



Krzysztof Palikowski

Własna strona WWW?
Sklep internetowy lub forum?
Z Drupalem to możliwe!

Drupal

**Poznaj go
z każdej strony!**



Jak sprawnie poruszać się w środowisku systemu Drupal?
Jak pracować z modułami, skórkami i zawartością witryny?
Jak od podstaw zbudować i uruchomić własny serwis?

» Idź do

- Spis treści
- Przykładowy rozdział

» Katalog książek

- Katalog online
- Zamów drukowany katalog

» Twój koszyk

- Dodaj do koszyka

» Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

» Czytelnia

- Fragmenty książek online

» Kontakt

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 32 230 98 63
e-mail: helion@helion.pl
© Helion 1991–2010

Drupal. Poznaj go z każdej strony!

Autor: Krzysztof Palikowski
ISBN: 978-83-246-2652-6
Format: 158×235, stron: 496



Własna strona WWW? Sklep internetowy lub forum? Z Drupalem to możliwe!

- Jak sprawnie poruszać się w środowisku systemu Drupal?
- Jak pracować z modułami, skórkami i zawartością witryny?
- Jak od podstaw zbudować i uruchomić własny serwis?

Jak twierdzi autor tej książki, Drupal to coś więcej niż CMS. I ma rację, biorąc pod uwagę fakt, że ten system zarządzania treścią umożliwia to, co nie zawsze łatwe jest w przypadku innych systemów, takich jak Joomla! czy Wordpress – swobodę tworzenia w pełni dopasowanych stron. Dzięki modularnej strukturze i tysiącom dodatków Drupal zapewnia projektantom niemal nieograniczone możliwości kreowania funkcjonalności i wyglądu witryny. Ten potencjał w połączeniu z opinią niezawodnego i elastycznego systemu oraz wielkim wsparciem światowej społeczności sprawił, że wciąż mało popularny w Polsce Drupal na świecie wyrósł na lidera rynku CMS. Dowód? W oparciu o niego powstały tak prestiżowe witryny, jak strona amerykańskiego Białego Domu, oficjalny serwis miasta Londyn, strona króla Belgii czy polskiego Ministerstwa Gospodarki. Jednocześnie od 2007 roku Drupal regularnie zdobywa tytuł najlepszego CMS w licznych konkursach.

I choć jest to system oparty na otwartym kodzie źródłowym, nic za darmo! Drupal uchodzi bowiem za jeden ze skryptów trudniejszych do opanowania. Aby więc ułatwić sobie to zadanie, podejmij naukę z kompletnym podręcznikiem, dzięki któremu szybko i bez trudu ujarzmisz ten niezwykle system. Błyskawicznie opanujesz jego środowisko i przejdiesz do budowania własnej, profesjonalnej witryny. Przeczytasz o tym, jak opracować strukturę menu i wykonać tłumaczenie interfejsu oraz dowiesz się wszystkiego na temat modułów, „skórek” i szablonów. Nauczysz się sprawnie zarządzać zawartością strony i jej użytkownikami oraz optymalnie ją konfigurować. Krok po kroku prześledzisz proces budowania przykładowego serwisu z wykorzystaniem modułów CCK i Views – zobaczysz, jak atrakcyjnie prezentować treść, zapewnić stronie pełnię funkcjonalności i estetyczny wygląd oraz zadbać o jej bezpieczeństwo. Dowiesz się, jak poprawnie uruchomić swoją witrynę i administrować nią, a także poznasz rozwiązania typowych problemów, na jakie możesz trafić przy każdym z realizowanych etapów.

Przekonaj się, że nawet tak potężne narzędzie jak Drupal z dobrym podręcznikiem staje się łatwe do opanowania!

Spis treści

Rozdział 1. Drupal: podstawowe informacje	15
Wprowadzenie	15
Co to jest ten Drupal?	15
Dlaczego właśnie Drupal?	16
Czy z Drupalem odniosę sukces?	17
Drupal od strony technicznej	20
Pliki systemowe	20
Baza danych	21
Pliki użytkowników	22
CMS czy CMF?	22
Drupal jako projekt	23
Trochę historii	24
Trochę liczb	24
Społeczność i wsparcie	25
Polska społeczność	26
Zasoby pomocne w nauce i poznawaniu Drupala	26
Podręcznik	27
IRC	27
Fora i grupy dyskusyjne	27
Blogi i mikroblogi	28
Podcasty	28
Screencasty	28
Rozwiązywanie problemów	29
Przyszłość Drupala	30
Drupal 7	30
Rozwój firm świadczących usługi	32
Dystrybucje	33
Przewidywania dotyczące przyszłości	37
Licencje i koszty	37
Licencje	37
Koszty	39
Podsumowanie	40
Rozdział 2. Poznajemy środowisko Drupala	41
Wprowadzenie	41
Podstawy HTML i CSS	42
HTML	42
CSS	44
Dokumenty statyczne i dynamiczne	45

Podstawy serwowania stron WWW	46
Jak serwowane są strony WWW?	46
Jak działa serwer WWW?	47
Drupal jako baza danych z interfejsem WWW	49
Budujemy środowisko testowe do nauki Drupala	50
Instalacja XAMPP LITE	50
Weryfikujemy działanie serwera stron WWW	53
Weryfikujemy działanie interpretera PHP	53
Weryfikujemy działanie serwera bazy danych	54
Tworzymy bazę danych dla przyszłej instalacji systemu Drupal	54
Instalujemy Drupala	57
Pobieramy potrzebne pliki	57
Poznajemy systemowe pliki Drupala	58
Pobieramy tłumaczenie i instalujemy Drupala	60
Poszczególne kroki instalacji	61
Podsumowanie	66

Rozdział 3. Pierwsze kroki po instalacji 67

Wprowadzenie	67
Ważne terminy	67
Moduł	67
Rodzaj zawartości	68
Segment	69
Strony niebędące segmentami	70
Skórka i szablon	70
Blok i obszar skórki	71
Ranga	71
Ścieżka do strony	72
Ścieżka do pliku	73
Co widać na stronie powitalnej	74
Dział zarządzania	75
Zarządzanie zawartością	75
Budowa witryny	76
Zarządzanie użytkownikami	76
Konfiguracja witryny	76
Raporty	76
Moduły dodatkowe	76
Wymijmy z szafki wszystkie zabawki	77
Czym są moduły	78
Skąd pobierać moduły	79
Włączamy moduł zewnętrzny	80
Skórki	81
Skórki systemowe	81
Skórki dodatkowe	82
Dodawanie zawartości	82
Strona „Dodaj zawartość”	83
Artykuł (wpis na stronie głównej)	84
Strona z linkiem w menu	86
Dodawanie obrazków do wpisu	87
Atrybuty segmentu	90
Co dalej	91
Uprawnienia	91
Odpowiedzi i ich moderacja	92
Rejestracja i logowanie	92
Podsumowanie	93

Rozdział 4. Budowa witryny	95
Wstęp	95
Aliaszy adresów	95
Bloki	97
Bloki dostępne po instalacji	99
Konfiguracja bloku	101
Własne bloki	103
Formularz kontaktowy	105
Menu	107
Struktura menu	107
Dodawanie elementu do menu	108
Dodawanie menu	110
Bloki zawierające menu	111
Ustawienia modułu menu	112
Moduły	112
Zależności modułów	114
Instalowanie nowych modułów	115
Wersje modułów	115
Strony modułów (projektów)	117
Wsparcie do modułów	120
Przestrzeń nazw	121
Aktualizacja modułów	121
Odinstalowywanie modułów	122
Skórki	123
Dodawanie skórek	123
Konfiguracja skórek	123
Ustawienia globalne skórek	124
Konfiguracja przykładowej skórki — Garland	126
Skórka działu zarządzania	127
Tłumaczenie interfejsu	127
Jak sprawdzić, w jakim stopniu nasz interfejs jest zlokalizowany	128
Jak zdobyć pliki .po	128
Jak zaimportować plik z tłumaczeniem do naszej strony	129
Wyszukiwanie i tłumaczenie pojedynczych napisów	129
Zmienne w napisach	130
Eksport i obróbka plików .po	131
Struktura plików .po	132
Formuła liczby mnogiej	134
Wyzwalacze	134
Podsumowanie	136
Rozdział 5. Zarządzanie zawartością	137
Wprowadzenie	137
Rodzaje zawartości	137
Nowy rodzaj zawartości	143
Rodzaje (typy) zawartości raz jeszcze	144
Podstawowe atrybuty segmentu	145
Prezentacja segmentów	146
Skrót i pełna wersja wpisu	147
Różnice między rodzajami zawartości	148
Kategorie	150
Dodawanie terminów	152
Adresy z listą wpisów przypisanych do terminu	154

Zawartość	154
Ustawienia dotyczące wpisów	156
Książki	157
Fora	160
Odpowiedzi	161
Kanały RSS	162
Subskrybent kanałów	163
Bloki	164
Podsumowanie	165
Rozdział 6. Konfiguracja witryny	167
Wprowadzenie	167
Podstawowe dane	167
Przerwa techniczna	168
Formaty danych	169
Konfiguracja formatu	170
Konfiguracja filtrów	171
Kolejność filtrów	172
Języki	173
Przyjazne adresy	175
Skórka działu zarządzania	176
System plików	176
Wysyłanie plików	177
API blogów	181
Data i czas	182
Działania zautomatyzowane	184
Wstrzymywanie publikacji na podstawie słów kluczowych	185
Wyświetlanie komunikatu dla użytkownika	186
Wysyłanie e-maila	186
Zmienne w akcjach	187
Dzienniki zdarzeń i alerty	188
Dzienniki zdarzeń w bazie danych	188
Syslog	188
Narzędzie przetwarzania grafik	191
Instalacja ImageMagick	192
Powiadamianie o błędach	192
Ustawienia wyszukiwarki	193
Wydajność	195
Zapobieganie przeciążeniu	197
Podsumowanie	198
Rozdział 7. Zarządzanie użytkownikami	199
Wprowadzenie	199
Rangi	200
Uprawnienia	201
Przykład deklaracji uprawnień	201
Za mało uprawnień?	203
Najważniejsze uprawnienia	203
Profile	206
Dodatkowe pola w profilach użytkowników	207
Reguły dostępu	211
Ustawienia dotyczące użytkowników	213
Tworzenie kont	213
Logowanie z OpenID	214
Wiadomości do użytkowników	215

Użytkownicy	216
Dodawanie nowego użytkownika	217
Użytkownicy testowi	217
Podsumowanie	218
Rozdział 8. Raporty	219
Wprowadzenie	219
Raport o stanie witryny	220
Nowe wersje	221
Ostatnie wpisy do dziennika	222
Najczęstsze błędy nieznalezienia strony	223
Najczęstsze błędy odmowy dostępu	224
Ustawienia dziennika odwiedzin	225
Najczęściej odsyłające strony	226
Najpopularniejsze słowa kluczowe	226
Ostatnie odsłony	226
Najpopularniejsze strony	227
Najczęściej odwiedzający	228
Podsumowanie	228
Rozdział 9. Budujemy przykładowy serwis	229
Wstęp	229
Typowy cykl życia projektu internetowego	229
Wymagania funkcjonalne	230
Projekt graficzny i rozplanowanie elementów	230
Wybór rozwiązania	231
Projekt techniczny	231
Przeniesienie na środowisko produkcyjne	232
Utrzymanie i rozwój	232
Serwis „Nasza Okolica”	232
Konfiguracja strony	232
Przygotowanie systemu	233
Moduły	233
Dane podstawowe	235
Konfiguracja skórki	235
Wydajność	235
Ranga „pomocnik redakcji”	236
Format Full HTML dla pomocników	237
Ustawienia użytkowników	238
Uprawnienia	239
Testowi użytkownicy	240
Forum dyskusyjne	241
Profile użytkowników	242
Menu	244
Pierwszy artykuł na stronę główną	246
Pierwsza strona	246
Błogi	247
Formularz kontaktowy	247
Wyszukiwarka	249
Podsumowanie	249
Rozdział 10. Zawartość serwisu	251
Wstęp	251
Nowe rodzaje zawartości	252
Rodzaj zawartości Wieści z miasta	252
Rodzaj zawartości Zdjęcie	255

Rodzaj zawartości Ogłoszenie	255
Uprawnienia dodatkowe	256
Kategoryzacja treści	257
Instalacja CCK i innych potrzebnych modułów	258
Spolszczenie CCK	260
Przegląd możliwości CCK	260
Nowe pola w typach zawartości	262
Pole Telefon w typie zawartości Ogłoszenie	262
Pole Obraz w typie zawartości Wieści z miasta	265
Współdzielenie pól między rodzajami zawartości	269
Zmiana etykiety i kontrolki pola już po jego utworzeniu	270
Pole Obraz w typie zawartości Zdjęcie	271
Pole Powiązane zdjęcia w typie zawartości Wieści z miasta	272
Uprawnienia do pól — moduł Content Permissions	276
Wprowadzanie i edycja zawartości	277
Formularz dodawania i edycji zawartości	277
Pola i sekcje formularza dodawania treści	279
Kolejność pól	279
Wersje	279
Formaty danych i filtry	281
Zawartość testowa	285
Drobne poprawki w strukturze zawartości	286
Uprawnienia do pól CCK	287
Podsumowanie	288
Rozdział 11. Prezentacja treści	289
Wstęp	289
Wpis w wersji skróconej i pełnej	289
Wersja skrócona (podsumowanie)	289
Wersja pełna segmentu	290
Jak zmienić wygląd segmentów	290
CCK i formatery	290
Skin i Fusion	292
Display Suite	292
Panels	293
ImageCache	293
Moduł ImageCache	293
Uprawnienia	296
Konfiguracja formaterów dla rodzaju zawartości Ogłoszenie	297
Położenie plików obsługiwanych przez ImageField i ImageCache	298
Konfiguracja formaterów dla rodzaju zawartości Zdjęcie	300
Odrobina CSS	301
Jak zmienić wygląd wykazów i list	302
Wykazy segmentów	302
Wykazy i tabele elementów	303
Własne listy i wykazy	304
Moduł Views	304
Przykłady widoków	305
Instalacja i polonizacja modułu Views	307
Views i wbudowane widoki	307
Interfejs modułu Views	307
Formaty	310
Obszar Podstawowe ustawienia	312
Ikona kółka zębatego	312

Ikony plusa i strzałek	314
Obszar Filtry	315
Obszar Pola	316
Obszar Kryteria sortowania	317
Obszar Powiązania	318
Obszar Argumenty	319
Budowanie widoków dla strony Nasza Okolica	320
Widok Ostatnio dodane zdjęcia	320
Widok Ogłoszenia	326
Blok 5 ostatnich ogłoszeń	328
Podsumowanie	331
Rozdział 12. Moduły i przepisy	333
Wstęp	333
Popularne moduły zewnętrzne	333
Content Construction Kit (CCK)	334
Views	335
Flag	337
Token	337
Pathauto	338
Administration menu	338
FileField	339
ImageCache	339
Rules	340
Przepisy na konkretne funkcje	340
Jak wybrać odpowiednie moduły	341
Galeria zdjęć	342
Wiki	343
Sklep internetowy	344
Biuletyny	345
Forum dyskusyjne	345
Grupy	347
Zarządzanie projektami, zadaniami, zgłoszeniami	347
Formularze i zbieranie danych	348
Katalogi	348
Zabezpieczenie przed spamem	348
System CAPTCHA	349
Systemy antyspamowe	350
Blokowanie IP	350
Moderacja treści	351
Regulamin	351
Podsumowanie	351
Rozdział 13. Wygląd	353
Wprowadzenie	353
Czym są skórki	353
Skórki to nie szablony	354
Skąd pobierać gotowe projekty	355
Skórki komercyjne	355
Szablony HTML	356
Inne źródła	356
Samodzielna produkcja	356
Drupal liderem słabych skórek?	357
Rodzynki, czyli ładne skórki do Drupala	358

Co składa się na wygląd drupalowej strony	359
Treść, forma, funkcje	359
Kaskadowe arkusze stylów (CSS)	360
Obszary i bloki	361
Obszar treści i wygląd segmentów różnego typu	364
Silnik PHPTemplate	367
Czym są silniki skórek	367
Skórka i szablon — to nie to samo!	367
Skórki PHPTemplate	368
Instalacja skórek	368
Szablony i funkcje szablonów	369
Przykładowy szablon	369
Nadpisywanie funkcji i szablonów	370
Reguły nazewnictwa plików przy nadpisywaniu szablonów	371
Rozwiązywanie konfliktów szablonów	372
Szablony w rdzeniu i możliwości ich nadpisania	373
Zasady nadpisywania funkcji szablonów	375
Zmienne dostępne w szablonie page.tpl.php	376
Zmienne szablonu block.tpl.php	378
Zmienne szablonu comment.tpl.php	378
Zmienne szablonu node.tpl.php	379
Contemplate	380
Budowa skórek	381
Pliki składające się na skórę	381
Plik .info	383
Pliki typowych szablonów	386
Skórki potomne	388
Własna skórka	389
Przerabiana czy całkiem własna?	389
Idealna skórka na start — Zen	390
Robimy skórę potomną na podstawie Zen	391
Dostosowujemy skórę z szablonu HTML/CSS	394
Podsumowanie	397
Rozdział 14. Uruchomienie witryny	399
Wstęp	399
Instalacja Drupala w środowisku produkcyjnym	399
Wybór serwera WWW	400
Zakładamy konto WWW	402
Zakładamy bazę danych MySQL	403
Zakładamy skrzynkę mailową	405
Konfigurujemy domenę na serwerze	406
Odszukujemy katalog na pliki strony	407
Umieszczenie plików Drupala na serwerze — metoda FTP	409
Umieszczenie plików Drupala na serwerze — wiersz poleceń	411
Uruchamiamy instalator	412
Uprawnienia do plików	412
Sprawdzamy działanie poczty	413
Sprawdzamy parametry życiowe	413
Ustawiamy zadania cron	414
Ukrywamy stronę przed ciekawskimi	415
Drupal w trybie wielowitrynowym (multisite)	415
Wiele witryn, jeden Drupal	416
Znaczenie katalogu sites	417

Instalacja nowej witryny	419
Dodanie istniejącej witryny do instalacji wielowitrynowej	422
Konfiguracja współdzielenia tabel między serwisami	424
Instalacja wielowitrynowa na środowisku testowym (Windows)	427
Podsumowanie	429
Rozdział 15. Utrzymanie witryny	431
Wstęp	431
Zasada nienaruszalności kodu rdzenia	431
Kopie zapasowe	432
Bieżące prace przy witrynie	432
Aktualizacja systemu i modułów	432
Przenosiny na inny serwer	432
Zmiana adresu strony	433
Utrzymanie środowiska rozwojowego	433
Wykorzystanie SSH i Drush	433
Kopie zapasowe	434
Kopia plików z użyciem FTP	435
Kopia plików z użyciem Direct Admin	436
Kopia bazy danych z użyciem phpMyAdmin	437
Kopie bezpieczeństwa z poziomu skryptów PHP i modułów Drupala	438
Kopie z użyciem mechanizmów wbudowanych w panel zarządzania kontem serwera WWW	439
Kopia bazy danych z użyciem mysqldump	439
Kopie różnicowe plików z użyciem narzędzia rsync	442
Planowanie wykonania się kopii z użyciem crontab	445
Przywrócenie systemu z kopii bezpieczeństwa	446
Odtworzenie plików	446
Odtworzenie bazy danych z użyciem phpMyAdmin	447
Odtworzenie bazy danych z użyciem programu mysql	448
Aktualizacje	450
Skąd mam wiedzieć, że są nowe wersje?	450
Kiedy aktualizacja jest konieczna?	451
Proces aktualizacji — najważniejsze zasady	452
Aktualizacja systemu	453
Aktualizacja modułów	454
Aktualizacja skórki	455
Kiedy będę wiedział, że wszystkie elementy mojej strony są aktualne?	455
Uruchomienie środowiska testowego	456
Jeden serwer testowy — piaskownica	456
Więcej serwerów testowych	456
Jak przygotować piaskownicę	457
Piaskownica generowana automatycznie	458
Podsumowanie	459
Rozdział 16. Typowe problemy	461
Wprowadzenie	461
Popularne problemy	461
Komunikat „Internal server error 500”	461
Biała strona zamiast serwisu	462
Komunikat „Fatal error: Allowed memory size of X bytes exhausted (tried to allocate Y bytes)”	463
Komunikat „The translation import failed”	464
Komunikat „Maximum execution time of xx seconds exceeded”	465

Błąd 404 protokołu HTTP	465
Powolne działanie systemu na localhost	466
Połączenie z serwerem zostało zresetowane	466
Komunikat „Cannot modify header information — headers already sent by ...” ...	467
Błędne ścieżki do plików	467
Utrata hasła do konta administratora	469
Na serwerze z PHP 5.3 nie działa podgląd ani generowanie miniatur w ImageCache	470
Podsumowanie	471
Skorowidz	473

Rozdział 4.

Budowa witryny

Wstęp

W tym rozdziale omówimy szereg opcji, które decydują o tym, jak witryna ma się prezentować i z czego jest zbudowana. Wybierzemy więc skórkę, skonfigurujemy wyświetlane w jej obrębie bloki, zbudujemy nawigację, a także dokonamy tłumaczenia interfejsu. Dowiemy się, jak włączyć moduły, a dla dowolnej strony ustawić alternatywny adres (alias). Jest to zestaw opcji składających się na część działu zarządzania nazwanego *Budowa witryny*.

Aliasy adresów

.../admin/build/path

Alias jest alternatywną, ustawioną przez nas ścieżką (URL), której wywołanie spowoduje wyświetlenie strony pierwotnie znajdującej się pod innym adresem.

Każda strona naszej witryny ma swój unikalny adres. Przykładowo każdy segment jest dostępny po podaniu ścieżki *.../node/<numer kolejny segmentu>*, a więc *.../node/1*, *.../node/2* i tak dalej. Formularz kontaktowy znajdziemy pod adresem *.../contact*, a stronę logowania pod *.../user*.

Bywają sytuacje, że chcemy nadać inny adres jednej ze stron — bo obecny jest trudny do zapamiętania, nie wygląda zbyt dobrze albo wolelibyśmy zmienić słówko angielskie na polskie (*contact* — kontakt, *taxonomy/2* — kategoria/2). Do tego właśnie służy zakładanie aliasów adresów.



Jeśli interesuje nas automatyczne generowanie aliasów dla każdej dodawanej treści, kategorii czy użytkownika, powinniśmy zainteresować się modułem *pathauto*, który oferuje takie właśnie możliwości.

Klikamy *Dodaj alias*, co przeniesie nas na stronę `.../admin/build/path/add`. Następnie musimy wybrać, dla jakiej wersji językowej chcemy go założyć, a potem podać istniejący adres i jego alias (rysunek 4.1).

Rysunek 4.1.

Tworzenie aliasu, czyli niestandardowego adresu do strony

Obecna ścieżka: *

Obecna ścieżka, której ma być nadany alias. Na przykład: node/28, forum/1, taxonomy/term/1+2.

Alias adresu: *

Alternatywna ścieżka, poprzez którą będzie można uzyskać dostęp do danych. Jeśli właśnie tworzona strona będzie traktować o poezji, można wpisać "wiersze". Ścieżka powinna być względna i nie może kończyć się ukośnikiem.



Ważne, aby adres, do którego chcemy utworzyć alias, faktycznie istniał. W powyższym przykładzie użyto adresu `.../node/4`, ale jeśli nie utworzono jeszcze żadnych wpisów albo dodano jedynie trzy lub mniej, zapisanie powyższych opcji wygeneruje błąd.

Podczas dodawania i edycji aliasów podajemy, dla jakiej wersji językowej je tworzymy. O językach i ich obsłudze więcej dowiemy się w rozdziale 6. „Konfiguracja witryny”. Teraz niech wystarczy nam wiedza, że jeśli będziemy prowadzić taką stronę, to dla poszczególnych wersji językowych danego wpisu możemy założyć inne aliasy.

Bez kłopotu założymy aliasy zawierające polskie litery, spacje, czy nawet wykrzyknik. Jednak pewne znaki (np. znak zapytania, znak &, gwiazdka) mają specjalne znaczenie w adresacji używanej na serwerach WWW bądź w systemach operacyjnych i mogą spowodować błędne rozwiązywanie adresów — strona po wpisaniu `.../główna @#%!*^&*` nie zostanie znaleziona. Aby uniknąć niepotrzebnych problemów, zaleca się stosowanie w adresach tylko ograniczonego zestawu znaków — a – Z, 0 – 9, podkreślenia i myślnika.

Po dodaniu kilku aliasów możemy przejrzeć ich wykaz na stronie głównej zarządzania aliasami (rysunek 4.2).

Rysunek 4.2.

Wykaz aliasów

Alias ▲	System	Język	Czynności	
group/kult	node/4	Angielski	edytuj	usuń
grupa/kult	node/5	Polski	edytuj	usuń

Należy zwrócić uwagę na to, że system nie dopuści do sytuacji, kiedy mielibyśmy dwa takie same adresy. Próba założenia aliasu, który spowodowałby istnienie dwóch identycznych URL, zakończy się błędem. Nie ma jednak przeciwwskazań, aby kilka aliasów wskazywało na jeden URL, zatem dla formularza kontaktowego możemy zarezerwować kilka adresów — kontakt, napisz-do-nas, napisz i jakie tylko chcemy.

Warto zapamiętać, że aliasy będą działać również jako identyfikatory stron wpisywane w definiowaniu widoczności bloków — o czym już za chwilę w kolejnej części, gdzie omówimy system ich wyświetlania.

Bloki

`.../admin/build/block`

Blok to pewien obszar na stronie, zawierający zazwyczaj niewielką porcję treści (np. ostatnie komentarze), jakąś funkcjonalność (np. pole wyszukiwania) bądź inne dane (np. listę obecnie zalogowanych użytkowników), dostarczaną przez jeden z zainstalowanych w systemie modułów.

Bloki możemy umieszczać w zdefiniowanych dla każdej skórki obszarach, a także sterować ich widocznością na podstawie adresu, uprawnień czy zdefiniowanej przez nas funkcji PHP.

Dzięki zarządzaniu blokami możemy sterować wszystkimi tymi funkcjami, tak aby każdy blok wyświetlał się w pożądanym miejscu, w ściśle zdefiniowanych sytuacjach.

Bloki w większości są dostarczane przez moduły — w miarę jak będziemy instalować kolejne, ich wykaz stanie się bogatszy. Jeden moduł może dostarczać zero, jeden bądź więcej bloków. Sami możemy też dodać blok z dowolną treścią (zgodnie z wybranym formatem danych) bądź kodem PHP.

Po wejściu na tę stronę zarządzania zobaczymy wykaz dostępnych w systemie bloków, zawierający w kolejnych kolumnach:

- ♦ uchwyt do przeciągania do innego obszaru lub do zmiany kolejności bloku w danym obszarze,
- ♦ nazwę bloku,
- ♦ przypisanie do jednego z obszarów, jakie zdefiniował autor aktualnie używanej skórki,
- ♦ opcję wykorzystywania modułu zapobiegania przeciążeniu,
- ♦ link do konfiguracji,
- ♦ ewentualnie (w przypadku bloków stworzonych przez nas) link do usunięcia bloku.

Rozmieszczanie bloków jest realizowane metodą „przeciągnij i upuść” lub poprzez przydział do obszaru na podstawie listy wartości. Formularz zarządzania blokami (dostępny pod adresem `.../admin/build/block`) należy zapisać, aby zmiany dały jakiś efekt (rysunek 4.3).

Na rysunku widzimy, że każdy blok przydzielony jest do jednego z obszarów (w wypadku skórki Garland może to być *Lewa kolumna boczna*, *Prawa kolumna boczna*, *Zawartość*, *Nagłówek*, *Stopka* bądź *<brak>*, czyli blok nigdzie niewyświetlany).

Powyższy rysunek przedstawia też blok własny, który możemy utworzyć za pomocą formularza tworzenia nowego bloku (co opisano kilka stron dalej).

Rysunek 4.3.
Rozmieszczanie
bloków w obszarach

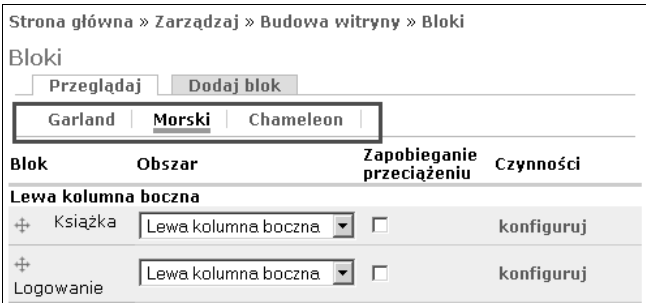
Blok	Obszar	Zapobieganie przeciążeniu	Czynności
Lewa kolumna boczna			
+ Komunikat do redakcji	Lewa kolumna boczna	☐	konfiguruj usuń
+ Najświeższa ankieta	Lewa kolumna boczna	☒	konfiguruj
+ Nawigacja	Lewa kolumna boczna	☐	konfiguruj
Prawa kolumna boczna			
+ Informacje o autorze	Prawa kolumna boczna	☒	konfiguruj
+ Subskrybuj	Prawa kolumna boczna	☐	konfiguruj
+ Logowanie	Prawa kolumna boczna	☐	konfiguruj
Zawartość			
+ Aktywne tematy	Zawartość	☐	konfiguruj
+ Książka*	Zawartość	☐	konfiguruj
Nagłówek			
Brak bloków w tym obszarze		Lewa kolumna boczna Prawa kolumna boczna Zawartość Nagłówek Stopka <brak>	

System wyświetla nam ponadto w ramach skórki szereg specjalnych pól, zawierających nazwy obszarów, dla łatwiejszej orientacji, gdzie dokładnie pojawi się blok, kiedy przydzielimy go np. do obszaru *Nagłówek* (rysunek 4.4).

Rysunek 4.4. Pola wyświetlane tylko w dziale zarządzania blokami pomagają w ich rozmieszczeniu

Jeśli mamy uruchomioną więcej niż jedną skórke, to zobaczymy dodatkowo linki do wyświetlenia konfiguracji bloków w każdej z nich, czyli dla każdej ze skórek konfiguracja rozmieszczenia bloków jest unikalna (rysunek 4.5).

Rysunek 4.5.
Rozmieszczenie bloków możemy konfigurować niezależnie dla każdej aktywnej skórki



Zauważmy też, że jeśli wybraliśmy inną skórke działu zarządzania, to strona konfiguracji bloków pokaże się w skórcie domyślnej, a nie skórcie działu zarządzania. Jest to logiczne, ponieważ przy zarządzaniu usytuowaniem bloków w 95% przypadków będzie nam chodziło o bloki widoczne w skórcie, jaką widzi typowy użytkownik witryny, a nie administrator.

Bloki dostępne po instalacji

Po instalacji Drupala 6 i uruchomieniu wszystkich dostępnych modułów mamy do dyspozycji bloki pokazane w poniższej tabeli (tabela 4.1).

Tabela 4.1. *Bloki dostępne po instalacji*

Nazwa bloku (Adres konfiguracji (admin/build/block/configure/...))	Co robi? (konfiguracja ponad standard)
Przełącznik wersji językowej (locale/0)	Pozwala wybrać język witryny. Jeśli właśnie oglądany segment ma tłumaczenie na kliknięty język, zostanie ono zaprezentowane.
Ostatnio w blogach (blog/0)	Pokazuje wszystkie najnowsze wpisy z blogów użytkowników.
Ostatnie wiadomości z kanału (NAZWA)(aggregator/feed-<identyfikator kanału>)	Pokazuje listę najnowszych wiadomości w kanale subskrybenta kanałów. Możemy skonfigurować liczbę elementów w bloku.
Ostatnie wiadomości w kategorii (NAZWA)(aggregator/category-< identyfikator kategorii>)	Pokazuje listę najnowszych wiadomości w kategorii subskrybenta kanałów. Możemy skonfigurować liczbę elementów w bloku.
Ostatnie odpowiedzi (comment/0)	Lista najnowszych komentarzy do dowolnego typu treści.
Odnośniki podstawowe (menu/primary-links)	Pokazuje menu odnośników podstawowych w strukturze drzewa.
Odnośniki dodatkowe (secondary-links)	Pokazuje menu odnośników dodatkowych w strukturze drzewa.

Tabela 4.1. Bloki dostępne po instalacji — ciąg dalszy

Nazwa bloku (Adres konfiguracji (admin/build/block/configure/...))	Co robi? (konfiguracja ponad standard)
Nowi użytkownicy (<i>user/2</i>)	Pokazuje listę nowo zarejestrowanych użytkowników. W konfiguracji możemy zdefiniować liczbę wyświetlanych użytkowników.
Nowe tematy (<i>forum/1</i>)	Wykaz najnowszych tematów założonych na forum. Możemy podać, jaka liczba tematów ma się pokazywać.
Aktywne tematy (<i>forum/0</i>)	Pokazuje najnowszą aktywność na forum — nowo założone tematy oraz tematy, w których właśnie pojawiła się nowa odpowiedź. Możemy podać, jaka liczba tematów ma się pokazywać.
Najświeższa ankieta (<i>poll/0</i>)	Wyświetla najnowszą ankietę z możliwością zagłosowania.
Kto przegląda (<i>user/3</i>)	Wyświetla nazwy użytkowników zalogowanych i aktywnych przez ustawiony w bloku czas w minutach oraz informację o liczbie odwiedzających (z podziałem na zarejestrowanych i nieznajomych). Aktywność użytkownika w minutach: Długość listy użytkowników:
Książka (<i>book/0</i>)	Pokazuje drzewiastą strukturę książki. Możemy zdecydować, czy pokazujemy tylko w sytuacji, kiedy trafiliśmy na stronę książki (wtedy pokazujemy strukturę książki, do której należy dana strona), czy może pokazujemy wszystkie książki w systemie, niezależnie od tego, jaka obecnie strona jest wyświetlana.
Informacje o autorze (<i>profile/0</i>)	Pozwala wyświetlić dane o autorze danego wpisu, wraz z polami wypełnionymi w profilu (zgodnie z konfiguracją bloku możemy wybrać tylko niektóre pola).
Formularz wyszukiwania (<i>search/0</i>)	Wyświetla pole wyszukiwarki.
Subskrybuj (<i>node/0</i>)	Wyświetla ikonę będącą linkiem do kanału RSS witryny — http://adres.pl/rss.xml .
Logowanie (<i>user/0</i>)	Udostępnia formularz logowania — oczywiście tylko użytkownikom niezalogowanym. Jeśli włączono moduł <i>OpenID</i> , blok umożliwia zamianę formularza tradycyjnego (użytkownik+hasło) na pole do wpisania identyfikatora <i>OpenID</i> .
Nawigacja (<i>user/1</i>)	Pokazuje podręczne menu dla zalogowanego użytkownika, zawierające link do edycji profilu, dodawania zawartości, wylogowania, linki do działu zarządzania (jeśli użytkownik ma do niego dostęp) i tym podobne.
Witrynę napędza Drupal (<i>system/0</i>)	Wyświetla informację o systemie użytym do budowy strony i logo Drupala. Możemy wybrać kilka wersji kolorystycznych i wielkości.

Konfiguracja bloku

Każdy blok może udostępniać specyficzne dla niego opcje konfiguracyjne, ale jest też szereg atrybutów wspólnych, które posiada każdy blok w systemie, niezależnie od tego, czy stworzyliśmy go sami, czy dostarcza go jeden z modułów.

Po wejściu w konfigurację dowolnego bloku (odnośniki do konfiguracji są przy każdym z nich na stronie zarządzania) zobaczymy zatem opcje specyficzne dla niego oraz standardowe, omówione niżej.

Opis bloku jest tekstem, jaki pojawi się na stronie zarządzania blokami. Możemy go ustawić tylko dla bloków, które dodamy własnoręcznie.

Tytuł bloku to tekst nagłówka, jaki zostanie zaprezentowany użytkownikowi (rysunek 4.6).

Rysunek 4.6.

*Blok o tytule
„Muzyka”*



Możemy też nie wyświetlać żadnego tytułu — wtedy w pole nań przeznaczone wstawiamy `<none>`. Jeśli pole pozostawimy puste, system wyświetli domyślny tytuł, o ile dany blok taki posiada.

W dalszej konfiguracji bloku, w sekcji **Użytkownicy**, mamy do wyboru trzy opcje decydujące o tym, czy blok ma być pokazywany obowiązkowo, bez możliwości wyłączenia (opcja pierwsza), czy zezwolić użytkownikom na wyłączenie go sobie w profilu (opcja druga), czy może w ogóle go nie wyświetlać, dopóki użytkownik z pełną świadomością nie włączy go na stronie edycji profilu.

Sekcja **Wyświetlanie zależne od rangi** oznacza, że blok zobaczą tylko osoby przynależne do zaznaczonych tutaj rang. W ten sposób możemy zaprezentować bloki tylko wybranej grupie — na przykład osobom niezalogowanym bądź moderatorom komentarzy.

Ostatnia sekcja, **Wyświetlanie zależne od strony**, wymaga szerszego omówienia. Za jej pomocą możemy ustawić pokazywanie bloku tylko, gdy odwiedzamy wybrane i wprost tu podane adresy (opcja *Pokazuj tylko na stronach z listy*), bądź przeciwnie — przy wskazanych ścieżkach go ukryć, a wyświetlać na wszystkich innych (opcja *Pokazuj na wszystkich stronach prócz tych z listy*).

Trzecia opcja, *Wyświetlaj, jeśli poniższy kod PHP zwraca prawdę* — `TRUE` jest przeznaczona do sterowania wyświetlaniem bloku za pomocą dowolnego kodu PHP zwracającego prawdę lub fałsz.

Jeśli chodzi o pierwsze dwie opcje, przykładem niech będzie sytuacja, kiedy chcemy wyświetlać blok z reklamą na każdej stronie, ale:

- ◆ nie na stronie głównej,
- ◆ nie w dziale administracyjnym,
- ◆ nie podczas edycji zawartości.

Wystarczy wypełnić formularz jak na rysunku i blok będzie zachowywał się dokładnie tak, jak chcemy (rysunek 4.7).

Rysunek 4.7.

*Manewrowanie
widocznością bloku
w zależności
od ścieżki do strony*

Wyświetlanie zależne od strony

Pokazuj blok na określonych stronach:

☒ Pokazuj na wszystkich stronach prócz tych z listy.

☐ Pokazuj tylko na stronach z listy.

☐ Wyświetlaj, jeśli poniższy kod PHP zwraca prawdę - TRUE (tryb PHP, tylko dla ekspertów).

Strony:

<front>
admin/*
node/* /edit

Jak widać, z wykorzystaniem znaku wieloznacznego * zastępującego dowolny (również pusty) ciąg znaków, jesteśmy w stanie dość precyzyjnie zaadresować zarówno końcówki, jak i dowolną wewnętrzną część adresu. Wiedząc, że wszystkie strony edycji zawartości mają format .../node/<identyfikator segmentu>/edit, mogliśmy wprowadzić maskę do wykluczenia wyświetlania bloku pod tymi adresami.

Warto też zauważyć słówko <front>. Służy ono do sterowania widocznością bloku na stronie głównej. Mimo iż strona ta może się z czasem zmienić (konfiguracja tego, co jest stroną główną, pozwala na wskazanie dowolnej ścieżki w systemie), nie będziemy musieli poprawiać widoczności wszystkich bloków.

Jeszcze szerszego omówienia wymaga trzecia opcja sterowania widocznością bloku. Kod PHP, jaki możemy tu wstawić, jest zupełnie dowolny. Jeśli zechcemy pokazywać blok z wynikami Dużego Lotka, możemy tu sprawdzać dzień tygodnia i zwracać TRUE, jeśli jest akurat wtorek, czwartek bądź sobota. Z bardziej przydatnych przykładów, możemy sprawdzić, jaki jest typ aktualnie wyświetlanej strony, i pokazywać blok tylko, jeśli jest to wpis w blogu (rysunek 4.8).

Rysunek 4.8.

*Manewrowanie
widocznością
bloku w zależności
od rezultatu,
jaki zwraca kod PHP*

Wyświetlanie zależne od strony

☒ Wyświetlaj, jeśli poniższy kod PHP zwraca prawdę - TRUE (tryb PHP, tylko dla ekspertów).

Strony:

```
<?php
if ($arg(0) == 'node' && is_numeric($arg(1))) {
    $node = node_load(array('nid' => $arg(1)));
    return $node->type == 'blog';
}
?>
```

Powyższy rysunek zawiera tylko kilka linijek kodu, więc przy okazji go omówimy. Zawiera on instrukcję `if`, która — jeśli adres obecnie wyświetlanej strony składa się ze słówka „node” i dowolnego numeru — zwróci `TRUE` pod warunkiem, że właśnie wyświetlany segment jest typu *blog*. Tablica `arg(0)` przechowuje kolejne części adresu rozdzielone ukośnikami, zatem w wypadku adresu `.../node/5` zmienna `arg(0)` będzie równa `node`, a `arg(1)` równać się będzie 5.

Przykłady skryptów do gotowego zastosowania w tym miejscu znajdują się w specjalnej części dokumentacji na stronie <http://drupal.org/node/60317>. Warto przejrzeć te propozycje, bo bywają tam naprawdę ciekawe sposoby wzbogacenia funkcji strony.

Własne bloki

Zarządzanie blokami pozwala nam po kliknięciu opcji *Dodaj blok* (`.../admin/build/block/add`) dodać do systemu blok zawierający dowolną treść bądź, znowu, kod PHP.

Od najprostszego umieszczenia komunikatu od administratora do użytkowników w randze moderator, do skomplikowanego bloku wyświetlającego dane pobrane z bazy, mechanizm ten daje potężne możliwości. Po zapisaniu pojawi się on na liście dostępnych bloków, różniąc się od wbudowanych tym, że widoczna będzie przy nim opcja *usuń* (rysunek 4.9).

Rysunek 4.9.

Opcja usuwania bloku jest dostępna tylko dla tych, które sami utworzyliśmy

Blok	Obszar	Zapobieganie przeciążeniu	Czynności
Lewa kolumna boczna			
+ Komunikat do redakcji	Lewa kolumna boczna	☐	konfiguruj usuń
+ Ostatnie wiadomości z kanału filmweb - newsy	Lewa kolumna boczna	☐	konfiguruj
+ Najświeższa ankieta	Lewa kolumna boczna	☐	konfiguruj
+ Nawigacja	Lewa kolumna boczna	☐	konfiguruj

Bloki zawierające kod PHP to funkcja, dzięki której możemy wyświetlić na stronie niemal dowolną informację, jaką możemy sobie wyobrazić i uda nam się zaimplementować jej przygotowanie — na przykład poprzez zapytanie do bazy danych, sięgnięcie do systemu plików czy napisanie dowolnie skomplikowanego kodu.

Przykładowe pomysły na bloki zbudowane z kodu PHP możemy znaleźć na stronie <http://drupal.org/node/21867>, zawierającej dziesiątki pomysłów na wykorzystanie tej funkcji.

Jeden z najprostszych pomysłów to wyświetlenie losowego tekstu — cytatu, wskazówki, porady. Osiągniemy to, wywołując opcję *Dodaj blok* (`.../admin/build/block/add`) i wpisując kod PHP w pole *Treść bloku* (rysunek 4.10).

Rysunek 4.10.
Wprowadzanie
zawartości bloku

Tytuł bloku:
Komunikat do redakcji
Tytuł bloku, który zobaczy użytkownik.

Treść bloku:

```
<?php
// porada ze strony http://drupal.org/node/21867
global $user;
$cytat = array(
  'Hej, czy nie wiecie, nie macie władzy na świecie <br /> (Kult)',
  'Mniej czytajcie/Mniej się uczcie/Więcej myślćie/Więcej czujcie <br /> (Dezertter)'.
  'Możesz wyłączyć blok z cytatem w <a
href="http://localhost/drupal?q=user/' . $user->uid . ">swoim profilu</a>',
);
return $cytat[rand(0,count($cytat)-1)];
?>
```

Zawartość bloku, którą zobaczy użytkownik.

Kod, jaki wkleiliśmy w powyższym rysunku, ma niniejszą postać:

```
<?php
// porada ze strony http://drupal.org/node/21867
global $user;
$cytat = array(
  'Hej, czy nie wiecie, nie macie władzy na świecie <br /> (Kult)',
  'Mniej czytajcie/Mniej się uczcie/Więcej myślćie/Więcej czujcie <br /> (Dezertter)'.
  'Możesz wyłączyć blok z cytatem w <a
href="http://localhost/drupal?q=user/' . $user->uid . ">swoim profilu</a>',
);
return $cytat[rand(0,count($cytat)-1)];
?>
```

Nie możemy zapomnieć wybrać PHP jako format danych do zawartości bloku (rysunek 4.11).

Rysunek 4.11.
Wybór formatu, jaki
ma zostać zastosowany
do wyświetlenia
zawartości bloku

Format danych

- ☐ Filtrowany HTML
 - ☐ Adresy internetowe są automatycznie zamieniane w odnośniki, które można kliknąć.
 - ☐ Dozwolone znaczniki HTML: <a> <cite> <code> <dl> <dt> <dd>
 - ☐ Znaki końca linii i akapitu dodawane są automatycznie.
- ☐ HTML bez ograniczeń
 - ☐ Adresy internetowe są automatycznie zamieniane w odnośniki, które można kliknąć.
 - ☐ Znaki końca linii i akapitu dodawane są automatycznie.
- ☒ PHP code
 - ☐ Można wprowadzać kod PHP. Powinien być on otoczony znacznikami <?php i ?>.

Więcej informacji na temat formatowania

Aby pozostać w zgodzie z ostatnią poradą, powinniśmy jeszcze ustawić atrybut *Do-myślnie wyświetlaj ten blok, ale pozwól użytkownikom go ukryć* w opcji *Ustawienia wyświetlania*:

Po zapisaniu tak spreparowanego bloku będziemy mogli przydzielić go do dowolnego obszaru, gdzie będzie wyświetlał swoje losowe teksty (rysunek 4.12).

Rysunek 4.12.

*Blok z treścią
wygenerowaną
przez kod PHP*

Komunikat do redakcji

Możesz wyłączyć blok z
cytatem w swoim profilu

Formularz kontaktowy

.../admin/build/contact

Trudno sobie w dzisiejszych czasach wyobrazić stronę, która nie udostępnia odwiedzającym jakiejś formy kontaktu. Wiele stron istnieje tylko po to, aby taki kontakt umożliwić. Często są to wizytówki firm i instytucji, ministrony dedykowane prezentacji produktów czy ofert specjalnych albo blogi, gdzie autor liczy na odzew ze strony czytelników.

Podstawowa dystrybucja Drupala daje dość prostą funkcjonalność formularza kontaktowego. Wyróżnia się on możliwością zdefiniowania dowolnej liczby kategorii, z których każda może mieć innych odbiorców, automatyczną odpowiedź i nazwę. Pozwala to zbudować prosty system, gdzie odwiedzający naszą witrynę ma do wyboru kilka opcji kontaktowych, na przykład z działem sprzedaży, pomocą techniczną czy sekretarką prezesa.

Nową kategorię dodajemy za pomocą opcji *Dodaj kategorię* (rysunek 4.13).

Rysunek 4.13.

*Konfiguracja kategorii
formularza
kontaktowego*

Formularz kontaktowyPrzeglądaj**Dodaj kategorię**Ustawienia

Kategoria: *

Na przykład: "uwagi do strony" lub "informacje o produktach".

Odbiorcy: *

Na przykład: "webmaster@twojastrona.com" lub "dzialsprzedazy@twojastrona.com". Adresy odbiorców rozdziela się za pomocą przecinka.

Automatyczna odpowiedź:

Odpowiedź wysyłana automatycznie. Wypełnienie pola jest opcjonalne. Jeśli nie zostanie wypełnione, użytkownik nie będzie otrzymywał żadnej automatycznie wysyłanej odpowiedzi.

Oprócz tych ustawień możemy przypisać kategorii wagę (cięższe, z większą wagą, będą niżej na liście, z której użytkownik wybiera kategorię kontaktu) oraz ustawić, czy kategoria ma być domyślnie wybrana po przejściu na stronę z formularzem kontaktowym.

Niestety, mamy do dyspozycji tylko jeden główny i wspólny dla wszystkich kategorii opis formularza, możliwy do skonfigurowania przez opcję *Ustawienia*. W tym samym miejscu ustalimy, ile wiadomości na godzinę można wysłać za pomocą naszego formularza. Ma to na celu przeciwdziałanie automatom spamującym (rysunek 4.14).

Rysunek 4.14.
Konfiguracja cech formularza kontaktowego wspólnych dla wszystkich kategorii

Na samym końcu ustawień jest jeszcze jedna istotna opcja. Jej włączenie uruchomi formularz kontaktowy na stronie każdego profilu użytkownika. Formularze będą dostępne pod adresem `.../user/<identyfikator użytkownika>/contact`. Użytkownicy będą mogli wyłączyć swój formularz w opcjach edycji własnego profilu.

O ile przy stronach prowadzonych indywidualnie włączanie osobistych formularzy kontaktowych nie ma większego sensu, to na stronach z większą liczbą użytkowników (np. redaktorów, bloggerów, dyskutantów na forum) może mieć ona bardzo praktyczne zastosowanie. Przede wszystkim użytkownicy, z którymi się kontaktujemy, nie muszą nigdzie podawać własnego adresu e-mail, a także w każdej chwili będą mogli wyłączyć formularz, jeśli z jego powodu otrzymywać będą niechciane wiadomości.



Wskazówka

Tak dla głównego formularza kontaktowego, jak i dla formularzy osobistych, a nawet dla rejestracji użytkownika czy dodawania komentarzy warto rozważyć wprowadzenie techniki weryfikującej, czy wypełniający jest na pewno człowiekiem. Moduły udostępniające tę funkcję, popularnie zwaną CAPTCHA¹, są dostępne w kilku wersjach — od całkowicie samodzielnego captcha, po dodatki integrujące naszą witrynę z systemami Mollom czy Askimet.

Kiedy już utworzymy kilka kategorii, możemy je przejrzeć i edytować na głównej stronie zarządzania modułem *Kontakt* (rysunek 4.15).

Rysunek 4.15.
Domyślna kategoria dla formularza kontaktowego

Kategoria	Odbiorcy	Wybrany	Czynności
Kontakt z Administratorem strony	admin@naszadomena.pl	Nie	edytuj usuń
Dział serwisu	serwis@naszadomena.pl	Nie	edytuj usuń
Uwagi do redakcji	mail@o2.pl, inny@wp.pl	Tak	edytuj usuń

¹ <http://pl.wikipedia.org/wiki/CAPTCHA>

Jeden z wpisów będzie zaznaczony jako *Wybrany*, co oznacza, że po załadowaniu strony `.../contact` będzie on wyświetlany jako domyślna kategoria kontaktowa. Ponadto można zauważyć, że jedna z kategorii ma wpisane dwa adresy odbiorców. Jest to jak najbardziej prawidłowe i czasami niezbędne ustawienie.

Menu

`.../admin/build/menu`

Menu to bardzo ciekawy element systemu Drupal. Z punktu widzenia osoby początkującej, budującej pierwsze strony, moduł ten jest tylko zbiorem interfejsów administracyjnych pozwalających zarządzać nawigacją na stronie. Oferuje też bloki dla każdego stworzonego menu (możemy ich bowiem stworzyć tyle, ile zechcemy).

Tak naprawdę `menu.module` jest jednym z kluczowych elementów systemu Drupal. Każda strona wyświetlana w ramach serwisu — czy to wpis na blogu, formularz kontaktowy, lista artykułów czy wreszcie formatka administracyjna do zarządzania użytkownikami — musi mieć swoją ścieżkę (adres, URL), a wywołanie dowolnej ścieżki obsługuje właśnie ten moduł.

Na szczęście szczegóły tych mechanizmów nie przydadzą się nam, dopóki nie zaczniemy pisać własnych modułów, zatem przejdźmy do tego, co możemy skonfigurować za pomocą tego obszaru działu zarządzania.

Po wejściu na główną stronę, *Wyświetl menu*, widzimy trzy menu stworzone tuż po instalacji systemu.

Pierwsze z nich — *Nawigacja* — jest specyficzną grupą odnośników użytecznych głównie dla zalogowanego użytkownika. Jest tam przede wszystkim link do wyświetlenia strony profilu oraz łącze służące do wylogowania się. Znajdziemy także łącza do dodawania zawartości i do zarządzania stroną — chociaż ich widoczność ma ścisły związek z uprawnieniami. Zwykły użytkownik (nie administrator) nie zobaczy w ogóle odnośnika do działu zarządzania.

Kolejne dwa standardowe menu są przeznaczone do umieszczania w nich odnośników podstawowych i dodatkowych. To, co się tam znajdzie, zależy wyłącznie od nas. Możemy tam umieścić dowolne adresy — po kliknięciu dowolnego z nich pojawi się interfejs służący do zarządzania pozycjami, jakie się w danym menu znajdują.

Struktura menu

Menu w Drupalu ma strukturę drzewiastą. Oznacza to, że każdy element może być umieszczony albo na szczycie menu (równorzędnie z innymi pozycjami), albo jako jeden z „liści”, czyli posiadać jako rodzica inny element menu. Liczba zagnieżdżeń nie jest ograniczona.

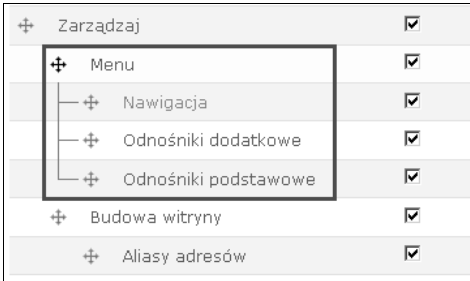
Dobrym przykładem struktury drzewiastej jest menu *Nawigacja*, a w nim element *Zarządzaj* — zawierający pod sobą wszystkie główne obszary działu zarządzania, a pod każdym obszarem poszczególne strony służące do sterowania konkretnymi funkcjami (rysunek 4.16).

Rysunek 4.16.
Widok drzewiastej struktury menu

Element menu	Włączony	Rozwinięte	Czynności
+ Zarządzaj	☒	☐	edytuj
+ Budowa witryny	☒	☐	edytuj
+ Aliasy adresów	☒	☐	edytuj
+ Bloki	☒	☐	edytuj
+ Formularz kontaktowy	☒	☐	edytuj
+ Menu	☒	☐	edytuj
+ Nawigacja	☒	☐	edytuj usuń
+ Odnośniki dodatkowe	☒	☐	edytuj usuń
+ Odnośniki podstawowe	☒	☐	edytuj usuń
+ Moduły	☒	☐	edytuj

Elementy w każdym menu możemy przesuwać metodą „przeciągnij i upuść”. Podczas tego procesu widzimy dokładnie, co i w jakie miejsce zamierzamy przenieść — zarówno element, który złapaliśmy, jak i wszystkie elementy podrzędne (rysunek 4.17).

Rysunek 4.17.
Przenoszenie gałęzi drzewa ze wszystkimi elementami podrzędnymi



Dodawanie elementu do menu

Aby dodać element do menu, wystarczy wybrać opcję *Dodaj element*, widoczną pod warunkiem, że jesteśmy na stronie wyświetlającej obecną zawartość jednego z menu.

Aby dodać element, musimy podać przynajmniej ścieżkę, na jaką wskazuje, oraz tekst odnośnika, który pojawi się w menu. Opcjonalnie możemy podać opis, jaki zostanie dodany do atrybutu `title` odnośnika w menu (rysunek 4.18).

Jak widać na przykładzie, podaliśmy ścieżkę, która jest tylko częścią pełnego adresu. Jest to tak zwana ścieżka wewnętrzna. Oczywiście możemy używać pełnego adresowania, na przykład `http://localhost/drupal/contact`, ale zdecydowanie odradzam taki wybór. Podając tylko ścieżkę wewnętrzną, unikniemy żmudnego poprawiania każdej

Rysunek 4.18.
*Konfiguracja nowego
 odnośnika w menu*

Ścieżka: *

 Ścieżka, do której ma prowadzić menu. Może to być wewnętrzna ścieżka Drupala, taka jak `node/add`, lub adres zewnętrznej strony, jak na przykład `http://drupal.org`. Wpisanie `<front>` sprawi, że odnośnik będzie prowadził do strony głównej.

Tekst odnośnika menu: *

 Tekst odnośnika, który pojawi się w menu.

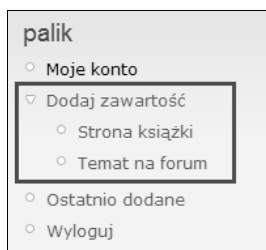
Opis:

 Opis wyświetlany po przesunięciu kursora nad menu.

pozycji menu w sytuacji, gdybyśmy nagle musieli zmienić adres, pod którym znajduje się nasza witryna. Ścieżki wewnętrzne gwarantują nam bezbolesne przeniesienie serwisu pod zupełnie inny adres — na przykład z komputera osobistego do Internetu, z domeny darmowej na płatną czy z podfolderu do katalogu głównego.

Poza podaniem ścieżki, opisu i tytułu powinniśmy zdecydować, czy w danej chwili element ma być widoczny (atrybut włączony), a także czy ma być domyślnie rozwinięty (atrybut rozwinięte). To drugie ustawienie jest pomocne przy prezentowaniu menu w blokach — przykładowo, jeśli zaznaczymy atrybut *rozwinięte* przy pozycji *Dodaj zawartość* w menu *Nawigacja*, nasi zalogowani użytkownicy od razu będą widzieli, jaką treść mogą dodać do witryny (rysunek 4.19).

Rysunek 4.19.
*Menu Nawigacja
 w postaci bloku, jaki
 zobaczy użytkownik
 po zaznaczeniu
 opcji „rozwinięte”
 przy pozycji
 „Dodaj zawartość”*



Co warte odnotowania, nawet klikając *Dodaj element* możemy umieścić nową pozycję w dowolnym miejscu całej struktury. Jedną z opcji do ustawienia jest bowiem element nadrzędny, który zawiera wszystkie menu (wyróżnione poprzez otoczenie ich nazwy ostrymi nawiasami) oraz wszystkie występujące w nich pozycje, z zaznaczoną hierarchią (rysunek 4.20).

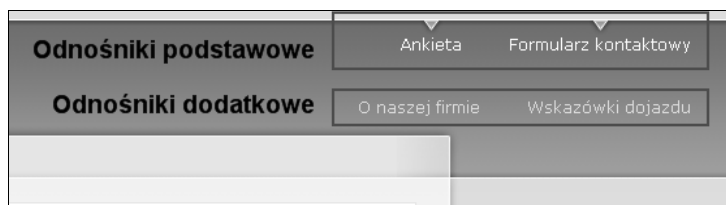
Rysunek 4.20.
*Wybór rodzica dla
 nowej pozycji w menu*



Na samym końcu formularza jest jeszcze możliwość ustawienia wagi elementu, ale robienie tego ręcznie, kiedy mamy do dyspozycji mechanizm „przeciągnij i upuść” (który w miarę przedstawiania pozycji względem siebie sam nadaje im właściwe wagi), graniczy z niedorzecznością.

Po dodaniu pozycji do odnośników podstawowych i dodatkowych, przy standardowych ustawieniach skórki Garland, zobaczymy je w specjalnym obszarze na szczycie strony (rysunek 4.21).

Rysunek 4.21.
Menu odnośników podstawowych i dodatkowych



To, czy odnośniki mają być wyświetlane, skonfigurujemy w ustawieniach każdej skórki. Każda skórka może mieć zdefiniowane nieco inne miejsce do wyświetlania tych odnośników.

Warto wiedzieć, że dodanie pozycji w menu jest możliwe także za pomocą innych sposobów. Jeden z najczęściej wykorzystywanych i najwygodniejszych jest dostępny z poziomu formularza dodawania zawartości. Możemy wtedy określić (za pomocą pól dostępnych w sekcji *Menu*), jaki ma być tekst odnośnika i gdzie ma się pokazać. Będzie on kierował do właśnie dodawanej treści, a więc nie musimy martwić się o ręczne wpisywanie adresu (rysunek 4.22).

Rysunek 4.22.
Formularz dodawania i edycji zawartości zawiera też opcję utworzenia odnośnika w menu

Dodawanie menu

Menu, jakie znajdziemy po instalacji Drupala, to jednak tylko przykładowe, gotowe do zagospodarowania struktury. Sami możemy założyć kolejne, klikając opcję *Dodaj menu*. Jedyne, co musimy podać, zakładając nowe menu, to wewnętrzna nazwa, tytuł i opis (rysunek 4.23).

Do nowo utworzonego menu możemy dodać nowe pozycje lub przenieść dowolny istniejący odnośnik. Aby tego dokonać, musimy wejść w edycję odnośnika i zmienić jego rodzica na nowo utworzone menu (rysunek 4.24).

Rysunek 4.23.
Tworzenie nowego menu

Nazwa menu: *

moje-menu

Nazwa menu wykorzystywana wewnętrznie. Tekst posłuży do budowy adresu URL strony z *przeglądem menu*. Nazwa może się składać jedynie z małych liter, liczb oraz łączników i nie może się powtarzać.

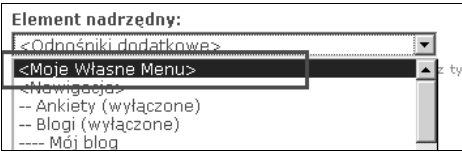
Tytuł: *

Moje Własne Menu

Opis:

Menu zawierające testowe odnośniki

Rysunek 4.24.
Nowe menu pojawi się jako jedna z pozycji listy Element nadrzędny podczas edycji dowolnego odnośnika



Bloki zawierające menu

Jeśli zechcemy wyświetlić nowo stworzone menu na stronie, musimy posłużyć się blokiem. Każde menu, jakie istnieje w systemie, jest możliwe do pokazania jako blok — automatycznie zakładany w momencie tworzenia menu. Na stronie zarządzania blokami ustawiamy widoczność tego bloku na wybrany przez nas obszar i menu z odnośnikami, jakie w nim zawarliśmy, pojawi się na stronie. Przy okazji możemy zaobserwować wpływ ustawienia atrybutu *rozwinięte* na prezentację menu (rysunek 4.25).

Moje Własne Menu

▼ element 1

▼ element 1.1

○ element 1.1.1

▷ element 2

▷ element 3

Moje Własne Menu

Wyświetl elementy

Dodaj element

Edycja menu

Układ elementów menu można zmienić metodą "przeciągnij i upuść", za pomocą uchwytów umieszczonych w kolumnie *Element menu*. Do wprowadzenia poczynionych zmian służy przycisk *Zachowaj konfigurację* u dołu strony.

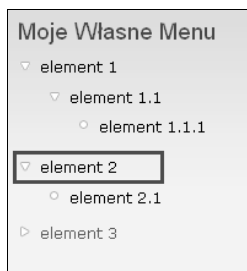
Element menu	Włączony	Rozwinięte	Czynności
+ element 1	☒	☒	edytuj usuń
+ element 1.1	☒	☒	edytuj usuń
+ element 1.1.1	☒	☐	edytuj usuń
+ element 2	☒	☐	edytuj usuń
+ element 2.1	☒	☐	edytuj usuń
+ element 3	☒	☐	edytuj usuń

Rysunek 4.25. *Przykładowa struktura menu*

Po kliknięciu elementu, który nie ma ustawionego atrybutu *rozwinięte*, jego elementy podrzędne stają się widoczne (rysunek 4.26).

Rysunek 4.26.

*Blok pokazujący
odnośniki należące
do nowego menu*



Ustawienia modułu menu

.../admin/build/menu/settings

Na sam koniec warto zajrzeć do ustawień samego modułu *menu*. Cała konfiguracja sprowadza się do wskazania trzech atrybutów. Pierwszy określa, jakie menu będzie domyślnie wybrane w trakcie dodawania nowego wpisu i wykorzystania wspomnianej wcześniej możliwości umieszczenia odnośnika do menu bezpośrednio podczas dodawania treści. Po prostu — to menu, które tu wybierzemy, będzie domyślnie wybrane jako menu, do którego zostanie wstawiony odnośnik.

Kolejne dwie opcje są ściśle związane z wyświetlaniem odnośników podstawowych i dodatkowych. Ponieważ każda standardowa skórka umożliwia skonfigurowanie jej tak, aby pokazywała jeden zestaw lub oba zestawy odnośników, możemy tu wskazać, które menu zawiera odnośniki podstawowe, a które dodatkowe. Jest to przydatne, jeśli z jakichś powodów wolimy stworzyć zupełnie własne menu, gdzie będziemy trzymać odnośniki, które zechcemy następnie wyświetlić w miejscu, gdzie zwyczajowo prezentowane są pozycje z menu odnośników podstawowych bądź dodatkowych.

Moduły

.../admin/build/modules

System modułów jest elementem, który prawdopodobnie zdecydował o sukcesie Drupala. Nie chodzi tylko o możliwość rozszerzania witryny o nowe funkcje, ale o elastyczność interfejsu programistycznego (API) stworzonego do dyspozycji twórców dodatków. Bardzo często podkreśla się zaletę, jaką jest prostota wpasowania się w ramy opisane przez twórców systemu i łatwość pisania nowych modułów. Dzięki temu możemy cieszyć się ich ogromną liczbą i różnorodnością. W momencie pisania tego rozdziału na stronie http://drupal.org/project/modules?filters=drupal_core%3A87&solrsort=sis_project_release_usage%20desc można było znaleźć ponad 3600 modułów do Drupal 6.

Jak zarządzać modułami w Drupalu? Pierwszym miejscem, jakie powinniśmy odwiedzić, jest oczywiście strona z interfejsem do ich włączania i wyłączania. Po przejściu na nią zobaczymy obszerny wstęp wyjaśniający podstawowe mechanizmy dodawania, aktualizowania, instalowania i konfigurowania modułów.

Pod wstępem, który warto przeczytać, zobaczymy pogrupowany wykaz wszystkich modułów, jakie mamy w tej chwili do dyspozycji. Po instalacji systemu te grupy to: *Moduły rdzenia*, *opcjonalne* oraz *Moduły rdzenia, wymagane*. Jak łatwo się domyślić, nie możemy wyłączyć żadnego z modułów wymaganych, dlatego dla wygody cała sekcja z nimi jest po załadowaniu strony zwinięta. Po rozwinięciu widać, że do działania Drupala wymagane jest włączenie tylko kilku modułów (rysunek 4.27).

Rysunek 4.27.
Wymagane moduły nie mają opcji ich wyłączenia

Moduły rdzenia, wymagane				
Włączony	Zapobieganie przeciążeniu	Nazwa	Wersja	Opis
☒	☐	Block	6.15	Daje kontrolę nad blokami wyświetlanymi wokół głównej zawartości strony.
☒	☐	Filter	6.15	Zajmuje się filtrowaniem zawartości przed jej opublikowaniem.
☒	☐	Node	6.15	Umożliwia dodawanie i wyświetlanie zawartości.
☒	☐	System	6.15	Umożliwia administratorom podstawową konfigurację witryny.
☒	☐	User	6.15	Obsługuje system kont użytkowników.

W miarę instalowania modułów zewnętrznych będą powstawać nowe sekcje grupujące moduły zgodnie z ich przeznaczeniem — na przykład zestaw modułów wchodzących w skład Content Construction Kit, służącego do budowania nowych rodzajów zawartości, będzie zgrupowany w jednym miejscu.

Wróćmy do listy modułów rdzenia, które są w grupie *opcjonalne*. Jeśli po instalacji nie włączyliśmy wszystkich dostępnych modułów, zrobmy to teraz. Po tej operacji wróćmy do strony `.../admin/build/modules` i obejrzymy listę (rysunek 4.28).

Rysunek 4.28.
Opcjonalne moduły rdzenia Drupala

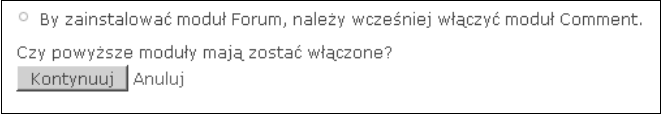
Moduły rdzenia, opcjonalne				
Włączony	Zapobieganie przeciążeniu	Nazwa	Wersja	Opis
☒	☐	Aggregator	6.15	Pobiera wiadomości z kanałów informacyjnych (RSS, RDF, bądź Atom) innych witryn.
☒	☐	Blog	6.15	Umożliwia prowadzenie łatwych w obsłudze stron domowych użytkowników bądź blogów.
☒	☐	Blog API	6.15	Umożliwia użytkownikom dodawanie zawartości za pomocą aplikacji obsługujących protokół XML-RPC.
☒	☐	Book	6.15	Umożliwia umieszczanie stron w hierarchicznej strukturze.
☒	☐	Color	6.15	Pozwala użytkownikom zmieniać schemat kolorów niektórych skórek.
☒	☐	Comment	6.15	Daje użytkownikom możliwość wypowiedzenia się na temat zamieszczanych w witrynie treści. Wymagany przez: Forum (enabled), Tracker (enabled)

Jak widać, dostępne opcje są minimalne i sprowadzają się do możliwości włączenia bądź wyłączenia modułu. Możemy jeszcze ustawić dla niektórych atrybut *Zapobieganie przeciążeniu*. Ta druga opcja jest częścią modułu *throttle.module* i omówiona zostanie w dalszej części książki, więc możemy ją w tym momencie zignorować.

Zależności modułów

Z listy modułów możemy jeszcze dowiedzieć się, czy dany moduł jest wymagany do działania innych — tak jak w przypadku modułu *Comment*. Co więcej, system nie pozwoli nam wyłączyć modułu, jeśli inne moduły, które go wymagają, są obecnie włączone. System zależności działa też w drugą stronę — jeśli przykładowo wyłączymy moduły *Forum* i *Tracker*, następnie wyłączymy *Comment*, a potem zechcemy włączyć ponownie moduł *Forum*, zostaniemy poczęstowani stosownym komunikatem (rysunek 4.29).

Rysunek 4.29.
Przykład działania
pilnowania zależności
między modułami



Po kliknięciu *Kontynuuj* zostaną włączone wymagane moduły.

Jak zauważysz, przeglądając listę modułów, system zależności bazuje na dwóch nazwanych relacjach — *Jest zależny od* i *Wymagany przez* (rysunek 4.30).

Rysunek 4.30.
Opisy zależności
między modułami

Włączony	Zapobieganie przeciążeniu	Nazwa	Wersja	Opis
☒	☐	Comment	6.15	Daje użytkownikom możliwość wypowiedzenia się na temat zamieszczanych w witrynie treści. Wymagany przez: Forum (enabled), Tracker (disabled)
☒	☐	Contact	6.15	Daje dostęp do formularzy kontaktowych.
☒	☐	Content translation	6.15	Umożliwia tłumaczenie zawartości na różne języki. Jest zależny od: Locale (enabled)
☒	☐	Database logging	6.15	Zapisuje w bazie danych informacje o zdarzeniach występujących w systemie.
☒	☐	Forum	6.15	Umożliwia prowadzenie dyskusji na różne tematy. Jest zależny od: Taxonomy (enabled), Comment (enabled)

Dzięki temu prostemu systemowi zależności nie uda nam się popsuć jakiejś funkcjonalności, wyłączając moduł dostarczający „usługi” innemu modułowi. Klasyczny przykład to wspomniany moduł *Comment*, pozwalający na uruchomienie modułu *Forum* — którego uruchomienie bez funkcji komentarzy nie ma wielkiego sensu.

Z powyższych rozważań wynika bardzo ważna obserwacja. Moduły w Drupalu bardzo często przenikają się, świadczą sobie różnego rodzaju usługi, rozszerzają funkcje dostępne w innych modułach. Przykładowo — możemy dodawać treść (wpisy w blogu, artykuły, strony książki) bez włączonego moduły *Taxonomy*, ale jego uruchomienie wzbogaca nasze możliwości o kategoryzację treści. Włączenie *Upload* pozwoli nam dodawać do treści załączniki, a moduł *Content translation* doda naszym wpisom jeszcze jeden wymiar — wersję językową.

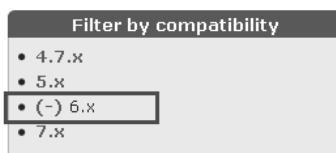
Instalowanie nowych modułów

Co jednak, jeśli chcemy dodać nowy moduł? Operacja ta jest stosunkowo prosta. Wystarczy pobrać archiwum modułu ze strony <http://drupal.org/project/Modules>, umieścić rozpakowany katalog zawierający pliki w katalogu `./sites/all/modules`, a następnie na stronie `.../admin/build/modules` odszukać nazwę załadowanego na serwer modułu i go włączyć.

Aby wyświetlić listę wszystkich dostępnych modułów do aktualnie używanej wersji Drupala, należy na stronie <http://drupal.org/project/Modules> skorzystać z filtra zawierającego wykaz do wybranej wersji, który znajdziemy w bloku *Filter by compatibility* (rysunek 4.31).

Rysunek 4.31.

Filtr wersji, do której wyświetlane są moduły



Listę możemy następnie zawęzić, korzystając z wyszukiwania (blok *Search modules*), wyboru jednej z kategorii (blok *Projects*), a następnie posortować ją według jednego z kryteriów w bloku *Sort by*: *Title* (tytuł), *Creation date* (data powstania projektu), *Last release* (ostatnie wydanie), *Recent activity* (projekty, w których dokonano zmian) czy też *Usage statistics* (liczba użytkowników danego modułu).

Co dzieje się podczas instalacji modułu i co przez nią rozumiemy? Instalacja to proces, jaki zachodzi tak naprawdę tylko przy pierwszym włączeniu pobranego modułu. Każdy moduł składa się z szeregu plików, a jednym z nich jest plik z rozszerzeniem *install*. Plik ten zawiera szereg funkcji wywoływanych przy włączaniu modułu — przede wszystkim ustawienie zmiennych systemowych oraz stworzenie wymaganej struktury w bazie danych. Jeśli moduł wyłączymy, dane te nie są kasowane, dzieje się tak dopiero po wyinstalowaniu modułu. Dopóki tego nie zrobimy, ponowne włączenie modułu spowoduje tylko zmianę jego statusu na „włączony”, a nie ponowną jego instalację.

Wersje modułów

Schemat nazewnictwa wersji projektów (modułów, skórek, rdzenia) dostępnych na stronie drupal.org jest oparty o kilka zasad, spisanych na stronie <http://drupal.org/handbook/version-info>.

Po pierwsze — w danym momencie istnieją dwie aktywnie wspierane główne, stabilne wersje Drupala. W chwili pisania tej książki główne wersje to Drupal 5 i Drupal 6.

Co to znaczy, że wersja jest wspierana? W skrócie oznacza to, że w sytuacji wykrycia luki w bezpieczeństwie lub krytycznego błędu są wydawane małe wersje — na przykład 6.1, 6.2 i tak dalej. Małe wersje mogą niejako „przy okazji” zawierać szereg napraw

drobniejszych błędów, ale zasada ogólna jest taka, że wydawane są tylko, jeśli zostanie opracowana łąta naprawiająca jakąś lukę bezpieczeństwa.

Co oznacza, że w danej chwili wspierane są dwie główne wersje? W uproszczeniu oznacza to tyle, że po wydaniu Drupala 7 Drupal 5 przestanie być wydaniem wspieranym.

Wersja Drupala, jaką mamy zainstalowaną na naszej witrynie, jest więc bazowym kryterium, jakim musimy się posługiwać, szukając do niej rozszerzeń i skórek. Schemat wersjonowania tychże jest taki, że wszystkie projekty pasujące do Drupala 6 będą miały numer wersji zaczynający się od „6.x”. Przykładowo, na stronie modułu `admin_menu`, w części *Downloads*, przy sekcji *Recommended releases* (czyli wydania rekomendowane) zobaczymy, że jest dostępny dla trzech wersji głównych Drupala (rysunek 4.32).

Rysunek 4.32.
*Tabela z rekomendowanymi
wydaniami modułu wyróżnia
się zielonym tłem*

Downloads	
Recommended releases	
Version	Downloads
6.x-1.5	Download (44.22 KB)
5.x-2.8	Download (31.09 KB)
4.7.x-1.4	Download (11.17 KB)

Jak widać — wersja każdego modułu składa się ze złożenia wersji głównej Drupala (np. 6.x) oraz wersji samego modułu. Zatem z rysunku 4.42 wynika, że moduł `administration menu` możemy pobrać i zainstalować w wersji 1.5, przeznaczonej dla Drupala 6, oraz wersjach 2.8 i 1.4, przeznaczonych odpowiednio dla Drupala 5.x i 4.7.x.

To jednak nie koniec schematu. Nieco niżej znajdziemy sekcje z wersjami „niezalecanymi”, które mogą zawierać różne przyrostki, takie jak „alpha” czy „dev” (rysunek 4.33).

Rysunek 4.33.
*Reszta wydań
oznaczona jest innymi
kolorami, najmniej
stabilne mają kolor
czerwony*

Other releases			
Version	Downloads	Date	Links
6.x-3.0-alpha3	Download (59.35 KB)	2009-Aug-16	Notes
5.x-3.0-alpha1	Download (36.94 KB)	2009-Jun-11	Notes
Development releases			
Version	Downloads	Date	Links
7.x-3.x-dev	Download (57.72 KB)	2010-Feb-02	Notes
6.x-3.x-dev	Download (62.59 KB)	2009-Dec-31	Notes
6.x-1.x-dev	Download (47.19 KB)	2009-Dec-25	Notes
5.x-3.x-dev	Download (38.19 KB)	2009-Nov-14	Notes

Zauważymy też, że do Drupala 6 istnieją dwie różne „linie” modułu, czyli wersje 1.x i 3.x. Jak to rozumieć? Najczęstszym wyjaśnieniem takiego stanu jest fakt, że programiści rozwijający moduł w wersji 1.x postanowili rozpocząć pracę nad wersją zawie-

rającą znaczne zmiany w kodzie. W tym celu pozostawili wersję 1.x jako „stabilną”, czyli rekomendowaną dla typowego użytkownika, oraz utworzyli linię 3.x, którą możemy pobrać w wersjach 6.x-3.x-dev i 7.x-3.x-dev, a więc niekoniecznie stabilnych, w stanie takim, jaki pozostawił ostatni programista, zapisując zmiany do CVS. Mamy też wersję 6.x-3.0-alpha3, co należy czytać jako wersję do testowania, którą (najlepiej po dokładnym przeczytaniu komentarzy do wydania, dostępnych pod linkiem *Notes*) możemy śmiało zainstalować na własną odpowiedzialność.

Z pewnością natkniemy się jeszcze na inne przyrostki, a mianowicie *beta* i *rc*, które oznaczają odpowiednio wersję *beta* (nieco bardziej stabilną niż *alpha*) oraz *release candidate* (czyli kandydata do wydania stabilnego). Każdy z tych przyrostków może mieć numer — na przykład 6.x-3.x-rc4 będzie oznaczało czwarte wydanie kandydujące do bycia stabilną trzecią wersją modułu przeznaczonego do współpracy z Drupalem 6.

Strony modułów (projektów)

Sama instalacja modułu to jednak niewielki problem. Najpierw powinniśmy w ogóle go znaleźć i wybrać właściwą wersję. Służy do tego zaawansowany system wersjonowania (oparty o integrację z CVS), śledzenia błędów i generowania „stron startowych” dla każdego projektu, dostępny w ramach witryny *drupal.org*.



Wskazówka

System ten jest de facto zbiorem modułów do Drupala, których centralną częścią jest moduł *Project*. Możemy go sobie pobrać i uruchomić dla własnych potrzeb. Co prawda zestaw ten jest bardzo mocno dopasowany do wymagań witryny *drupal.org*, ale warto zwrócić uwagę, że twórcy Drupala stworzyli na bazie swojego dziecka zaawansowany system o funkcjach porównywalnych ze znanym projektem Sourceforge. Czyż nie jest to świetny dowód elastyczności i rozszerzalności tego narzędzia?

System, jaki zbudowali twórcy witryny *drupal.org*, narzuca szereg wymagań co do nazewnictwa, położenia plików i dostępu do narzędzia wspomagającego wersjonowanie — CVS. Ponieważ twórcy sami stosują się do schematu, proponuję zacząć poznawanie go od najważniejszego projektu, jakim jest sam rdzeń Drupala.

Przejdźmy na „stronę startową” projektu Drupal — <http://drupal.org/project/Drupal>. Każdy moduł i każda skórka, jakie możemy pobrać na *drupal.org*, posiada podobną stronę, zawierającą szereg elementów pozwalających zapoznać się z projektem. Do najważniejszych elementów można zaliczyć obszary zaznaczone na rysunku (rysunek 4.34).

1. Metryczka z autorem i datą utworzenia strony projektu.
2. Oznaczenie kategorii, do której zalicza się projekt (może należeć do kilku).
3. Część opisowa, zawierająca nie tylko ogólny opis funkcji, jakie zawiera w sobie projekt, ale również często różne uwagi dotyczące niestandardowych wymagań (inne moduły, wersje PHP).
4. Sekcja *Downloads*, w której znajdziemy poszczególne wydania projektu. Pierwszą grupą są wydania rekomendowane, czyli najstabilniejsze.
5. Druga grupa plików do pobrania prezentuje wydania w wersjach rozwojowych (alfa, beta, rc).

The screenshot shows the Drupal.org homepage. Numbered annotations are as follows:

- 1:** Points to the 'View' and 'Revisions' tabs at the top.
- 2:** Points to the 'Drupal project' link in the top right.
- 3:** Points to the main content area describing Drupal as an open-source platform.
- 4:** Points to the 'Downloads' section.
- 5:** Points to the 'Other releases' section.
- 6:** Points to the 'Getting Started' sidebar menu.
- 7:** Points to the 'Maintainers for Drupal' sidebar section.
- 8:** Points to the 'Issues for Drupal' sidebar section.
- 9:** Points to the 'Recent issues' sidebar section.

Getting Started

- Getting Started
- Hosting
- Set-up development environment
- Installing Drupal forum

Maintainers for Drupal

- Dries - 23124 commits (last: 2 hours ago, first: 9 years ago)
- webchick - 5283 commits (last: 5 hours ago, first: 1 year ago)
- Gábor Hojtsy - 2839 commits (last: 2 weeks ago, first: 3 years ago)
- drumm - 1215 commits (last: 6 months ago, first: 3 years ago)

[View all committers](#)

Issues for Drupal

To avoid duplicates, please search before submitting a new issue.

All issues
8035 open, 29782 total

Bug reports
4044 open, 16664 total

Subscribe via e-mail
Issue statistics
Oldest open issue: 28 Feb 02

Recent issues

- Create theme.api.php to help

Drupal

Drupal - September 28, 2003 - 1/1/31

Drupal is an open-source platform and content management system for building dynamic web sites. It offers a broad range of features and services including user administration, publishing workflow, discussion capabilities, news aggregation, metadata functionalities using controlled vocabularies and XML publishing for content sharing purposes. Equipped with a powerful blend of features and configurability, Drupal can support a diverse range of web projects ranging from personal weblogs to large community-driven sites.

Help test the forthcoming Drupal 7.0!

7.0-alpha1 release: [Download](#) | [Notes](#)
7.x-dev snapshot: [Download](#) | [Notes](#)

Before installing Drupal, please review the system requirements and the Installation Guide.

If you are using Internet Explorer and WinZip, save the file first, as otherwise IE will corrupt the file's extension.

Information on version numbers can be found in the [handbook](#). Information about themes and modules used to build on drupal.org itself can be found in [What theme and modules does Drupal.org use?](#).

Downloads

Recommended releases

Version	Downloads	Date	Links
6.15	Download (1.04 MB)	2009-Dec-16	Notes
5.21	Download (749.18 KB)	2009-Dec-16	Notes

Other releases

Version	Downloads	Date	Links
7.0-alpha1	Download (2.18 MB)	2010-Jan-15	Notes

Rysunek 4.34. Obszary strony z przykładowym projektem utrzymywanym na witrynie drupal.org

6. Odnośniki do stron, z jakimi warto się zapoznać.
7. Lista programistów biorących udział w rozwoju projektu, wraz z ich aktywnością — liczbą wysłanych do CVS zmian w kodzie (*commits*), pierwszym i ostatnim zapisem zmian do CVS.
8. Wyszukiwarka zgłoszonych spraw w systemie śledzenia błędów oraz podsumowanie, ile i jakiego typu sprawy były zgłaszane.
9. Lista ostatnio zgłoszonych spraw w systemie śledzenia błędów.

Przewinięcie strony z modułem w dół uwidoczni kolejny zestaw opcji, który może różnić się nieznacznie dla poszczególnych projektów (rysunek 4.35).

Rysunek 4.35.
Przykładowe linki,
jakie można spotkać
przy każdym
z projektów
na drupal.org

Resources	Development
• Read license	• View pending patches
• Read complete log of changes	• Browse the CVS repository
• Try out a demonstration	• View CVS messages
• Look at screenshots	• Report a security issue
• View usage statistics	

Powyższe linki prowadzą do najważniejszych zasobów dodatkowych, czyli:

Resources (zasoby)

- ♦ *Read license* (czytaj licencję),
- ♦ *Read complete log of changes* (przejrzyj cały dziennik zmian),
- ♦ *Try out a demonstration* (wypróbuj wersję demonstracyjną),
- ♦ *Look at screenshots* (zobacz ekrany — jak wygląda projekt),
- ♦ *View usage statistics* (zobacz statystyki użycia projektu).

Development (rozwój)

- ♦ *View pending patches* (przejrzyj oczekujące łatki),
- ♦ *Browse the CVS repository* (przejrzyj repozytorium CVS),
- ♦ *View CVS messages* (zobacz komunikaty CVS),
- ♦ *Report a security issue* (zgłoś problem z bezpieczeństwem).

Z punktu widzenia wyboru modułu, szczególnie kiedy odszukamy kilka podobnych do siebie, warto zweryfikować, czy moduły są obecnie mocno rozwijane, czy może okres ich świetności już minął. Pomocne mogą być w tym celu trzy linki:

View usage statistics (odnośnik widoczny na rysunku 4.35) pokazujący dla danego modułu dane zbierane co tydzień ze wszystkich stron raportujących do *drupal.org* listę używanych modułów. Da nam to ogólne pojęcie o wykorzystaniu projektu (z podziałem na wersje) na przestrzeni ostatnich tygodni (rysunek 4.36).

Rysunek 4.36.

Statystyka ilości stron wykorzystujących dany projekt

• <i>Drupal project page</i> • <i>Usage statistics for all projects</i>				
Weekly project usage				
Weekly Drupal usage by API version				
Week	5.x	6.x	7.x	Total
February 7th	17,042	268,701	820	286,563
January 31st	16,765	257,683	725	275,173
January 24th	17,908	257,426	805	276,139
January 17th	18,517	240,778	1,274	260,569
January 10th	18,351	237,522	829	256,702
January 3rd	17,615	227,152	407	245,174
December 27th	15,230	203,306	318	218,854

Odnośnik *View all committers* (odnośnik widoczny na rysunku 4.34 w obszarze numer 7), który prowadzi do strony z pełną listą programistów danego projektu. Często pozwala zorientować się, czy projekt jest „żywy” i czy jego rozwój nie zatrzymał się kilka(naście) miesięcy temu (rysunek 4.37).

Rysunek 4.37.

Wykaz
programistów
zapisujących zmiany
w kodzie do CVS
danego projektu

Committers for <i>Drupal</i>			
User	Last commit ▼	First commit	Commits
Dries	2 hours ago	9 years ago	23124 commits
webchick	5 hours ago	1 year ago	5283 commits
Gábor Hojtsy	2 weeks ago	2 years ago	2839 commits
drumm	8 weeks ago	3 years ago	1215 commits
killes@www.drop.org	2 years ago	3 years ago	715 commits
Steven	2 years ago	9 years ago	2083 commits
Kjartan	3 years ago	9 years ago	694 commits
Natrak	6 years ago	9 years ago	174 commits
Jeroen	8 years ago	9 years ago	253 commits

Issue statistics (odnośnik widoczny na rysunku 4.34 w obszarze numer 8), gdzie dowiemy się, ile spraw w systemie śledzenia błędów zgłoszono ogółem i w ostatnim miesiącu (*Issue activity*), a także jaki był średni czas obsługi spraw w danej kategorii (*Average lifetime*). Te dane również mogą być pomocne w oszacowaniu, czy moduł jest aktywnie używany, naprawiany i rozwijany (rysunek 4.38).

Rysunek 4.38.

Statystyki obsługi
zgłoszeń do danego
projektu

Statistics

Create a new issue · Advanced search · Statistics · Subscribe

Average lifetime

Category	Overall	Last month
bug reports	7 weeks 1 day	4 days 19 hours
feature requests	12 weeks 2 days	N/A
support requests	4 weeks 5 days	2 days 5 hours
tasks	13 weeks 6 days	N/A

Issue activity

Status	Overall	%	Last month
active	45	7%	19
duplicate	162	25%	1
won't fix	99	15%	2
by design	85	4%	0

Wsparcie do modułów

Drupal.org, a konkretnie dział zarządzający projektami, jest przy okazji zaawansowaną platformą śledzenia błędów, podobną do znanej z projektów fundacji Mozilla — Bugzilla. Możemy z jej pomocą zarówno wyszukiwać rozwiązania naszych problemów, jak i pomóc innym, odpowiadając na ich zapytania.

W sytuacji kiedy mamy problem z modułem, powinniśmy zacząć od kliknięcia *Advanced search* w bloku *Issues for...* na stronie startowej modułu. Prowadzi on do zaawansowanego formularza, dzięki któremu wyszukamy zgłoszone sprawy dotyczące projektu — zarówno błędy, propozycje zmian, jak i prośby o wsparcie (rysunek 4.39).

Search issues for Drupal

Create a new issue · Advanced search · Statistics · Subscribe

Search for

Assigned

Enter a comma separated list of users.

Submitted by

Enter a comma separated list of users.

Participant

Enter a comma separated list of users.

Status

- Open issues -
- active
- needs work
- needs review
- reviewed & tested by the community

Priority

- critical
- normal
- minor

Category

- bug report
- task
- feature request
- support request

Version

- 7.x issues -
- 6.x issues -
- 5.x issues -
- 4.7.x issues -
- x.y.z

Component

- base system
- browser system
- cron system
- database system
- field system

Issue tags

Is one of

Search

Rysunek 4.39. Zaawansowane wyszukiwanie w zgłoszeniach dotyczących projektów

Po wyszukaniu zobaczymy listę spraw pasujących do zadanych kryteriów — w wielu z nich znajdziemy wręcz gotowe kawałki kodu PHP, tak zwane łatki (*patch*), które rozwiązują jakiś problem. Usprawnienia takie, po sprawdzeniu i ewentualnym poprawieniu kodu, często trafiają do kolejnej wersji modułu.

Przestrzeń nazw

Każdy projekt utrzymywany na *drupal.org* ma przydzieloną tak zwaną przestrzeń nazw, czyli unikalny ciąg znaków stosowany do oznaczenia projektu, który będzie przewijał się w wielu miejscach — na przykład jako nazwa katalogu z rozpakowanym modulem, arkusza stylów skórki czy pliku *.info* zawierającego informacje o module. Przykładowo, pobierany w rozdziale 3. „Pierwsze kroki po instalacji” moduł *Administration Menu* ma do swojej dyspozycji ciąg znaków `admin_menu`, który odnajdziemy w bardzo wielu miejscach. Pierwszym z nich będzie ostatnia część adresu, pod którym znajdziemy stronę projektu w ramach witryny *drupal.org* — http://drupal.org/project/admin_menu.

Aktualizacja modułów

W trakcie rozwijania witryny nieraz będziemy musieli wykonać operację aktualizacji modułu. Stanie się tak, jeśli postanowimy zmienić jego wersję na nowszą — bo zawiera ona nowe funkcje, naprawę błędów czy łatkę bezpieczeństwa.

Proces aktualizacji możemy podzielić na kilka kroków, które powinny być zawsze wykonane, jeśli chcemy uniknąć kłopotów:

1. Ustawiamy stronę w tryb przerwy technicznej.
2. Czyścimy wszelkie pamięci podręczne.
3. Wykonujemy kopię zapasową plików i danych.

4. Upewniamy się, że jesteśmy zalogowani jako administrator (użytkownik o identyfikatorze 1).
5. Czytamy dokumentację dostarczoną do aktualizowanego modułu. Szczególnie ważne jest przeczytanie plików *README.txt*, *INSTALL.txt* czy innych, z których nazwy wynika, że mogą nieść ważną informację (np. *READ-THIS-BEFORE-UPGRADE!!!.txt*).
6. Czytamy notę, jaką programiści zamieścili przy pobraniu przez nas wydaniu (czasem nic tam nie ma, czasem znajdziemy listę usuniętych błędów, czasem jednak bardzo ważne informacje o procesie aktualizacji).
7. Pobieramy pliki z modułem.
8. Nadpisujemy stare pliki modułu nowymi, pobranymi w punkcie 7.
9. Uruchamiamy skrypt *.../update.php* i wykonujemy proces aktualizacji.
10. Notujemy ewentualne błędy.
11. Sprawdzamy, czy wszystko działa.
12. Wyłączamy tryb przerwy technicznej.

W 95% przypadków aktualizacja modułu w obrębie danej wersji (np. z wersji 6.x-1.1 na 6.x-1.5) nie niesie żadnych zagrożeń i niespodzianek i trwa, po wykonaniu kroków 1 - 8, kilka sekund. Przy migracji na wyższą linię danego modułu (np. z wersji 6.x-1.1 na 6.x-2.0, czy wręcz 7.x-1.0) musimy już bardzo uważnie przeczytać wszystkie dostarczone dokumenty, wpisy na stronie modułu, noty przy wydaniach. Bardzo często proces aktualizacji jest możliwy, ale tylko z ostatniej wersji „starej” linii, tak więc zanim wrzucimy pliki nowej gałęzi, musimy dostosować swoją instalację i wgrać najnowszą wersję modułu z linii, którą chcemy porzucić.

Czym jest tak naprawdę aktualizacja? Proces ten niewiele różni się od instalacji. Jeśli jest taka potrzeba, tworzone są nowe kolumny bądź tabele w bazie danych, wykonują się operacje modyfikacji danych w istniejących tabelach bądź zmiany atrybutów pól i tabel — na przykład dodanie indeksu.

Dlaczego system aktualizacji jest tak niezawodny? Twórcy modułów są zobowiązani do ścisłego wpasowania się w schemat i utrzymywania ścieżek aktualizacyjnych, czyli specjalnych funkcji w pliku o rozszerzeniu *.install*, które wykonują odpowiednie zmiany w bazie danych, dla każdej nowej wersji. Podczas pracy ze stroną *.../update.php* wykonywane jest sprawdzenie, jaką wersję modułu mamy zainstalowaną, a jaka jest wgrana na serwer. Jeśli okaże się, że nasza wersja to 6.x-1.0 (informacje o tym są zapisane w tabeli *system*), a na serwerze są pliki z wersją 6.x-1.15, zostaną wykonane instrukcje z pliku *install* dla wszystkich pośrednich wersji, aż do obecnej.

Odinstalowywanie modułów

Na głównej stronie z listą dostępnych modułów jest też niepozorna zakładka *Odinstaluj*. Po jej kliknięciu zostaniemy zabrani na stronę z listą modułów, które możemy odinstalować. Jak wspomniano wcześniej, odinstalowanie modułu wiąże się z wykonaniem

specjalnej funkcji z pliku *install* — najczęściej usunięciem zmiennych systemowych oraz stworzonych na potrzeby modułu tabel w bazie danych. Operacja ta jest oczywiście nieodwracalna — chyba że mamy kopie zapasowe naszej bazy danych.

Czasem jednak taki krok może być wymagany i przydatny, szczególnie jeśli przestaliśmy używać jakiegoś modułu, a teraz chcemy zacząć z niego korzystać, ale zależy nam na zupełnie świeżym starcie, bez starych ustawień konfiguracyjnych, które mogły przeleżeć w bazie danych nawet po skasowaniu plików modułu z dysku serwera czy wymianie Drupala z wersji 5 na 6.

Skórki

.../admin/build/themes

Skórki to zestawy plików, dzięki którym nasza strona, zachowując strukturę nawigacyjną i zawartość, może zmieniać wygląd jak kameleon. Wystrój witryny — pliki graficzne tła, logo, ikon, style czcionek, obramowania — można zmienić za pomocą kilku kliknięć, ustawiając inną skórę. Efekt można porównać do zmiany wystawy sklepowej przy pozostawieniu tych samych produktów. Naszym produktem jest zawartość (treść wpisów, nawigacja, formularze), a wystawą jest projekt graficzny.

Dodawanie skórek

Podobnie jak moduły, możemy dodawać do systemu nowe skórki. Operacja ta polega na pobraniu archiwum z plikami skórki i rozpakowaniu go do właściwego folderu — w najprostszej sytuacji będzie to *./sites/all/themes*.

Po umieszczeniu plików we właściwym miejscu zobaczymy nowy wpis na wykazie dostępnych skórek. Teraz wystarczy ją włączyć, zaznaczając pole w kolumnie *Włączony*, i wybrać jako domyślną, aby od razu cieszyć się z nowego wyglądu.

Konfiguracja skórek

System skórek jest w Drupalu bardzo rozbudowany i stwarza projektantom ogromne możliwości. Na szczęście za pomocą interfejsu administracyjnego jesteśmy w stanie skonfigurować większość typowych opcji, bez schodzenia w pliki skórki. Przyjrzyjmy się, co to za opcje.

Po wyświetleniu głównej strony tego działu zobaczymy dostępne skórki, czyli te, których pliki znajdują się w katalogach *./themes*, *./sites/all/themes*, *./sites/default/themes* i katalogach dedykowanych dla poszczególnych witryn (jeśli używamy trybu multisite, omówionego w rozdziale 14. „Uruchomienie witryny”, w punkcie „Drupal w trybie wielowitrynowym (multisite)”).

Wykaz ten prezentuje nam miniaturkę, nazwę i opis skórki, wersję, możliwość ustawienia atrybutu *Włączony*, zaznaczenia skórki domyślnej i konfiguracji włączonych skórek (rysunek 4.40).

Rysunek 4.40.
Dział zarządzania — lista dostępnych skórek

Zrzut ekranu	Nazwa	Wersja	Włączony	Domyślny	Czynności
	Bluemarine Marynistyczna, wielokolumnowa skórka zbudowana z użyciem tabel.	6.15	☒	☐	konfiguruj
	Chameleon Minimalistyczna, jasna skórka zbudowana z użyciem tabel.	6.15	☐	☐	
	Garland Wolna od tabel, dająca się kolorować, wielokolumnowa skórka o zmiennej szerokości (domyślna).	6.15	☒	☒	konfiguruj

W danej chwili możemy mieć kilka włączonych skórek, ale tylko jedna może być domyślna — wyświetlana niezalogowanym i nowym użytkownikom. Po instalacji Drupala jedyną włączoną skórą jest Garland. Jeśli włączymy kolejne, umożliwimy użytkownikom, którym nadamy uprawnienie „wybieranie skórki”, wybór skórki, jaka im najbardziej odpowiada. Wyboru spośród włączonych skórek będą mogli dokonać na stronie edycji swojego profilu (rysunek 4.41).

Rysunek 4.41.
Wybór skórki przez użytkownika

Konfiguracja skórki		
Wybranie innej skórki spowoduje zmianę wyglądu witryny.		
Zrzut ekranu	Nazwa	Wybrany
	bluemarine: themes/bluemarine	☐
	garland: themes/garland (domyślna skórka witryny)	☒

Ustawienia globalne skórek

.../admin/build/themes/settings

Konfigurację skórek możemy podzielić na ustawienia globalne i ustawienia dla konkretnej skórki. Te pierwsze będą stosowane do wszystkich uruchomionych skórek,

dopóki nie zdecydujemy się skonfigurować którejs z nich wprost. *Ustawienia globalne* zawierają tylko opcje, które implementowane są w każdej zbudowanej według standardu skórce Drupala. Możemy więc zdecydować, czy skórka ma wyświetlać poszczególne dane:

- ♦ logo,
- ♦ nazwa witryny,
- ♦ hasło strony,
- ♦ misja strony,
- ♦ portrety użytkowników w treści wpisów,
- ♦ portrety użytkowników w odpowiedziach,
- ♦ pole wyszukiwania,
- ♦ ikona skrótu,
- ♦ odnośniki podstawowe,
- ♦ odnośniki dodatkowe.

Ponadto możemy zdecydować, czy skórka ma wyświetlać informacje o autorze i dacie dodania wpisu dla poszczególnych rodzajów zawartości. Przy założeniu, że włączyliśmy wszystkie moduły z podstawowej dystrybucji, będzie to wykaz, jaki prezentuje rysunek 4.42.

Rysunek 4.42.

Wyświetlanie informacji o autorze i dacie można zdefiniować osobno dla każdego z rodzajów zawartości

Wyświetlaj informacje o dacie dodania i autorze dla

Pozwala włączyć lub wyłączyć wyświetlanie informacji o autorze i dacie dodania dla wpisów określonego typu.

- ☒ Ankieta
- ☒ Artykuł
- ☐ Strona
- ☒ Strona książki
- ☒ Temat na forum
- ☒ Wpis w blogu

Powyższe opcje będą przeniesione na strony konfiguracji wszystkich aktywnych skórek, pod warunkiem że dana opcja nie była tam wcześniej konfigurowana.

Nieco niżej możemy skorzystać z prostego interfejsu do wskazania bądź załadowania na serwer grafiki na logo i ikonę skrótu. Oba formularze pozwalają użyć materiałów dostarczonych ze skórka, wskazać ścieżkę do pliku z grafiką albo wgrać go za pomocą dedykowanego pola (rysunek 4.43).

Rysunek 4.43.
Konfiguracja logo
używanego w skórcie

Konfiguracja przykładowej skórki — Garland

Ustawienia globalne nie są może imponujące, ale poszczególne wersje wystroju witryny mogą zawierać zdecydowanie bogatsze opcje, czego dobrym przykładem jest skórka Garland. Po przejściu na konfigurację tej właśnie skóry widzimy dodatkową sekcję pod tytułem *Paleta kolorów*. Dzięki umieszczonym tam opcjom możemy wybrać jeden z dostępnych schematów kolorystycznych albo skomponować własny. Co więcej, przed zatwierdzeniem zmian schematu mamy bieżący podgląd na to, jak będzie prezentowała się nasza strona w wybranych kolorach (rysunek 4.44).

Rysunek 4.44.
Przybornik kolorów,
lista predefiniowanych
wersji kolorystycznych
i podgląd zmian
na żywo w skórcie
Garland

Poniżej sekcji *Paleta kolorów* znajdziemy standardowe opcje, jakie mogliśmy skonfigurować w ustawieniach globalnych. Jeśli jednak zmienimy którąś z nich (na przykład wyłączymy wyświetlanie logo), to ustawienia globalne przestaną wpływać na tę opcję, dopóki nie użyjemy przycisku *Przywróć domyślne wartości* na samym dole strony konfiguracji danej skórki. Po kliknięciu tego przycisku wszystkie standardowe opcje są ustawiane tak jak na zakładce *Ustawienia globalne* i dopóki nie zdecydujemy się ponownie skonfigurować skórki wprost, zmiany ustawień globalnych będą przenoszone do poszczególnych skórek.

Warto odnotować, że opcja *Przywróć domyślne wartości* nie tylko przywróci ustawienia globalne i ich synchronizację, ale też spowoduje przestawienie konfiguracji skórki do jej wyjściowych parametrów, czyli takich jak świeżo po zainstalowaniu skórki.

Skórka działu zarządzania

Konfiguracja Drupala pozwala nam ustawić inną skórkę dla wyświetlania stron, które są częścią panelu administracyjnego.

Takie rozwiązanie ma kilka zalet, z których bodaj największą jest taka, że jeśli zmienimy skórkę domyślną na taką, która nie działa właściwie i powoduje całkowitą niedostępność serwisu, to nie będziemy mieli kłopotu z jej zmianą. Wystarczy pamiętać, że skórkę zmienimy pod adresem `.../admin/build/themes`. Ponieważ tej operacji dokonamy na stronie działu zarządzania, dla którego wybraliśmy inną niż uszkodzona skórka, to nie będzie kłopotu z wycofaniem zmiany.

Wyboru skórki działu zarządzania dokonamy pod adresem `.../admin/settings/admin`.

Tłumaczenie interfejsu

`.../admin/build/translate`

Drupal został napisany w sposób umożliwiający łatwą zmianę języka interfejsu. Oznacza to, że wszystkie napisy (etykiety pól, listy wartości, komunikaty, tekst pomocy), jakie pokazywane są użytkownikowi, możemy przetłumaczyć.

Napisy, jakie pokazywane są w różnych miejscach systemu, są przechowywane w bazie danych, a specjalny mechanizm (obsługiwany przez moduł *Locale*) pozwala zapisywać w tym miejscu również ich tłumaczenia. W ramach jednej witryny możemy więc mieć kilka wersji językowych interfejsu.

Nie należy mylić interfejsu z zawartością (treścią) wpisów. Tłumaczenie napisów nie obejmuje bowiem tłumaczenia treści. Ta operacja jest również możliwa, ale obsługiwana jest przez inny moduł (*Content translation*), który opisany będzie w rozdziale 6. „Konfiguracja witryny” w punkcie „Języki”.

Jak sprawdzić, w jakim stopniu nasz interfejs jest zlokalizowany

Na stronie głównej (.../admin/build/translate) jest informacja o zainstalowanych językach oraz o kompletności tłumaczenia. Jeśli przy języku, który nas interesuje, znajduje się wartość niższa niż 100%, oznacza to, że istnieją w systemie ciągi znaków, które nie doczekały się tłumaczenia na dany język.

Aby uzupełnić braki w napisach, możemy posłużyć się narzędziami na zakładkach *Szukaj*, *Importuj* i *Eksportuj*. Pierwsza opcja pozwala nam wyszukać dowolny napis i dodać bądź edytować jego tłumaczenie. Drugi z odnośników pozwala na zaimportowanie tłumaczenia. Trzeci służy do eksportowania wszystkich napisów do pliku *.po*, który możemy następnie wygodnie edytować z użyciem notatnika systemowego bądź dedykowanego narzędzia, takiego jak POEdit.

Jak zdobyć pliki *.po*

Do niedawna proces pozyskiwania plików *.po* był zagmatwany i niejasny. Szukało się ich albo na stronach z lokalnym wsparciem, albo w katalogach z pobranym modulem, albo wykonywało się samemu. Tłumaczenia były zazwyczaj autorskie i nie podlegały moderacji, która dbałaby o ich wzajemną korelację i konsekwencję w tłumaczeniu terminologii.

Problemy te skończyły się po uruchomieniu serwera <http://localize.drupal.org>. Jest to centralna platforma dla tłumaczy interfejsu Drupala. Umożliwia grupową pracę nad lokalizacją modułów i samego rdzenia.

Aby skorzystać z umieszczanych tam tłumaczeń, należy zarejestrować się na drupal.org, a następnie wykonać instrukcję:

1. Logujemy się do <http://localize.drupal.org> (z użyciem konta założonego na drupal.org).
2. Przechodzimy na <http://localize.drupal.org/translate/languages/pl/export>.
3. Wpisujemy część nazwy modułu (np. *View*), z podpowiedzi wybieramy interesujący nas moduł (np. *Views*).
4. Wybieramy wersję modułu, jaki mamy zainstalowany.
5. Zostawiamy domyślną opcję z grupy *Type*.
6. Zmieniamy *Packaging* na *All in one file*, tak aby pobrać tłumaczenie w jednym pliku.
7. Reszta opcji bez zmian.

Pobrane pliki możemy od razu zaimportować do swojego serwisu.

Jak zaimportować plik z tłumaczeniem do naszej strony

Aby zaimportować plik tłumaczenia, udajemy się na zakładkę *Importuj* (.../admin/build/translate/import) i w polu *Plik języka*: wskazujemy zbiór z naszego komputera.

Wybieramy, do jakiego języka chcemy zaimportować napisy, oraz decydujemy, czy import ma nadpisać już istniejące tłumaczenia, czy tylko dodać te, które występują w pliku, a w naszej witrynie figurują jako zupełnie nieprzetłumaczone (zalecana jest ta druga opcja, chyba że świadomie chcemy nadpisać istniejące napisy).



Wskazówka

Import pliku z tłumaczeniem może być procesem długotrwałym i w niektórych przypadkach (zależnie od konfiguracji PHP na serwerze) zakończyć się niepowodzeniem. Najczęstsze komunikaty to błąd przekroczenia czasu wykonywania się skryptu bądź błąd braku pliku. Przypadki te zostaną omówione w rozdziale 16. „Typowe problemy”.

Wyszukiwanie i tłumaczenie pojedynczych napisów

Opcja *Szukaj* to narzędzie do odnajdywania ciągów znaków. Przykładowo, jeśli widzimy w systemie nieprzetłumaczony tekst, możemy skopiować go i wpisać w wyszukiwarkę na tej stronie. System (jeśli zostawimy domyślne opcje wyszukiwania) znajdzie nam wszystkie wystąpienia wpisanego ciągu znaków — w każdym języku, jaki jest zainstalowany w witrynie.

Jeśli wyszukiwarka znalazła jakieś pasujące do kryteriów ciągu znaków, pojawi się nam ich wykaz, wraz z oznaczeniem, czy dany ciąg został przetłumaczony. Jeśli tak, jego kod w kolumnie *Języki* nie będzie przekreślony. Dla przykładu, słówko *node* po wyszukaniu w witrynie z uruchomionymi językami polskim i niemieckim i po imporcie polskiego tłumaczenia zwraca kilkadziesiąt wyników z przekreślonym kodem *de* i nieprzekreślonym *pl* (rysunek 4.45).

Rysunek 4.45.
Wyszukany napis i wersje językowe, dla których został przetłumaczony

Kategoria tłumaczenia	Wyrażenie	Języki	Czynności
Standardowy interfejs	Failed to connect to your MySQL database server. MySQL reports the following message: %error. Are you sure you have the correct username an ... includes/install.mysql.inc:44 includes/install.mysql.inc:39	de pl	edytuj usuń

Jeśli zechcemy przetłumaczyć wyszukany ciąg znaków (bądź edytować jego istniejące tłumaczenie), klikamy *Edytuj*, co spowoduje wyświetlenie formularza z oryginalnym wyrażeniem i polami do wprowadzenia bądź edycji jego tłumaczeń (rysunek 4.46).

Rysunek 4.46.
Tłumaczenie napisu

Edycja napisu

Tekst w oryginale:

Users with the `<em>administer nodes</em>` permission will be able to override these options.

Niemiecki:

Polski:

Użytkownicy z uprawnieniami do `<em>zarządzania segmentami</em>` będą mogli nadpisać poczynione tu ustawienia.

Po wprowadzeniu i zapisaniu tłumaczenia będzie ono natychmiast dostępne w systemie i widoczne dla odwiedzających witrynę, którzy wybrali w konfiguracji swojego profilu odpowiednią wersję językową (bądź została im ona wyświetlona na podstawie adresu, subdomeny czy ustawień przeglądarki).



Opisane powyżej opcje są bardzo przydatne, kiedy chcemy przetłumaczyć pojedynczy ciąg znaków bądź ich niewielką liczbę. Jeśli zabieramy się do tłumaczenia dużego obszaru — na przykład całego modułu — zdecydowanie polecam inne metody, na przykład przez opisane niżej pliki `.po` bądź (według mnie najwygodniejsze, bo bez odrywania napisów od kontekstu) tłumaczenie z wykorzystaniem modułu *l10n client* (http://drupal.org/project/l10n_client). Jeśli interesuje Cię, jak działa moduł *l10n client*, zapraszam do obejrzenia pokazu na jego temat na stronie <http://elimu.pl/drupal-6-wygodne-tlumaczenie>.

Zmienne w napisach

W oryginalnych napisach stosowane są bardzo często specjalne ciągi znaków, których podczas tłumaczenia nie powinno się zmieniać. Są to zmienne przechowujące jakieś dane, które mogą zmieniać się w czasie. W tabeli 4.2 przedstawiłem bardzo typowe przypadki.

Umiejętność poprawnego przetłumaczenia napisów, bez uszkodzenia zmiennych, jakie zostały użyte w oryginalnej wersji, jest bardzo przydatna, jeśli sami tłumaczymy interfejs użytkownika. Napisy bez tych zmiennych będą oczywiście działać, ale mogą utracić swój pierwotny sens (i zazwyczaj tak się dzieje).

Łatwo zauważyć, że zmienne są poprzedzane różnymi znakami. Czasem jest to %, a czasem &, kiedy indziej !. Nie wynika to z braku konsekwencji czy przypadku. Jeśli nie będziemy tworzyć modułów do Drupala, to niekoniecznie musimy znać różnicę między tymi trzema sposobami umieszczania zmiennych w łańcuchach tłumaczeń, ale warto wiedzieć, że takowe istnieją. Nie wchodząc w szczegóły, zmienne są przetwarzane na trzy sposoby:

Tabela 4.2. Przykłady zmiennych występujących w napisach

Oryginalny napis	Tłumaczenie polskie	Znaczenie zmiennych
!name cannot be longer than %max characters but is currently %length characters long.	Nazwa !name nie może się składać z więcej niż %max znaków, a w tej chwili ma %length znaków.	!name — nazwa pola w wypełnianym formularzu. %max — maksymalna liczba znaków w polu formularza. %length — liczba wprowadzonych znaków. Stosowane jako ogólny komunikat błędu w formularzach.
Cron has not run. Please visit the status report for more information.	Zadania cron nie zostały wykonane. Więcej informacji w raporcie o stanie witryny.	@status — link (URL) do strony z raportem o statusie witryny.
!datetime – !username	!datetime – !username	Ekstremalny przypadek napisu, który składa się z samych zmiennych połączonych myślnikiem. Wyświetlany w segmentach, które mają w skrócie ustawione pokazywanie informacji o dacie i autorze wpisu.
!name sent a message using the contact form at !form.	Użytkownik !name wysłał wiadomość, korzystając z formularza kontaktowego na stronie !form.	!name — nazwa użytkownika, który wypełnił formularz kontaktowy !form — nieco myląco nazwana zmienna, przechowująca adres strony, na której został wypełniony formularz. Używane w treści wiadomości e-mail powiadamiającej odbiorcę o wypełnieniu formularza kontaktowego.

- ♦ Wszystkie zmienne poprzedzone znakiem @ będą przed wyświetleniem przekazane do funkcji `check_plain` celem sprawdzenia, czy tekst składa się ze znaków zakodowanych utf8, a następnie zamiany znaków specjalnych na encje HTML (zgodnie z opisem funkcji na http://api.drupal.org/api/function/check_plain).
- ♦ Wszystkie zmienne poprzedzone % są traktowane jak te z @, ale dodatkowo otaczane znacznikami `<em>`.
- ♦ Wszystkie zmienne poprzedzone ! są wyświetlane bez żadnego sprawdzenia, co jest potencjalnie groźnym sposobem, jeśli nie mamy 100% pewności, że zostały już na wcześniejszym etapie sprawdzone.

Eksport i obróbka plików .po

Jeśli chcemy przetłumaczyć większą liczbę napisów, wyżej opisany sposób (tłumaczenie pojedynczych fraz przez interfejs Drupala) nie jest optymalny, jeśli chodzi o szybkość czy wygodę. Mając do zlokalizowania cały moduł, na który nierzadko

składa się kilkaset napisów, warto sięgnąć do metody eksportowania i importowania plików *.po* lub *.pot* — w których umieszczone są wszystkie napisy obecne na dany moment w naszej instalacji.

Pliki *.po* są łatwe do obróbki zarówno w zwykłym edytorze tekstowym, jak i w dedykowanych programach — na przykład dostępnym na platformę Windows poEdit.

Na stronie *Eksport* możemy pobrać plik w dwóch formatach. Sekcja *Eksportowanie tłumaczenia* zwróci nam napisy w formacie Gettext Portable Object (rozszerzenie *.po*), a *Eksportowanie szablonu* w formacie szablonu Gettext Portable Object (rozszerzenie *.pot*).

Jaka jest różnica między tymi plikami? Plik *.pot* to szablon zawierający tylko angielskie teksty i miejsce na ich przetłumaczenie, nie zawiera jednak samych tłumaczeń. Jest więc idealnym materiałem na całkiem nowe, niezawierające obcych przekładów tłumaczenie. Plik *.po* dla odmiany będzie zawierał zarówno angielskie, jak i polskie napisy, o ile oczywiście mamy już w serwisie jakieś tłumaczenia, które zaimportowaliśmy bądź wykonaliśmy samodzielnie.



Jeśli chcemy przetłumaczyć jakiś moduł zewnętrzny, warto przeszukać jego katalog CVS w poszukiwaniu szablonu *.pot*. Link do katalogu CVS każdego modułu znajdziemy na jego stronie startowej. Znajdąc unikalną nazwę modułu (nadaną mu przestrzeń nazwiczną, przykładowo *<nazwa>*), możemy od razu udać się pod adres <http://drupalcode.org/viewvc/drupal/contributions/modules/<nazwa>>.

Jeśli nie znajdziemy pliku *.pot* (zazwyczaj umieszczany jest w katalogu *translations*), możemy go wygenerować sami. Aby to zrobić, należy zainstalować specjalny moduł o nazwie *Translation template extractor*, który znajdziemy pod adresem <http://drupal.org/project/potx>. Z jego pomocą po zainstalowaniu modułu do tłumaczenia wyeksportujemy szablon do tłumaczenia.

Struktura plików *.po*

Pliki *.po* i *.pot* posiadają prostą strukturę, dzięki czemu tłumaczenie możemy wykonać nawet za pomocą najprostszego edytora tekstowego obsługującego kodowanie UTF-8. W plikach tych wyróżnia się część nagłówkową — poprzedzoną zestawem dwóch pustych fraz:

```
msgid ""
msgstr ""
```

po których znajdziemy definicję atrybutów pliku, takich jak autor tłumaczenia, data ostatniej zmiany, kodowanie znaków czy nazwa tłumaczonego projektu.

Po nagłówku następuje część zawierająca właściwe napisy wraz z tłumaczeniami, umieszczone w zestawach:

- ♦ komentarz (poprzedzony znakiem #, w plikach generowanych przez Drupal będzie tam informacja, z którego modułu pochodzi napis);

- ♦ napis oryginalny (umieszczony w cudzysłowach poprzedzonych kodem msgid);
- ♦ napis przetłumaczony (umieszczony w cudzysłowach poprzedzonych kodem msgstr).

Rzut oka na plik .po powinien wyjaśnić wiele wątpliwości, zobaczmy więc przykładowe tłumaczenia napisów (tabela 4.3).

Tabela 4.3. Przykładowe tłumaczenia, jakie znajdziemy w plikach .po

L.p.	Fragment pliku .po	Komentarz
1	<code>#: modules/node/node.module:2555</code> <code>msgid "Remove post from front page"</code> <code>msgstr "Usuń ze strony głównej"</code>	Najprostszy przypadek tłumaczenia — zestaw msgid i msgstr.
2	<code>#: modules/node/node.module:2635</code> <code>msgid "Promoted @type %title to front page."</code> <code>msgstr "Wpis %title (@type) umieszczono na stronie głównej."</code>	Tłumaczenie zawierające zmienne @type (rodzaj zawartości) oraz %title (tytuł wpisu).
3	<code>#: modules/poll/poll-bar.tpl.php:25</code> <code>modules/poll/poll.pages.inc:20</code> <code>msgid "1 vote"</code> <code>msgid_plural "@count votes"</code> <code>msgstr[0] "1 głos"</code> <code>msgstr[1] "@count głosy"</code> <code>msgstr[2] "@count głosów"</code>	Tłumaczenie zawierające formy mnogie, zależne od liczby, jaka znajduje się w zmiennej @count.
4	<code>#: /drupal/node/add</code> <code>msgid ""</code> <code>"Strona, podobna w formie do</code> <code>artykułu, służy do "</code> <code>"umieszczania informacji, które rzadko są</code> <code>aktualizowane, na przykład "</code> <code>"stron typu \"0 nas\". Domyślne ustawienia</code> <code>strony nie "</code> <code>"pozwalają na umieszczanie odpowiedzi ani na</code> <code>publikację na stronie "</code> <code>"głównej."</code> <code>msgstr ""</code>	Przykład zestawu, gdzie po słowie msgid występuje kilka wierszy z zawartością w cudzysłowach. Taka sama konstrukcja zachowana jest w tłumaczeniu. Widać również znaki ucieczkowe.

W trzecim przykładzie widać tak zwane znaki ucieczki. Stosuje się je, aby w napisie wyświetlanym użytkownikowi umieścić znak cudzysłowu. W tym celu znak cudzysłowu musimy poprzedzić znakiem \, tak aby silnik gettext nie uznał cudzysłowu za koniec danej frazy. Podobnie rzecz ma się z nową linią — aby zawrzeć w napisie nowy wiersz, musimy wstawić kod sterujący \n.

Formuła liczby mnogiej

Drupal, a dokładniej mechanizm gettext, który wykorzystano w Drupalu, obsługuje liczby mnogie, których zasady są różne dla różnych języków. Dzięki temu możemy poprawnie przetłumaczyć komunikat, w którym występuje zmienna przyjmująca postać liczby, na przykład „Oddano 1 głos”, „Oddano 2 głosy”, „Oddano 10 głosów”.

Aby możliwe było obsłużenie takich form, w pliku *.po* musimy przede wszystkim umieścić definicję form mnogich właściwą dla języka, na który tłumaczymy. Dla polskiego będzie to definicja (za <http://translate.sourceforge.net/wiki/l10n/pluralforms#p>):

```
"Plural-Forms: nplurals=3; plural=(n==1 ? 0 : n%10>=2 && n%10<=4 &&
(n%100<10 || n%100>=20) ? 1 : 2);\n"
```

Definicję taką umieszczamy w części nagłówkowej pliku, zastępując tę, która znajduje się w wygenerowanym pliku *.pot*.

Zgodnie z dokumentacją gettext (<http://www.gnu.org/software/hello/manual/gettext/Plural-forms.html>) mamy w przypadku naszego języka tylko trzy przypadki do obsłużenia:

- ◆ 1 plik,
- ◆ 2, 3, 4 pliki,
- ◆ 5 – 21 plików,
- ◆ 22 – 24 pliki,
- ◆ 25 – 31 plików,
- ◆ ...i tak dalej.

Zatem dla każdego napisu mającego „obsłużyć” liczby mnogie musimy dostarczyć trzy wersje tłumaczenia, jak w przykładzie z tabeli 4.3, pozycja 3:

```
#: modules/poll/poll-bar.tpl.php:25 modules/poll/poll.pages.inc:20
msgid "1 vote"
msgid_plural "@count votes"
msgstr[0] "1 głos"
msgstr[1] "@count głosy"
msgstr[2] "@count głosów"
```

Jak widać, Anglicy mają prościej — albo 1 głos, albo wiele głosów. My mamy jeszcze 2 – 4, 22 – 24, 32 – 34 (i tak dalej) głosy, stąd zamiast jednej formy mnogiej musimy wprowadzić w pliku zawierającym tłumaczenie dwa przypadki.

Wyzwalacze

.../admin/build/trigger/node

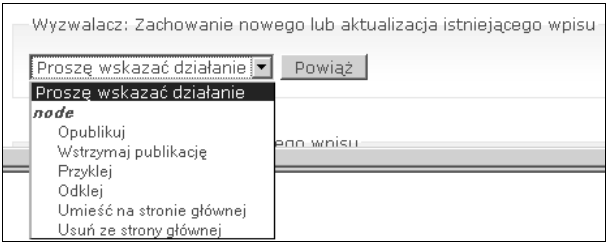
Drupal 6 posiada wbudowany mechanizm wykonywania działań (akcji) pod pewnymi warunkami, zwanymi wyzwalaczami.

Wyzwalaczem może być zatem dodanie nowego wpisu, komentarza, terminu (kategorii w słowniku), czy nawet zalogowanie się użytkownika.

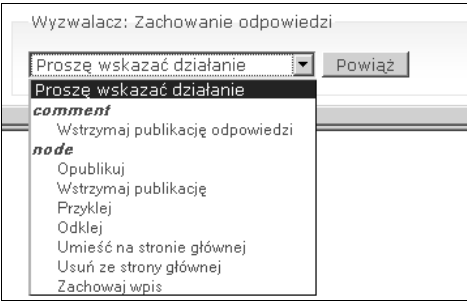
Mimo iż moduł *trigger.module* posiada panel administracyjny, to po instalacji niewiele możemy w nim ustawić. Klikając po dostępnych zakładkach: *Odpowiedzi*, *Zawartość*, *Cron*, *Kategorie*, *Użytkownicy*, zauważymy szereg zdefiniowanych wyzwalaczy, ale tylko do kilku z nich będziemy mogli przypisać działanie.

Z dostępnych zaraz po instalacji działań wyzwalanych przez zdarzenia w systemie możemy wyróżnić te, które mają zdarzyć się po zachowaniu bądź aktualizacji nowego wpisu (dostępne na zakładce *Zawartość*, rysunek 4.47) oraz po zachowaniu, aktualizacji bądź usunięciu odpowiedzi do wpisu (dostępne na zakładce *Odpowiedzi*, rysunek 4.48).

Rysunek 4.47.
Działania, jakie możemy wyzwoić

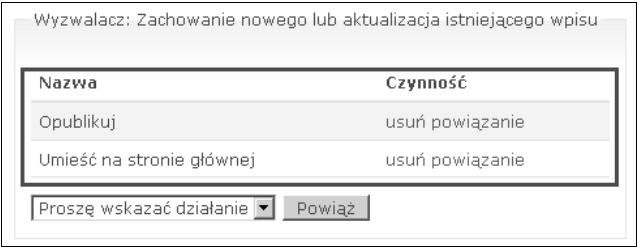


Rysunek 4.48.
Wyzwalacze dostępne w rdzeniu Drupala



Jak widać, opcji nie mamy za wiele. Przykładowo, możemy zdecydować, że każdy wpis ma zostać po zapisaniu opublikowany, i to na stronie głównej, niezależnie od ustawień, jakie wprowadziliśmy na formularzu dodawania zawartości. Tak zdefiniowane akcje będą wyglądały następująco (rysunek 4.49).

Rysunek 4.49.
Działania dla konkretnego wyzwalacza



Możemy przekonać się, że niezależnie od tego, jak ustawimy wpis, zawsze zostanie on umieszczony w statusie „opublikowano” na stronie głównej naszej strony.

Na szczęście to, co widzimy, to tylko przykłady, a nie koniec możliwości. Siłą tego modułu tkwi bowiem w ścisłej współpracy z innym — działaniami zautomatyzowanymi. To właśnie tam zdefiniujemy działania, jakie możemy powiązać z danym zdarzeniem systemowym (wyzwalaczem). Aby to zrobić, musimy przejść do interfejsu zarządzania modułem działań — pod adres `.../admin/settings/actions`.

Jeśli w dziale zarządzania działaniami zautomatyzowanymi dodamy akcję *Wyślij e-mail po dodaniu wpisu* (dodawanie akcji opisano w rozdziale 6. „Konfiguracja witryny”, punkt „Działania zautomatyzowane”), to nasza paleta działań dostępnych w ramach tego wyzwalacza powiększy się (rysunek 4.50).

Rysunek 4.50.
Dołączenie nowego działania



Po dodaniu tej akcji do wyzwalacza *Zachowanie nowego wpisu* zostanie wysłany mail o temacie, treści i adresacie zdefiniowanych w module akcji zautomatyzowanych.

Kolejne akcje i wyzwalacze będą pojawiać się w miarę instalowania nowych modułów, tak więc warto co jakiś czas zaglądać w ten obszar. Bywa, że moduły służące do wykonywania pewnych czynności mogłyby z powodzeniem zostać zastąpione wygenerowaniem odpowiedniej akcji i zaplanowaniem jej w określonych warunkach — tak jak ma to miejsce z powiadomieniem administratora witryny o dodaniu nowego komentarza czy wpisu.

Podsumowanie

Budowa witryny jest działem podstawowym, jeśli chodzi o konfigurację klocków, z jakich strona jest zbudowana — modułów, skórek, bloków, odnośników w menu i tak dalej. To, co ustawimy w tym miejscu, będzie miało wpływ na całą witrynę.

Jak twierdzi autor tej książki, Drupal to coś więcej niż CMS. I ma rację, biorąc pod uwagę fakt, że ten system zarządzania treścią umożliwia to, co nie zawsze jest łatwe w przypadku innych systemów, takich jak Joomla! czy Wordpress — swobodę tworzenia w pełni dopasowanych stron. Dzięki modularnej strukturze i tysiącom dodatków Drupal zapewnia projektantom niemal nieograniczone możliwości kreowania funkcjonalności i wyglądu witryny. Ten potencjał w połączeniu z opinią niezawodnego i elastycznego systemu oraz wielkim wsparciem światowej społeczności sprawił, że wciąż mało popularny w Polsce Drupal na świecie wyrósł na lidera rynku CMS. Dowód? W oparciu o niego powstały tak prestiżowe witryny, jak strona amerykańskiego Białego Domu, oficjalny serwis miasta Londyn, strona króla Belgii czy polskiego Ministerstwa Gospodarki. Jednocześnie od 2007 roku Drupal regularnie zdobywa tytuł najlepszego CMS w licznych konkursach.

I choć jest to system oparty na otwartym kodzie źródłowym, nic za darmo! Drupal uchodzi bowiem za jeden ze skryptów trudniejszych do opanowania. Aby więc ułatwić sobie to zadanie, podejmij naukę z kompletnym podręcznikiem, dzięki któremu szybko i bez trudu ujarzmisz ten niezwykle system. Błyskawicznie opanujesz jego środowisko i przejdziesz do budowania własnej, profesjonalnej witryny. Przeczytasz o tym, jak opracować strukturę menu i wykonać tłumaczenie interfejsu, oraz dowiesz się wszystkiego na temat modułów, „skórek” i szablonów. Nauczysz się sprawnie zarządzać zawartością strony i jej użytkownikami oraz optymalnie ją konfigurować. Krok po kroku prześledzisz proces budowania przykładowego serwisu z wykorzystaniem modułów CCK i Views — zobaczysz, jak atrakcyjnie prezentować treść, zapewnić stronie pełnię funkcjonalności i estetyczny wygląd oraz zadbać o jej bezpieczeństwo. Dowiesz się, jak poprawnie uruchomić swoją witrynę i administrować nią, a także poznasz rozwiązania typowych problemów, na jakie możesz natrafić przy każdym z realizowanych etapów.

Z tej książki dowiesz się:

- Czym jest Drupal, jak działa i jak jest zbudowany
- Jak sprawnie poruszać się w jego środowisku
- Jak poprawnie go zainstalować i uruchomić
- Czym są moduły i jak z nimi pracować
- Jak stworzyć strukturę menu
- Jak dodawać i konfigurować „skórki”
- Jak przeprowadzić tłumaczenie interfejsu
- Jak wprowadzać i edytować zawartość witryny oraz zarządzać nią
- Jak pracować nad prezentacją zamieszczanych treści
- Czym są segmenty i jakie mają atrybuty
- Jak przeprowadzić konfigurację witryny
- Jak zarządzać profilami i uprawnieniami użytkowników
- Jak generować raporty i korzystać z nich
- Jak od podstaw zbudować i uruchomić przykładowy serwis
- Jak utrzymywać gotową witrynę
- Jak sprawnie rozwiązywać typowe problemy
- Jak budować nowe typy zawartości z modułem CCK
- Jak prezentować treść za pomocą modułu Views

Przekonaj się, że nawet tak potężne narzędzie jak Drupal z dobrym podręcznikiem staje się łatwe do opanowania!

Cena: 69,00 zł

Nr katalogowy: 5705



Księgarnia internetowa

<http://helion.pl>



Zamówienia telefoniczne

0 801 339900



0 601 339900

Zamów najnowszy katalog:

➤ <http://helion.pl/katalog>

Zamów informacje o nowościach:

➤ <http://helion.pl/novosci>

Zamów cennik:

➤ <http://helion.pl/cennik>



**Wydawnictwo
Helion**

ul. Kościuszki 1c, 44-100 Gliwice

✉ 44-100 Gliwice, skr. poczt. 462

☎ 32 230 98 63

<http://helion.pl>

e-mail: helion@helion.pl

helion.pl
księgarnia
internetowa

PATRON MEDIALNY



ISBN 978-83-246-2652-6



Informatyka w najlepszym wydaniu