Template Engines für PHP

im Rahmen der Vorlesung Web-Engineering

Cornelius Knall

5. September 2007

Template Engines für PHP
Cornelius Knall
Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Inhaltsverzeichnis

- Literatur
- Ziele der Präsentation
- Warum Templates
- Templates
- Template-Engines
  - Smarty
    - Installation
    - Funktionsweise und Demo
  - EasyTemplateSystem
  - Kajona
- Templates ohne Engine
- Fazit

Literatur

Ziele der Präsentation

Warum Templates

Templates

Template-Engines

Smarty

Installation

Funktionsweise und Demo

EasyTemplateSystem

Kajona

Templates ohne Engine

Fazit

Literatur


Möhrke, Carsten (2004), Besser PHP programmieren, 1. korrigierter Nachdruck, Bonn (Galileo Press)


Krause, Jörg (2004): PHP 5, Grundlagen und Profiwissen, Webserverprogrammierung unter Windows und Linux, München/ Wien (Carl Hanser Verlag)


Lubkowitz, Mark (2006), Webseiten programmieren und gestalten, 4. korrigierte Nachdruck, Bonn (Galileo Press)


Schröer, Michael (2005), Web Content Management mit PHP und MySQL, 1. Auflage , Bonn (Galileo Press)

Literatur

Ziele der Präsentation

Warum Templates

Templates

Template-Engines

Smarty

Installation

Funktionsweise und Demo

EasyTemplateSystem

Kajona

Templates ohne Engine

Fazit

Literatur

- Smarty Template Engine, http://smarty.php.net/manual/de/
- Template Engine nur mit PHP, http://php-coding-standard.de/php_template_engine.php
- Beyond The Template Engine, http://www.sitepoint.com/article/beyond-template-engine
- Easy Template System, http://ets.sourceforge.net/ets.pdf

Literatur

Ziele der Präsentation

Warum Templates

Templates

Template-Engines

Smarty

Installation

Funktionsweise und Demo

EasyTemplateSystem

Kajona

Templates ohne Engine

Fazit

Einleitung

- Überblick verschiedener Template-Engines geben
- Hilfestellung zur Benutzung/Entscheidung geben
- Warum Templates einsetzten

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Einleitung

„Software design is hard, and we need all the help we can get.„

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Einleitung

„Software design is hard, and we need all the help we can get.„
-Bjarne Stroustrup

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Einleitung

- Smarty

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Einleitung

- Smarty
- EasyTemplateSystem

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Einleitung

- Smarty
- EasyTemplateSystem
- Kajona

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Einleitung

- Smarty
- EasyTemplateSystem
- Kajona
- Template Engine nur in PHP

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Warum Templates?

- Früher sehr ineffizient
- Da Layout und Logik vermischt
- Trennung von Design und Logik
- Übersichtlichkeit und Wartung werden erhöht
- Produktivität wird gesteigert

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Warum Templates?

- Früher sehr ineffizient
- Da Layout und Logik vermischt
- Trennung von Design und Logik
- Übersichtlichkeit und Wartung werden erhöht
- Produktivität wird gesteigert
- Designer schlechte Programmierer

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Warum Templates?

- Früher sehr ineffizient
- Da Layout und Logik vermischt
- Trennung von Design und Logik
- Übersichtlichkeit und Wartung werden erhöht
- Produktivität wird gesteigert
- Designer schlechte Programmierer
- Programmierer schlechte Designer

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Code ohne Trennung

Listing 1: Code ohne Trennung

```
1  ...
2  echo ("<table width=627 border=0
3    cellpadding=0 cellspacing=0>");
4  echo ("<tr>");
5  echo ("<td width=327>"
6    .$row["titel"]."</td>");
7  echo ("</td>");
8  ...
```


Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Was ist ein Template?

- Einsatz in verschieden Situationen
- Bekannt aus Textverarbeitungsprogrammen
- Dokumentenvorlage (*engl. template*)
- Gleiche Vorlage Inhalt wird über sog. Platzhalter eingebunden
- Einsatz auch bei Webanwendungen

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Was ist eine Template-Engine?

- Software Modul
- Wird zur Generierung von HTML-Seiten oder andere Dokumente eingesetzt
- Layout/Design wird von Applikationslogik getrennt
- Beide Dateien werden zur Laufzeit zusammen gefügt

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Template Engine

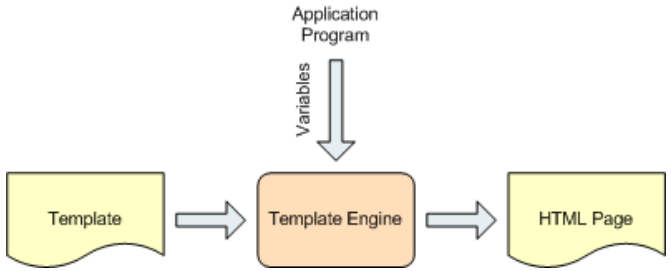

Abbildung: Template Engine Diagramm

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Template Engine

Listing 2: FastTemplate PHP-Code

```
1 <?
2 $tpl = new FastTemplate("pfad");
3 $tpl->assign(PAGETITLE, "Musterseite");
4 $tpl->parse(MAIN, "bask");
5 $tpl->FastPrint(MAIN);
6 ?>
```


Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Template Engine

Listing 3: FastTemplate HTML-Code

```
1  <html>
2    <head>
3      <title>{PAGETITLE}</title>
4    </head>
5    <body>
6      <h1>Willkommen</h1>
7    </body>
8  </html>
```


Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Smarty

- 1999 erster Ansatz
- Entwickelt in C
- Komplikationen mit Anforderungen und Umsetzung mit C
- Neuentwicklung mit PHP
- So entstand SmartTemplate, wurde aber nie veröffentlicht
- Durch Weiterentwicklung entstand Smarty

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Smarty - Installation

- Source unter http://smarty.php.net
- An gewünschten Ort entpacken
- */libs* relevanter Ordner
- Spezielle Rechte auf folgende Ordner vergeben:

Smarty - Installation

- Source unter http://smarty.php.net
- An gewünschten Ort entpacken
- */libs* relevanter Ordner
- Spezielle Rechte auf folgende Ordner vergeben:
- templates (rw)

Smarty - Installation

- Source unter http://smarty.php.net
- An gewünschten Ort entpacken
- */libs* relevanter Ordner
- Spezielle Rechte auf folgende Ordner vergeben:
- templates (rw)
- configs (rw)

Smarty - Installation

- Source unter http://smarty.php.net
- An gewünschten Ort entpacken
- */libs* relevanter Ordner
- Spezielle Rechte auf folgende Ordner vergeben:
- templates (rw)
- configs (rw)
- templates_c (rw)

Smarty - Installation

- Source unter http://smarty.php.net
- An gewünschten Ort entpacken
- */libs* relevanter Ordner
- Spezielle Rechte auf folgende Ordner vergeben:
- templates (rw)
- configs (rw)
- templates_c (rw)
- cache (rw)

Listing 4: PHP-Code für Smarty

```
1  <?
2     require 'smarty/MySmarty.class.php';
3
4     //Instanziieren eines neuen Objektes
5     $smarty = new MySmarty;
6
7     // Initialisierung der Variablen
8     $smarty->assign ('titel', 'Hallo Welt!');
9     $smarty->assign ('text', 'Die erste Seite');
10
11    // Template übergeben
12    $smarty->display('eins.tpl')
13 ?>
```


Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Demo

- DEMO

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

EasyTemplateSystem - ETS

- Überführung von Daten in ein beliebiges Dokument
- SQL-Statments, XML-Dokumente oder z.B. ASCII
- HTML-Dokumente

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

EasyTemplateSystem - ETS

Zur Benutzung mit PHP stehen folgende Funktionen bereit:

Listing 5: ETS Funktionen für PHP

```
1  void printt {
2    (mixed datatree, mixed containers,
3        [, string entry]}
4  //gibt das Template aus
5
6  void sprintt {
7    (mixed datatree, mixed containers,
8        [, string entry]}
9  //gibt das Template zurück
```

EasyTemplateSystem - ETS

- *datatree* kann ein Objekt sein, oder ein Array von Objekten

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

EasyTemplateSystem - ETS

- *datatree* kann ein Objekt sein, oder ein Array von Objekten
- *containers* können entweder Arrays enthalten oder nur einfache Strings

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

EasyTemplateSystem - ETS

- *datatree* kann ein Objekt sein, oder ein Array von Objekten
- *containers* können entweder Arrays enthalten oder nur einfache Strings
- *entry* ist der Abschnitt in einem Template, der angesprochen werden soll. Standard ist main

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

EasyTemplateSystem - Funktionsweise

Listing 6: ETS datatree

```
1  $page->title = 'Home page';
2  $page->lastmodified = 'Thursday,
3    January 23, 2003';
4
5  printt($page, 'test.tpl');
```

Template Engines für PHP
Cornelius Knall
Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

EasyTemplateSystem - Funktionsweise

Listing 7: ETS Template-Code

```
{mask:main}
<html>
  <head>
        <title>{title}</title>
  </head>
  <body>
    <h1>{title}</h1>
    <hr>
    <div align="center">
      Last modified: {lastmodified}
    </div>
  </body>
</html>
{/mask}
```

EasyTemplateSystem - Funktionsweise

Listing 8: ETS datatree

```
1  $page->title = 'Home page';
2  $page->lastmodified = 'Thursday,
3                         January 23, 2003';
4  $page->partner->name = 'foobar.com';
5  $page->partner->id = 123;
6  $page->menu[1]->url = "download.php";
7  $page->menu[1]->label = "Downloads";
8  $page->menu[2]->url = "links.php";
9  $page->menu[2]->label = "Links";
10
11 printt($page, 'test.tpl',menu,partner);
```


Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

EasyTemplateSystem - Funktionsweise

Listing 9: ETS Template-Code

```
{mask:main}
<html>
    <head><title>{title}</title></head>
    <body>
        <h1>{title}</h1><hr>
{mask:menu}
  <a href="{url}">{label}</a>
{/mask}<hr>
      <div align="center">
       Last modified: {lastmodified}</div>
{mask:partner}
  <div>with our partner {name} ({id})</div>
{/mask}
    </body>
</html>
{/mask}
```


Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

CMS Kajona

- Eigenentwicklung im Rahmen eines CM-Systems
- Strickte Trennung von Layout/Design und Applikationslogik
- Einzige Ausnahme *JavaScript etc*

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

CMS Kajona

- Eigenentwicklung im Rahmen eines CM-Systems
- Strickte Trennung von Layout/Design und Applikationslogik
- Einzige Ausnahme *JavaScript etc*
- Funktionalität wird anhand des Systems erklärt

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Templates ohne Template Engine

Es gibt zwei Ansätze, die ohne eine Engine auskommen:

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Templates ohne Template Engine

Es gibt zwei Ansätze, die ohne eine Engine auskommen:

- Template Engine nur in PHP

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Templates ohne Template Engine

Es gibt zwei Ansätze, die ohne eine Engine auskommen:

- Template Engine nur in PHP
- Beyond the Template Engine

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Template Engine nur in PHP

- Auch hier Layout/Design und Applikationslogik getrennt
- Jedoch keine strickte Trennung
- Blöcke, Schleifen etc werden im Template implementiert

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Listing 10: Code für das Template

```
<body>
  <ul>
    <?php foreach ($block as $value): ?>
    <li>
      <?=$value?>
    </li>
    <?php endforeach; ?>
  </ul>
  <table width="100%" border="1">
    <?php foreach ($block as $value): ?>
    <td>
      <?=$value?>
    </td>
    <?php endforeach; ?>
  </table>
</body>
```


Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Listing 11: PHP-Code

```
1 <?
2   $title_text = 'TITLE: This example
3           contains two blocks';
4   $block = array('Claus', 'Kelvin',
5           'Skrol', 'Daniel');
6
7   require_once 'two_blocks.htm';
8 ?>
```

Vorteile

- Trennung von Layout/Design von Programmlogik
- Übersichtlichkeit und Wartung
- Gemeinsames arbeiten von Designern und Entwickler
- Durch Caching Performance sparend

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Nachteile

- Gewisse Einarbeitungszeit
- Entscheidung für eine für sich optimale Lösung
- Sehr schwerer Wechsel von einer Technologie zur anderen
- Kompliziertes durchschleifen von Inhalten

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit

Danksagung

Vielen Dank für die Aufmerksamkeit

Literatur
Ziele der Präsentation
Warum Templates
Templates
Template-Engines
Smarty
Installation
Funktionsweise und Demo
EasyTemplateSystem
Kajona
Templates ohne Engine
Fazit