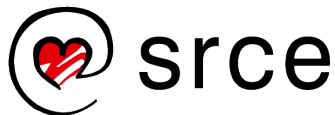


Zašto smo odabrali Laravel (4)?

Denis Stančer



Sveučilište u Zagrebu
Sveučilišni računski centar



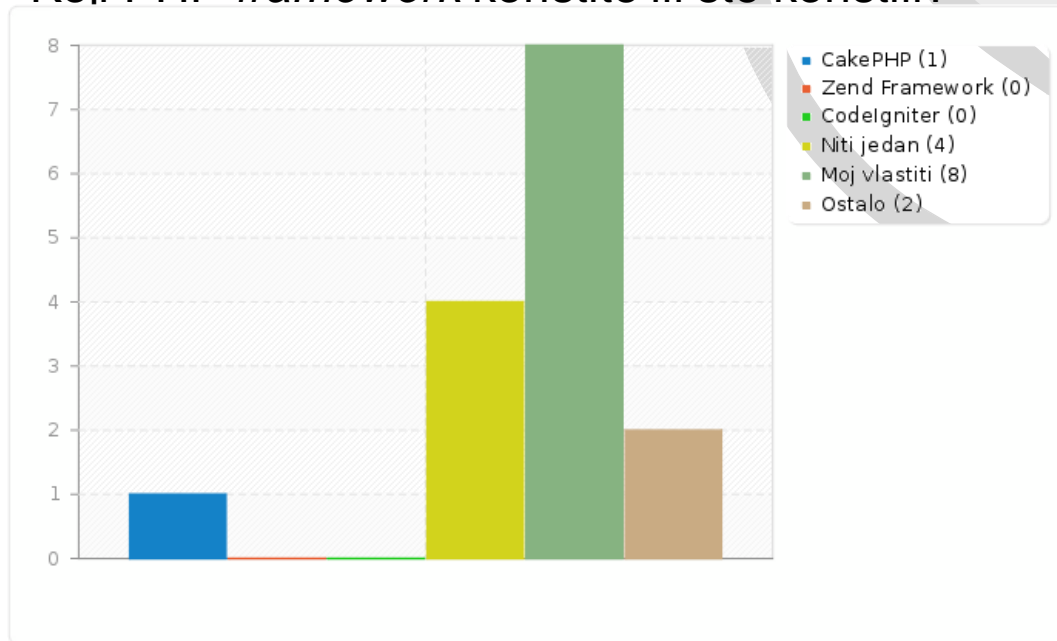
srce
otvoreni pristup

Prije *framework-a*

- Razvijate web aplikacije od ranih početaka (kraj XX stoljeća)
 - Perl – CGI
 - PHP (3.0 - 6/1998, 4.0 - 5/2000, 5.0 - 7/2004, 5.3 - 6/2009)
- Tijekom vremena sami razvijete elemente *frameworka*
 - Prednosti:
 - Brži razvoj
 - Neka vrsta standarda
 - Nedostaci:
 - Još uvijek velika količina *spaghetti* kôda
 - Pojedini *developer* ima svoj *framework*
 - Ne razvijaju se svi jednako brzo
 - Nama pravovremenih sigurnosnih zakrpi

U Srcu

- Koji PHP *framework* koristite ili ste koristili?

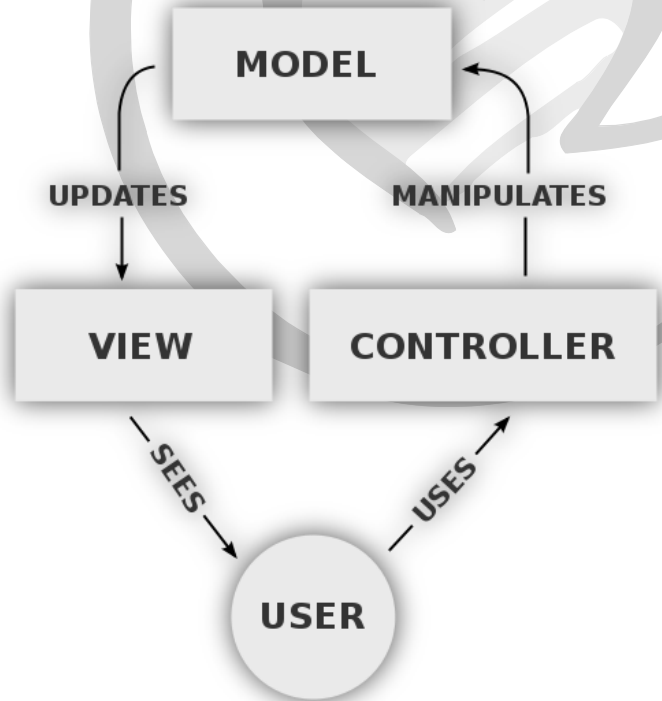


Zašto *framework*?

- Brži razvoj
- Standardizirana organizacija kôda
- Pojednostavljeno
 - Pristupu bazi/bazama
 - Zaštita od osnovnih sigurnosnih propusta
 - Modularnost
- Razmjena gotovih rješenja među *developerima*
 - *Copy/paste* ili *Composer*
- U MVC *framework*-u razdvojen HTML/JS od PHP-a
- U konačnici - bolja suradnja unutar tima = efikasniji razvoj i održavanje

MVC – Model-View-Controller

- Programski predložak kojim se komunikacija s korisnikom dijeli na tri dijela:
 - data model: podaci
 - najčešće baza
 - user interface: prikaz stanja u modelu
 - najčešće templating engine
 - bussines model: šalje naredbe modelu



Koji *framework* odabrati?

- Koji su najpopularniji?
- Koji imaju mogućnosti koje nama trebaju?
- Popis općih kriterija
 - *Composer*
 - *ORM*
 - Testna okruženja
 - Migracije i *seeding*
 - *Templating engine*
 - *Bootstrap*
 - *Git*
 - Kvaliteta dokumentacije
 - Stanje zajednice: forumi, članci, konferencije,...

Koji *framework* odabrati? (2)

- Popis specifičnih kriterija
 - Mali (rijetko srednje veliki) projekti
 - *simpleSAMLphp*: jednostavno uključivanje
 - Pristup odvojenim (i različitim) bazama
 - Generiranje PDF dokumenata
 - *Upload* datoteka

Opći kriteriji - *Composer*

- Alat za razrješavanje ovisnosti paketa
 - Zamjena za PEAR
 - Dostupan od 01.03.2012.
 - *Command line* alat
 - Konfigurira se pomoću jedne datoteke
 - **composer.json**
 - Uključuje *autoloader* koji omogućuje povezivanje raznih izvora klasa
 - *De facto* standard
 - Oslanja se na GitHub
 - Ali za pretraživanje posebna stranica
 - **<https://packagist.org/>**

Opći kriterij: ORM

- Object-relational mapping
- PHP je objektni jezik
- SQL DBMS-i nisu objektno orijentiran
- ORM omogućava pretvorbu objekata u skupinu jednostavnih naredbi za SQL
- Najpoznatiji za PHP:
 - Doctrine <http://www.doctrine-project.org/>
 - RedBeanPHP <http://www.redbeanphp.com/>
 - Propel <http://propelorm.org/>
- CakePHP, Symfony, CodeIgniter, Laravel imaju svoj ORM baziran na PHP ActiveRecord-u

Opći kriterij: Testna okruženja

- Automatizirano testiranje je važan dio razvoja i održavanja aplikacija
- Postoji filozofija razvoja koja uključuje pisanje testova kao primarnu aktivnost – *Test Driven Development* – TDD
- U PHP svijetu su najrašireniji
 - PHPUnit (<https://phpunit.de/>)
 - Codeception (<http://codeception.com/>)
 - Behat (<http://docs.behat.org/>)
 - PHPSpec (<http://www.phpspec.net/>)

Opći kriterij: Migracije i *seeding*

- *Migration* je programatski opis strukture baze podataka
 - Poželjan jer uz aplikaciju se bilježi i struktura baze
 - U sustav za verzioniranje se pohranjuje svaka izmjena u strukturi baze
 - Ne ovisi o vrsti baze
- *Seeding* je programatski upis podataka u bazu
 - Poželjan za punjenje inicijalnih podataka
 - Može se iskoristiti i za punjenje razvojnih podataka
 - Podaci se pohranjuju u sustav za verzioniranje
 - Ne ovisi o vrsti baze
- Ako su povezani – efikasno se može vratiti stanje na početak za potrebe testiranja

Opći kriterij: *Templating engine*

- PHP je *templating* jezik
- Prilikom ispreplitanja s HTML-om dokument postaje slabo čitak

```
<h1><?=$title?></h1>
<ul>
  <?php foreach ($items as $item) {?>
    <li><?=$item?></li>
  <?php } ?>
</ul>
```

- Mnogi *templating* sustavi povećavaju čitljivost

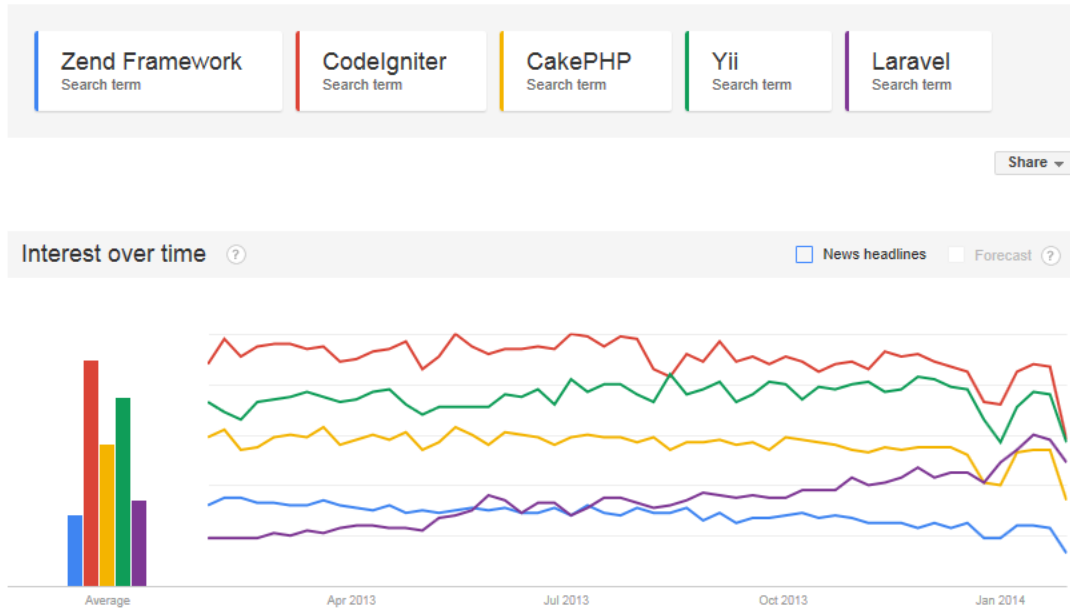
```
<h1>{{ $title }}</h1>
<ul>
  {% for $item in $items %}
    <li>{{ $item }}</li>
  {% endfor %}
</ul>
```

Posebni kriteriji

- Uglavnom mali projekti
 - Gotovo svi projekti su mali
 - U planu je jedan-dva srednje velične
- *simpleSAMLphp*
 - Autentikacija gotovo isključivo kroz AAI@EduHr
- Pristup odvojenim (i različitim) bazama
 - Često se kombiniraju podaci iz različitih sustava
 - Npr. MySQL i Microsoft SQL
- Generiranje PDF dokumenata
 - Gotovo sve aplikacije generiraju izvještaje ili obrasce
- *Upload* datoteka
 - Za određene aplikacije se podaci dostavljaju *uploada*-om datoteka određenog formata

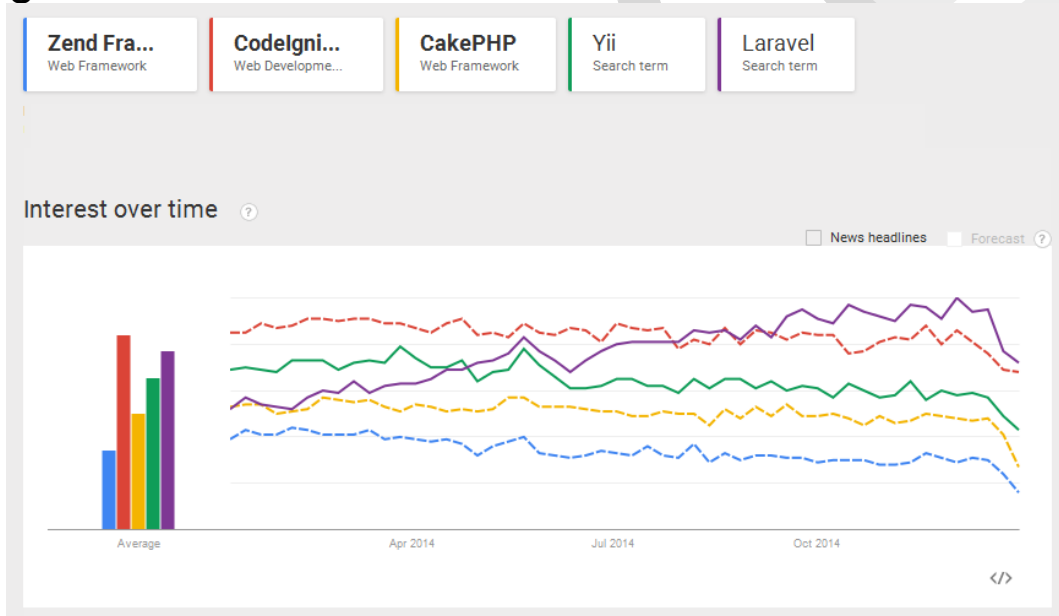
Kandidati (1)

- Google trends - 2013



Kandidati (2)

- Google trends - 2014



Kandidati (3)

- Analiza pojedinih blogova koji preporučaju PHP razvojna okruženja (kraj 2013.-te)

Okruženje	Broj preporuka
Yii	11
CodeIgniter	10
CakePHP	9
Zend	9
Symfony	8
Laravel	6
FuelPHP	3
Kohana	2

Kandidati (4)

- Wikipedia

Project	Start Date	Language	MVC framework	i18n & L10n?	ORM	Testing framework(s)	DB migration framework(s)	Security framework(s)	Template framework(s)	Caching framework(s)	Form validation framework(s)	Scaffolding
CakePHP	2005-08	PHP >= 5.2	Yes	Yes	Active record pattern (CakePHP 1.x), data mapper pattern (CakePHP 2.x)	Unit tests, object mocking, fixtures, code coverage, memory analysis with SimpleTest and XDebug PHPUnit (cakephp 2.0)	Yes	ACL-based	Themes, layouts, views, elements	Memcache, Redis, XCache, APC, File	Validation, security	Yes
CodeIgniter	2006-01	PHP >= 5.1	Yes	Mostly	Third party only	Ready for next release	Yes	Yes	Yes	Yes	Yes	No ^[27]
FuelPHP	2010-10	PHP >= 5.3.x	MVC, HMVC	Yes	Yes	PHPUnit	Yes	Yes, Plugins available	Yes, Plugins available	File, Redis, Memcache, more	Yes	Yes
Laravel	2011-06	PHP >= 5.3.7	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	?
Symfony 2	2010	PHP >= 5.3.3	Yes	Yes	Propel, Doctrine (YAML)	Yes	Plugin exists	Yes	PHP, Twig	Yes	Yes	Yes
Yii	2008-01	PHP >= 5.1.0	Yes	Yes	Data Access Objects (DAO), Active Record Pattern, Plugins (incl. Doctrine 2.0)	PHPUnit, Selenium	Yes	ACL-based, RBAC-based, plugins	PHP-based, PRADO-like, plugins	APC, Database, eAccelerator, File, Memcache, Redis, WinCache, XCache, Zend Platform	Yes	Yes
Zend Framework 2	2010-06	PHP >= 5.3.3	Yes	Yes	Table and row data gateway and Doctrine 2.0 for Zend Framework 2.0	Unit tests, PHP Unit or other independent	Yes	ACL-based	Yes	APC, Database, File, Memcache, Zend Platform	Yes	Yes

Analiza

- Odabrani su
 - CakePHP 2.4.x
 - CodeIgniter 2.1.x
 - Laravel 4.1.x
 - Symfony 2.4.x
 - Yii 1.1
 - ZendFramework 2.x
- Za svaki od odabranih je bilo potrebno napraviti jednostavan web site kroz koji se provjeravaju kriteriji

Analiza (2)

- Svaki od članova tima je dobio čisti VM sa standardnom instalacijom *Debian Linux-a*
- Svaki član tima je nakon implementacije web-a u određenom *framework-u* održao 45-minutnu prezentaciju na kojoj je prošao kroz sve kriterija koji su navedeni
- Na prezentacijama se raspravljalo o razlikama i sličnostima
- Na kraju su sudionici (koji su htjeli) sudjelovali u glasanju za odabir *framework-a*

Testni web

- Testni web site je morao imati sljedeće dijelove
 - Tri područja dostupna samo određenim vrstama korisnika
 - Javni dio dostupan svima
 - Dio za autenticirane korisnike
 - Dio za administratore
 - Autentikacija kroz AAI@EduHr
 - Primarna baza MySQL
 - Sekundarna baza (isto MySQL) – samo *read-only* pristup
 - Osnovne CRUD akcije na primarnoj bazi

Usporedba

Naziv	Composer	ORM	Test	Migracije	Templating engine	Boot strap	Git	Simple SAML php
Yii	Ne	Ne	Da Codeception	Ne	Ne	Ne	Ne	Da
CodeIgniter	Ne	Ne	Da	Ne	Ne	Ne	Ne	Da
CakePHP	Da	Da	Da	Plug-in	Da	Da	Ne	Da
Symfony	Da	Da	Da Codeception	Plug-in	Da	Da	Da	Da
Laravel	Da	Da	Da Codeception	Da	Da	Da	Da	Da
Zend	Da	Da	Da Codeception	Ne	Ne	Da	Ne	Da

Odabran



laravel

- Laravel je dobar za male i srednje projekte (većina u Srcu)
- Symfony je dobar za velike projekte

Prednost Laravel-a

- Relativno nov pa se brzo implementiraju noviteti uvedeni u sâm PHP (npr. Laravel 4.2 zahtjeva PHP 5.4)
- Većina jezgre su gotovi (Symfony) moduli pa se okruženje može brže razvijati
- Prihvaćanje novih tehnologija
- Otvorenost prema drugim komponentama (koristeći *Composer*)
- Osmišljen kako bi se zaštitile osjetljive informacije (lozinke i ključevi)

Laravel 4



Povijest

- Autor: Taylor Otwell
- Laravel 1 – 6/2011
 - Eloquent ORM, autentikacija, lokalizacija, caching, sessions, views, moduli i lib-ovi
- Laravel 2 – 11/2011 – privatna inačica
 - Kontroleri – pravi MVC, blade templating engine, Inversion of Control
- Laravel 3 – 2/2012
 - Unit testing, artisan, DB migracije, paketni sustav – bundles
- Laravel 4 – 5/2013
 - Napisan iznova, umjesto bundlova – Composer, DB seeding, mail, fasade
- Laravel 5 – 2/2015
 - Nova struktura direktorija, middleware, method injection, scheduler,...

Održavanje

- Baziran na gotovim modulima
 - 13 Symfony modula
 - Carbon
 - Monolog
 - Doctrine
 - Predis
- Relativno redovita objava novih verzija



Instalacija - preduvjeti

- PHP
 - Laravel 4.1 - PHP \geq 5.3.7
 - Laravel 4.2+ - PHP \geq 5.4.0
 - MCrypt PHP Extension
 - PHP JSON extension
- Composer (<https://getcomposer.org/>)

```
$ curl -sS https://getcomposer.org/installer | php
```

```
$ mv composer.phar /usr/local/bin/composer
```

Instalacija

```
$ composer create-project laravel/laravel <naziv_projekta> <verzija>
```

- Struktura 4.2 (spremna za Git)

app/

config/

konfiguracijske datoteke

controllers/

kontroleri

databases/

migracijske datoteke

lang/

datoteke s prijevodima sistemskih poruka

models/

datoteke s definicijama modela

storage/

cache, logovi, ...

tests/

datoteke za testiranje

views/

prezentacijske datoteke

bootstrap/

autoload datoteke

public/

javno dostupne datoteke – JavaScript, CSS, slike

vendor/

datoteke svih modula

artisan

konfiguracijski alat

composer.json

konfiguracija modula

Postupak instalacije

```
$ composer create-project laravel/laravel <app_dir> 4.2
$ cd <app_dir>
$ composer update
$ cd app
$ touch storage/logs/laravel.log
$ chgrp -R <apache_group> storage
$ chmod -R g+w storage
$ ln -s <app_dir>/public <apache_document_root>/<app_dir>
```

- Po potrebi u datoteku <app_dir>/public/.htaccess dodati redak `RewriteBase /<app>` (iza # Handle Front Controller...)

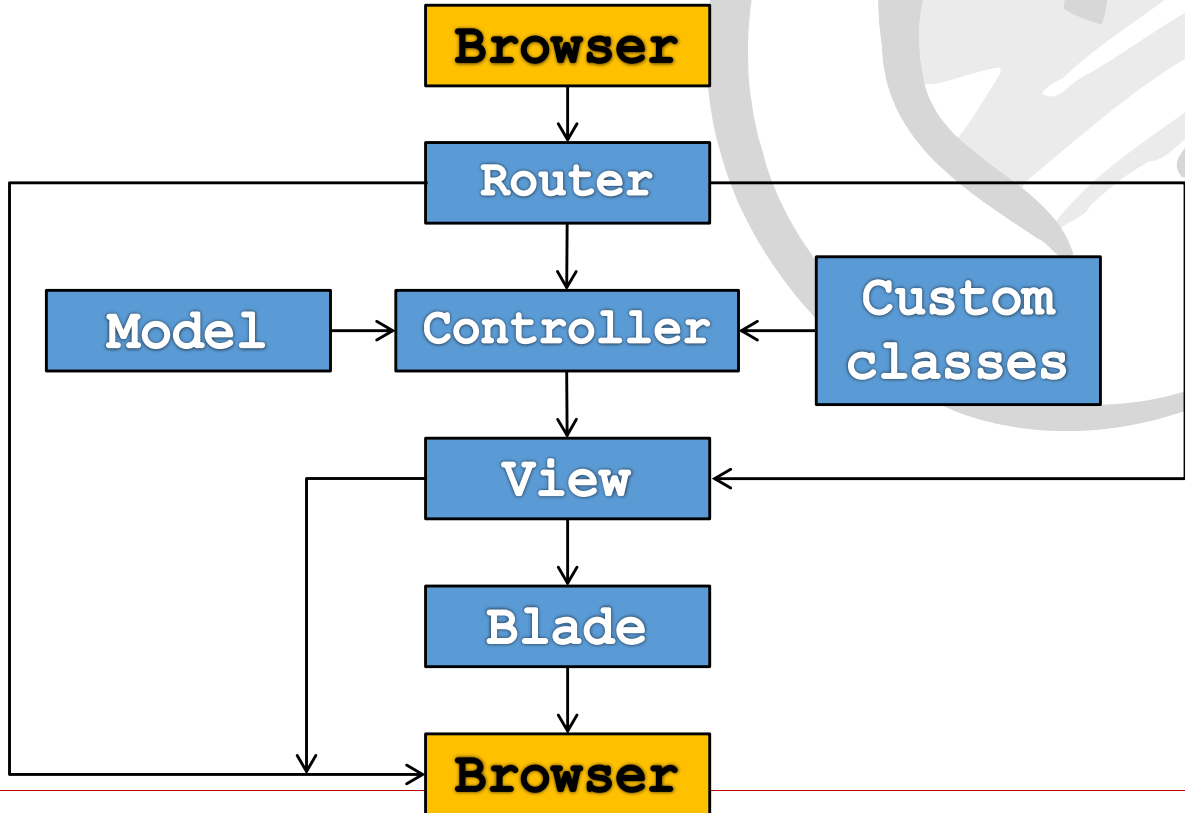
Konfiguracija Apache servera

- Potrebno je instalirati *rewrite* modul
`a2enmod rewrite`
- Za putanju u kojemu će biti povezani **public** direktorij obavezno dozvoliti čitanje iz **.htaccess** datoteke
`AllowOverride All`
- Preporuka – koristiti HTTPS

Više aplikacija

- Za svaku instalaciju je potrebno obaviti punu instalaciju
- Prednosti
 - svaka aplikacija je neovisna o verziji paketa ostalih aplikacija
 - nadogradnju paketa radi održavatelj/autor aplikacije neovisno o ostalima
- Nedostatci
 - svaka instalacija je velika oko 30MB
 - ako su održavatelji/autori nemarni neće se nadograditi sve aplikacije

Laravel - MVC





Controller

MVC - Putanje

- GET putanja

```
Route::get('/', function()
{
    return 'Hello World';
});
```

- POST putanja

```
Route::post('foo/bar', function()
{
    return 'Hello World';
});
```



MVC – Putanje (2)

- Putanja s parametrima

```
Route::get('user/{id}', function($id = null)
{
    return 'User '.$id;
});
```

- Putanja s ograničenjima

```
Route::get('user/{id}/{name}', function($id, $name)
{
    //
})->where(array('id' => '[0-9]+', 'name' => '[a-z]+'));
```

MVC – Putanje (3)

- *RESTfull* definicija

```
Route::resource('resource', 'PhotoController');
```

- Daje putanje

URI	Metoda
GET HEAD photo	PhotoController@index
GET HEAD photo/create	PhotoController@create
POST photo	PhotoController@store
GET HEAD photo/{photo}	PhotoController@show
GET HEAD photo/{photo}/edit	PhotoController@edit
PUT photo/{photo}	PhotoController@update
DELETE photo/{photo}	PhotoController@destroy

MVC – Putanje (4)

- Implicitna definicija putanje
`Route::controller('test', 'TestController');`
- Za metode definirane u kontroleru
`getIndex, getEdit, postEdit`
- Daje putanje

URI	Metoda
GET HEAD test/index/{one?}/{two?}/{three?}/{four?}/{five?}	TestController@getIndex
GET HEAD test	TestController@getIndex
GET HEAD test/edit/{one?}/{two?}/{three?}/{four?}/{five?}	TestController@getEdit
POST test/edit/{one?}/{two?}/{three?}/{four?}/{five?}	TestController@postEdit
GET HEAD POST PUT PATCH DELETE test/{_missing}	TestController@missingMethod

MVC – Filteri

- Definiraju se u `app/filters.php` datoteci

```
Route::filter('auth', function()
{
    if (Auth::guest())
        return Redirect::guest('login');
});
```

MVC – Filteri (2)

- Primjenjuju se na putanje

```
Route::get('user', array('before' => 'auth|csrf',  
function()  
{  
    return 'You are authenticated and protected!';  
}));
```

- Najčešće se grupiraju putanje

```
Route::group(array('before' => 'auth'), function()  
{  
    Route::get('/', function(){...});  
})
```



Model

MVC – Konfiguracija

app/config/database.php

```
'mysql' => array(  
    'driver'      => 'mysql',  
    'host'        => 'localhost',  
    'database'    => 'database',  
    'username'    => 'user',  
    'password'    => 'pwd',  
),
```

MVC – Eloquent

- Za definiciju tablica
`php artisan migrate:make create_posts_table`
- Stvori se nova datoteka u `app/database/migrations`

```
public function up()
{
    Schema::create('posts', function(Blueprint $table)
    {
        $table->increments('id');
        $table->string('name');
        $table->text('body');
        $table->timestamps(); //created_at, updated_at
    });
}
```

MVC – Model

- Definicije modela se pohranjuju u `app/models/`
- Već postoji `User.php` kao osnova za izradu prilagođenih modela
- Veza između modela i tablice se ostvaruje naredbom
`protected $table = 'posts';`
- Veza između modela i tablice u drugoj bazi
`protected $connection = 'drugaBaza';`
- Polja koja smiju popunjavati korisnici
`protected $fillable = array('name', 'body');`
- Polja čije vrijednosti se ne smiju vidjeti
`protected $hidden = array('password');`

MVC – Upiti

- Dohvati sve

```
$posts = Posts::all();
```

- Dohvati određeni

```
$posts = Posts::where('name', 'like', '%good%')  
->get();
```

```
$posts = Posts::where('name', 'like', '%good%')  
->orWhere('name', 'like', '%excell%')  
->get();
```

- Ručni način

```
$posts = DB::table('posts')  
->select(DB::raw('count(*)', name'))  
->where('id', '>', 10)  
->groupBy('name')  
->get();
```

MVC – Upiti (2)

- Insert

```
$user = new User;  
$user->name = 'John';  
$user->save();  
$insertedId = $user->id;
```

- Update

```
$user = User::find($id);  
$user->name = 'Matt';  
$user->save();
```



MVC – Upiti (3)

- Delete

```
$user = User::find($id);  
$user->delete();
```

- Brisanje većeg broja zapisa

```
$affectedRows = User::where('votes', '>', 100)  
->delete();
```

- Čisti SQL

```
$posts = DB::select(DB::raw("  
    SELECT count(*) as Count, name  
    FROM posts  
    WHERE some_col = 1"));
```

MVC – Upiti (4)

- Soft delete
- U definiciji modela navesti
`protected $softDelete = true;`
- U definiciji scheme dodati
`$table->softDeletes();`
- Kao bi se u tablici pojavio stupac `deleted_at`
- Svi upiti će zanemariti sve zapise koji imaju postavljenu vrijednost u `deleted_at` polju

MVC – *Caching*

- Pri dohvatu podataka navesti koliko minuta se pamte podaci

```
$posts = Posts::remember(10, 'posts_cache_key')  
->all();
```

- Općenito za bilo koji objekt

```
Cache::put('post_111', $post, 60);  
$post = Cache::get('post_111');
```

- Za brisanje

```
Cache::forget('posts_cache_key');
```


MVC – Mutatori

- Automatski se pozivaju na modelima

```
public function setProdDateAttribute($prod_date)
{
    $this->attributes['prod_date'] = $prod_date ?
    date('Y-m-d', (strtotime($prod_date))) : '';
}
```

- Za prikaz datuma se može koristiti modul **Carbon**

```
use Carbon\Carbon;
. . .
Carbon::setToStringFormat('d.m.Y');
```



View

MVC – *Templating engine*

- Po defaultu se koriste standardne `*.php` datoteke koje se nalaze u `app/views/` direktoriju
- Datoteke `*.blade.php` datoteke koje se nalaze u `app/views/` direktoriju su Laravel template datoteke
- Vlastiti *templating engine* – *blade*
- Standardni PHP uz dodatak nekoliko kontrolnih naredbi
- Varijable

`{{ $user }}` ili `{{{ $user }}}}`

MVC – Templating engine (2)

- Uvjeti

```
@if ()  
  ...  
@elseif ()  
  ...  
@endif  
  
@unless ()  
  ...  
@endunless
```

- Petlje

```
@for ($i = 0; $i < $len; $i++)  
  ...  
@endfor  
  
@foreach ($users as $user)  
  {{ $user->id }}  
@endforeach  
  
@while (true)  
  ...  
@endwhile
```

MVC – Templating engine (3)

- Struktura
- Glavna struktura se definira zasebno

```
@section('title')
```

```
    Laravel AAI@EduHr
```

```
@show
```

```
...
```

```
@yield('content')
```

- Koristi se u pojedinim stranicama

MVC – Templating engine (4)

- Uporaba

```
@extends('layouts.master')
```

```
@section('title')
```

```
    Naslov
```

```
@stop
```

```
@section('title', 'Naslov')
```

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

```
    Tijelo stranice
```

```
@stop
```



MVC – Templating engine (5)

- Makroi
 - Najčešće u zasebnoj datoteci
 - Potrebno ih je povezati s projektom (npr. u datoteci putanja)

```
require __DIR__.'/macros.php';
```

- Uporaba Bootstrapa

```
Form::macro('bstext',  
function ($name, $value, $class = "col-sm-9"){  
    return '<div class="'. $class. "'>'.  
        Form::text($name, $value,  
            $attributes = array('class' => 'form-control')).  
            '</div>';  
});
```

MVC – Povezivanje *Controller - Model*

- Podaci se proslijede prezentacijskom dijelu

```
public function index()
{
    $posts = Post::all();
    ...
}
```


MVC – Povezivanje *Controller* - *View*

- Podaci se proslijede prezentacijskom dijelu

```
public function index()
{
    $posts = Post::all();
    return View::make('posts.index')
        ->with(compact('posts'));
}
```

MVC – Povezivanje Controller – View (2)

- Podaci se prikažu

```
<ul>
```

```
@foreach($posts as $post)
```

```
    <li>$post->title ($post->answer_no)</li>
```

```
@endforeach
```

```
</ul>
```

Ostala svojstva

Provjera podataka

- Provjera vrijednosti u obrascima

```
$validation = Validator::make($input, $rules);  
if ($validation->passes()){...}  
if ($validation->fails()){...}
```

- Pravila se definiraju kao nizovi

```
public static $rules = array(  
    'hrEduPersonUniqueID' => 'required',  
    'localRole' => 'required|alpha');
```

- Upozorenja se nalaze u `app/lang/<jezik>/validation.php`

Provjera podataka (2)

- Velika rasprava o tome gdje bi trebalo pisati pravila
 - U *controller*-u
 - U *model*-u
 - U zasebnom servisu
- 37 vrsta provjera
- Najčešće

`required`

`alpha`

`min:<val>`

`between:<min>,<max>`

`not_in:<stg1>,<stg2>,<stg3>,...`

`regex:<pattern>`

`numeric`

`email`

`max:<val>`

`in:<stg1>,<stg2>,<stg3>,...`

Provjera podataka (3)

- Izrada vlastitog pravila za provjere

```
class CustomValidator extends Illuminate\Validation\Validator {  
  
    public function validateSrcediff($field, $value, $parameters){  
        $other = $parameters[0];  
        if(($value == 0) or ($this->data[$other] == 0)){ return true; }  
        return isset($this->data[$other]) and $value != $this-  
>data[$other];  
    }  
}  
  
Validator::resolver(  
    function($translator, $data, $rules, $message){  
        return new CustomValidator($translator, $data, $rules, $message);  
    });
```

Događaji

- Gdje ih definirati
 - Mali broj: `app/start/global.php`
 - Puno: `app/events.php` koji se *include*-a u `global.php`
- Definicija događaja koji se prati

```
Event::listen('auth.login',  
    function($user){  
        ...  
    }  
);
```

Događaji (2)

- Definicija događaja koji se prati kroz *handler* i uz prioritet

```
Event::listen(  
    'auth.login',  
    'LoginHandler',  
    10  
);
```

- Okidanje događaja

```
Event::fire('auth.login', [$arg1, $arg2]);
```


Zaključak



Jesmo li pogriješili?

- Nismo
 - Koristimo jedan *framework*
 - Koristimo javni *framework*
 - Nemamo dovoljno resursa za održavanje vlastitog *frameworka*
 - Nemamo dovoljan broj vrsta uporabe da bismo bili sigurni da je naš *framework* siguran (*battle tested*)
 - Koristimo jedan od najpopularnijih *frameworka*
 - Koristimo *framework* koji se brzo razvija i prati trendove

Jesmo li pogriješili? (2)

```
(fisset($this->student_data[$stanova] ['top_student']) && count
```

Developers Hrvatska

I takoooo... prijava stipendije 😊

March 13

and 53 ot

barem laravel koriste 😊

March 13 at 10:09am · Like · 24

sve pršti od izvrsnosti

March 13 at 10:14am · Like · 3

Mislio sam da je nekakva Ali da, barem je Laravel u pitanju... 😊

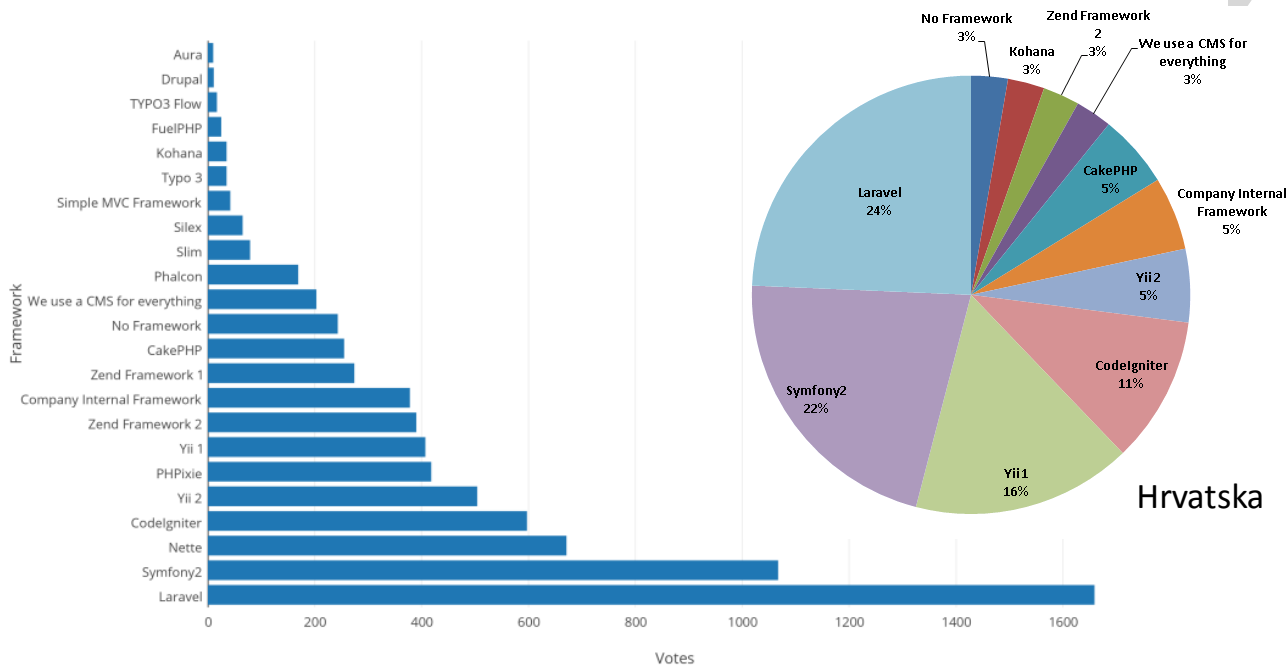
March 13 at 10:31am · Edited · Like

Mislio sam da je nekakva Ali da, barem je Laravel u pitanju... 😊

March 13 at 10:31am · Edited · Like

Najpopularniji PHP FW za poslovnu uporabu

PHP Framework Popularity at Work - SitePoint, 2015

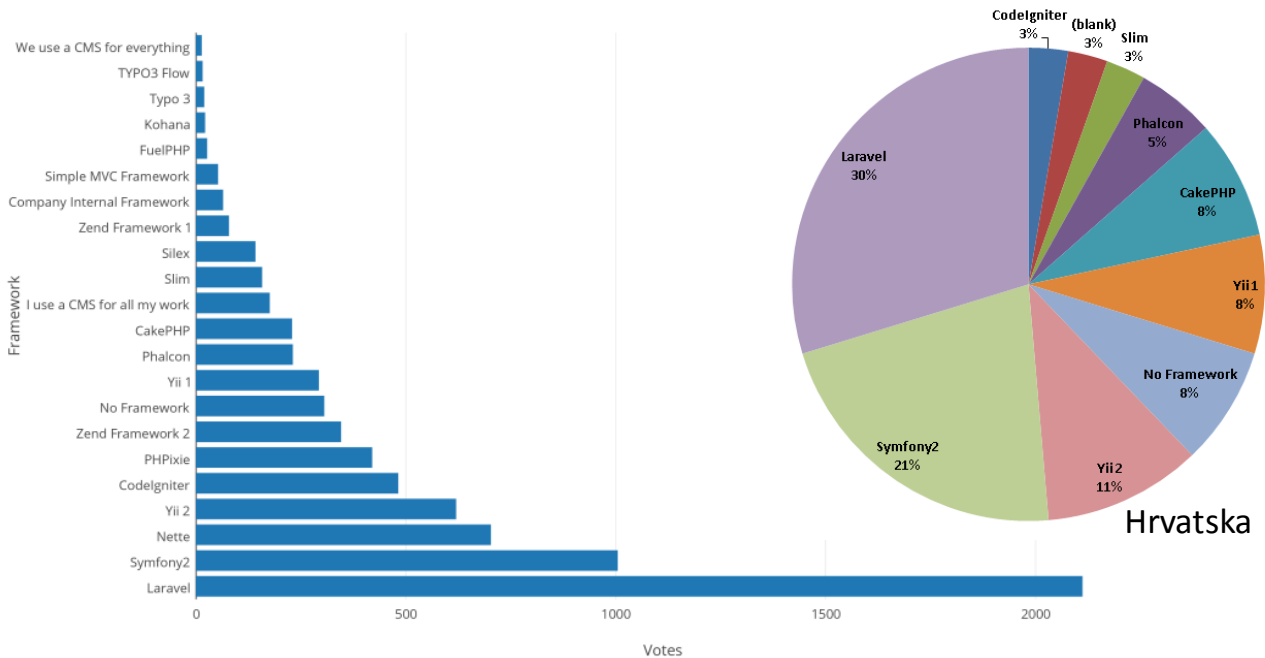


Hrvatska

<http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>

Najpopularniji PHP FW za osobnu uporabu

PHP Framework Popularity in Personal Projects - SitePoint, 2015



Hrvatska

<http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>

Laravel 5



Laravel 5

- Nova struktura direktorija
 - Odmak od RoR strukture
 - Korisnici su tražili mogućnost jednostavnog načina prilagodbe strukture direktorija
 - Kad je to postignuto – zašto ne intuitivnije nije posložiti direktorije
- *Route cache*
 - Korisno za velike projekte, s više od 100 stavki u `routes.php` datoteci
- *Route Middleware*
 - Filteri su postali punopravni objekti
 - Uključeni su CSRF i auth
- *Controller Method Injection*
 - *Injection* je dostupan u svim metodama, ne samo u `__construct`-u
 - Koristi se za provjeru (*validation*)

Laravel 5 (2)

- *Event Objects*
 - Događaji su postali punopravni objekti, ne samo nizovi znakova
- *Commands / Queueing*
 - Implementacija *command bus-a*
- *Laravel Scheduler*
 - Zamjena za `cron` – nije potreban SSH pristup i zajedno je u Git-u s cijelom aplikacijom
- *Tinker / Psysh*
 - Novi *shell* za interaktivni PHP
- *DotEnv*
 - Jedinstvena `.env` datoteka
 - Jednostavna struktura

Laravel 5 (3)

- *Laravel Elixir*
 - gulp (node.js) baziran sustav za održavanje CSS i JavaScript datoteka
- *Laravel Socialite*
 - Sučelje prema OAuth providerima
 - Facebook, Twitter, Google, GitHub, Bitbucket
- *Flysystem Integration*
 - Sučelje prema razni datotečnim sustavima
 - Local, Amazon Web Services - S3, Rackspace Cloud Files, Dropbox, FTP, SFTP, Zip, WebDAV, Azure Blob Storage
- Simple Controller Request Validation
- New Generators

Laravel 5 (4)

- *Annotated Controllers*
 - PHP-FIG radi na preporuci neovisnih *controller*-a, odn. način da se *route* datoteka generira iz popisa *controller*-a
 - *Symfony* ima implementirano pomoću *DocBlock annotation*-a
 - Implementirano u početku Laravel-a 5.0, ali izbačeno zbog preporuke da izmjena komentara ne smije utjecati na aplikaciju
 - PHP radi na preporuci za *annotation*-e koji su dio jezika
 - <https://wiki.php.net/rfc/annotations>

Pitanja?



srce

Sveučilište u Zagrebu
Sveučilišni računski centar

www.srce.unizg.hr

Ovo djelo je dano na korištenje pod licencom
Creative Commons *Imenovanje-Nekomercijalno*
4.0 međunarodna.

creativecommons.org/licenses/by-nc/4.0/deed.hr



Srce politikom otvorenog pristupa široj javnosti osigurava dostupnost i korištenje svih rezultata rada Srca, a prvenstveno obrazovnih i stručnih informacija i sadržaja nastalih djelovanjem i radom Srca.

www.srce.unizg.hr/otvoreni-pristup

 **srce**
otvoreni pristup