

# **Mini-CMS (EJS + Express)**

Przykładowa, lekka aplikacja CMS napisana w Node.js z wykorzystaniem Express i EJS. Umożliwia przeglądanie listy artykułów, wyświetlanie pojedynczych wpisów oraz dodawanie nowych, które zapisywane są w pliku JSON (data/articles.json).

## **Struktura katalogów**

artykuly/

- |- src/ – kod źródłowy
  - |- server.js – punkt startowy
  - |- app.js – konfiguracja Express i middleware
  - |- routes/ – definicje tras
    - |articles.js – definicje tras podstron artykułów
  - |- controllers/ – logika kontrolerów
    - |articlesController.js – render, walidacja, wyświetlanie
  - |- services/ – usługi aplikacyjne
    - |articleServices.js – generowanie slugów, tworzenie artykułów
  - |- models/ – dostęp do danych
    - |storage.js – zapis i odczyt JSON
- |- views/ – szablony EJS
  - |- articles/ – zawiera elementy strony związane z tworzeniem i wyświetaniem artykułów
  - |- error.ejs – odpowiada za wyświetlanie błędów podczas korzystania z aplikacji

- | - list.ejs – wyświetla listę artykułów
- | - new.ejs – wpisanie nowego artykułu
- | - show.ejs – wyświetla zawartość artykułu
- | - partials/ – customowe elementy używanie na jako części innych szablonów
  - | - footer.ejs – customowa stopka
  - | - header.ejs – customowy nagłówek
- | - public/ – pliki statyczne (CSS, grafiki)
- | - data/articles.json – lokalne repozytorium artykułów

## Najważniejsza funkcjonalność

- Zarządzanie artykułami: listowanie, podgląd, tworzenie
- Widoki oparte na EJS (views/articles/\*)
- Dane przechowywane lokalnie w data/articles.json
- Prosta i czytelna struktura projektu – idealna do nauki

## Najważniejsze funkcje

- readArticles() - czyta informacje z pliku JSON
- writeArticles() - wpisuje dane do pliku JSON
- slugify(title) - uproszcza tytuł według podanego regresa
- listArticles() - zwraca dane z pliku JSON
- getArticlesBySlug() - szuka artykułu po slugu
- createArticle() - sprawdza unikalność slua i tworzy dane do pliku JSON

- index(req,res) - redneruje stronę listującą artykuły
- show(req,res) - renderuje stronę podglądu artykułu
- newForm(req,res) - renderuje stronę nowego formularza
- create(req,res) - waliduje podane wartości i przekazuje je by stworzyć artykuł

## **Wymagania**

- Node.js (zalecane v16 lub nowsze)
- npm (do instalacji i uruchamiania skryptów)

## **Szybki start**

### 1. Zainstaluj zależności:

```
komenda: "npm install"  
# lub zgodnie ze skryptem projektu: npm prepare
```

### 2. Uruchom serwer:

```
komenda: "npm start"  
# uruchamia node src/server.js
```

### 3. Otwórz w przeglądarce adres <http://localhost:3001> (port domyślnie ustawiony w src/server.js).

Podczas pracy nad kodem możesz użyć nodemon (globalnie lub jako devDependency) i odpalić nodemon src/server.js – serwer będzie się automatycznie restartować przy każdej zmianie plików.

## **Dostępne trasy**

- GET / – lista wszystkich artykułów
- GET /articles/new – formularz dodawania nowego wpisu
- POST /articles – zapis nowego artykułu (wymagane pola: title, content, author)
- GET /articles/:slug – wyświetlenie artykułu po slug

## **Zarządzanie artykułami**

- Artykuły możesz dodawać przez formularz dostępny pod /articles/new.
- Wszystkie wpisy są zapisywane w pliku data/articles.json. Możesz też edytować go ręcznie, ale zachowaj poprawny format JSON.