Zend Framework od podstaw

Wykorzystaj gotowe rozwiązania PHP do tworzenia zaawansowanych aplikacji internetowych

Włodzimierz Gajda

Helion

Zend Framework od podstaw. Wykorzystaj gotowe rozwiązania PHP do tworzenia zaawansowanych aplikacji internetowych

Autor: Włodzimierz Gajda
ISBN: 978-83-246-3052-3
Format: 158×235, stron: 536

Elastyczne PHP – twórz nowe strony z wykorzystaniem Zend Framework!

- Tworzenie stron WWW w Zend Framework – zacznij od „Hello, World”
- Bazy danych i formularze – opanuj wszystkie elementy
- Zabezpieczanie dostępu do aplikacji – nie daj się zaskoczyć

Zend Framework to nowoczesna biblioteka ułatwiająca tworzenie stron WWW w języku PHP. Ten wygodny system pozwala projektantowi stron internetowych znacznie ograniczyć bądź nawet wyeliminować konieczność żmudnego wpisywania kodu na rzecz posługiwania się gotowymi elementami, niezależnie od tego, czy chce zaimplementować podstawowe mechanizmy aplikacji, czy też wzbogacić ją o konkretne funkcjonalności. Wsparcie ze strony twórców języka PHP, stabilna wersja, gotowa do pomocy społeczność programistów-entuzjastów oraz ogromna elastyczność to główne atuty tego rozwiązania, sprawiające, że jego popularność rośnie w dużym tempie.

Niniejsza publikacja ma za zadanie przybliżyć Ci Zend Framework, począwszy od absolutnych podstaw, aż po kwestie związane z pieczołowitym zabezpieczaniem dostępu do aplikacji. Znajdziesz tu jasne i czytelne przykłady zastosowania frameworka w różnych sytuacjach oraz propozycje gotowych rozwiązań konkretnych problemów programistycznych. Nauczysz się tworzyć zarówno proste strony WWW, jak i zaawansowane aplikacje, wymieniać szablony oraz implementować wtyczki. Poznasz sposoby tworzenia i wykorzystywania klas, współpracy z bazą danych, publikowania aplikacji w Internecie oraz używania formularzy. Krótko mówiąc, masz w ręku kompletny przewodnik po jednym z najlepszych frameworków PHP!

- Pierwszy projekt w Zend Framework i praca w środowisku NetBeans
- Tworzenie i usuwanie kontrolerów oraz akcji
- Wymiana szablonu HTML/CSS i dołączanie zewnętrznych zasobów
- Zasoby i implementacja inicjalizującej je wtyczki
- Bazy danych, tabele i relacje
- Identyfikacja rekordów na podstawie wartości slug
- Menu generowane na podstawie zawartości tabeli bazy danych
- Publikowanie aplikacji wykorzystującej bazę danych na serwerze hostingowym
- Przetwarzanie formularza, czyli implementacja interfejsu CRUD i dostosowywanie kodu HTML
- Walidatory oraz filtry i przesyłanie plików na serwer
- Zabezpieczanie haseł funkcjami skrótu
- Rejestracja i ograniczanie uprawnień użytkowników
- Modularyzacja aplikacji

Programuj swobodnie, wykorzystując uniwersalne biblioteki PHP!

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 32 230 98 63
e-mail: helion@helion.pl

Spis treści

Rozdział 4.

Wymiana szablonu HTML/CSS

Teraz zajmiemy się udekorowaniem aplikacji gotowym szablonem HTML/CSS. Najpierw podzielimy widoki na dwa fragmenty: układ oraz treść. Wyodrębniony układ będzie zapisany w jednym pliku *layout.phtml*, widoki akcji będą zaś generowały wyłącznie treść zawartą na stronie. W ten sposób uprościmy nadanie spójnego wyglądu wszystkim stronom aplikacji.

Dekorację aplikacji pisanej w ZF szablonem HTML/CSS rozpoczynamy od wydania polecenia:

```
zf enable layout
```

W wyniku wydania powyższej komendy w folderze aplikacji pojawi się plik:

```
application/layouts/scripts/layout.phtml
```

zaś w pliku *configs/application.ini* pojawi się wpis:

```
resources.layout.layoutPath = APPLICATION_PATH "/layouts/scripts/"
```

Powyższa reguła modyfikuje przebieg wykonania aplikacji: generowanie strony WWW wysyłanej do przeglądarki będzie odbywało się przy użyciu pliku *layout.phtml*[1].

W pliku *layout.phtml* umieszczamy szablon strony WWW, czyli m.in. znaczniki `html`, `head` i `body` oraz elementy ustalające podział na nagłówek, menu, treść czy panele (np. `div`). Treść zawarta na stronie WWW jest generowana przez widoki akcji, które tym razem nie zawierają już elementów `html`, `head` czy `body`.

[1] Jeśli instrukcję zawartą w pliku *application.ini* zakomentujesz, poprzedzając ją średnikiem:

```
;resources.layout.layoutPath = APPLICATION_PATH "/layouts/scripts/"
```

wówczas plik *layout.phtml* nie będzie wykorzystywany. Generowanie stron WWW będzie przebiegało identycznie jak w pierwszych trzech rozdziałach. Kod HTML wysyłany do przeglądarki będzie w całości pochodził z widoku akcji.

Przeanalizujmy stronę WWW, która była generowana w przykładzie trzecim jako wynik przetwarzania akcji `wierszyk/pokaz` (listing 3.3). W widoku akcji zawarty był kompletny kod HTML strony wysyłanej do przeglądarki.

Wprowadzając podział na szablon i treść, plik z listingu 3.3 dzielimy na dwa oddzielne pliki: szablon *layout.phtml* widoczny na listingu 4.1 oraz widok przedstawiony na listingu 4.2.

Listing 4.1. *Szablon layout.phtml dla strony z listingu 3.3*

```
<!DOCTYPE...>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
  <head>
    <title>Wyliczanka</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  </head>
<body>
<?php echo $this->layout()->content ?>
</body>
</html>
```

Listing 4.2. *Treść wyodrębniona ze strony 4.1*

```
<h1>Ene, due</h1>
<p>
Ene, due, rike, fake<br />
Torbe, borbe, ósme, smake<br />
Eus, deus, kosmateus<br />
I morele baks.
</p>
```

Zwróć uwagę, że szablon aplikacji widoczny na listingu 4.1 zawiera specjalną instrukcję:

```
<?php echo $this->layout()->content ?>
```

Powoduje ona umieszczenie w szablonie wyniku przetwarzania akcji. W ten sposób podczas wykonywania aplikacji dwa oddzielne pliki z listingów 4.1 oraz 4.2 są łączone w jeden dokument z listingu 3.3, który jest następnie wysyłany do przeglądarki.

Przykład 4. Wierszyk pt. Dwa kabele

Przygotuj stronę WWW, która będzie prezentowała treść wiersza pt. *Dwa kabele*. Zadanie wykonaj tak, by wierszyk był prezentowany na stronie akcji `index/index`. W rozwiązaniu utwórz szablon *layout.phtml*. Układ strony WWW zapisz w pliku *layout.phtml*, treść wiersza zaś — w widoku akcji `index/index` (tj. w pliku *index.phtml*).

Wykorzystaj szablon zawarty w folderze *html-css-template/*. Treść wiersza oraz folder *html-css-template/* są zawarte w pliku *04-start.zip*, który znajdziesz pod adresem *ftp://ftp.helion.pl/przyklady/zendfr.zip*. Wykonana strona powinna wyglądać tak jak na rysunku 4.1.

Rysunek 4.1. *Strona z wierszem pt. Dwa kabele*

ROZWIĄZANIE

Krok 1. Utwórz nowy projekt ZF

Utwórz nowy projekt PHP, który wykorzystuje ZF. Projekt nazwij `dwa-kabele`. Utworzenie nowego projektu będzie równoważne wykonaniu komendy:

```
zf create project . dwa-kabele
```

Następnie w pliku *public/.htaccess* dodaj reguły:

```
SetEnv APPLICATION_ENV development
DirectoryIndex index.php
```

W zależności od swoich preferencji wszystkie projekty omówione w książce możesz wykonać w wierszu poleceń lub w środowisku NetBeans.

Krok 2. Włącz przetwarzanie szablonu layout.phtml

Wydaj komendę:

```
zf enable layout
```

W folderze aplikacji pojawi się plik *application/layouts/scripts/layout.phtml*. Jego zawartość jest przedstawiona na listingu 4.3.

Listing 4.3. *Domyślna zawartość pliku layout.phtml*

```
<?php echo $this->layout()->content ?>
```

Krok 3. Dostosuj widok akcji index/index

W widoku akcji `index/index` wprowadź treść wierszyka. Zarys pliku *application/views/scripts/index/index.phtml* jest przedstawiony na listingu 4.4.

Listing 4.4. *Fragment widoku index.phtml*

```
<h2>Włodzimierz Gajda</h2>
<h3>Dwa kabele</h3>
<p>
Czasem tak sie dziwnie składa,<br />
Że gdy nic nie zapowiada<br />
Żadnych nieszczęść czy frustracji,<br />
Jakiś smyk wkroczy do akcji<br />
I, być może, bez złych chęci,<br />
Sielankę ojcu zamąci.<br />
<br />
...
```

Krok 4. Zmień szablon HTML/CSS

W pliku *layout.phtml* dodaj znaczniki HTML ustalające wygląd generowanej strony WWW. Zadanie mamy rozwiązać, wykorzystując szablon zapisany w folderze *03-start/html-css-template/*. Plik *04-start/html-css-template/index.html* jest przedstawiony na listingu 4.5.

Listing 4.5. *Szablon index.html, którego chcemy użyć w projekcie dwa-kabele*

```
<!DOCTYPE...>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
  <head>
    <title>template</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="css/style.css" media="screen" />
    <link rel="stylesheet" type="text/css" href="css/print.css" media="print" />
  </head>
<body>

<div id="pojemnik">
    <h1 id="naglowek">Wiersze i wierszyki...</h1>
    <div id="tekst">
        <p>
            Lorem ipsum...
        </p>
    </div>
```

```
    <div id="dol"></div>
</div>

</body>
</html>
```

Znaczniki z listingu 4.3 należy umieścić w pliku *layout.phtml*, tak jak pokazano na listingu 4.6.

Listing 4.6. *Szablon layout.phtml, który spowoduje, że strona z wierszem będzie wyglądała tak jak na rysunku 4.1*

```
<!DOCTYPE...>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl" lang="pl">
  <head>
    <title>Wierszyk</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="css/style.css" />
    <link rel="stylesheet" type="text/css" href="css/print.css" media="print" />
  </head>
<body>

<div id="pojemnik">
  <h1 id="naglowek">Wiersze i wierszyki...</h1>
  <div id="tekst">
    <?php echo $this->layout()->content; ?>
  </div>
  <div id="dol"></div>
</div>

</body>
</html>
```

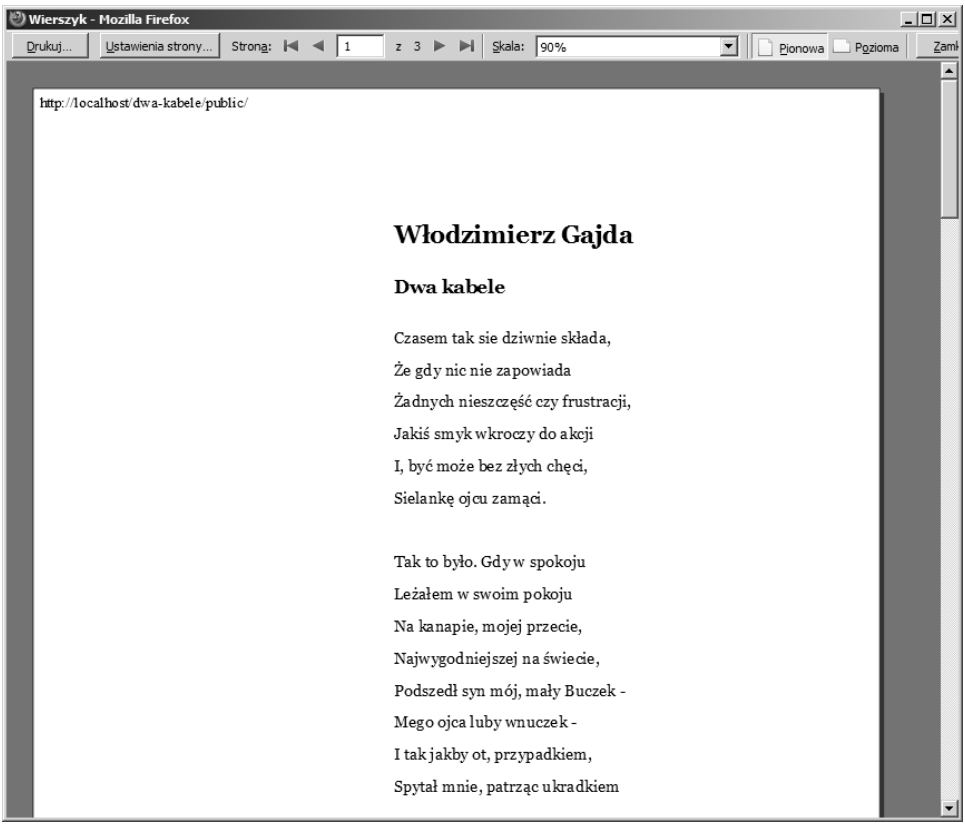
Kod HTML generowanej strony WWW należy sformatować stylami CSS zawartymi w plikach *html-css-template/css/style.css* oraz *html-css-template/css/print.css*. Style z pliku *style.css* są przeznaczone dla monitorów, a style z pliku *print.css* — dla urządzeń drukarek. Pliki *style.css* oraz *print.css* kopiujemy z folderu *html-css-template/css/* do folderu *public/css/*[2]. Modyfikowanie szablonu aplikacji kończymy, kopiując pliki graficzne z folderu *html-css-template/images/* do folderu *public/images/*.

W celu sprawdzenia wyglądu aplikacji odwiedź za pomocą przeglądarki adres:

```
http://localhost/dwa-kabele/public/
```

Powinieneś ujrzeć stronę taką jak na rysunku 4.1. Jeśli chcesz poznać funkcję stylów *print.css*, wykonaj w przeglądarce operację *Plik/Podgląd wydruku*. Ujrzysz taką stronę jak na rysunku 4.2.

[2] Foldery *public/css/* oraz *public/images/* należy utworzyć samodzielnie.



Rysunek 4.2. *Podgląd wydruku strony z rysunku 4.1*

Dzięki zastosowaniu w pliku *print.css* właściwości:

```
display: none
```

elementy dekoracyjne zostały usunięte i nie pojawią się na wydruku.

Pliki graficzne tworzące szablon HTML/CSS z rysunku 4.1 nazywają się *gora.png*, *naglowek.png*, *pojemnik.png* i *stopka.png*. Pliki te nie występują w kodzie HTML, a wyłącznie w kodzie CSS. W projekcie nie korzystamy ze znacznika `img`:

```
<img src="" alt="" />
```

Pliki PNG stanowią tło wybranych elementów HTML i są dołączane do strony przy użyciu właściwości `background`, np.:

```
body {
    background: #e8edef url('../images/gora.png') repeat-x;
    margin: 0;
    font-family: Georgia, serif;
}
```

Ścieżki do plików graficznych są względne. Prowadzą z pliku *style.css* do folderu *images/*.

Adresy strony z wierszem

W wykonanej aplikacji strona akcji `index/index` jest dostępna pod sześcioma adresami URL:

```
http://localhost/dwa-kabele/public/
http://localhost/dwa-kabele/public/index.php
http://localhost/dwa-kabele/public/index.php/
http://localhost/dwa-kabele/public/index.php/index
http://localhost/dwa-kabele/public/index.php/index/index
http://localhost/dwa-kabele/public/index.php/index/index/
```

Tylko dwa pierwsze adresy będą powodowały wyświetlenie strony WWW ozdobionej stylami CSS. Cztery dolne adresy, czyli:

```
http://localhost/dwa-kabele/public/index.php/
http://localhost/dwa-kabele/public/index.php/index
http://localhost/dwa-kabele/public/index.php/index/index
http://localhost/dwa-kabele/public/index.php/index/index/
```

będą wyświetlały stronę WWW pozbawioną stylów CSS. Dzieje się tak dlatego, że ścieżka prowadząca do stylów CSS jest zapisana w pliku *layout.phtml* jako względna:

```
<link rel="stylesheet" type="text/css" href="css/style.css" />
```

Style zostaną więc odnalezione wyłącznie wtedy, gdy znajdują się w stosunku bieżącego adresu URL w podfolderze *css/*.

Skorowidz

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

Ś

T

U

V

W

Z

Programuj swobodnie, wykorzystując uniwersalne biblioteki PHP!

Zend Framework to nowoczesna biblioteka ułatwiająca tworzenie stron WWW w języku PHP. Ten wygodny system pozwala projektantowi stron internetowych znacznie ograniczyć bądź nawet wyeliminować konieczność żmudnego wpisywania kodu na rzecz posługiwania się gotowymi elementami, niezależnie od tego, czy chce zaimplementować podstawowe mechanizmy aplikacji, czy też wzbogacić ją o konkretne funkcjonalności. Wsparcie ze strony twórców języka PHP, stabilna wersja, gotowa do pomocy społeczność programistów-entuzjastów oraz ogromna elastyczność to główne atuty tego rozwiązania, sprawiające, że jego popularność rośnie w dużym tempie.

Niniejsza publikacja ma za zadanie przybliżyć Ci Zend Framework, począwszy od absolutnych podstaw, skończywszy na kwestiach związanych z pieczołowitym zabezpieczaniem dostępu do aplikacji. Znajdziesz tu jasne i czytelne przykłady zastosowania frameworka w różnych sytuacjach oraz propozycje gotowych rozwiązań konkretnych problemów programistycznych. Nauczysz się tworzyć zarówno proste strony WWW, jak i zaawansowane aplikacje, wymieniać szablony oraz implementować wtyczki. Poznasz sposoby tworzenia i wykorzystywania klas, współpracy z bazą danych, publikowania aplikacji w internecie oraz używania formularzy. Krótko mówiąc, masz w ręku kompletny przewodnik po jednym z najlepszych frameworków PHP!

- Pierwszy projekt w Zend Framework i praca w środowisku NetBeans
- Tworzenie i usuwanie kontrolerów oraz akcji
- Wymiana szablonu HTML/CSS i dołączanie zewnętrznych zasobów
- Zasoby i implementacja inicjalizującej je wtyczki
- Bazy danych, tabele i relacje
- Identyfikacja rekordów na podstawie wartości slug
- Menu generowane na podstawie zawartości tabeli bazy danych
- Publikowanie aplikacji wykorzystującej bazę danych na serwerze hostingowym
- Przetwarzanie formularza, czyli implementacja interfejsu CRUD i dostosowywanie kodu HTML
- Walidatory oraz filtry i przesyłanie plików na serwer
- Zabezpieczanie haseł funkcjami skrótu
- Rejestracja i ograniczanie uprawnień użytkowników
- Modularyzacja aplikacji

Cena 89,00 zł

ISBN 978-83-246-3052-3

9 788324 630523

Informatyka w najlepszym wydaniu