

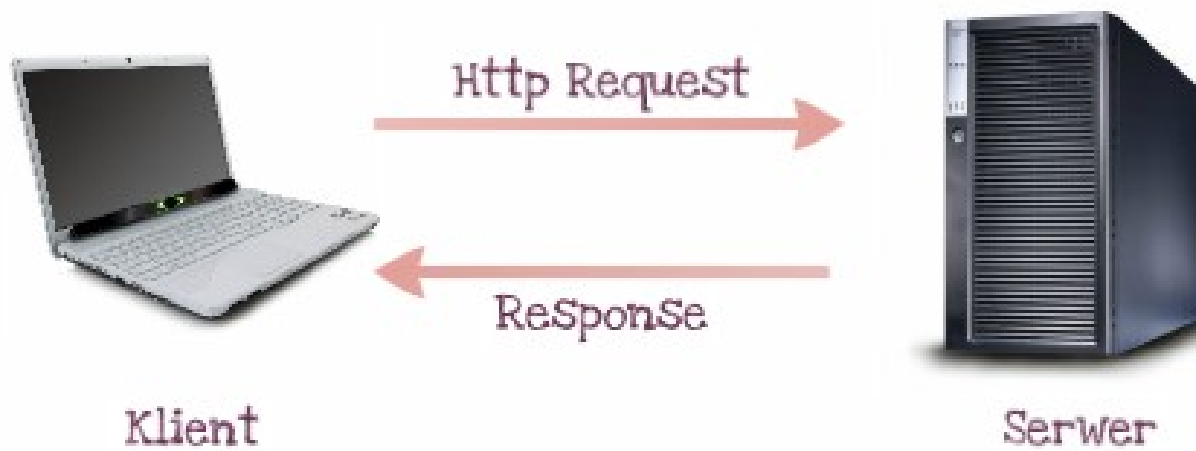
**Ewolucja projektowania
aplikacji
w PHP na bazie frameworka
Symfony 2**



Symfony

Statyczne strony HTML

Wczytanie statycznej strony HTML
sprowadza się do odebrania żądania przez
serwer, odnalezienia właściwego pliku i
wysłania odpowiedzi



Dynamiczne strony PHP

Dynamiczne strony internetowe są dodatkowo interpretowane.





- obiektowy język programowania zaprojektowany do generowania stron internetowych w czasie rzeczywistym
- 5-ty najpopularniejszy język świata
- wykorzystywany przez 77,9% wszystkich stron w Internecie

PHP umożliwia współpracę z wieloma rodzajami źródeł danych, takich jak systemy zarządzania bazami danych, pliki tek



flickr



YAHOO!



y WWW.

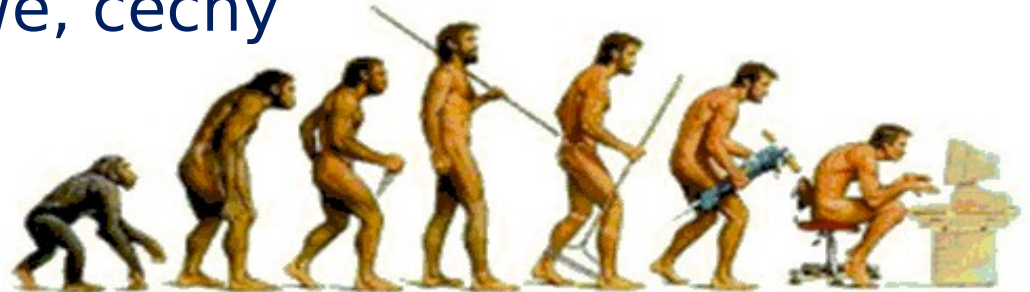
Co sprawia, że PHP jest tak popularny?

- Open-Source
- bardzo dobra dokumentacja
<http://www.php.net/manual/pl/>
- bardzo duża ilość dostępnych bibliotek
- dostępny na wielu platformach
- najłatwiejszy język programowania dla osób „nie-technicznych”



Ewolucja PHP

- PHP (1994r.)
- PHP 3 (1997r.)
- PHP 4 (2000r.)
- PHP 5(2004r.) – rewolucja w programowaniu aplikacji internetowych, era frameworków
- PHP 5.3, 5.4, 5.5 ...(2009) – przestrzenie nazw, funkcje anonimowe, cechy
- PHP 6 ?



Składnia PHP

Składnia PHP jest oparta na składni języków C, Java oraz Perl

```
<?php
$a = 5;
$b = 3;
function suma($zmienna1, $zmienna2)
{
    $wynik = $zmienna1 + $zmienna2;
    return $wynik;
}
echo suma($a, $b);
?>
```

Składnia PHP

```
<?php
class Przyklad
{
    public function powiedzCzesc($imie)
    {
        return 'Cześć, ` $imie;
    }
}
$objPrzyklad = new Przyklad;
echo $objPrzyklad->powiedzCzesc('Adam');
?>
```



```
1 <!DOCTYPE html>
2 <html lang="pl">
3   <head>
4     <meta charset="utf-8" />
5     <title>e-mail php</title>
6   </head>
7   <body>
8     <h1>E-mail v.0.1</h1>
9
10    <p>Zamówienia<p>
11
12    <form action="http://www.fizyka.umk.pl/~254134/UMK/PHP/2.php" method="post">
13      <p>Nazwisko:<br />
14        <input type="text" name="nazwisko" size="40" />
15      </p>
16
17      <p>Adres poczty elektronicznej:<br />
18        <input type="text" name="mail" size="40" />
19      </p>
20
21      <p>Komentarz:<br />
22        <textarea name="kom" cols="20" rows="10"></textarea>
23      </p>
24
25      <p>
26        <input type="submit" name="przycisk" value="Nacisnij aby wyslac" />
27      </p>
28    </form>
29
30  </body>
31 </html>
```

E-mail v.0.1

Zamówienia

Nazwisko:

Adres poczty elektronicznej:

Komentarz:

Nacisnij aby wyslac

```
<?php echo '<p>e-mail został wysłany </p>';  
    $to      = $_POST['mail'];  
    $subject = 'automatycznie wysyłany';  
    $message = $_POST['kom'];  
    $headers = 'From: '.$_POST['nazwisko'].'" za pomocą  
X-Mailer: PHP/" .phpversion();;  
  
    mail($to, $subject, $message, $headers);  
?>
```

Czym jest framework ?

- Framework jest szkieletem do budowy aplikacji
 - definiuje strukturę aplikacji oraz ogólny mechanizm jej działania, a także dostarcza zestaw komponentów i bibliotek ogólnego przeznaczenia do wykonywania określonych zadań
-
- ⊕ efektywność
 - ⊕ poprawa jakości kodu
 - ⊕ niezawodność
 - ⊖ złożoność
 - ⊖ wydajność

Symfony



- bazuje na wzorcu projektowym MVC
- programowanie zorientowane obiektowo
- niezależność od systemu bazodanowego
- zgodność z najlepszymi standardami
- walidacja formularzy i treści
- zarządzanie sesjami
- łatwość rozbudowy oraz możliwość integracji z innymi bibliotekami
- <http://symfony.com/what-is-symfony>

Symfony



- + system szablonów,
 - + automatyczne generowanie projektów
 - + pluginy
- trudna konfiguracja,
wymaga serwera

Kontroler

Kontroler to funkcja PHP, którą tworzy się, aby pobierała informacje z żądania HTTP, a następnie konstruowała i zwracała odpowiedź HTTP (jako obiekt Response Symfony2). Odpowiedź może być stroną HTML, dokumentem XML.

use

```
Symfony\Component\HttpFoundation\Response;
```

```
Public function helloAction()
```

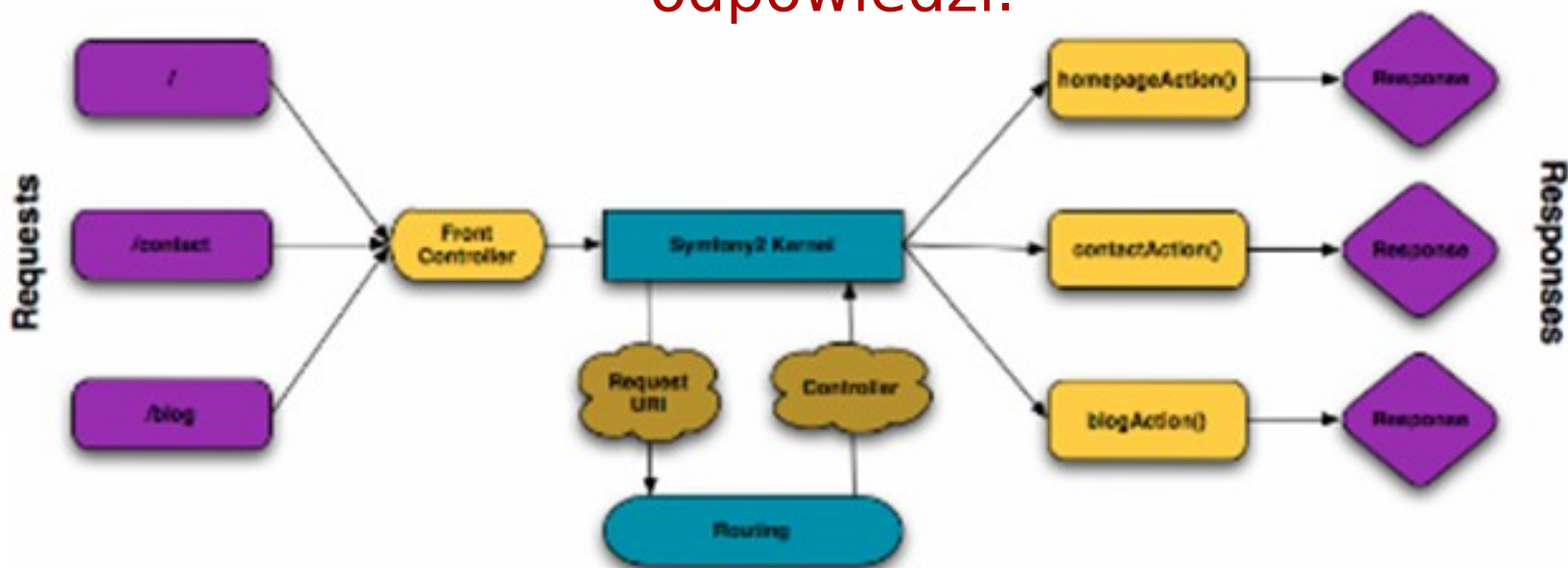
```
{
```

```
    return new Response('Hello world!');
```

```
}
```



Głównym zadaniem Symfony2 jest odbieranie żądań, odpowiadanie, ich przewarżanie i zwrócenie odpowiedzi.



Symfony 2 - innowacje

- **PHP 5.3**
- **Composer**
- **Twig**
- **Debug Toolbar**
- **Doctrine2 - mapowanie obiektów, zapisywanie obiektów**
- **DIC -Dependency Injection Container zarządzanie usługami(obiektami)**



Composer

Narzędzie do zarządzania zależnościami w PHP. Pozwala zadeklarować biblioteki potrzebne do danego projektu



Dokumentacja <http://getcomposer.org/doc/>

:

Twig

System szablonów napisany w PHP:

- szybki i prosty
- dziedziczenie szablonów
- dołączanie arkuszy stylów i skryptów
- JavaScript
- filtry
- możliwość definiowania makr

Szablony w PHP

```
<!DOCTYPE html>
<html>
  <head>
    <title>Welcome to Symfony!</title>
  </head>
  <body>
    <h1><?php echo $page_title ?></h1>

    <ul id="navigation">
      <?php foreach ($navigation as $item): ?>
        <li>
          <a href="<?php echo $item->getHref() ?>">
            <?php echo $item->getCaption() ?>
          </a>
        </li>
      <?php endforeach; ?>
    </ul>
  </body>
</html>
```

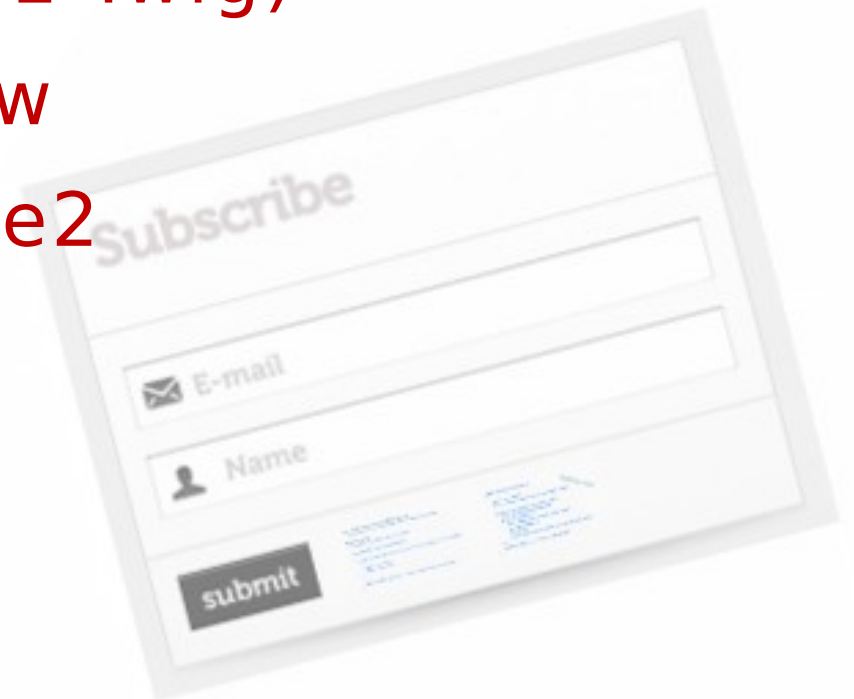
Szablony w Twig

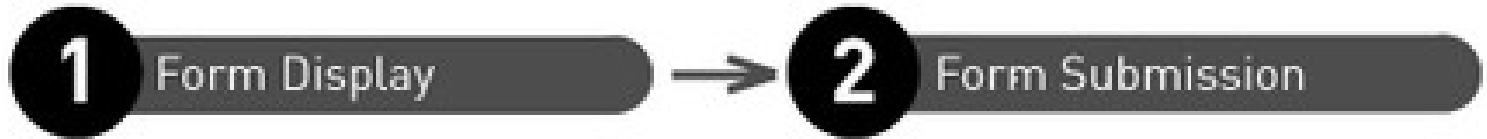
```
<!DOCTYPE html>
<html>
  <head>
    <title>Welcome to Symfony!</title>
  </head>
  <body>
    <h1>{{ page_title }}</h1>

    <ul id="navigation">
      {% for item in navigation %}
        <li><a href="{{ item.href }}">{{ item.caption }}</a></li>
      {% endfor %}
    </ul>
  </body>
</html>
```

Formularze

- złożony proces przetwarzania danych
- widgety (integracja z Twig)
- mapowanie obiektów
- integracja z Doctrine2





Name	<input type="text"/>
Email	<input type="text"/>
Message	<input type="text"/>
<input type="submit" value="Submit Query"/>	



- Name: Fabien
- Email: fabien.potencier@symfony-project.com
- Message: Hello World!

Widżety

Klasy SfForm i sfWidget

Użytkownicy wpisują dane w celu wypełnienia formularza. W Symfony, formularz jest obiektem dziedziczącym z klasy sfForm. W naszym przykładzie, stworzymy Formularz kontaktowy dziedziczący z klasy sfForm

```
// lib/form/ContactForm.class.php
class ContactForm extends sfForm
{
    public function configure()
    {
        $this->setWidgets(array(
            'name' => new sfWidgetFormInput(),
            'email' => new sfWidgetFormInput(),
            'message' => new sfWidgetFormTextarea(),
        ));
    }
}
```


Wyświetlenie formularza:

```
cd ~/PATH/TO/THE/PROJECT  
php symfony generate:module frontend  
contact
```

Teraz musimy stworzyć template, aby wyświetlić formularz w sposób pokazany na wcześniejszym rysunku

```
// apps/frontend/modules/contact/templates/indexSuccess.php  
<form action="<?php echo url_for('contact/submit') ?>" method="POST">  
  <table>  
    <?php echo $form ?>  
    <tr>  
      <td colspan="2">  
        <input type="submit" />  
      </td>  
    </tr>  
  </table>  
</form>
```

```
<form action="/frontend_dev.php/contact/submit" method="POST">
  <table>

    <!-- Beginning of generated code by <?php echo $form ?>
-->
    <tr>
      <th><label for="name">Name</label></th>
      <td><input type="text" name="name" id="name" /></td>
    </tr>
    <tr>
      <th><label for="email">Email</label></th>
      <td><input type="text" name="email" id="email" /></td>
    </tr>
    <tr>
      <th><label for="message">Message</label></th>
      <td><textarea rows="4" cols="30" name="message" id="message"></textarea></td>
    </tr>
    <!-- End of generated code by <?php echo $form ?>
-->

    <tr>
      <td colspan="2">
        <input type="submit" />
      </td>
    </tr>
  </table>
</form>
```

Instalacja i konfiguracja Symfony

Instalacja za pomocą composera:

```
php composer.phar create-project  
symfony/framework-standard-edition path/ 2.4.1
```

Biblioteki

```
php composer.phar install
```

Weryfikacja poprawności konfiguracji Serwera:

```
http://localhost/project/config.php
```

Konfiguracja

```
http://localhost/project/app.dev.php
```

Welcome!

Congratulations! You have successfully installed a new Symfony application.



[READ THE QUICK TOUR](#)



[CONFIGURE](#)



[RUN THE DEMO](#)

Documentation

[The Book](#)

[The Cookbook](#)

[The Components](#)

[Reference](#)

[Glossary](#)

Sensio

[Trainings](#)

[Books](#)

Community

[IRC channel](#)

[Mailing lists](#)

[Forum](#)

[Contributing](#)

Dziękuję za uwagę



Natalia Rybarczyk