

Frameworks oraz Serwery Aplikacji

Igor Wojnicki

Katedra Automatyki
Akademia Górniczo-Hutnicza w Krakowie

28 maja 2012

Spis Treści

1 Smarty Template Engine

- Smarty: Cele
- Przykłady
- Użyteczne funkcje i modyfikatory

2 Serwery Aplikacji

- Zope
- Inne

3 Zarządzanie Treścią

Outline

1 Smarty Template Engine

- Smarty: Cele

- Przykłady

- Użyteczne funkcje i modyfikatory

2 Serwery Aplikacji

- Zope

- Inne

3 Zarządzanie Treścią

Charakterystyka

- Template/Presentation Framework – tworzenie interfejsu.
- Buforowanie.
- Programowanie PHP → Definiowanie szablonów (ograniczony język znaczników).
- Szablony do typowych zastosowań: Tabele, Formularze.
- Kompilacja on-the-fly do kodu PHP.
- <http://smarty.php.net>

Cel

- Separacja modelu oraz interfejsu (wizualizacji): Model/View (zastosowanie koncepcji MVC).
- Model: PHP.
 - przyporządkowanie logicznej struktury generowanym danym: tytuł, paragraf, zawartość tabeli, autor etc.
- View: Szablony.
 - prezentacja struktury: HTML(statyczne oraz generowane przez PHP)+CSS.

Zalety Model/View

- Projektant interfejsu nie może *zepsuć* modelu.
- Interfejs może być przeprojektowany bez konieczności zmiany/modyfikacji modelu.
- Programista koncentruje się na modelu, nie na interfejsie.

Outline

1 Smarty Template Engine

- Smarty: Cele
- Przykłady
- Użyteczne funkcje i modyfikatory

2 Serwery Aplikacji

- Zope
- Inne

3 Zarządzanie Treścią

I. Wojnicki, Tech.Inter.

Model

index.php

```
include('Smarty.class.php');

// create object
$smarty = new Smarty;

// assign some content. This would typically come from
// a database or other source, but we'll use static
// values for the purpose of this example.
$smarty->assign('name', 'george smith');
$smarty->assign('address', '45th & Harris');

// display it
$smarty->display('index.tpl');
```


View

index.tpl

```
<html>
<head>
<title>User Info</title>
</head>
<body>
```

```
User Information:<p>
```

```
Name: {$name}<br>
```

```
Address: {$address}<br>
```

```
</body>
</html>
```

wyjście

```
<html>
<head>
<title>User Info</title>
</head>
<body>
```

```
User Information:<p>
```

```
Name: george smith<br>
```

```
Address: 45th & Harris<br>
```

```
</body>
</html>
```

Konkurencyjny View

Użycie modyfikatorów – logika prezentacji.

index.tpl

```
<html>
<head>
<title>User Info</title>
</head>
<body>
```

User Information:<p>

```
Name: {$name|capitalize}<br>
Addr: {$address|escape}<br>
Date: {$smarty.now|
  date_format:"%Y-%m-%d"}<br>
```

```
</body>
</html>
```

wyjscie:

```
<html>
<head>
<title>User Info</title>
</head>
<body>
```

User Information:<p>

```
Name: George Smith<br>
Addr: 45th & Harris<br>
Date: 2003-12-19<br>
```

```
</body>
</html>
```

Funkcje View I

index.tpl – wartość `title` tylko dla wskazanego pliku.

```
{include file="header.tpl" title="User Info"}
```

User Information:<p>

Name: {\$name|capitalize}

Address: {\$address|escape}


```
{include file="footer.tpl"}
```

Funkcje View II

header.tpl – wartość domyślna `title` jeżeli nie jest przekazana.

```
<html>
<head>
<title>{$title|default:"no title"}</title>
</head>
<body>
```

footer.tpl

```
</body>
</html>
```

Funkcje View III

wyście:

```
<html> <head> <title>User Info</title> </head> <body>
User Information:<p>
Name: George Smith<br>
Address: 45th & Harris<br>
</body> </html>
```

Funkcje generujące HTML I

index.php

```
include('Smarty.class.php');

// create object
$smarty = new Smarty;

// assign options arrays
$smarty->assign('id', array(1,2,3,4,5));
$smarty->assign('names', array('bob','jim','joe','jerr

// display it
$smarty->display('index.tpl');
```

Funkcje generujące HTML II

index.tpl

```
<select name="user">
{html_options values=$id output=$names selected="5"}
</select>
```

wyjście:

```
<select name="user">
<option label="bob" value="1">bob</option>
<option label="jim" value="2">jim</option>
<option label="joe" value="3">joe</option>
<option label="jerry" value="4">jerry</option>
<option label="fred" value="5" selected="selected">fred</option>
</select>
```

Pętle w View I

index.php

```
include('Smarty.class.php');

// create object
$smarty = new Smarty;

// assign an array of data
$smarty->assign('name', array('bob', 'jim', 'joe', 'jerry'));

// assign an associative array of data
$smarty->assign('users', array(
    array('name' => 'bob', 'phone' => '555-3425'),
    array('name' => 'jim', 'phone' => '555-4364'),
    array('name' => 'joe', 'phone' => '555-3422'),
```


Pętle w View II

```
array('name' => 'jerry', 'phone' => '555-4973'),  
array('name' => 'fred', 'phone' => '555-3235')  
));
```

```
// display it  
$smarty->display('index.tpl');
```

Pętle w View III

index.tpl

```
<table>
{section name=mysec loop=$name}
{strip}
  <tr class="{cycle values="odd,even"}">
    <td>{$name[mysec]}</td>
  </tr>
{/strip}
{/section}
</table>
```

strip usunięcie wielokrotnych spacji, znaku nowej linii, tabulatora

Pętle w View IV

```
<table>
{section name=mysec loop=$users}
{strip}
  <tr class="{cycle values="odd,even"}">
    <td>{$users[mysec].name}</td>
    <td>{$users[mysec].phone}</td>
  </tr>
{/strip}
{/section}
</table>
```

Pętle w View V

wyjście:

```
<table>
<tr class="odd"><td>bob</td></tr>
<tr class="even"><td>jim</td></tr>
<tr class="odd"><td>joe</td></tr>
<tr class="even"><td>jerry</td></tr>
<tr class="odd"><td>fred</td></tr>
</table><table>
<tr class="odd"><td>bob</td><td>555-3425</td></tr>
<tr class="even"><td>jim</td><td>555-4364</td></tr>
<tr class="odd"><td>joe</td><td>555-3422</td></tr>
<tr class="even"><td>jerry</td><td>555-4973</td></tr>
<tr class="odd"><td>fred</td><td>555-3235</td></tr>
</table>
```

Outline

1 Smarty Template Engine

- Smarty: Cele
- Przykłady
- **Użyteczne funkcje i modyfikatory**

2 Serwery Aplikacji

- Zope
- Inne

3 Zarządzanie Treścią

Modyfikatory I

`count_characters` ilość znaków

`count_paragraphs` ilość akapitów (oddzielone pustą linią)

`count_sentences` ilość zdań

`lower` zamiana znaków na małe

`upper` zamiana znaków na duże

`replace` zastąpienie ciągu znaków

`string_format` formatowanie danych, podobnie jak format dla `printf`

`strip` usunięcie wielokrotnych spacji, znaku nowej linii, tabulatora

`strip_tags` usuwa znaczniki HTML

`truncate` obcinanie łańcucha znaków

`escape` zastąpienie znaków encjami

`unescape` zastąpienie encji znakami

Funkcje I

{capture} przypisanie fragmentu wzorca do zmiennej

{for} {foreach} {while} pętle

{if} {elseif} {else} konstrukcje warunkowe

{function} funkcje

Outline

1 Smarty Template Engine

- Smarty: Cele
- Przykłady
- Użyteczne funkcje i modyfikatory

2 Serwery Aplikacji

- Zope
- Inne

3 Zarządzanie Treścią

Zope: Charakterystyka

- Kompletny serwer aplikacji: Model, View.
- Oparty o Python.
- Koncepcja obiektowej bazy danych.
- Kombajn, integruje języki wysokiego poziomu, szablony, bazę danych.
- Szeroka gama dostępnych modułów: obsługa dokumentów, galerii zdjęć, blogów, wiki.
- Zarządzany poprzez interfejs Web.
- Ścisła kontrola dostępu (użytkownicy).
- Open Source.
- <http://www.zope.org>

Zope: zarządzanie

The screenshot shows the Zope web management interface accessed via Netscape. The browser title is "Zope on http://machiavelli.digicool.com:8080 - Netscape". The address bar shows the URL "http://machiavelli.digicool.com:8080/manage".

The interface has a menu bar with "Contents", "View", "Properties", "Import/Export", "Security", "Undo", "Find", and "Help". The left sidebar shows a tree view of the site structure:

- Control_Panel
- QuickStart
 - Outline
 - Applications
 - Content
 - Database
 - Objects
 - acl_users
 - ecp
 - work

At the bottom of the sidebar, it says "© Digital Creations Refresh".

The main content area displays "Folder at /QuickStart / Outline" and a list of items:

- Applications (Application Development)
- ClosedBook.gif
- Content (Content Management)
- Database (Database Integration)
- Objects (Web to Objects)
- OpenBook.gif
- dtContent
- dtcTemplate
- index.html

Below the list are buttons for "Rename", "Cut", "Copy", "Delete", and "Export...".

At the bottom of the main area, it says "To add a new item, select an item type ." followed by a dropdown menu labeled "Available Objects".

The status bar at the bottom of the browser shows "Document: Done" and various navigation icons.

Zope: aplikacja

Zope on http://machiavelli.digicool.com:8080 - Netscape

File Edit View Go Communicator Help

Bookmarks Location: http://machiavelli.digicool.com:8080/manage What's Related

Zope Edit Upload Properties View Proxy Security Help

DTML Document at /QuickStart / Outline / index.html

Id index.html

Title

Size 154 bytes

Last modified 1999/08/02 13:57:49.958 US/Eastern

```
<!--#var standard_html_header-->
<h2><!--#var document_title--></h2>
<p>
This is the edit view of a DTML ocutment.
</p>
<!--#var standard_html_footer-->
```

Change Taller Shorter Wider Narrower

Document Done

Outline

1 Smarty Template Engine

- Smarty: Cele
- Przykłady
- Użyteczne funkcje i modyfikatory

2 Serwery Aplikacji

- Zope
- **Inne**

3 Zarządzanie Treścią

I. Wojnicki, Tech.Inter.

Inne

- J2EE (JSP), Java EE, (SUN) – specyfikacja, produkty:
 - Sun Java System Application Server, <http://www.sun.com>
 - JBoss (Open Source), <http://www.jboss.org>.
- WebSphere (IBM),
<http://www.ibm.com/software/websphere/>.
- Ruby on Rails (Open Source) – silne użycie koncepcji MVC,
<http://www.rubyonrails.org>.

Systemy zarządzania treścią

CMS Content Management System (nie mylić z Course Management System!) System Zarządzania Treścią.

- Drupal.
- Joomla.

Wiki

- MediaWiki.
- DokuWiki.

Systemy Zarządzania Dokumentami

- CMS: Content Management System!!! (nazwa poważnie przeciążona),
- bardziej szczęśliwa nazwa: ECM: Enterprise Content Management.
 - Documentum, EMC, www.emc.com.
 - Alfresco, www.alfresco.com, (Open source).
 - Nuxeo, www.nuxeo.org, (Open source).