



wydarzen.io - dokumentacja projekt



Link do repozytorium GitHub: <https://github.com/wojtekweg/wydarzen-io/>

Wstęp

Użyte technologie

Backend

Frontend

Testy

CI

Podstawowe funkcjonalności

Baza danych wydarzeń

Tablica wydarzeń

Import wydarzeń z zewnętrznych serwisów

Asynchroniczne zadania aplikacji

Wygodny interfejs użytkownika

Dalsza praca

Błędy

Planowane funkcjonalności

Funkcjonalności, które nie są przewidywane

Potencjalne usprawnienia

Dodatkowa dokumentacja

Dokumentacja API backendu

Diagram relacji tabel bazy danych

Wstęp

wydarzen.io to prosta aplikacja internetowa, mająca za zadanie pełnić funkcjonalność kalendarza osobistego lub planera wydarzeń dla niewielkich instytucji lub osób organizujących wydarzenia, takie jak różne warsztaty, koncerty lub inne spotkania.

Użyte technologie

Aplikacja posiada własny backend oraz frontend, a więc jest to projekt full-stack.

Backend

Backend aplikacji zasila `Django` - framework do tworzenia aplikacji webowych w `Python`, pozwalający na zbudowanie pełnej aplikacji internetowej na podstawie wzorca projektowego Model-View-Template.

W moim projekcie zrezygnowałem z używania silnika templatek wbudowanego w Django na rzecz widoków opartych na REST API. Do tego celu służy paczka `django-rest-framework`.

Kolejne zewnętrzne biblioteki z których korzystam to `beautifulsoup4` i `requests`, które pomagają w pobieraniu danych z sieci. Paczki `icalendar` oraz `python-dateutil` pomagają w importowaniu wydarzeń z innych kalendarzy (Facebook, Google, USOS) do aplikacji. Biblioteka `celery` używana jest przeze mnie do tworzenia asynchronicznych zadań, takich jak pobieranie zdjęć dla wydarzeń czy notyfikowanie użytkowników o nadchodzącym wydarzeniu.

Frontend

Frontend aplikacji został napisany w `react`. Jest to moje pierwsze spotkanie z tą biblioteką. Zapytania do serwera wykonywane są za pomocą REST API przy wykorzystaniu biblioteki `axios`. Aplikacja `gh-pages` wykorzystywana jest przeze mnie to publikacji frontendu aplikacji na darmowym hostingu GitHub pages. Wykorzystuję biblioteki `date-fns` oraz `final-form` do wygodniejszej obsługi kolejno dat oraz formularzy. Paczki `react-big-calendar` oraz `react-toastify` pomagają mi w bardziej estetycznym stylowaniu elementów aplikacji (kolejno duży widok kalendarza oraz obsługa frontendowych notyfikacji).

Sam widok frontendowy jest napisany przy wykorzystaniu biblioteki `tailwindcss`, która w znaczący sposób ułatwia stylowanie aplikacji oraz nadaje spójny i estetyczny wygląd.

Frontend aplikacji, bez połączenia z bazą danych i kilkoma przykładowymi obiektami zapisanymi jako odpowiedź z bazy można zobaczyć pod poniższym adresem:



Frontend serwisu: <https://wojtekweg.github.io/wydarzen-io/>

Testy

Aplikacja posiada unit testy backendowe (napisane przy pomocy domyślnej biblioteki `unittest` wbudowanej w Django) oraz testy end-to-end napisane w frameworku `cypress`. Testy te sprawdzają końcowy widok interfejsu użytkownika oraz integrację strony backendowej oraz frontendowej.

Nagranie z wykonania testów E2E w wygodny sposób prezentuje podstawowe funkcjonalności aplikacji oraz jej interfejs:

<https://youtu.be/j5W0vDGsXGE>

CI

Repozytorium kodu na GitHubie posiada zainstalowane aplikacje Dependabot oraz GitHuba Actions, które pozwalają na automatyczną aktualizację paczek, bibliotek i frameworków w momencie jak tylko zostaną opublikowane nowe wersje (tworzony wtedy jest `Pull request`). Każda potencjalna aktualizacja uruchamia testy jednostkowe backendu oraz testy interfejsu użytkownika w chmurze (za pomocą GitHub Actions), co daje natychmiastową informację o potencjalnych problemach, wynikających z aktualizacji.

Ostateczna decyzja o przyjęciu aktualizacji danej biblioteki należy jednak do właściciela repozytorium.

Podstawowe funkcjonalności

Baza danych wydarzeń

Każde wydarzenie może posiadać swój opis, zdjęcie, bazodanową relację do miejsca, datę oraz status aktywności (czy jest aktywne, wydarzenia są aktualizowane jako nieaktywne, jeśli są przeszłe za pomocą asynchronicznego zadania).

Tablica wydarzeń

Z możliwością sortowania obiektów, filtrowania po nazwie, wyświetlania według danego miejsca, filtrowania po wydarzeniach aktywnych/nieaktywnych oraz w danym zakresie dat. Dostępny jest też widok kalendarza.

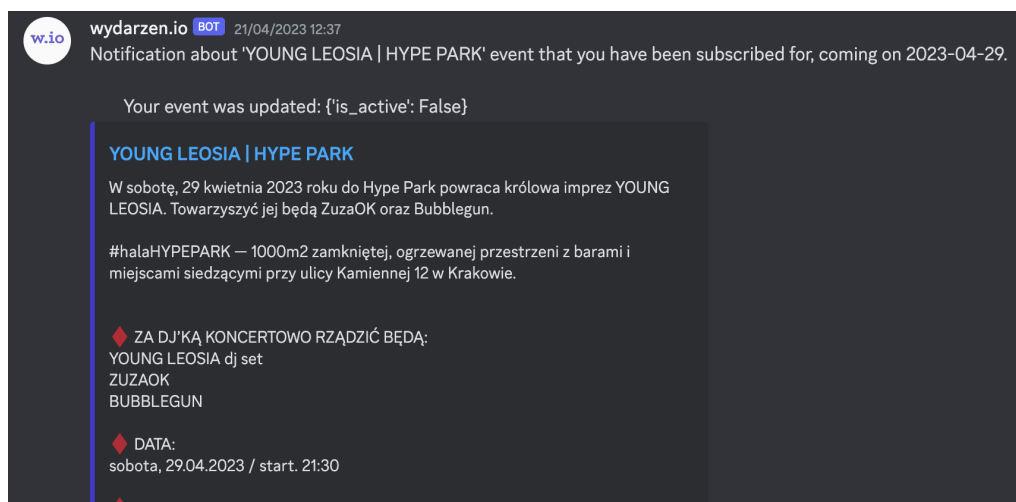
Import wydarzeń z zewnętrznych serwisów

Aplikacja wspiera import wydarzeń z Facebooka, różnych kalendarzy z rozszerzeniem .ical (jest to dość popularne rozszerzenie, wspierane też przez np. Google Calendar, iCalendar, USOS) lub plików .json.

Asynchroniczne zadania aplikacji

Aplikacja posiada trzy zadania asynchroniczne:

- Aktualizowanie wydarzeń jako nieaktywne, jeśli ich data już minęła
- Pobieranie zdjęć z wyszukiwarki Bing według tytułu wydarzenia dla obiektów, które nie mają własnych grafik (np. importowane z zewnętrznych kalendarzy)
- Notyfikacje dla nadchodzących wydarzeń w aplikacji Discord
 - admin może dodać kolejne kanały Discord do danego wydarzenia oraz włączyć lub wyłączyć powiadomienia informujące o nadchodzącym wydarzeniu na kanale Discord z podanym adresem
 - pozostali użytkownicy mogą zobaczyć dane subskrypcje, lecz nie mogą ich edytować
 - przykładowe powiadomienie:



Wygodny interfejs użytkownika

Styl aplikacji jest responsywny, co oznacza, że aplikacja jest wygodna do używania zarówno na smartfonach, jak i na komputerach.

Strona wydarzen.io posiada też system powiadomień o błędach, zwijany pasek nawigacji oraz tryb jasny i ciemny, do wyboru wedle preferencji użytkownika.

Widok kalendarza aplikacji pozwala na wyświetlanie wydarzeń w trybie dziennym, miesięcznym oraz rocznym.

Dalsza praca

Błędy

Znane błędy aplikacji zapisane są w zakładce **Issues** w repozytorium Github:

<https://github.com/wojtekweg/wydarzen-io/issues>

Planowane funkcjonalności

- System miejsc nie jest wspierany w wystarczający sposób - brakuje opcji dodawania zdjęć miejsc oraz wygody w wybieraniu ich lokalizacji
- Testy jednostkowe kody frontendowego poprawiłyby komfort pracy z kodem
- Poprawa frontendowego widoku aplikacji - informacja o braku połączenia z bazą danych, wyłączenie błędów, wynikających z braku połączenia, przygotowanie bardziej reprezentatywnych przykładowych danych

Funkcjonalności, które nie są przewidywane

- Zarządzanie użytkownikami - aplikacja nie została stworzona we wsparciu dla wielu użytkowników, stąd też brak systemu rejestracji czy rozróżnienia działania aplikacji dla różnych użytkowników

Potencjalne usprawnienia

- Jakość kodu we frontendowej części aplikacji - był to mój pierwszy projekt przy użyciu React, stąd też wiele części kodu wymaga poprawy i usprawnień
- Optymalizacja wydajności - obecnie, w wielu miejscach aplikacji pobierane są przez API wszystkie wydarzenia oraz filtrowane po stronie frontendowej. Jeśli aplikacja miałaby być publikowana komercyjnie, musiałaby posiadać filtrowanie obiektów po stronie backendowej
- Mimo wszelkich starań, wciąż jest wiele miejsc, gdzie styl strony pozostawia wiele do życzenia - był to mój pierwszy projekt ze stylowaniem większego projektu, stąd

też proszę o wyrozumiałość 😊

Dodatkowa dokumentacja

Dokumentacja API backendu

wydarzen-io/api-documentation.pdf at main · wojtekweg/wydarzen-io

Simple website for managing personal events created during Design Patterns lecture at Jagiellonian University 2021/22 created with Django + React -

wydarzen-io/api-documentation.pdf at main · wojte...

<https://github.com/wojtekweg/wydarzen-io/blob/main/docs/api-documentation.pdf>

wojtekweg/
wydarzen-io



Simple website for managing personal events created during Design Patterns lecture at Jagiellonian University 2021/22 created with Django

React
5 Contributors 12 Issues 0 Stars 0 Forks

Diagram relacji tabel bazy danych

