

Web Frameworks



Ciprian Dobre
ciprian.dobre@cs.pub.ro

Problematica

- Web Frameworks...

Zend Framework: <http://framework.zend.com/>

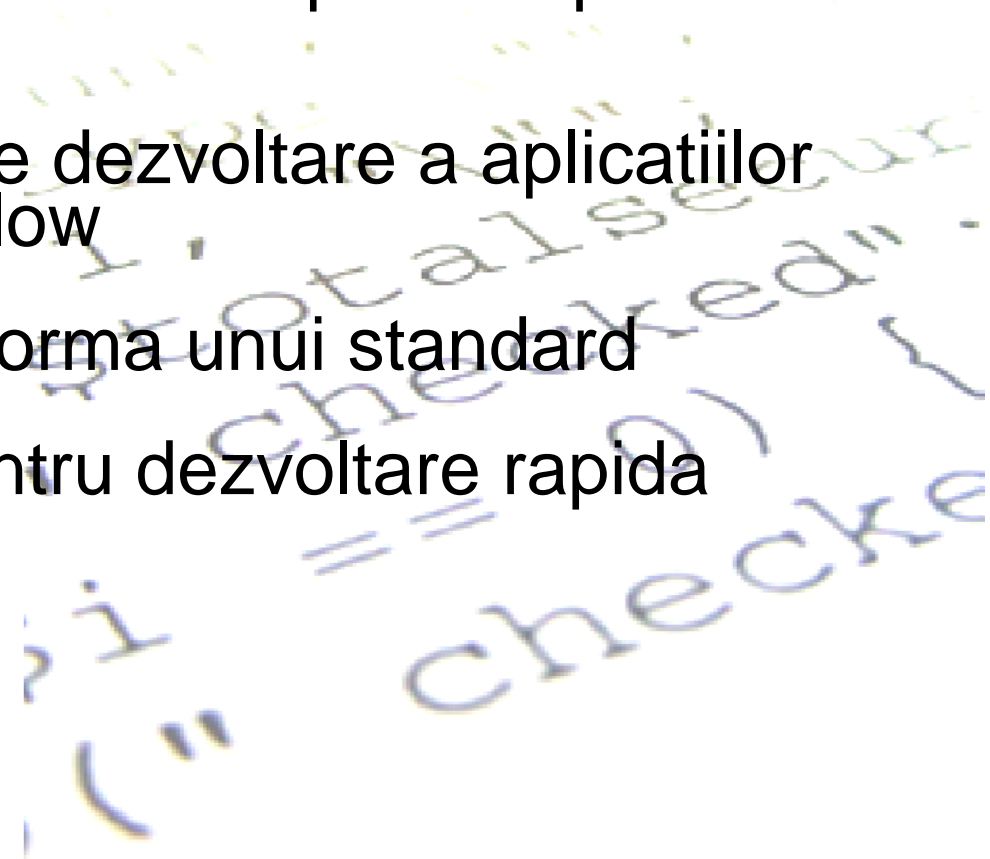
Symfony: <http://www.symfony-project.org>

CakePHP: <http://cakephp.org/>



Framework PHP

- Ce e un framework?
 - Cod ce permite scrierea mai rapida a aplicatiei Web.
 - Defineste un mediu de dezvoltare a aplicatiilor Web sub forma workflow
 - Implica codarea sub forma unui standard
 - Furnizeaza suport pentru dezvoltare rapida





Frameworks...

- [Agavi Development](#)
- [Akelos Development](#)
- [CakePHP Development](#)
- [CodeIgniter Development](#)
- [Flow3 Development](#)
- [Fuse Development](#)
- [Horde Development](#)
- [Kohana Development](#)
- [Lion Development](#)
- [Midgard Development](#)
- [PHP MVC Development](#)
- [PHPDevShell Development](#)
- [PHPOpenBiz Development](#)
- [Prado Development](#)
- [QPHP Development](#)
- [Qcodo Development](#)
- [Recess Development](#)
- [Seagull Development](#)
- [Smarty Development](#)
- [Solar Development](#)
- [Symfony Development](#)
- [WASP Development](#)
- [Yii Development](#)
- [Zend Development](#)
- [Zoop Development](#)
- [EvoCore Development](#)



PHP Frameworks

- Ce inseamna un framework bun?
 - Usor de invatat
 - Simplu de folosit
 - Intuitiv
 - Usor de extins si modificat
 - Ajutor pentru dezvoltare rapida si mentenanta
 - Trebuie sa aiba suportul unei comunitati de dezvoltatori



Frameworks PHP

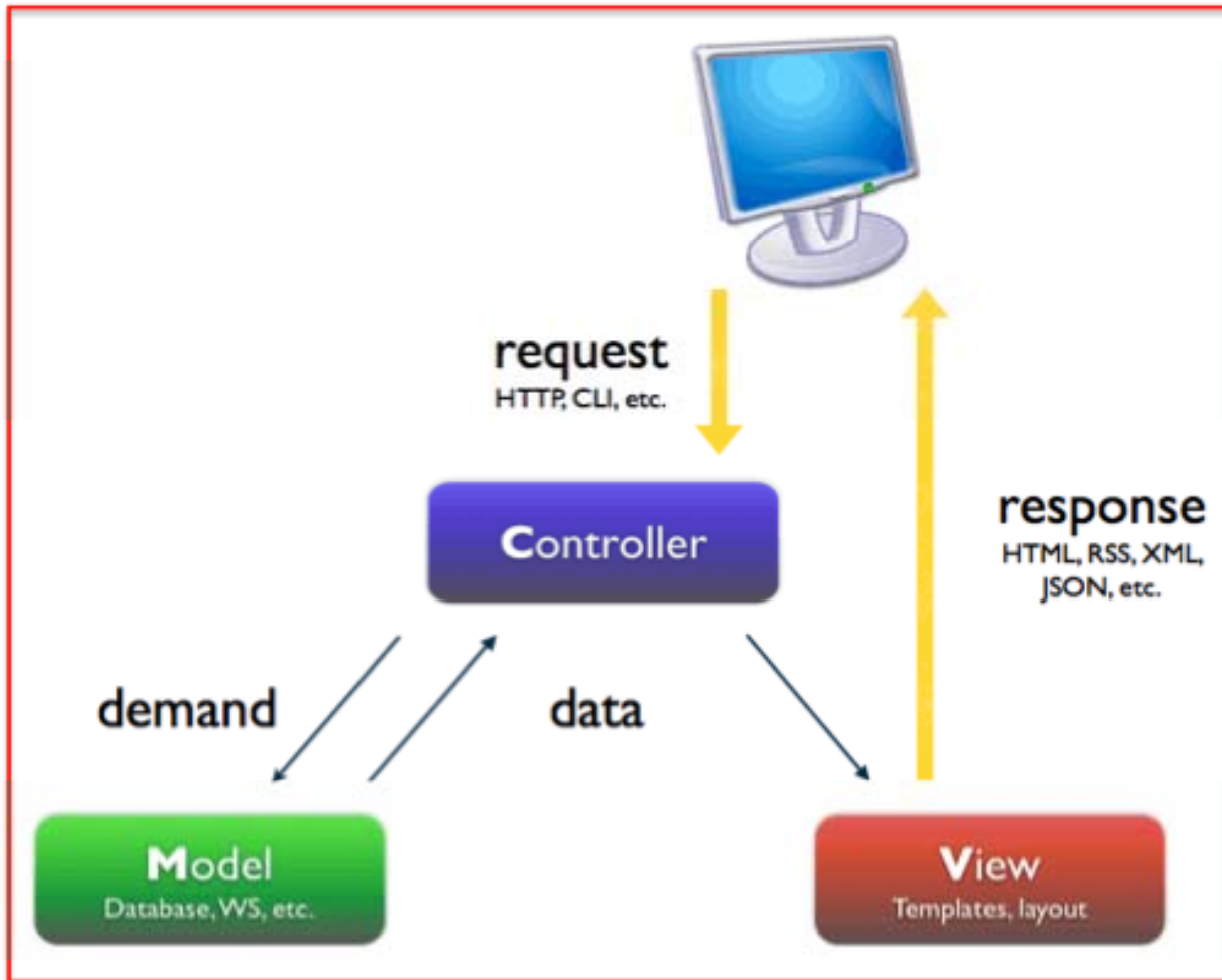
- 10. WACT (component toolkit)
- 9. Seagull
- 8. Prado
- 7. PHPDevShell
- 6. Zend Framework (recomandat)
- 5. CodeIgniter (recomandat)
- 4. Symfony (recomandat)
- 3. Yii (recomandat)
- 2. Kohana (recomandat)
- 1. CakePHP (recomandat)



MVC

- Design pattern software ce permite separarea datelor (model) de interfata utilizatorului (view)
- Model (nivelul datelor)
 - Logica bazei de date: wrappere in jurul datelor, elemente de securitate.
- View (interfata utilizator)
 - Redarea modelului intr-o forma specifica de interactiune, de obicei sub forma unui element interfata cu utilizatorul
- Controller (gestiunea interactiunilor)
 - Proceseaza si raspunde la evenimente, adesea sub forma unor actiuni ale utilizatorului, ce pot conduce la invocari ale modelului

MVC



Symphony





Symfony

- symfony-project.com
- Scris in PHP5
- Documentatie
 - Include o serie de aplicatii de test
 - Carte open-source disponibila online
 - Forumuri, wiki, mailing list...
- Multe plugin-uri include
 - CMS
 - Forum
 - Blog



Symfony ...

- Configuratia recomandata
 - Apache cu mod_rewrite
 - PHP5.2.x
- Simplu de instalat
 - Pear
 - #> pear channel-discover pear.symfony-project.com
 - #> pear install symfony/symfony
 - Apache, adaugare alias in VirtualHost:
 - Alias /sf /usr/share/php/data/symfony/web/sf



Symfony ...

- Dependinte minimale, toate incluse
 - Creole
 - Abstractizare DB
 - Suport pentru MySQL, PostgreSQL, SQLite, MSSQL
 - Propel
 - Object Relationship Mapping
 - PAKE
 - Biblioteca pentru taskuri simple din linia de comanda
 - LIME
 - Biblioetca pentru testare functionala si de unitate
 - Plus...
 - Prototype
 - Script.aculo.us
 - TinyMCE (WYSIWYG)
 - DynArch.com Rich Calendar



Implementarea MVC a Symfony

- Model
 - Acces la date
 - Bazat pe Propel (implicit)
 - Instrumente pre-construite pentru paginare, sortare si filtrare
 - Modelul poate fi configurat folosind XML sau YAML
 - Symfony construiește modelul!
 - `#> symfony propel-build-model`
 - Symfony furnizează clase pregătite pentru extinderea modelului



Implementarea MVC a Symfony...

- View
 - Nivelul prezentare
 - Layout si configuratie globale
 - Fiecare actiune poate avea propriul template
 - Symfony foloseste PHP ca limbaj back-end
 - Partial si Slot
 - Fragmente de template fara logica de business
 - Componente
 - Fragmente de template cu logica de business



Implementarea MVC a Symfony...

- Controller
 - Intermediar pentru logica de business
 - Fiecare 'modul' are o clasa controller asociata ce contine o serie de metode specifice pentru diversele 'actiuni'
- Conversia automata a URL-ului in modulul/actiunea corespunzatoare
- Metode de executie pre si post executie



Object Brokering (nivelul Model

- Generarea automata a claselor PHP pornind de la fisierul config
- Inlatura nevoia de efectuare a unor apeluri repetate catre DB, furnizeaza un nivel de abstractizare

```
schema.yml
propel:
  weblog_post:
    _attributes: { phpName: Post
}
  id:
  title:  varchar(255)
  excerpt:  longvarchar
  body:    longvarchar
  created_at:
```

PHP Code

```
$p = new Post();
$p->setTitle("Test Post");
$p->setBody("the body");
$p->save();
```

```
$posts =
PostPeer::doSelect(new
Criteria());
foreach($posts as $post)
{
print "Post: ".$post->getTitle();
}
```




CRUD Generare + Validare (Controller + View)

- Cel mai mare cosmar al unui programator Web : scrierea cod pentru multe formulare si elemente input, asociate cu procesoare si validatoare
- Symfony : genereaza template-uri Create/Read/Update/Delete
- Usor de customizat odata generate



symfony propel:generate-module admin findable FindableItem

Findable List

| Id | Name | Cardinality | Points | Description | S findable category |
|----|------------|-------------|--------|---------------------------------|---------------------|
| 1 | Alma Mater | | 1 | Take a picture with alma mater! | 1 |

[New](#)

New Findable

Name

Cardinality

Points

Description

S findable category id

[Cancel](#)

```
indexSuccess.php:
<h1>Findable List</h1>

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Cardinality</th>
<th>Points</th>
<th>Description</th>
<th>S findable category</th>
</tr>
</thead>
<tbody>
<?php foreach ($findable_item_list as $findable_item): ?>
<tr>
<td><a href="<?php echo url_for('findable/edit?id='.$findable_item->getId()) ?>"><?php echo $findable_item->getId() ?></a></td>
<td><?php echo $findable_item->getName() ?></td>
<td><?php echo $findable_item->getCardinality() ?></td>
<td><?php echo $findable_item->getPoints() ?></td>
<td><?php echo $findable_item->getDescription() ?></td>
<td><?php echo $findable_item->getSFindableCategoryId() ?></td>
</tr>
<?php endforeach; ?>
</tbody>
</table>

<a href="<?php echo url_for('findable/new') ?>">New</a>
```

```
actions.class.php:
public function executeIndex(sfWebRequest $request)
{
    $this->findable_item_list = FindableItemPeer::doSelect(new Criteria());
}
```

Validare

- Form “widgets” :

```
BaseFindableItemForm
$this->setValidators(array(
    'id'          => new sfValidatorPropelChoice(array('model' => 'FindableItem', 'column' => 'id',
'required' => false)),
    'name'        => new sfValidatorString(array('max_length' => 255, 'required' => false)),
    'cardinality' => new sfValidatorInteger(array('required' => false)),
    'points'      => new sfValidatorInteger(array('required' => false)),
    'description' => new sfValidatorString(array('required' => false)),
    's_findable_category_id' => new sfValidatorPropelChoice(array('model' => 'FindableCategory',
'column' => 'id')),
));
```

Noi validatori pot suprascrie acest comportament, prin scrierea codului corespunzator in FindableItemForm!



Dezvoltare Orientata spre Aspecte

- Symfony suporta “comportamente”
- Crearea automata de hook-spots
- Exemplu clasic: comportament “paranoid”
 - Inlocuirea stergerilor cu setarea unui flag
 - Extinderea selectului pentru a evita preluarea inregistratilor cu respectivul flag setat

Rutare

- Problema: Tuturor ne plac URL-urile simple, precum :
 - <http://cusearch09.com/user/image/5/320>
 - In loc de <http://cusearch09.com/user/upload/imageThumb.php?id=5&width=320>
- Solutia:
 - Apache mod_rewrite
 - Nu... prea complicat si greu de gestionat “on the fly”.
Necesita lucrul cu RegExp
 - Symfony:
 - routing.yml

```
resized_image:  
  url:          image/:id/:width  
  param: {module: upload, action: imageThumb}
```



Plugin-uri

- Symfony poate fi usor extins cu plugin-uri
- Respecta pattern-ul observer
 - static public function
listenToSomeSortOfEvent(sfEvent \$event)
 - Plugin-ul poate fi legat de oricare parte din
Symfony
- Folosire: obiecte, aspecte
 - Securitate? AJAX?

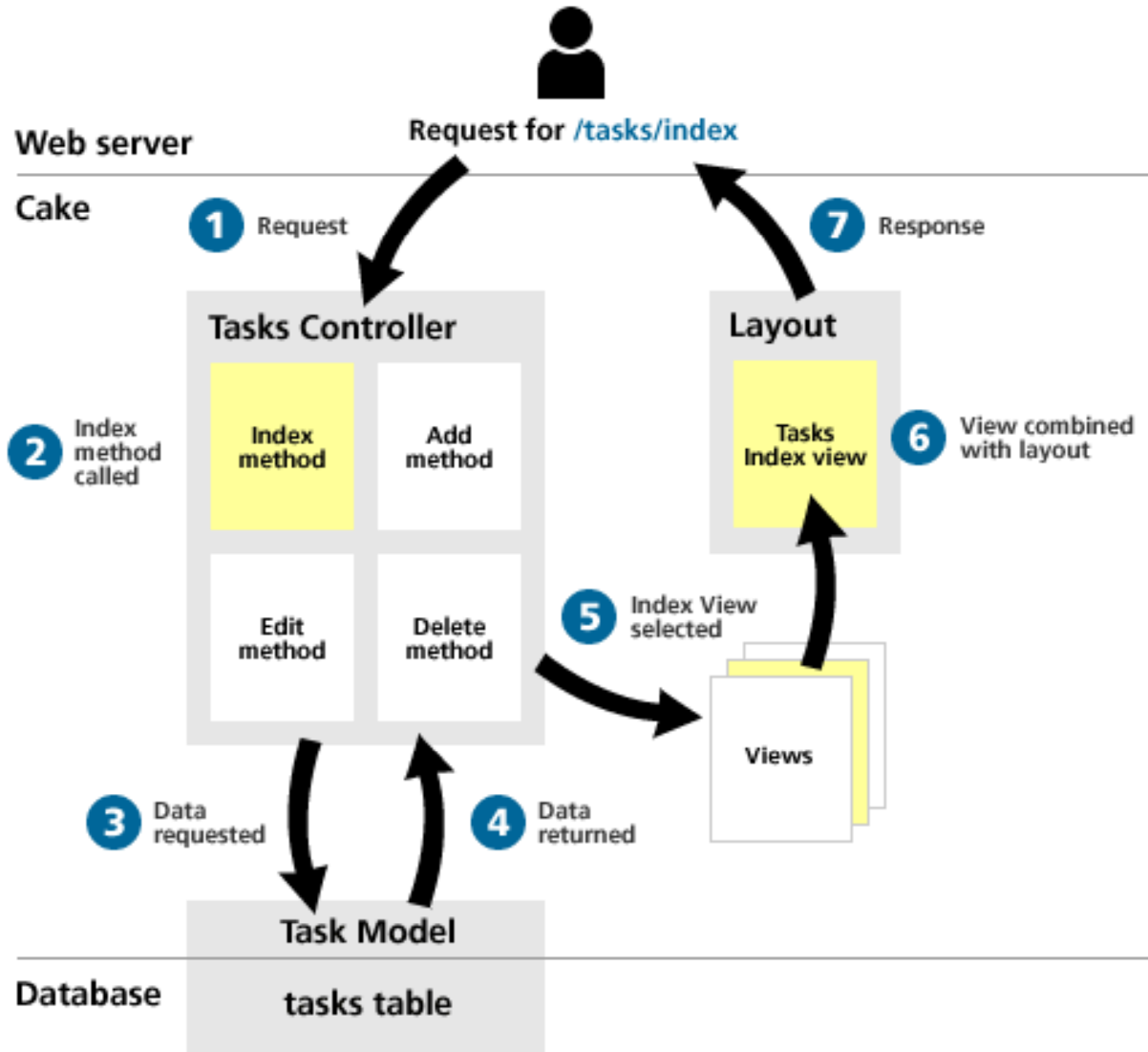
CakePHP





CakePHP

- CakePHP - framework open-source PHP
- Automatizarea task-urilor in Cake :
 - Usureaza crearea de functii de management standard (adaugare, editare sau stergere)
 - Foloseste Layout-uri si View-uri sub forma de template-uri pentru separarea logicii de business de nivelul prezentare
 - Validarea campurilor din formulare
 - Securizarea automata a datelor salvate in bazele de date (ex., previne atacuri SQL injection)
 - Include functii callback (ex., efectuarea unei actiuni inaintea sau dupa salvare)
 - Permite testarea atat a bazelor de date de test, cat si a celor de productie





Structura de Foldere

- **App**: codul aplicatiilor.
 - **Config**: toate fisierele de configurare
 - **Models**: modelele aplicatiilor, surse de date si comportamentul aplicatiilor
 - **Controllers**: controllerele si componentele aplicatiilor
 - **Views** : stocheaza toate template-urile, layout-urile (header, footer) si module auxiliare (ex., AJAX).
 - **Vendors**: clase si biblioteci 3rd party
 - **Webroot**: document root pentru aplicatie, stocheaza datele statice, precum CSS, imagini si codul JavaScript folosind de aplicatie.
 - **Locale**: fisiere pentru internationalizare
 - **Tmp**: date temporare generate de sistemul de cache si loguri.
 - **Plugins**: pachete plugin
- **lib/Cake** : stocheaza toate functiile de baza pentru CakePHP
- **Vendors**: alte biblioteci include intr-o aplicatie particulara

Mult succes la examene !!!

