

CodeIgniter PHP Framework

“Kako v krajšem času narediti več”



Tomaž Muraus
kami@k5-storitve.net
<?php konferenca, 5. in 6. junij 2009

O meni

- 5 let izkušenj s PHP
- Ukvarjam se tudi s administracijo Unix strežnikov
- Vodja slovenske FreeBSD skupnosti (**freebsd.si**), sodelujem tudi pri slovenski Ubuntu skupnosti (**ubuntu.si**)
- Avtor odprto kodnega sistema za blog - Open Blog (<http://www.open-blog.info>)
- Na spletu poznan pod vzdevkom “Kami”
- Študiram računalništvo in informatiko

Danes bomo spoznali

- Kaj je framework?
- Kaj je MVC?
- Zakaj izbrati CodeIgniter ?
- Zgodovina CodeIgniter PHP frameworka
- Predstavitev CodeIgniter PHP frameworka
- Uporabni viri in povezave
- Vprašanja

Kaj je framework?

Programsko ogrodje ali framework je zbirka knjižnjic, orodij in razredov, ki nam pomagajo pri razvijanju aplikacij

Kaj je framework?

Spletno programsko ogrodje pa je programsko ogrodje, ki nam pomaga pri razvoju dinamičnih spletnih aplikacij

Kaj je framework?

Kaj je framework?

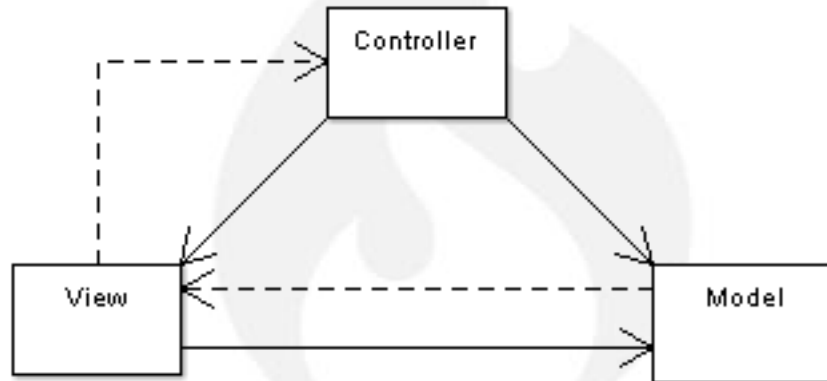
- Namen framework-a je, da nam olajša delo in prihrani čas
- Večina popularnih framework-ov uporablja **MVC** načrtovalski vzorec
- Pod framework-e štejemo tudi nekatere CMS sisteme (TYPO3, Drupal, ipd.)

Prednosti uporabe programskega ogrodja:

- Ponovna uporaba (reusable code)
- Krajši čas razvoja
- Razširljivost
- Skalabilnost

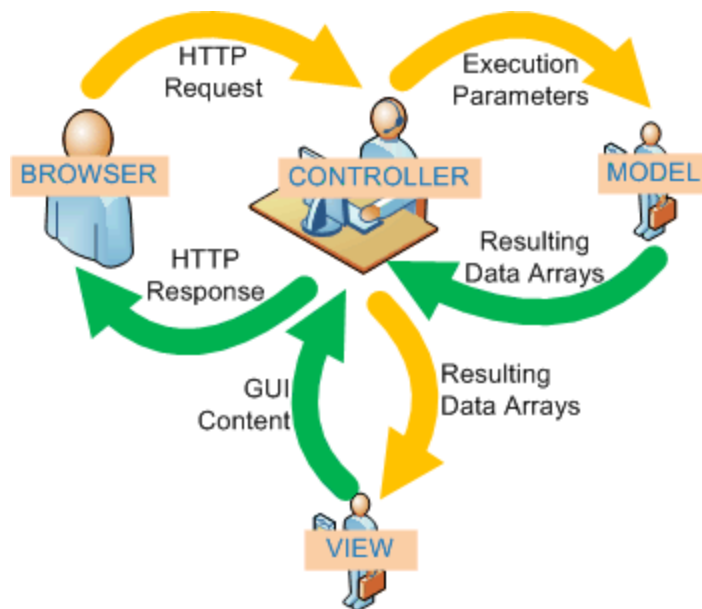
Kaj je MVC?

MVC je načrtovalski vzorec, ki striktno ločuje poslovno logiko od pogleda.



Kaj je MVC?

MVC je načrtovalski vzorec, ki striktno ločuje poslovno logiko od pogleda.



Zakaj izbrati CodeIgniter?

- MVC arhitektura
- Preprost za učenje in uporabo
- Zelo dobra dokumentacija
- Zelo prilagodljiv in razširljiv
- “Lahek” ampak zelo zmogljiv in hiter
- Velika in aktivna skupnost, ki z veseljem pomaga (forum, wiki, IRC)

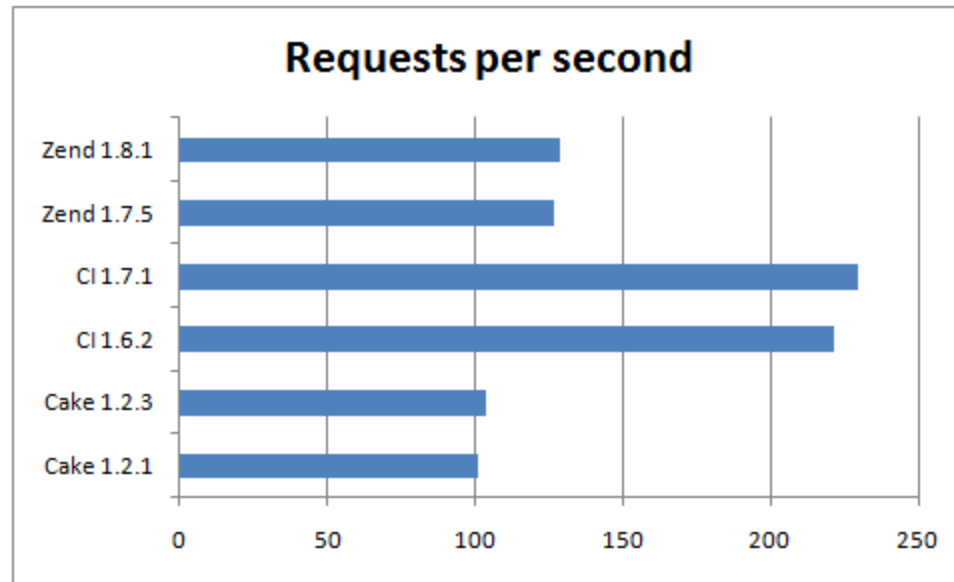
Zakaj izbrati CodeIgniter?

| | Run 1 | Run 2 | Run 3 | Run 4 | Average |
|----------------|--------|--------|--------|--------|---------------|
| Baseline HTML | 1327.5 | 1326.5 | 1328.6 | 1329.1 | 1327.9 |
| Baseline PHP | 331.6 | 332.1 | 331.4 | 332.0 | 331.8 |
| CakePHP | 3.6 | 3.7 | 3.8 | 3.5 | 3.7 |
| CodeIgniter | 21.5 | 21.2 | 21.7 | 21.7 | 21.5 |
| Zend Framework | 9.3 | 9.1 | 9.2 | 9.3 | 9.2 |

število obdelanih zahtev na sekundo

Vir: <http://avnetlabs.com/php/php-framework-comparison-benchmarks>

Zakaj izbrati CodeIgniter?



število obdelanih zahtev na sekundo

Vir: <http://leftblank.nl/php-framework-benchmark-zend-codeigniter-cakephp-481.html>

CodeIgniter

- Razvilo podjetje **EllisLab**
- Februar 2006 – prva javna različica (**1.0**)
- Februar 2008 – različica **1.6**, ki je prinesla veliko izboljšav in novosti
- Oktober 2008 – različica **1.7**
- Zadnja različica (**1.7.1**) je izšla februarja letos
- Q4 2009 – CodeIgniter 2 ?



CodeIgniter

Uradna stran:

www.codeigniter.com

Prenos:

www.codeigniter.com/downloads/

Dokumentacija:

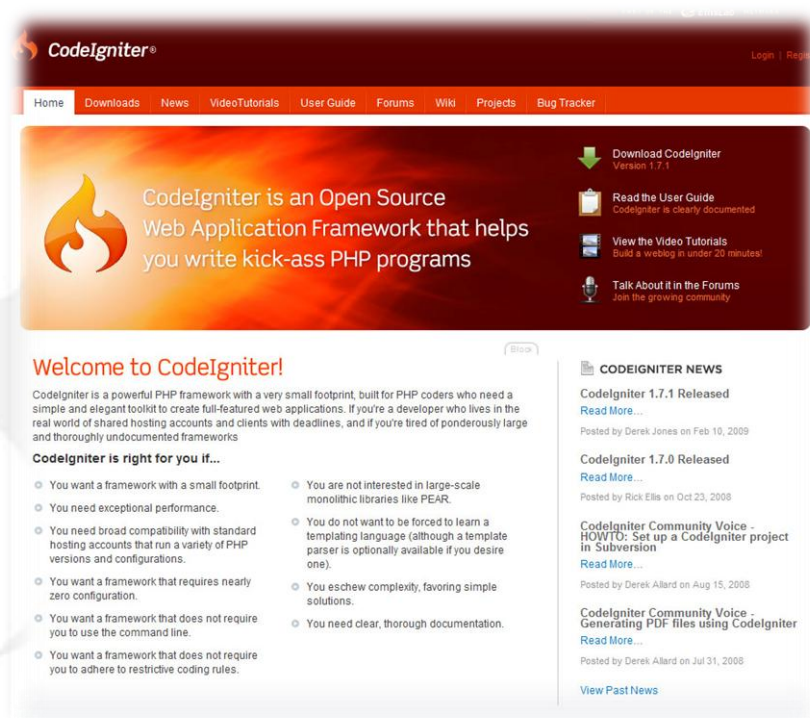
www.codeigniter.com/user_guide/

Wiki:

www.codeigniter.com/wiki/

Forum:

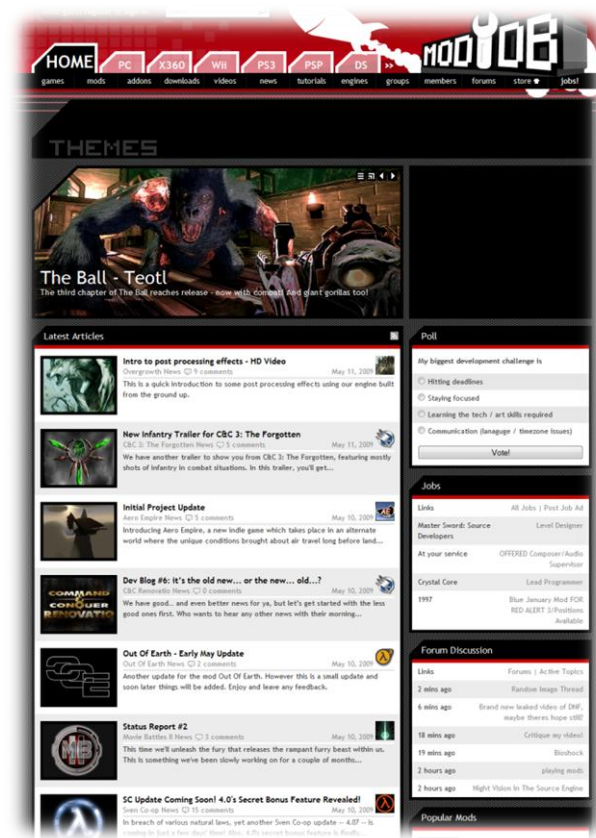
www.codeigniter.com/forums/



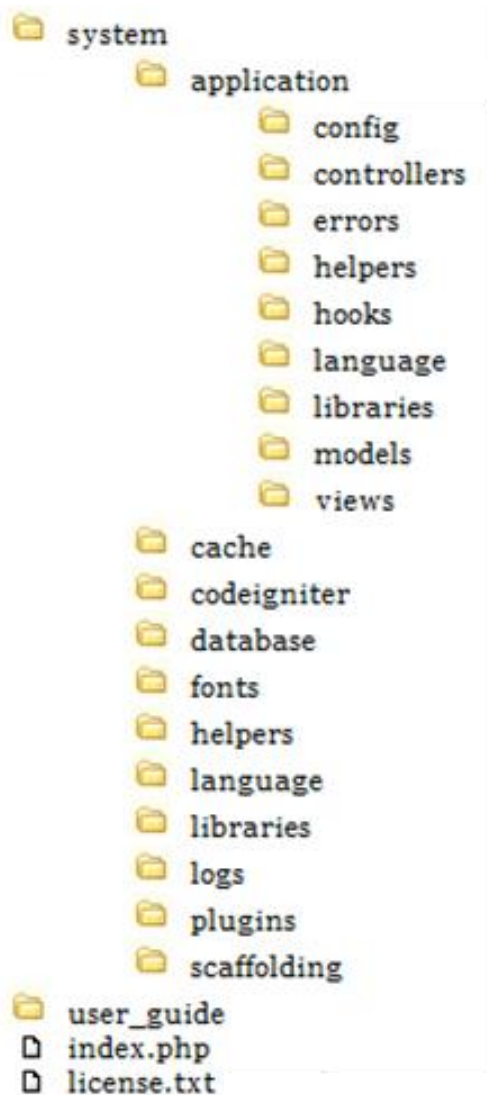
CodeIgniter

Večje spletne aplikacije razvite s pomočjo CodeIgniter-ja:

- <http://www.moddb.com>
- <http://www.gold.org>
- <http://www.opensourcefood.com>
- <http://www.campuslive.com>
- <http://www.snipprshop.de>
- <http://www.motortopia.com>
- <http://patterntap.com>
- <http://noknok.sanook.com>

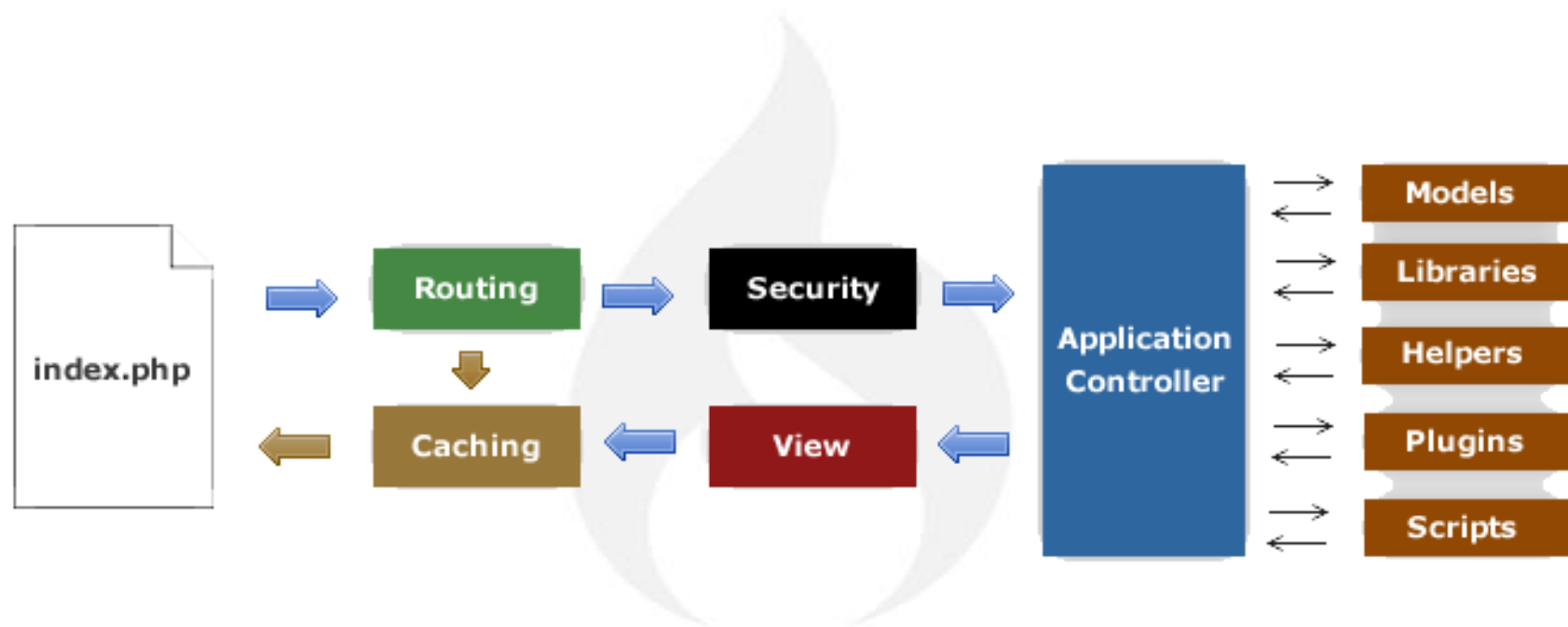


Struktura direktorijev



- **system/** – vsebuje jedro
- **application/** – tukaj se nahaja naša aplikacija
- **application/config/** - konfiguracijske datoteke
- **application/controllers/** - krmilniki
- **application/helpers/** - pripomočki
- **application/languages/** - jezikovne datoteke (prevodi)
- **application/libraries/** - knjižnjice
- **application/models/** - modeli
- **application/views/** - pogledi
- **user_guide/** - dokumentacija

Delovanje aplikacije



CodeIgniter URL-ji

Sestava URL naslova:

domena.com/**razred**/**metoda**/**parametri**

- **Prvi segment** – ime razreda (npr. **News**)
- **Drugi segment** – metoda razreda (npr. **view**)
- **Tretji in ostali segmenti** – parametri metode (npr. **35** – ID novice)

Primeri:

- domena.com/**users**
- domena.com/**users**/**profile**/**Kami**
- domena.com/**news**/**view**/**category**/**sport**/**article**/**formula-1-2009**

CodeIgniter URL-ji (primer)

Primer razreda (krmilnika) za prejšnji primer URL naslova:

```
class Users extends Controller
{
    // konstruktor
    public function __construct()
    {
        parent::Controller();
    }

    // metoda, ki se bo klicala, ko ne podamo metode razreda - "drugi konstruktor"
    public function index()
    {
        ...
    }

    // metoda profile, ki vrne podatke o uporabniku
    public function profile($username = null)
    {
        ...
    }
}
```

URI Routing (preusmerjanje)

Kje se pravila nahajajo?

Pravila se nahajajo v datoteki **application/config/routes.php**

Posebnosti:

- `:any`, `:num` – rezervirani besedi (`:any` – bilo kateri znak, `:num` – samo števila)
- Uporabimo lahko tudi regularne izraze, ki jih lahko poljubno mešamo s rezerviranimi besedami

Primeri pravil:

- domena.com/articles = domena.com/news:
`$route['articles'] = 'news';`
- domena.com/user/Kami:
`$route['user/(:any)'] = 'users/profile/$1';`
- domena.com/user/35:
`$route['user/(:num)'] = 'users/user_by_id/$1';`

Controllers (krmilniki)

Kaj je krmilnik?

Krmilniki so srce naše aplikacije in skrbijo, kako bodo HTTP zahteve obravnavane. So neke vrste vmesnik med modeli in pogledi.

Kje se krmilniki nahajajo?

Krmilniki se nahajajo v direktoriju **application/controllers/**. Ime datoteke je enako imenu razreda.

Za prejšnji primer bi se krmilnik nahajal v datoteki **application/controllers/users.php**

Controllers (krmilniki) - primer

Primer krmilnika (users.php):

```
class Users extends Controller
{
    // konstruktor
    public function __construct()
    {
        parent::__construct();
    }

    // metoda, ki se bo klicala, ko ne podamo metode razreda - "drugi konstruktor"
    public function index()
    {
        echo "Seznam uporabnikov";
        ...
    }

    // metoda profile, ki vrne podatke o uporabniku
    public function profile($username = null)
    {
        ...
    }

    // metoda, ki izbriše uporabnika
    public function delete($user_id = null)
    {
        // tukaj bi naložili in poklicali metodo modela za brisanje
        ...
    }
}
```

Models (modeli)

Kaj je model?

Model je tisti del aplikacije, ki vsebuje poslovno logiko - torej del, ki izvaja neke operacije ali transformacije nad podatki (podatki se lahko nahajajo v bazi, datoteki, seji, ipd.).

Modeli ponavadi vsebujejo tudi največ kode in so najbolj obsežni.

Kje se modeli nahajajo?

Modeli se nahajajo v direktoriju **application/models/**.

Ime datoteke je v obliki **imemodela_model.php**

Models (modeli) - primer

Primer modela (users_model.php):

```
class Users_model extends Model
{
    // konstruktor
    public function __construct()
    {
        parent::__construct();
    }

    // metoda, ki pridobi podatke o uporabniku
    public function get_user_by_id($user_id)
    {
        // tukaj se nahaja koda, ki dobi podatke o uporabniku (iz baze ?)
        ...
    }
}
```

Models (modeli) - nalaganje

Nalaganje modela:

Modele nalagamo na sledeč način:

```
$this->load->model('imemodela_model', 'ime_objekta');
```

Primer:

```
$this->load->model('users_model', 'users');
```

Do metod posameznega modela dostopamo na sledeč način:

```
$this->ime_objekta->ime_metode();
```

```
$this->users->get_user_by_id(15);
```

```
$this->users->delete_user(16);
```


Models (modeli) – avtomatsko nalaganje

Modele lahko tako kot tudi nekatere druge komponente nalagamo avtomatsko.

To pomeni, da se bodo naložili ob inicializaciji in nam bodo na voljo v celotnem življenjskem ciklu naše aplikacije.

Model avtomatsko naložimo tako, da ga dodamo v posebno polje v datoteki **application/config/autoload.php**.

Primer:

```
$autoload['model'] = array('model1', 'model2');
```

```
$autoload['model'] = array('users_model', 'news_model',  
'pages_model');
```

Models (modeli) – avtomatsko nalaganje

Primer autoload.php datoteke:

```
/*
| -----
|   Auto-load Language files
| -----
| Prototype:
|
|   $autoload['language'] = array('lang1', 'lang2');
|
| NOTE: Do not include the "_lang" part of your file.  For example
| "codeigniter_lang.php" would be referenced as array('codeigniter');
|
*/

$autoload['language'] = array();

/*
| -----
|   Auto-load Models
| -----
| Prototype:
|
|   $autoload['model'] = array('model1', 'model2');
|
*/

$autoload['model'] = array('users_model');
```

Views (pogledi)

Kaj je pogled?

Pogled je preprosto stran oziroma del strani (glava, noga, stranski meni, ...), ki prikazuje neke podatke.

CodeIgniter je zelo fleksibilen in nam omogoča tudi gnezdenje pogledov (možnost, da v enem pogledu vključimo oziroma naložimo drugega).

Kje se pogledi nahajajo?

Pogledi se nahajajo v direktoriju **application/views/**.

Ime datoteke je v obliki **imepogleda.php** (npr. **user_profile.php**).

Enako kot pri krmilnikih in modelih lahko tudi poglede poljubno organiziramo v direktorije.

Views (pogledi) - primer

Primers pogleda (users/list.php):

```
<html>
<head>
  <title>Seznam uporabnikov</title>
</head>
<body>
  <h1>Seznam uporabnikov</h1>
  ...
</body>
</html>
```

Views (pogledi) – nalaganje

Nalaganje pogleda

Poglede nalagamo v krmilnikih in sicer na sledeč način:

```
$this->load->view('ime_pogleda');
```

Primeri:

- `$this->load->view('about_us');` - naloži pogled **about_us.php**, ki se nahaja v direktoriju **application/views/**
- `$this->load->view('users/profile');` - naloži pogled **profile.php**, ki se nahaja v direktoriju **application/views/users/**
- `$this->load->view('users/profile', $data);` - naloži pogled **profile.php**, ki se nahaja v direktoriju **application/views/users/** in mu poda podatke, ki se nahajajo v spremenljivki **\$data** (polje ali objekt)

Active Record

Active Record je CodeIgniter-ov razred za delo s podatkovno bazo - torej branje, vstavljanje, posodabljanje in brisanje podatkov (ni ORM).

- **Prednosti:**

- Lahek za uporabo
- Preprosta sintaksa
- Varen (vse vrednosti so avtomatsko filtrirane)
- Poizvedbe so neodvisne od tipa podatkovne baze

- **Podprte platforme:**

- MySQL (4.1+), MySQLi (MySQL improved),
- PostgreSQL,
- Oracle,
- SQLite in
- ODBC

Active Record – primer (select)

Primer preproste poizvedbe:

```
$this->db->select('id, author, content');  
$this->db->order_by('date_posted', 'DESC');  
$this->db->limit(2, 5);
```

```
$query = $this->db->get('news');
```

ali z nizanjem metod (PHP 5):

```
$this->db->select('id, author, content')->order_by('date_posted',  
'DESC')->limit(2, 5);
```

Dobimo naslednje:

```
SELECT id, author, content FROM news ORDER BY date_posted DESC  
LIMIT 5, 2;
```

Active Record – primer (insert)

Primer vstavljanja podatkov v bazo:

```
$data = array(  
    'title' => 'Testna novica',  
    'author' => 'Tomaž Muraus',  
    'date_posted' => '2009-05-05',  
    'content' => 'Tukaj se nahaja vsebina...');  
  
$this->db->insert('news', $data);
```

Dobimo naslednje:

```
INSERT INTO news (title, author, date_posted, content) VALUES  
('Testna novica', 'Tomaž Muraus', '2009-05-05', 'Tukaj se nahaja  
vsebina...');
```


Libraries (knjižnjice)

Kaj so knjižnjice?

Knjižnjice niso nič drugega kot razredi, ki se nahajajo v direktoriju **libraries/**.

Nekatere knjižnjice, ki nam jih že ponuja CodeIgniter:

- Database – delo s podatkovno bazo (active record),
- Email – za pošiljanje e-sporočil,
- Form Validation – za validacijo obrazcev,
- File Uploading – za nalaganje datotek,
- Input,
- Language – za lokalizacijo,
- Session – za delo s sejami,
- URI – za delo z URI nizi in preusmerjanjem,
- ...

Helpers (pomočniki)

Kaj so pomočniki?

Pomočniki so preprosto zbirke kratkih in logično urejenih funkcij, ki opravljajo neke preproste operacije in se nahajajo v direktoriju **helpers/**.

Nekateri pomočniki, ki nam jih že ponuja CodeIgniter:

- Array – za delo s polji,
- Date – za zelo s datumi,
- Form – za delo z obrazci,
- Download – za prenos podatkov,
- File – za delo s datotekami (branje, pisanje, brisanje, ipd.),
- Smiley - 😊,
- URL – za delo s URL naslovi,
- ...

Uporabne povezave in viri - knjižnjice

Knjižnjice:

- **Organizacija aplikacije v module:**

- Matchbox (<http://code.google.com/p/matchbox/>)
- Modular Extensions (http://codeigniter.com/wiki/Modular_Extensions_-_HMVC/)

- **Object Relational Mapping (ORM):**

- IgnitedRecord (<http://www.assembla.com/wiki/show/IgnitedRecord>)
- Data Mapper (<http://stensi.com/datamapper/>)

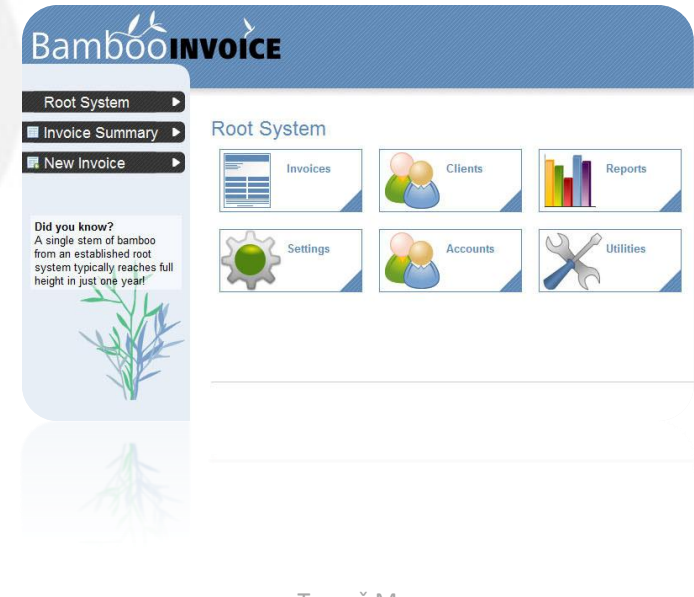
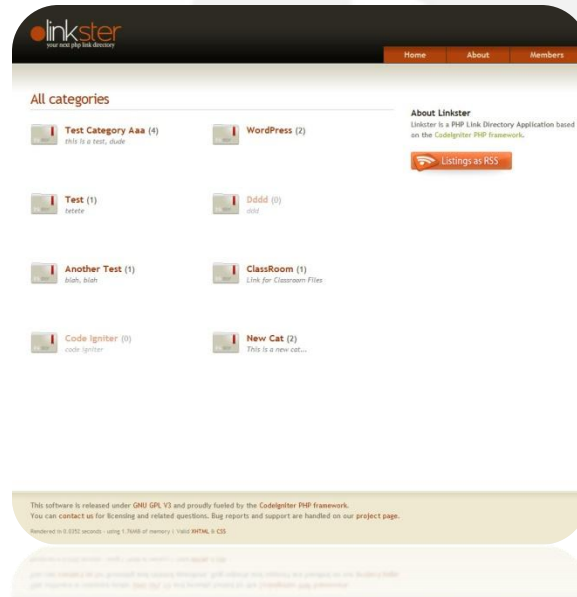
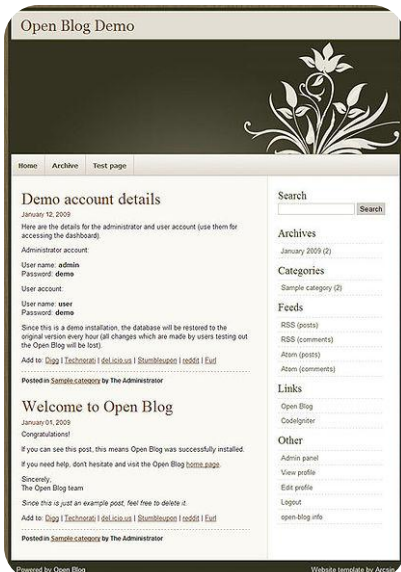
- **Avtentikacija, ACL:**

- Tank Auth (http://konyukhov.com/soft/tank_auth/)
- FreakAuth (<http://freakauth.4webby.com/>)
- DX Auth (http://dexcell.shinsengumiteam.com/dx_auth/)
- The Authentication Library (http://www.adamgriffiths.co.uk/user_guide/)
- KhACL (<http://codeigniter.com/forums/viewthread/70197/>)

Uporabne povezave in viri - aplikacije

Aplikacije:

- BambooInvoice (<http://bambooinvoice.org>) - spletna aplikacija za izdajo računov
- 64KB (<http://64kb.com>) – aplikacija za bazo znanj (knowledge database)
- Linkster (<http://www.assembla.com/wiki/show/linkster>) – imenik povezav
- Open Blog (<http://www.open-blog.info>) - sistem za blog

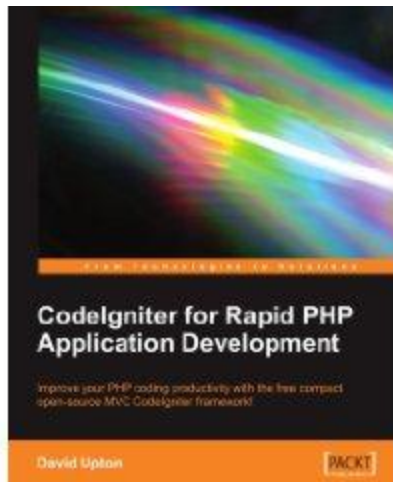


Uporabne povezave in viri - ostalo

Ostale uporabne povezave in viri:

- Dokumentacija - http://codeigniter.com/user_guide/
- Forum - <http://codeigniter.com/forums/>
- Wiki - <http://codeigniter.com/wiki/>
- CodeIgniter Directory - <http://www.codeigniterdirectory.com>
- Blog enega izmed razvijalcev - <http://www.derekallard.com>
- Blog enega izmed uporabnikov - <http://www.haughin.com>

Uporabne povezave in viri - knjige



CodeIgniter for Rapid PHP Application Development

Avtor: David Upton

Založnik: Packt Publishing

Datum izdaje: 23. julij, 2007



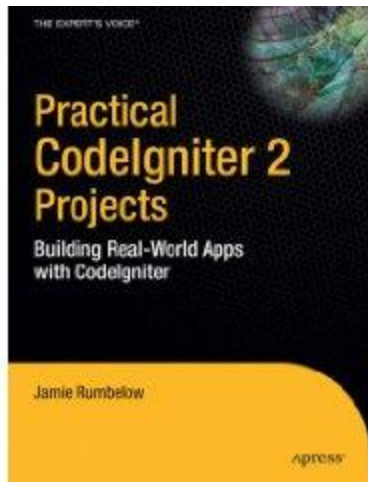
Professional CodeIgniter

Avtor: Thomas Myer

Založnik: Wrox

Datum izdaje: 28. julij, 2008

Uporabne povezave in viri - knjige



Practical CodeIgniter 2 Projects: Building Real-World Apps with CodeIgniter

Avtor: Jamie Rumbelow

Založnik: APRESS

Datum izdaje: 30. oktober, 2009

Hvala za pozornost

Vprašanja ?

**(lahko tudi kasneje na kami@k5-storitve.net ali
KamiSLO @ twitter)**