



„A web application framework in PHP“

Fehler dürfen korrigiert, Einwürfe spontan gemacht werden!

Über mich

Fabian Blechschmidt, Freelancer

- PHP seit 2003
- Symfony 1.4 seit 2011, Erfahrung bisher ~30h
- Eigenes Projekt: IkonoShirt wird mit Symfony umgesetzt

IKONO
SHIRT

Command Line Interface (CLI)

- CLI für alle wichtigen Aufgaben:
 - Neues Projekt, App, Module, Task (Skeleton) erzeugen
 - Log leeren
 - I18N extrahieren und finden
 - (funktionale/Unit-) Tests laufen lassen
 - Optimierung
 - Deploying auf einem anderen Server

Apps und Umgebungen

- Projekte bestehen aus Applications
 - Applications teilen sich die gleichen Models
- Umgebungen definieren Eigenschaften, z.B.
 - dev-Umgebung: ~~Cache~~, Error-Reporting, DB-Einstellungen, ...
 - live-Umgebung: Cache, ~~Error-Reporting~~, DB-Einstellungen, ...
 - test-Umgebung: ...
 - staging-Umgebung: ...

Model-View-Controller

- Models laufen auf Propel oder Doctrine (ORM)
 - Symfony generiert aus YAML Doctrine
- Views sind PHP-Dateien
 - Layout definiert Seitenaufteilung
 - Helper für sich wiederholende Aufgaben
 - `include_http metas()`, `include_title()`,
`include_component('shop', 'topMenu')`, ...

```
Attribute:
tableName: attributes
columns:
  id:
    type: integer(4)
    primary: true
    notnull: true
    autoincrement: true
    unsigned: true
  name:
    type: string(63)
    notnull: true
  attribute_number:
    type: string(31)
    mandatory:
      type: boolean
      notnull: true
  lft:
    type: integer(4)
    notnull: true
    unsigned: true
  rght:
    type: integer(4)
    notnull: true
    unsigned: true
  default_value:
    type: integer(4)
    notnull: true
    unsigned: true
relations:
  Attribute:
    class: Attribute
    local: lft
    foreign: id
    foreignAlias: Attribute
    onDelete: no action
    onUpdate: no action
  Attribute:
    class: Attribute
```

MVC: Controller

- Controller bietet Actions, Components und Partial
 - Actions
 - Aufruf durch URL (<http://example.com/controller/action>)
 - Auswertung des PHP-Codes
 - Erstellung des Templates
 - Component
 - Unabhängig von der URL, ansonsten ähnlich der Action
 - Partial
 - Template, welches eingebunden werden kann

YAML ... YAML ... YAML und XML

- Konfiguration über YAML

- Routen
- Cache
- Filter
- ...

```
# default rules
homepage:
  url: /
  param: { module: shop, action: index }

# generic rules
# please, remove them by adding more specific rules
default_index:
  url: /:module
  param: { action: index }

default:
  url: /:module/:action/*
```

```
<!-- apps/frontend/i18n/fr/messages.xml -->
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE xliiff PUBLIC "-//XLIFF/DTD XLIFF//EN"
"http://www.oasis-open.org/committees/xliiff/documents/xliiff.dtd">
<xliiff version="1.0">
  <file source-language="EN" target-language="fr" datatype="plaintext"
    original="messages" date="2008-12-14T12:11:22Z"
    product-name="messages">
    <header/>
    <body>
      <trans-unit id="1">
        <source>About Jobeet</source>
        <target>A propos de Jobeet</target>
      </trans-unit>
      <trans-unit id="2">
        <source>Feed</source>
        <target>Fil RSS</target>
      </trans-unit>
      <trans-unit id="3">
        <source>Jobeet API</source>
        <target>API Jobeet</target>
      </trans-unit>
      <trans-unit id="4">
        <source>Become an affiliate</source>
        <target>Devenir un affilié</target>
      </trans-unit>
    </body>
  </file>
</xliiff>
```

- *Convention over configuration*

- Global ← App ← Module/Action
- Übersetzung via ICU-Schema in XML

Funktionale und Unit-Tests

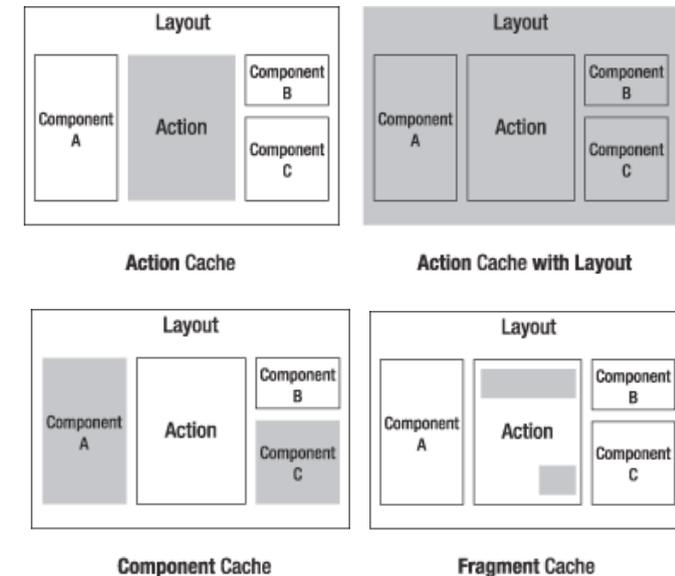
- Symfony bringt *lime* mit, ein Unit-Test-Framework
 - Nicht so umfangreich wie PHPUnit
 - damit schnell zu lernen
- SfBrowser und sfTestFunctional
 - Tester um Eigenschaften zu prüfen
 - Szenarien können definiert werden, wie in Selenium

Mehr?

- Routing
 - Vielseitige Regeln
 - Übergabe von weiteren Informationen/Parametern
 - Leichter Umgang in Templates dank Helper(n)
- Form-Framework
 - `<form>` aus Widgets aufbauen und validieren
 - CSRF Sicherung „gratis“

Noch mehr?

- Swift Mailer
 - global konfigurierbar via yaml
- AJAX via jQuery, Prototype, ...
- Caching
 - Action (mit oder ohne Layout)
 - Partial, Component oder Component-Slot
 - Template Fragment
 - Als HTML-Datei (bypass Symfony)



Fragen?

Gibt es Fragen?

The logo for IKONO SHIRT, featuring the words 'IKONO' and 'SHIRT' stacked vertically in a bold, white, sans-serif font. The text is set against a light orange background that is shaped like a t-shirt collar.

Danke und Links

Danke für die Aufmerksamkeit!

Für mehr Infos:

- Symfony: <http://www.symfony-project.org/>
- Doctrine: <http://www.doctrine-project.org/>
- Propel: <http://www.propelorm.org/>
- YAML: <http://www.yaml.org/>
- ICU-Standard: <http://site.icu-project.org/>
- Selenium: <http://seleniumhq.org/>
- JQuery: <http://jquery.com/>
- Prototype: <http://www.prototypejs.org/>
- script.aculo.us: <http://script.aculo.us/>