|  |  |  |
| --- | --- | --- |
| gfEngine | 15 stycznia  2012 | |
| Podręcznik gfEngine, wprowadzający do programowania aplikacji w silniku gfEngine. Zawiera opis funkcjonalności wraz z licznymi przykładami. | | Podręcznik wydanie I |

Spis treści

[1. O gfEngine 3](#_Toc298951963)

[1.1. Historia silnika gfEngine 3](#_Toc298951964)

[1.2. Wersja 1.0 3](#_Toc298951965)

[1.3. Wersja 2.0 4](#_Toc298951966)

[1.4. Przyszłość gfEngine 4](#_Toc298951967)

[2. Pierwsze kroki z gfEngine 5](#_Toc298951968)

[2.1. Struktura katalogów silnika 5](#_Toc298951969)

[2.2. Podstawowe skonfigurowanie gfEngine 5](#_Toc298951970)

[2.3. Przygotowania 7](#_Toc298951971)

[2.4. Pierwsza akcja 8](#_Toc298951972)

[2.5. Pierwszy widok 8](#_Toc298951973)

[2.6. Pierwszy model 8](#_Toc298951974)

[2.7. Pierwszy system języku 8](#_Toc298951975)

[2.8. Podsumowanie 8](#_Toc298951976)

[3. Tworzenie akcji, modeli, widoków z wykorzystaniem gfEngine 8](#_Toc298951977)

[4. Omówienie systemów gfEngine 8](#_Toc298951978)

[4.1. Biblioteki silnika 8](#_Toc298951979)

[4.2. Helpery silnika 8](#_Toc298951980)

[4.3. Biblioteki i helpery użytkownika 8](#_Toc298951981)

[4.4. Routing – czyli przyjazne adresy 8](#_Toc298951982)

[4.5. Access Control List – zarządzanie uprawnieniami użytkowników 8](#_Toc298951983)

[5. Zaawansowane systemy gfEngine 8](#_Toc298951984)

[5.1. Setters\_core 8](#_Toc298951985)

[5.2. Filtry (gfFilter) 8](#_Toc298951986)

[5.3. Wtyczki (plugins) 8](#_Toc298951987)

[5.4. Modyfikowanie adresów (rewriter) 8](#_Toc298951988)

[5.5. Własne funkcje i klasy (functions) 8](#_Toc298951989)

[5.6. Logi (Logs) 8](#_Toc298951990)

[6. Omówienie tworzenia akcji z 8](#_Toc298951991)

# O gfEngine

***gfEngine*** to *prosty, elastyczny, niezwykle* szybki Framework do tworzenia *nowoczesnych aplikacji i stron napisanych w języku PHP 5.2*. Silnik spełnia wymogi pozwalające na efektywne tworzenie stron internetowych. **gfEngine** wykorzystuje w swojej architekturze posiada zaimplementowany wzorzec **MVC** (**Model**, **View**, **Controller**), który pozwala na oddzielenie kodu **PHP** od **HTML**, a także wydzielenia *logiki* od *widoku* i *modelu*. Głównym atutem przemawiającym za **gfEngine** jest fakt, że jest szybkim systemem w odróżnieniu od swoich *głównych rywali*. Jednym z ważniejszych czynników podczas tworzenia **gfEngine** była elastyczność, która pozwala na własne modyfikowanie aplikacji i adaptowania rozwiązań z innych systemów.

## Historia silnika gfEngine

**gfEngine** został opracowany w ***2008***r. Początkowo był systemem dostępnym jedynie nielicznym osobom, które wykorzystywały go do tworzenia swoich aplikacji oraz tworzenia stron dla klientów biznesowych. Numeracja **gfEngine** ma swój początek od wersji **0.1** – która nie była wersją publiczną. **0.2** to rozkwit frameworka, który już wtedy był stosowany do tworzenia aplikacji biznesowych, wersja ta przyniosła wiele dobrych zmian, które przyczyniły się do dalszego rozwoju silnika. Wersja oznaczona numerkiem **0.3** była głównie drobnymi poprawkami względem wersji **0.2** oraz wprowadziła kosmetyczne zmiany w samym silniku. Wersją niezwykle udaną była wersja **0.4**, zostały dodane nowe narzędzia, silnik został przebudowany, właśnie ten build powiększył grono programistów korzystających z **gfEngine**.

## Wersja 1.0

**gfEngine 1.0** został wydany dnia *15 stycznia 2012r*. Silnik został całkowicie przebudowany i przepisany, owe zabiegi podwyższyły jego czas generowania strony. Jeszcze bardziej zmniejszono wagę całego silnika. Zaimplementowano system logowań zdarzeń (logging events) oraz dodano obsługę wyjątków (exceptions). Kolejne wydania są planowane co 6 miesięcy od poprzedniego, będą one wprowadzać poprawki do poprzednich wersji, będą dodawane nowe narzędzia i możliwości. Jednocześnie będzie ciągle utrzymana zasada **gfEngine: prostota, wygoda, szybkość**.

## Wersja 2.0

**Wersja 2.0** jest przyszłością silnika **gfEngine**, aktualnie jest w trakcie tworzenia, data oficjalnego wydania jest zaplanowana na dzień 11 listopada 2011 roku. **gfEngine 2** będzie *wydaniem wprowadzającym rewolucje, z nowym silnikiem, z nowymi bibliotekami i narzędziami pozwalającymi tworzenie jeszcze lepszych aplikacji*.

***Rewolucja nadejdzie 11.11.11!***

## Przyszłość gfEngine

**gfEngine** ma oficjalnie być rozwijany odpowiednio wersja 1.0 – do roku po premierze czyli do 15.08.2012r, zaś wersja 2.0 co najmniej 2 lata od premiery, z możliwością przedłużenia.

# Pierwsze kroki z gfEngine

**gfEngine v1.0** – dział ten skupia się na podstawach programowania w frameworku. Zostanie stworzony pierwszy kontroler.

Zanim zostanie stworzony pierwszy kontroler, należy ustalić parę terminów. Często w tym podręczniku będą używane takie pojęcia jak:

* **Dostęp akcyjny** – czyli poruszanie się wewnątrz modułu, akcji, czyli klasie odpowiedzialnej za logikę.
* **Dostęp globalny** – używany w filtrach, setters\_core, use\_functions itd.

Jeżeli zostanie użyty, któryś z tych terminów należy pamiętać co on oznacza.

## Struktura katalogów silnika

|  |  |
| --- | --- |
|  | |
| /apps/ | Główny katalog aplikacji |
| /apps/cache/ | Katalog z danymi z cache |
| /apps/cache/db/ | Katalog cache bazy danych |
| /apps/configs/ | Katalog zawierający konfiguracje silnika |
| /apps/filters/ | Katalog zawierający filtry |
| /apps/functions/ | Katalog zawierający funkcje I klasy użytkownika |
| /apps/libs/ | Katalog zawierający biblioteki użytkownika |
| /apps/modules/ | Katalog zawierający moduły tj. strony |
| /apps/setters\_core/ | Katalog zawierający system ustawiania silnika |
| /apps/web/ | Katalog zawierający motyw strony |

## Podstawowe skonfigurowanie gfEngine

Konfiguracja **gfEngine** w przeciwieństwie do innych frameworków jest niezwykle prosta, cala konfiguracja znajduje się w zaledwie kilku plikach. Główną konfigurację możemy znaleźć w pliku **/apps/configs/configs.ini** Plik ten powinien wyglądać mniej więcej tak:

|  |
| --- |
| filters = 0  rewrite = 0  rewrite\_rules = 0  setters\_core = 0  logs = 0  system\_conf = 0  functions\_use\_functions = 0  functions\_use\_classes = 0  access = 0  use\_access = "gf"  db\_use = 0  db\_host = "localhost"  db\_user = "gf"  db\_password = "gf"  db\_database = "gf" |

Opcja ***filters*** – dokładne omówienie w dalszych rozdziałach podręcznika.

Opcja ***rewrite*** – dokładne omówienie w rozdziale: **Routing**.

Opcja ***rewrite\_rules*** – dokładne omówienie w dalszych rozdziałach.

Opcja ***setters\_core*** – ustawienie zaawansowane silnika gfEngine, rozdzial „**setters\_core**”

Opcja ***logs*** – ustawienie rejestrowanie zdarzeń zachodzących podczas pracy.

Opcja ***system\_conf*** – ustawienie pracy silnika, przyjmuje 3 wartości:

*0* – ustawienie publiczne, powinno być ustawione, gdy aplikacja już widoczna w sieci.

*1* – ustawienie testowe, tylko niektóre błędy będą wyświetlane.

*2* – ustawienie w pełni testowe, wszystkie błędy będą odnotowywane.

Opcja ***functions\_use\_functions*** – informacje w dalszej części podręcznika.

*Opcja* ***functions\_use\_classes*** – informacje w dalszej części podręcznika.

Opcja ***access*** – dostępne wartości:

*0* - Kontrola uprawnień wyłączona

*1* – włączona manualna

2 – włączona automatyczna

Opcja ***use\_access*** – więcej informacji w dalszej części podręcznika

Opcje ***db\_\**** - dokładne informacje w sekcji: **Bazy danych**

## Przygotowania

Zanim zostanie stworzona pierwsza akcja, model, widok należy wyjaśnić parę spraw. Swoje kontrolery, czyli strony umieszczamy w katalogu **/apps/modules/** należy rozróżniać dwa terminy:

* **Kontroler** – główny magazyn akcji, można nazwać go grupą, jest kontenerem.
* **Akcja** – przynależy bezpośrednio do kontrolera, odpowiedzialna za logikę aplikacji

Czym dokładniej jest kontroler? Przeanalizujmy adres strony:

**gfengine.pl/user/register.htm**

oraz: **gfengine.pl/index.php?con=user&act=register**

Oba adresy prowadzą do tego samego miejsca, jednakże należy wybrać w ustawieniach, z których adresów masz korzystać silnik.

W tym przypadku w adresie jest ukryty kontroler, który ma wyszukać silnik oraz akcję W TYM kontrolerze, którą uruchomi **gfEngine**. Czyli odpowiednio: w tej sytuacji jako kontroler mamy: *user*, a jako akcję: *register*, która musi być w danym kontrolerze. Jeżeli silnik jest ustawiony na korzystanie z przyjaznych linków to budowa adresu wygląda tak: gfengine.pl/**KONTROLER**/**AKCJA**, zaś w przypadku korzystania z normalnych linków: gfengine.pl/index.php?con=**KONTOLER**&act=**AKCJA**

Jednakże jak to się ma w przypadku budowy katalogów i plików? Należy rozpatrywać folder: **/apps/modules/**

W domyślnej instalacji gfEngine znajdują się 3 foldery: **index, error, short.** Te właśnie foldery są **KONTROLERAMI**, tak więc odwołując się do strony: **gfengine.pl/index**, odwołamy się bezpośrednio do folderu **/apps/modules/index**, należy od razu wspomnieć o tym, że **AKCJA** tak jak w tym adresie nie została podana, jeżeli nie wskażemy silnikowi **AKCJI**, **gfEngine** będzie szukać *akcji index w kontrolerze index*.

Teraz już wiadomo co to jest kontroler, oraz po części akcja. gfEngine wyszukując dany kontroler, otwiera dany folder oraz przeszukuje daną akcję, czyli korzystając z adresu: **gfengine/index/index.htm** gfEngine wyszuka kontroler **INDEX**, a następnie zacznie szukać akcji **INDEX** – struktura kontrolera powinna zawierać w sobie foldery poniżej wymienione:

|  |
| --- |
| actions |
| models |
| langs |
| views |
| plugins |

W folerze action znajdują się owe akcje, których szuka **gfEngine** w danym kontrolerze. Dana akcja wiąże się z przynależnością **widoków, języków, modeli, wtyczek** do danej akcji danego kontrolera. Podsumowując: **Kontroler** – kontener zawierający akcje, **akcja** – odpowiedzialne za przetworzenie strony, logikę.

## Pierwszy kontroler

Aby przystąpić do budowy pierwszej strony należy stworzyć kontroler, który będzie kontenerem akcji. Tworzenie kontrolera nie jest trudnym zadaniem i wiąże się jedynie z utworzeniem folderów i plików pierwszej akcji – szablonu. Aby utworzyć kontroler należy wejść do folderu /apps/modules/ i stworzyć folder o dowolnej nazwie – nazwa ta będzie jednocześnie nazwą kontrolera i będzie wykorzystywana w adresie (chociaż jest możliwe pominięcie tej reguły poprzez zastosowanie Rewriter’a – omówienie w dalszej części podręcznika). Po stworzeniu nowego foldera, należy otworzyć go i stworzyć następujące katalogi:

* actions
* langs
* models
* plugins
* views

Po utworzeniu tych katalogów można powiedzieć, że schemat kontrolera jest już zbudowany, następnie należy stworzyć domyślne pliki kontrolera – akcji, modelów, widoków itd. Są to pliki

## Pierwsza akcja

## Pierwszy widok

## Pierwszy model

## Pierwszy system języku

## Podsumowanie

# Tworzenie akcji, modeli, widoków z wykorzystaniem gfEngine

# Omówienie systemów gfEngine

## Biblioteki silnika

## Helpery silnika

## Biblioteki i helpery użytkownika

## Routing – czyli przyjazne adresy

## Access Control List – zarządzanie uprawnieniami użytkowników

# Zaawansowane systemy gfEngine

## Setters\_core

## Filtry (gfFilter)

## Wtyczki (plugins)

## Modyfikowanie adresów (rewriter)

## Własne funkcje i klasy (functions)

## Logi (Logs)

# Omówienie tworzenia akcji z