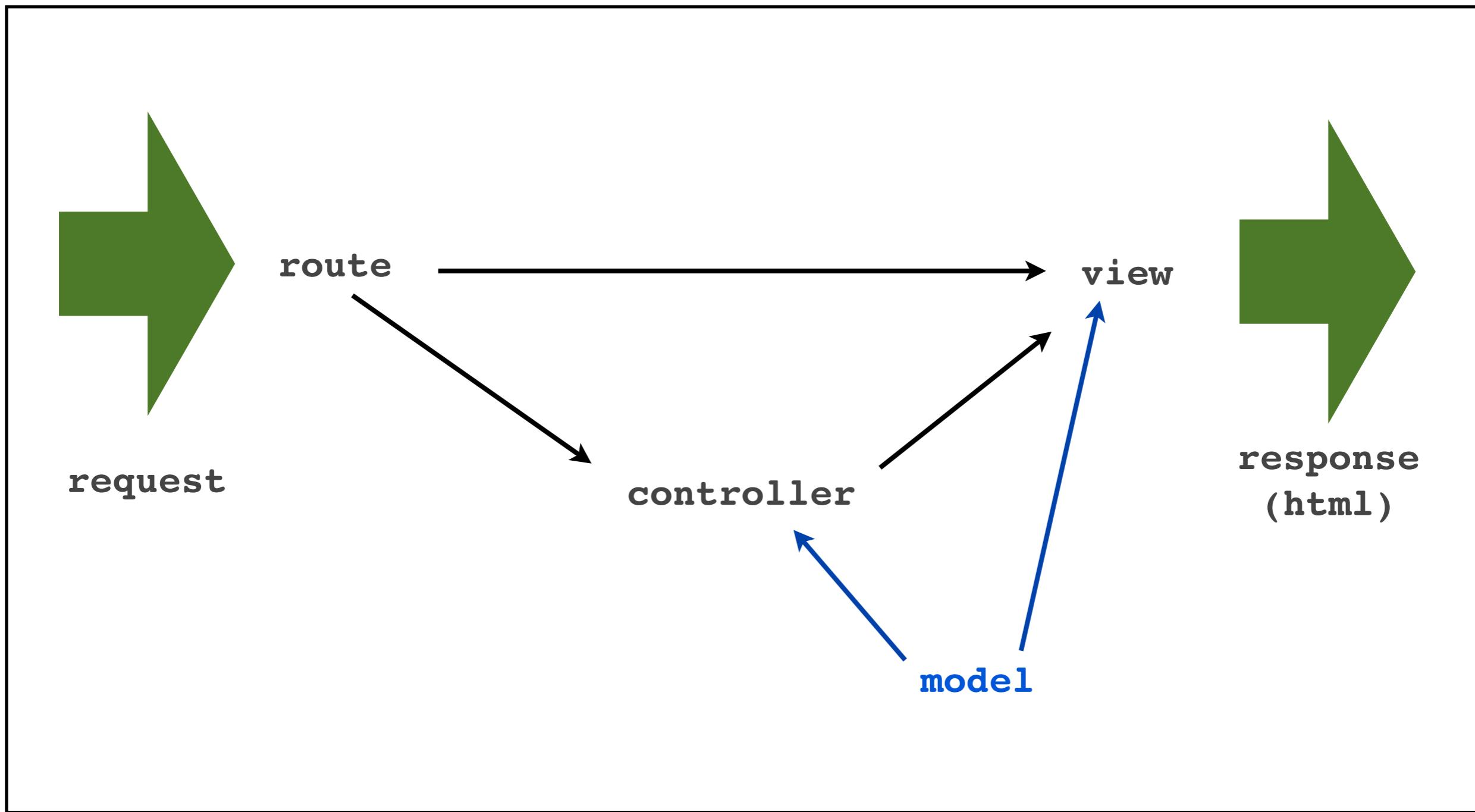
// Insert a new task by author
$task = new Task([ title: 'Go to store.' ]);
User::find(1)->tasks()->insert($task);
```

참조키

참조키로 연결되어 있는 경우 체이닝 처럼 사용 가능

Laravel > Route

`index.php`



Laravel > Route - basic

Method

example.com/users/1, users/profile.php view

```
Route::get('users/{id}', function ($id) {
    // find the user
    $user = User::find($id);

    // display view, and pass user object
    return View::make('users.profile')
        ->with('user', $user);
});
```

callback

resource

```
Route::get('users/{id}', 'Users@show');
```

어렵지 않아요

Laravel > Route - controller

Controllers/Users.php

```
class UsersController extends Controller {  
    /**  
     * Display the specified resource.  
     */  
    public function show($id)  
    {  
        // find the user  
        $user = User::find($id);  
  
        // display view, and pass user object  
        return View::make('users.profile')  
            ->with('user', $user);  
    }  
}
```

실제로는 컨트롤러를 쓰게 됩니다.

Laravel > Blade template

```
<ul>
    <?php foreach($people as $p) : ?>
        <li><?php echo $p; ?></li>
    <?php endforeach; ?>
</ul>
```



```
<ul>
    @foreach($people as $p)
        <li>{{ $p }}</li>
    @endforeach
</ul>
```

간단한 형식

Laravel > Blade template

```
// views/layout.blade.php
<!doctype html>
<html>
<head>
    <title></title>
</head>
<body>
    <div class="container">
        @yield('container')
    </div>
</body>
</html>
```

```
// views/home/hello.blade.php
@layout('layout')

@section('container')
    <h1>Hello, {{ $user->name }} !</h1>
@endsection
```

same!

blade template 는 심플하고 가볍습니다.

Laravel > Mail - config

```
// app/config/mail.php

return array(
    'host' => 'smtp.example.com',
    'port' => 2525,
    'from' => array('address' => null, 'name' => null),
    'encryption' => 'tls',
    'username' => null,
    'password' => null,
);
```

설정은 간편하게. SMTP 기본

Laravel > Mail - usage

```
Route::get('/', function()
{
    $user = User::find(1);
    $data = [ 'user' => $user ];

    // email view, data for view, closure to send email
    Mail::send('emails/welcome', $data,
        function($message) use ($user)
    {
        $message
            ->to($user->email)
            ->subject('Welcome Bieber Fan!')
            ->attach('images/bieberPhoto.jpg');
    });
}

return 'Welcome email sent!';
});
```

사용도 간단합니다.

Laravel > Queue for job

Supported: "sync", "beanstalkd", "sqS", "iron"

```
// app/config/queue.php

'connections' => array(
    'beanstalkd' => array(
        'driver' => 'beanstalkd',
        'host'   => 'localhost',
        'queue'  => 'default',
    ),
),
```

Job queue 를 간단하게 설정할 수 있습니다.

Laravel > Mail - basic

```
Route::get('/', function()
{
    $user = User::find(1);
    $data = [ 'user' => $user ];

    // email view, data for view, closure to send email
    Mail::send('emails/welcome', $data,
        function($message) use($user)
    {
        $message
            ->to($user->email)
            ->subject('Welcome Bieber Fan!')
            ->attach('images/bieberPhoto.jpg');
    });
}

return 'Welcome email sent!';
});
```

Laravel > Mail - queue

```
Route::get('/', function()
{
    $user = User::find(1);
    $data = [ 'user' => $user ];

    // email view, data for view, closure to send email
    Mail::queue('emails/welcome', $data,
        function($message) use($user)
    {
        $message
            ->to($user->email)
            ->subject('Welcome Bieber Fan!')
            ->attach('images/bieberPhoto.jpg');
    });
}

return 'Welcome email sent!';
});
```

queue!

sending time 을 간단하게 줄입니다.

Laravel > loc

Inversion Of Control

클래스 의존성을 관리하는 강력한 도구

DI - 의존성 삽입(Dependency injection)을 통해서

하드코딩된 클래스간 의존성을 제거하는 하나의 방법

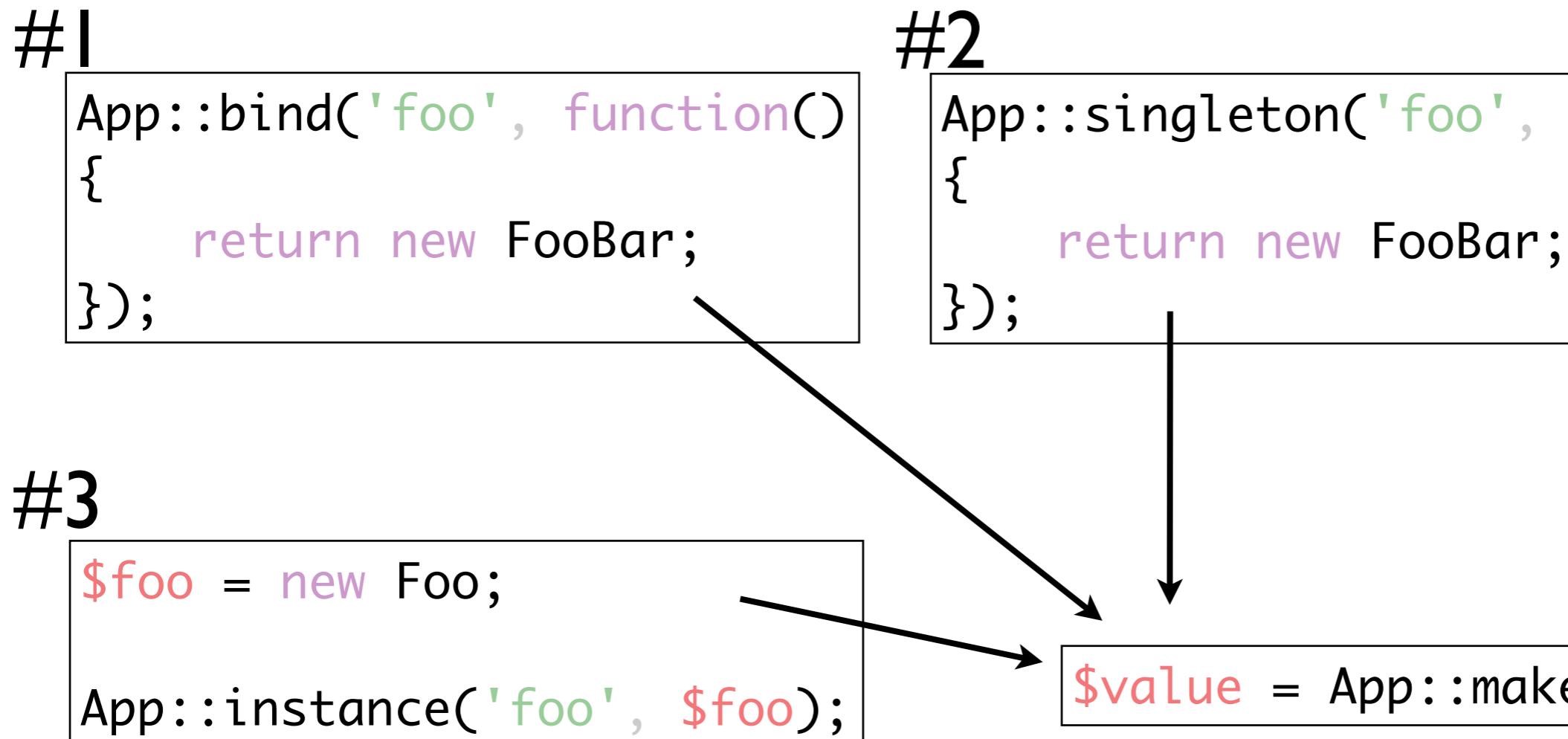
대신에 클래스의 의존성은 런타임시에 삽입되어

강력한 유연성을 제공하게 되고, 이로 인해

의존성 구현이 쉽게 변경될 수 있게 됨을 이야기 합니다.

뭔말이냐? - 내가 번역했지만 어렵다.

Laravel > loc



클로저를 통한 바인딩이 가능

Laravel > loc

```
class UserController extends BaseController {  
  
    public function __construct(UserRepositoryInterface  
$users)  
    {  
        $this->users = $users;  
    }  
  
}
```

```
App::make('UserRepositoryInterface', 'DbUserRepository');
```

컨트롤러에서 사용된 예 - DI

Laravel > loc

```
class User  
{  
    public  
    $users)  
}  
}
```

WHY?!?!

Interface

설계 관점에서 Testable이 핵심

```
App::make(
```

```
sitory');
```

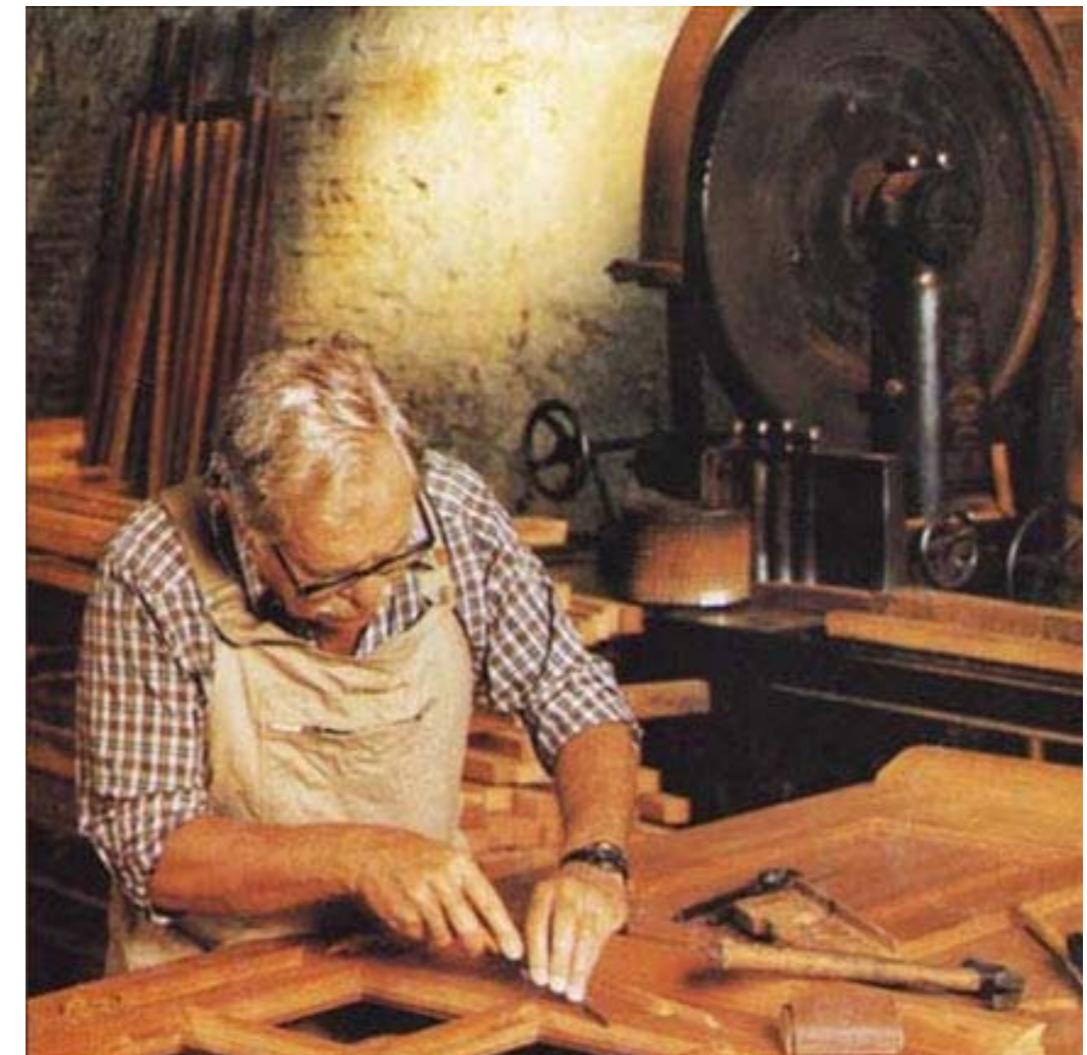
컨트롤러에서 사용된 예 - DI

Laravel > Artisan CLI

CLI : Command Line Interface

: 커맨드 라인으로 명령어 수행.

1. queue 핸들링
 2. DB 마이그레이션, seeding
 3. 컨트롤러 생성..
- 기타 등등



다양한 명령어를 통해서 개발편의를 향상시켜줍니다

Laravel > 실제적 문제들

이미 기존에 존재하는 시스템.



어느 세월에 바꾸나?



디비 마이그레이션 :<http://mwb.thinkers.it/>



Laravel > 유용한 github project

처음 시작시 예제

laravel4-starter-kit

Laravel-4-Generators

twitter bootstrap 기반

Laravel-4-Bootstrap-Starter-Site

핵심 개념 이해에 도움

fideloper.com

Laravel > IDE 지원

아직 많이 미흡합니다.

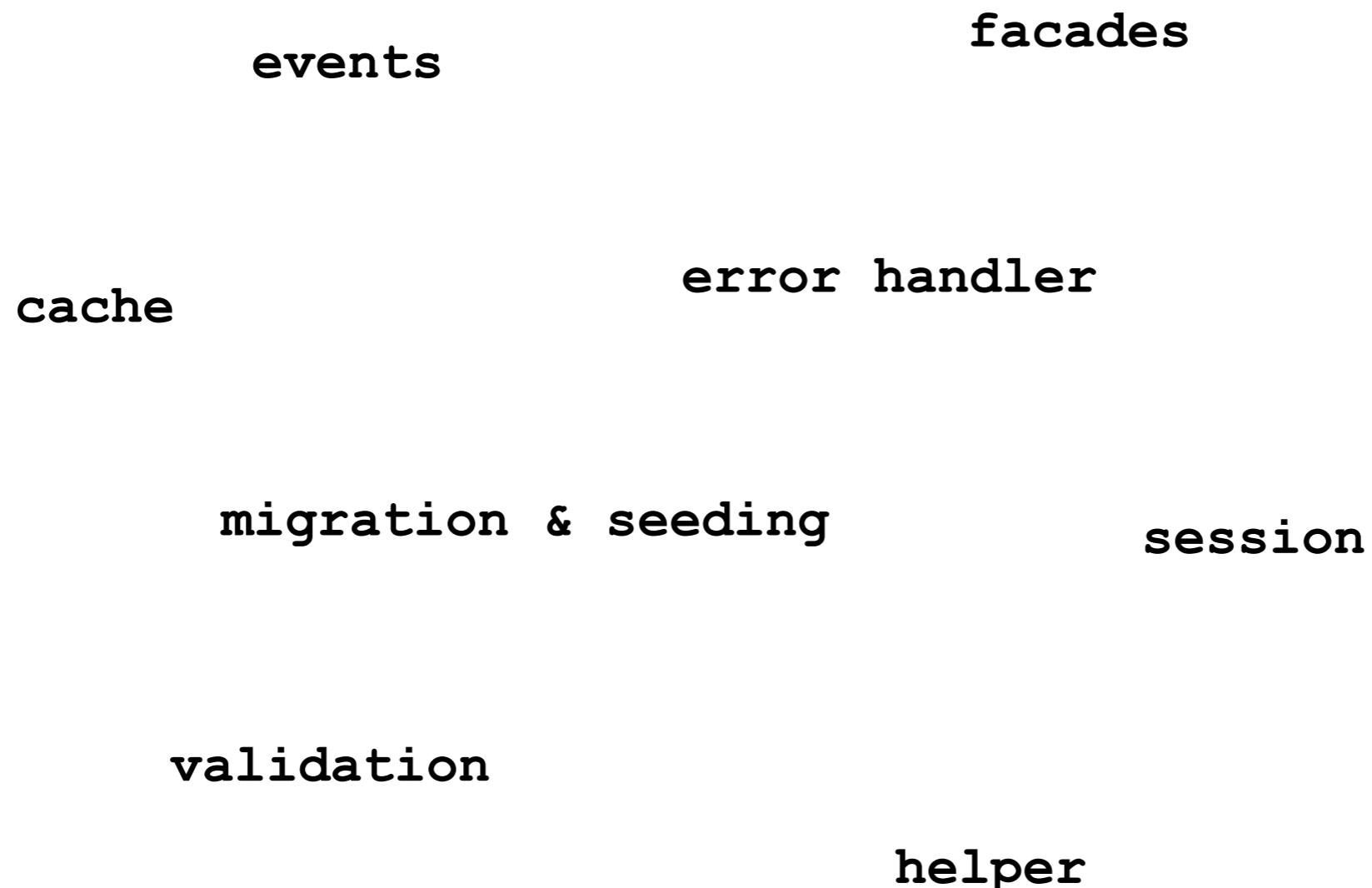


phpstorm

[laravel4-phpstorm-helper](#)

Sublime Text

Laravel > 아직 남은 이야기



Laravel > 아직 남은 이야기

cache



.on

va

정리

Laravel 은 쓸만한 프레임워크



최근 php 프레임워크중에 HOT



우리도 한번 배워 봅시다



근데 한글이 아니잖아..



아마 우린 안될꺼야

한글! - 그래서 준비했습니다.

한글 매뉴얼

머리말(PREFACE)	
소개(Introduction)	✓
빠른시작(Quickstart)	✓
기여(contributing)	✓
시작하기(GETTING STARTED)	
설치(Installation)	✓
설정(configuration)	✓
요청 사이클(Request Lifecycle)	✓
라우팅(Routing)	✓
요청 & 입력(Requests & Input)	✓
뷰 & 응답(Views & Responses)	✓
컨트롤러.Controllers)	✓
에러 & 로그(Errors & Logging)	✓
더 배우기(LEARNING MORE)	
캐싱(Cache)	✓
이벤트(Events)	✓
파사드(Facades)	✓
폼 & HTML	✓
헬퍼(Helpers)	✓
IoC 컨테이너(IoC Container)	✓
지역화(Localization)	✓

소개

- Laravel 철학
- Laravel 배우기

Laravel 철학

Laravel은 풍부하고 우아한 구문을 제공하는 웹어플리케이션 프레임워크입니다. 우리는 개발이 즐거워야 하고, 진심으로 성취감을 주는 창의적인 체험이 되어야 한다고 믿습니다. Laravel은 인증, 라우팅, 세션, 캐싱같이 대부분의 웹 프로젝트에서 사용되는 일반적인 작업들을 구현하여 개발할때 이러한 고통을 없애려고 시도합니다.

Laravel은 개발과정이 어플리케이션의 희생없이 개발자에게 즐겁도록 하는것이 목표입니다. 행복한 개발자가 최상의 코드를 만듭니다.

Laravel 배우기

Laravel을 배우는 가장 좋은 방법은 매뉴얼 전체를 꼼꼼히 읽어보는 것 입니다. 이 가이드는 프레임워크의 모든 면과 어떻게 어플리케이션에 적용하는지 상세히 나와 있습니다.

이 가이드 외에도 [책 페이지](#)에 있는 몇몇의 Laravel 책들을 확인해 볼수도 있습니다. 이 책들은 프레임워크를 배우는데 훌륭한 부가적인 리소스를 제공합니다.

저도 번역했습니다. (자랑질)

laravel-korea.org

정리 - 다시...

간단한 설정법

최근 php 프레임워크중에 HOT

다양한 지원 - mail, cache, queue, session

restful

composer, ioc, di, facade - 최신 기법

생각보다 쉬워요.

composer 결합

정리 - 다시...

간단한 설정법

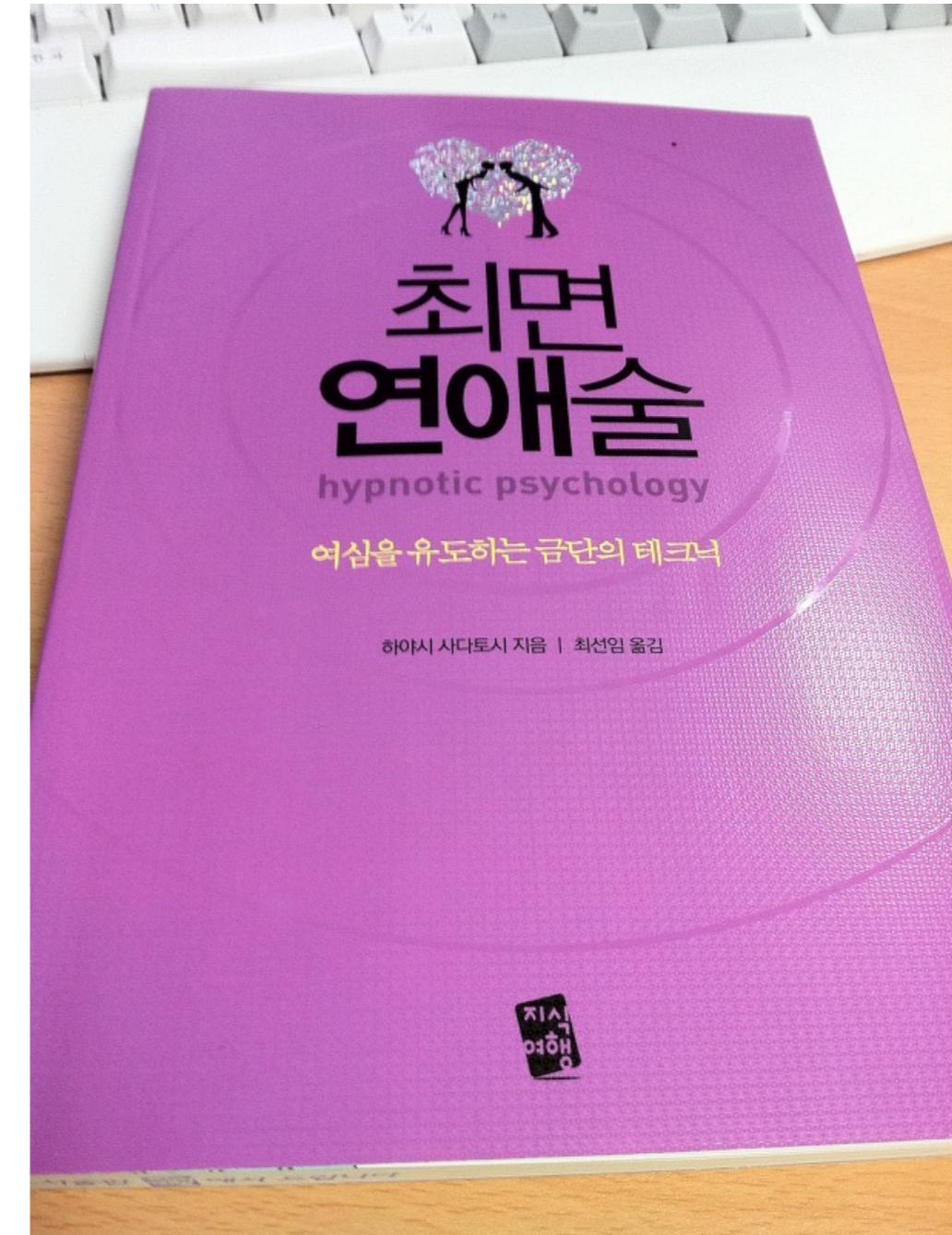


생각보나 귀워요.

composer 결합

결론

빠져 봅시다!
Laravel



참고

1. Kenny Meyers - Per Your Request, Laravel 4

<http://www.youtube.com/watch?v=bRyOrtCim70>

2. laravel queue

<http://fideloper.com/ubuntu-beanstalkd-and-laravel4>

3. Why Laravel is Taking the PHP Community by Storm

<https://tutsplus.com/tutorial/why-laravel-is-taking-the-php-community-by-storm>

4. 구글+ Laravel 커뮤니티 그룹

<https://plus.google.com/communities/106838454910116161868>

5. Laravel 공식 포럼 <http://forums.laravel.io>

6. Laravel IRC irc.freenode.net or chat.freenode.net #laravel

7. 한국 Laravel 사용자 모임 <http://laravel-korea.org>

참고

7. Mail::queue and Mail::later <http://vimeo.com/64779946>
8. Github project
9. slide share.
10. google image search.

꿋

감사합니다.

안정수
@findstar