

DJANGO CMS.

M. G. T. P. K. S.

Wydział Fizyki, Matematyki i Informatyki
Politechnika Krakowska im. Tadeusza Kościuszki

17.05.2019

Plan.

- 1 Wstęp.
- 2 Popularne CMSy.
- 3 Django.
- 4 Cechy.
- 5 Tworzenie stron.
- 6 Publikowanie stron i uprawnienia.
- 7 Przykład - blog.
- 8 Podsumowanie.
- 9 Źródła.

CMS to skrót od angielskiego **Content Management System** (System Zarządzania Treścią). Jest to strona internetowa, posiadająca panel administracyjny, który pozwala na samodzielne i bezpłatne zmienianie treści bez znajomości programowania. Dodawanie, modyfikowanie i usuwanie tekstów, tabel, grafik, zdjęć, filmów, a nawet całych podstron i działów odbywa się przez klikanie przycisków. Podczas pisania tekstu można używać przycisków do formatowania tekstu podobnych do tych z Worda. Najpopularniejsze systemy CMS posiadają (zazwyczaj darmowe) dodatki, zwane wtyczkami. Wtyczką może być np. galeria zdjęć, zestaw dodatków pomagających w pozycjonowaniu, forum dyskusyjne, system rezerwacji pokoi hotelowych, czy przycisk "Lubię to" Facebooka.

Populärne CMSy.

- WordPress
- Joomla
- Drupal
- ModX
- ExpressionEngine
- TextPattern
- Django

- Automatycznie generowany i kompletny panel administracyjny.
- Prosty lecz funkcjonalny system szablonów.
- Obsługuje następujące bazy danych: PostgreSQL, MySQL, SQLite oraz Oracle.
- Wsparcie dla wielojęzycznych aplikacji.
- Własny, prosty serwer do testowania aplikacji.
- Rozpowszechniany jest na otwartej licencji BSD.

Instalacja.

Przed instalacją Django należy zainstalować Pythona w wersji 3.4 lub nowszej. Warto również stworzyć **środowisko wirtualne** (virtualenv), którego zadaniem jest oddzielenie środowiska Pythona/Django dla każdego projektu z osobna. Oznacza to, że zmiany dokonane w obrębie jednej aplikacji nie wpłyną na działanie innych, nad którymi pracujemy. Aby utworzyć nowego środowisko wirtualne, należy otworzyć okno wiersza polecenia i wykonać następujące polecenie:

```
python -m venv myenv
```

Środowisko wirtualne uruchamiamy za pomocą polecenia:

```
myenv/Scripts/activate
```

Django instalujemy za pomocą polecenia:

```
pip install Django
```

Kiedy CMS jest już zainstalowany, skonfigurowany i uruchomiony można przystąpić do tworzenia strony. Aby to zrobić należy przejść do panelu administracyjnego i dodać nową stronę (Page) w sekcji CMS. Po podaniu tytułu strony, wybraniu języka i szablonu warto kliknąć **“save and continue editing”** po to by określić szczegółowe parametry strony. Języki jakie ma obsługiwać CMS, dostępne szablony stron, a także wiele innych parametrów CMS konfiguruje się w **settings.py**. Pełna lista opcji konfiguracyjnych django CMS znajduje się w dokumentacji.

Tworzenie stron.

Django administration Welcome, **ext**. [Change password](#) / [Log out](#)

[Home](#) > [Cms](#) > [Pages](#) > strona-glowna

✔ The page "strona-glowna" was added successfully. You may edit it again below.

Change page (you can perform actions on this page directly)

[History](#) [View on site](#) ➤

French | **German** | **English** | **Brazil**

Title: **Menu Title:**
The default title Overwrite what is displayed in the menu

Slug:
The part of the title that is used in the URL

Is published In navigation

Basic Settings

Note: This page reloads if you change the selection. Save it first.

Template: ▼
The template used to render the content.

Right Column

[...] ⓘ Plugin saved successfully.

▼ ➤ Add Plugin

Body

▼ ➤ Add Plugin No Plugins present. Add a plugin to this placeholder-slot.

Advanced Settings (Show)

SEO Settings (Show)

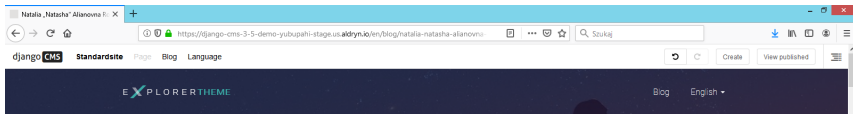
Page permissions

User	Group	Can edit	Can add	Can delete	Can change advanced settings	Can publish	Can change permissions	Can move	Can moderate	Grant on	Delete?
<input type="text" value="....."/>	<input type="text" value="....."/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="text" value="Page and descendants"/>	<input type="checkbox"/>
+	+										
<input type="text" value="....."/>	<input type="text" value="....."/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="text" value="Page and descendants"/>	<input type="checkbox"/>

Publikowanie stron i uprawnienia.

Strona po utworzeniu nie jest od razu widoczna dla użytkowników. Aby dokonać jej publikacji należy zaznaczyć znacznik **Is published**. Jeśli dodatkowo strona ma być widoczna w menu nawigacyjnym trzeba zaznaczyć także opcję **In navigation**. Dla każdej strony możemy zdefiniować uprawnienia, które określają co dany użytkownik lub grupa użytkowników może ze stroną robić (edytować, usuwać, moderować itp.). Co bardzo przydatne uprawnienia mogą się automatycznie propagować na strony umieszczone niżej w hierarchii. Dla każdej ze stron można także określić, czy dostęp do niej wymaga logowania oraz czy ma się ona pokazywać w menu wszystkim, czy tylko zalogowanym użytkownikom.

Przykład - blog.



Natalia „Natasha” Alianovna Romanova



Natalia urodziła się w 1927 lub 1928 roku w Stalingradzie, w rodzinie, której nazwisko sugeruje związek z ostatnimi carami Rosji, jednak pokrewieństwo to nigdy nie zostało potwierdzone. O jej rodzicach wiadomo tylko tyle, że matka zginęła, ratując będącą jeszcze niemowlęciem Natalię z płonącego budynku. Człowiekiem, który zabrał dziecko z rąk konającej kobiety i zaopiekował się nim, był oficer Armii Czerwonej, Ivan Petrovitch.

Ivan zajmował się Natalią przez około dziesięć lat, a gdy nie mógł tego robić, trafiła tymczasowo do ludowych sierotinców. By oszczędzić jej takiego losu, odszukał dalekiego krewnego dziewczynki i zamierzał oddać ją pod jego opiekę. Okazało się jednak, że Taras Romanoff jest wysokim urzędnikiem KGB i dyrektorem szkoły dla przyszłych szpiegów, zaś generalissimus Stalin osobiście zdecydował, że młoda Romanova ma zostać poddana szkoleniu, gdy tylko dowiedzial się o jej istnieniu.

W 1940 roku Natalia była już zdolną adeptką specjalnego wydziału szkoleniowego KGB. W tym właśnie czasie Romanoff sprowadził do swego ośrodka Kanadyjczyka nazwiskiem

Przykład - blog.



Przykład - blog.

The screenshot shows a web browser window with the following elements:

- Browser Address Bar:** URL is `https://django-cms-3-5-demo-yubupahi-stage.us.kddyn.io/en/blog/natalia-natasha-atanova-`. Search bar contains "Szukaj".
- Page Header:** "django CMS Standardsite" with navigation links for "Page", "Blog", and "Language".
- Navigation Bar:** "E X P L O R E R 110-M4" logo on the left, "Blog" and "English" on the right. Below it are buttons: "← Previous Article", "← Back to Overview →", and "Next Article →".
- Comments Section:**
 - Header: "0 Comments Xplorer" with a "Login" button.
 - Actions: "Recommend", "Tweet", and "Share". Sort by "Best".
 - Form: A text input field containing "Świetny artykuł, proszę o więcej!". Below it is a rich text editor toolbar with icons for Bold, Italic, Underline, Bulleted List, Numbered List, Link, Unlink, Image, and Embed.
 - Authentication: "LOG IN WITH" (with icons for D, f, t, G) and "OR SIGN UP WITH DISQUS" (with a help icon).
 - Input: A "Name" text field.
 - Text: "Be the first to comment."
- Footer:** "Subscribe", "Add Disqus to your site", "Disqus' Privacy Policy", and the "DISQUS" logo.
- Related Articles:** A section titled "Related Articles" is visible at the bottom.

Podsumowanie.

Rozwiązania oparte na Pythonie są zdecydowanie mniej narażone na większość typowych zagrożeń dla systemów, które są przygotowane w języku PHP (Wordpress, Drupal, Joomla). Powszechne skrypty spawowe, które szukają dziur w zabezpieczeniach popularnych CMSów na PHP zaczynają w 90% przypadków od wstrzykiwania kawałka kodu PHP. Kod PHP, który jest rozsyłany przez spambots zwyczajnie nie wywoła żadnego skutku na tym systemie zarządzania treścią. Zostanie zignorowany, nie powodując spustoszenia na naszej stronie i problemów z odzyskaniem pełnej sprawności serwisu. Po skonfigurowaniu systemu tak, by można było na nim pracować i dodawać treści Django CMS jest idealnym rozwiązaniem dla niejednego serwisu. Śmiało może konkurować z najpopularniejszymi, Open Source'owymi CMSami typu Wordpress, Joomla czy Drupal.

Źródła.

 <https://www.django-cms.org/en/>

 https://www.youtube.com/watch?v=6jn_-WyRcaY