

# Frameworks oraz Serwery Aplikacji

Igor Wojnicki

Katedra Informatyki Stosowanej  
Akademia Górniczo-Hutnicza w Krakowie

22 stycznia 2014

# Spis Treści

## 1 Smarty Template Engine

- Smarty: Cele
- Instalacja
- Przykłady
- Użyteczne modyfikatory
- Użyteczne funkcje

## 2 Serwery Aplikacji

- Zope
- Inne

## 3 Zarządzanie Treścią

# Outline

## 1 Smarty Template Engine

- Smarty: Cele
- Instalacja
- Przykłady
- Użyteczne modyfikatory
- Użyteczne funkcje

## 2 Serwery Aplikacji

- Zope
- Inne

## 3 Zarządzanie Treścią

I. Wojnicki, JiTW

# 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://www.smarty.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.

# Wersje

- Smarty 3.x: PHP 5.2+
- Smarty 2.x: PHP 4 lub 5.

I. Wojnicki, JiTW

# 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
- Instalacja
- Przykłady
- Użyteczne modyfikatory
- Użyteczne funkcje

## 2 Serwery Aplikacji

- Zope
- Inne

## 3 Zarządzanie Treścią

I. Wojnicki, JiTW



# Instalacja

```
tar xvfz Smarty-stable.tar.gz
mkdir templates templates_c configs cache
chmod a+rx cache/ configs/ templates/
chmod a+w templates_c/ cache/
```

I. Wojnicki, IITW

# Test instalacji

index.php

```
<?php
require ( './Smarty-3.1.7/libs/Smarty.class.php' );

$smarty = new Smarty();
$smarty->testInstall();
?>
```

# Test instalacji, rezultat, cd.

```
<PRE>
```

```
Smarty Installation test...
```

```
Testing template directory...
```

```
/home/users/vip/wojnicky/public_html/smarty/templates is OK.
```

```
Testing compile directory...
```

```
/home/users/vip/wojnicky/public_html/smarty/templates_c is OK.
```

```
Testing plugins directory...
```

```
/home/users/vip/wojnicky/public_html/smarty/Smarty-3.1.7/libs/plugins is OK.
```

```
Testing cache directory...
```

```
/home/users/vip/wojnicky/public_html/smarty/cache is OK.
```

```
Testing configs directory...
```

```
/home/users/vip/wojnicky/public_html/smarty/configs is OK.
```

```
Testing sysplugin files...
```

```
... OK
```

```
Testing plugin files...
```

```
... OK
```

```
Tests complete.
```

```
</PRE>
```

# Test instalacji, typowe błędy

```
<PRE>
```

```
Smarty Installation test...
```

```
Testing template directory...
```

```
/home/users/vip/wojnicki/public_html/smarty/templates is OK.
```

```
Testing compile directory...
```

```
FAILED: /home/users/vip/wojnicki/public_html/smarty/templates_c  
is not writable.
```

```
Testing plugins directory...
```

```
/home/users/vip/wojnicki/public_html/smarty/Smarty-3.1.7/libs/plugins is OK.
```

```
Testing cache directory...
```

```
FAILED: /home/users/vip/wojnicki/public_html/smarty/cache  
is not writable.
```

```
Testing configs directory...
```

```
/home/users/vip/wojnicki/public_html/smarty/configs is OK.
```

```
Testing sysplugin files...
```

```
... OK
```

```
Testing plugin files...
```

```
... OK
```

```
Tests complete.
```

```
</PRE>
```

# Test instalacji, prosta aplikacja

index.php

```
<?php
require (' ./Smarty-3.1.7/libs/Smarty.class.php' );

$smarty = new Smarty();

$smarty->assign('imie', 'Ned');
$smarty->display('index.tpl');
?>
```

# Test instalacji, szablon

templates/index.tpl

```
<html>
  <head>
    <title>Smarty</title>
  </head>
  <body>
    Czesc, {$imie}!
  </body>
</html>
```

I. Wojnicki, JiTW

# Test instalacji, rezultat

## templates/index.tpl

```
<html>
  <head>
    <title>Smarty</title>
  </head>
  <body>
    Czesc, Adam!
  </body>
</html>
```

I. Wojnicki, JiTW

# Outline

## 1 Smarty Template Engine

- Smarty: Cele
- Instalacja
- Przykłady
- Użyteczne modyfikatory
- Użyteczne funkcje

## 2 Serwery Aplikacji

- Zope
- Inne

## 3 Zarządzanie Treścią

I. Wojnicki, JiTW



# Model

index.php<sup>1</sup>

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

// utwórz obiekt
$smarty = new Smarty;

$smarty->assign('nazwisko', 'adam kowalski');
$smarty->assign('adres', '45th & Harris');

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

---

<sup>1</sup>Na podstawie: [http://www.smarty.net/crash\\_course](http://www.smarty.net/crash_course)

# View

## index.tpl

```
<html>
<head>
<title>Osoba</title>
</head>
<body>

<p>Informacje osobowe:
<p>
Nazwisko: {$nazwisko}<br>
Adres: {$adres}<br>

</body>
</html>
```

## wyjście

```
<html>
<head>
<title>Osoba</title>
</head>
<body>

<p>Informacje osobowe:
<p>
Nazwisko: adam kowalski<br>
Adres: 45th & Harris<br>

</body>
</html>
```

# Konkurencyjny View

Użycie modyfikatorów – logika prezentacji.

index.tpl

```
<html>
<head>
<title>Osoba</title>
</head>
<body>

<p>Informacje:
<p>
{$nazwisko|capitalize}<br>
Adres: {$adres|escape}<br>
Data: {$smarty.now|
  date_format:"%Y-%m-%d"}<br>

</body>
</html>
```

wyście:

```
<html>
<head>
<title>Osoba</title>
</head>
<body>

<p>Informacje
<p>
Adam Kowalski<br>
Adres: 45th & Harris<br>
Data: 2003-12-19<br>

</body>
</html>
```

# Funkcje View I

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

```
{include file="header.tpl" title="Informacje osobowe"}
```

```
<p>Informacje
```

```
<p>
```

```
Nazwisko: {$nazwisko|capitalize}<br>
```

```
Adres: {$adres|escape}<br>
```

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

# Funkcje View II

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

```
<html>
<head>
<title>{$title|default:"brak tytułu"}</title>
</head>
<body>
```

footer.tpl

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

# Funkcje View III

wyjście:

```
<html> <head> <title>Informacje osobowe</title> </head>

<p>Informacje
<p>

Nazwisko: Adam Kowalski<br>
Adres: 45th & Harris<br>

</body> </html>
```

# Funkcje generujące HTML I

index.php

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

// utwórz obiekt
$smarty = new Smarty;

// wartościami atrybutów mogą być tablice
$smarty->assign('id', array(1,2,3,4,5));
$smarty->assign('imiona', array('bob', 'jim', 'joe',
                                'jerry', 'fred'));

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

## Funkcje generujące HTML II

index.tpl

```
<select name="user">
{html_options values=$id output=$imiona 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');

// utwórz obiekt
$smarty = new Smarty;

// złożone dane
$smarty->assign('imie', array('bob', 'jim', 'joe',
                              'jerry', 'fred'));

// tablice asocjacyjne
$smarty->assign('users', array(
array('imie' => 'bob', 'tel' => '555-3425'),
array('imie' => 'jim', 'tel' => '555-4364'),
```

## Pętle w View II

```
array('imie' => 'joe', 'tel' => '555-3422'),  
array('imie' => 'jerry', 'tel' => '555-4973'),  
array('imie' => 'fred', 'tel' => '555-3235')  
));  
  
// display it  
$smarty->display('index.tpl');
```

# Pętle w View III

index.tpl

```
<table>
{section name=i loop=$imie}
{strip}
  <tr class="{cycle values="n,p"}">
    <td>{$imie[i]}</td>
  </tr>
{/strip}
{/section}
</table>
```

# Pętle w View IV

```
<table>
{section name=i loop=$users}
{strip}
  <tr class="{cycle values="n,p"}">
    <td>{$users[i].imie}</td>
    <td>{$users[i].tel}</td>
  </tr>
{/strip}
{/section}
</table>
```

# Pętle w View V

wyjście:

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

# Outline

## 1 Smarty Template Engine

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

## 2 Serwery Aplikacji

- Zope
- Inne

## 3 Zarządzanie Treścią

I. Wojnicki, JiTW

# 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` zastąp ciąg 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

# Outline

## 1 Smarty Template Engine

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

## 2 Serwery Aplikacji

- Zope
- Inne

## 3 Zarządzanie Treścią

I. Wojnicki, JiTW



# 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
- Instalacja
- Przykłady
- Użyteczne modyfikatory
- Użyteczne funkcje

## 2 Serwery Aplikacji

- Zope
- Inne

## 3 Zarządzanie Treścią

I. Wojnicki, JiTW

# 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

© 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: Rename, Cut, Copy, Delete, Export...

To add a new item, select an item type .

Available Objects

# 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
- Instalacja
- Przykłady
- Użyteczne modyfikatory
- Użyteczne funkcje

## 2 Serwery Aplikacji

- Zope
- **Inne**

## 3 Zarządzanie Treścią

I. Wojnicki, JiTW

# 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](http://www.emc.com).
  - Alfresco, [www.alfresco.com](http://www.alfresco.com), (Open source).
  - Nuxeo, [www.nuxeo.org](http://www.nuxeo.org), (Open source).