

Jak żyć z dokumentacją?

Paweł Wiszniewski



Agenda

1. Jest jak jest;
2. Nie musi być tak źle;
3. Jak to dokumentacja bez Worda? - Docs as code.

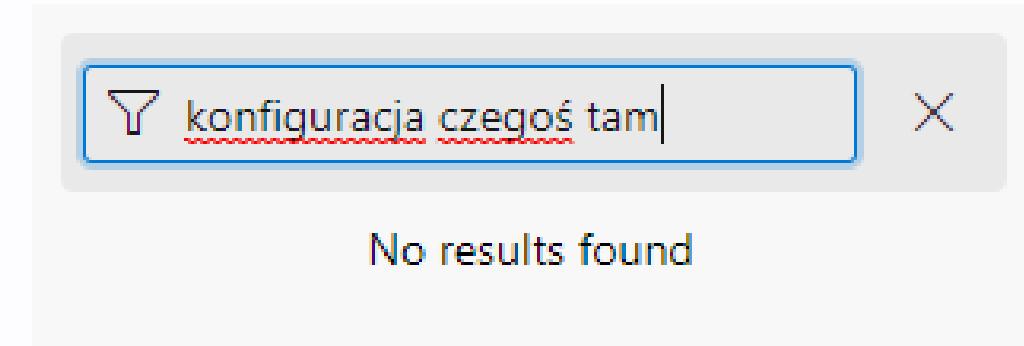


Jest jak jest

Czyli codzienne problemy

Kiedy ostatnio dokumentacja zawiodła?

Oczywiście w kryzysowej sytuacji,
gdy nikt nie wie jak coś powinno
działać...



Jak wy tłumaczyć brak dokumentacji?

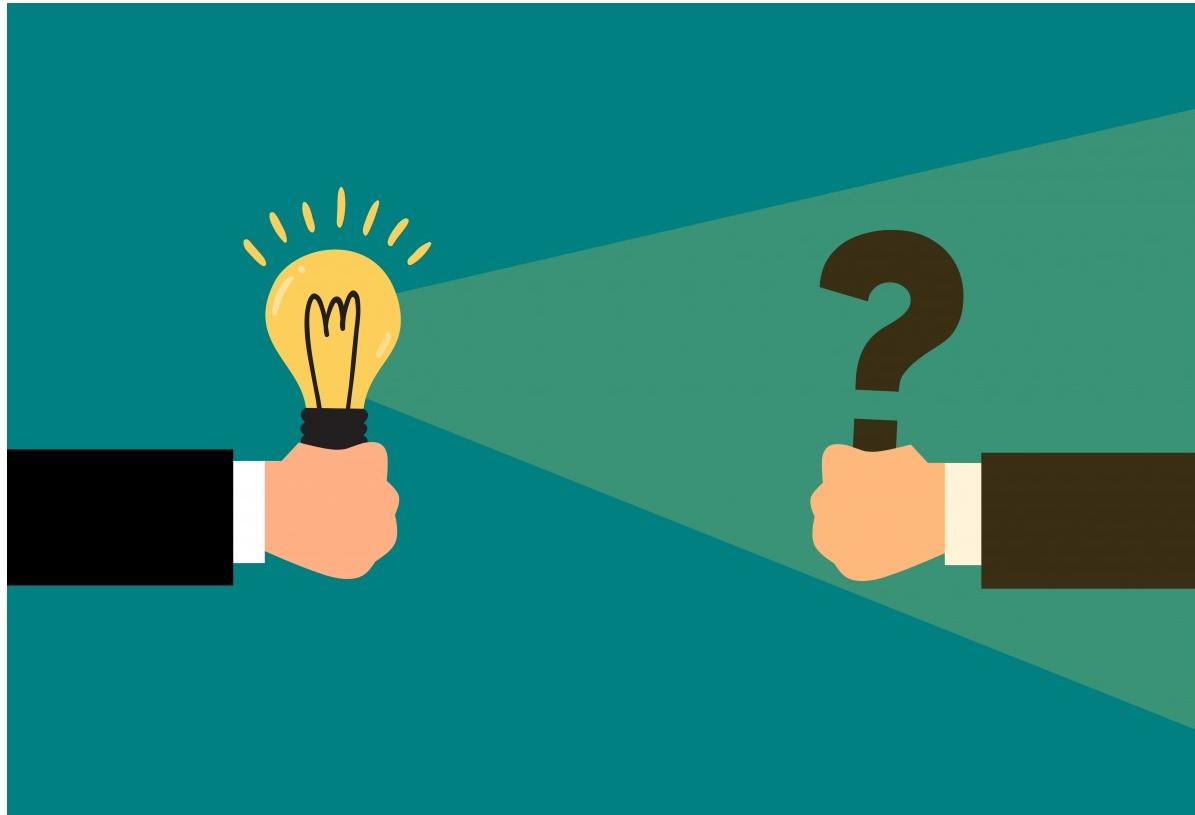
- brak czasu;
- niechęć do pisania dokumentacji;
- kod / sposób działania wydaje się oczywisty;
- "working software over comprehensive documentation";
- brak techwriterów w projekcie.

Co to jest wiedza plemienna?

- dziwne, niestandardowe, tzw. zdrutowane rozwiązania, które powstały przez brak czasu na realizację w prawidłowy sposób;
- niekompletna wśród wszystkich w zespole oraz nieudokumentowana;
- ktoś kto dołącza do projektu nie zdaje sobie z tego sprawy;
- odejście guru pewnego tematu z projektu, może wiązać się z jej utratą.



Nie musi być tak źle



A może jednak dokumentacja tworzy się sama? - dokumentacja niejawna

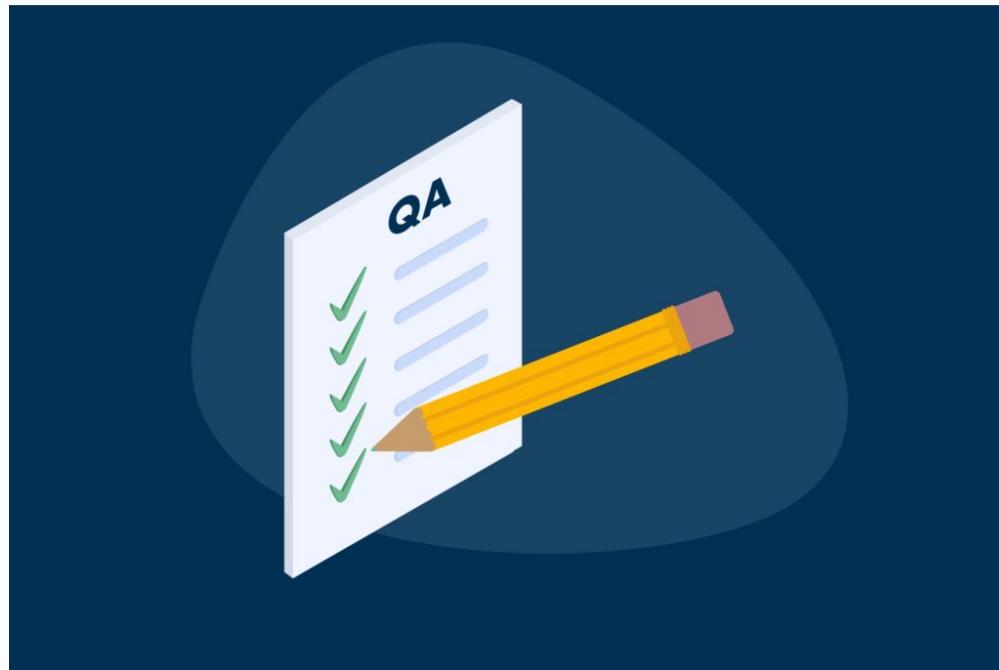
- powstaje podczas analizy, rozpisywana, rozdzielania zadań pomiędzy developerami oraz podczas komunikacji z klientem;
- skupia się na fragmencie projektu a nie na całym jego zakresie;
- nie koniecznie jest rozbudowanym opisem, zawierającym np. instrukcje używania funkcjonalności.

Wady dokumentacji niejawnej

- nie jest zebrana w jednym miejscu;
- możliwość przeoczenia wad rozwiązania, gdy całokształt nie jest widoczny;
- informacje mogą być niekompletne bądź nieaktualne w niektórych miejscach.

Co jeszcze okazuje się pomocne?

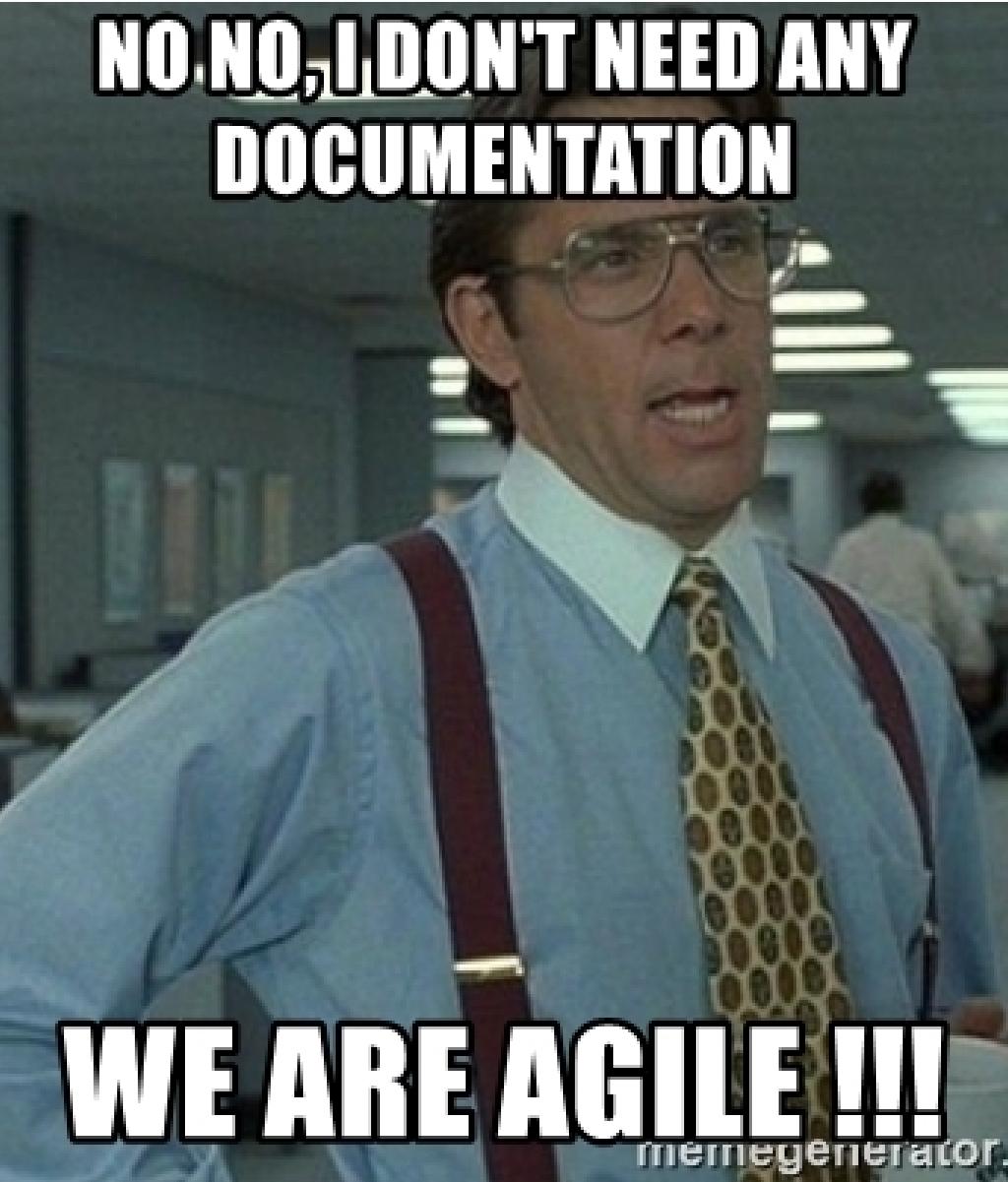
W projektach znajdują się przyjaciele dokumentacji - **testy**.



A testy nie powinny być pisane na podstawie dokumentacji?

Testy bez dokumentacji

- w rzeczywistości często testy tworzone są na podstawie komunikacji z developerami lub po prostu tworzą je sami;
- testy są również dokumentacją niejawną dla programistów:
 - testy jednostkowe - założenia działania poszczególnych elementów;
 - testy funkcjonalne i integracyjne - wymagania funkcjonalne oraz sposób współpracy całego systemu;
- aktualne;
- czytelne dla developerów.



Pomysły na dokumentację w Scrumie

- przygotowanie zadań przed sprint planingiem;
- wydzielony czas w każdym sprincie;
- proste opisy zadań;
- komunikacja.

Pozostałe drobne usprawnienia

- ustalenie co jest nam potrzebne;
- zwrócenie uwagi na to, że dokumentacja służy do komunikacji ze sobą z przyszłości/przeszłości;
- dokumentowanie rzeczy z pozoru oczywiste;
- dokumentowanie wiedzy plemiennej;
- dokumentowanie błędów i ich rozwiązania;
- decyzja czy dokumentację tworzymy w bloku czy sukcesywnie podczas realizacji funkcjonalności.

Co by było, gdyby stworzyć dokumentację przed realizowaniem projektu?

- nie zawsze możliwe;
- łatwiejsze wyodrębnienie zadań;
- obraz całego rozwiązania;
- zmniejszenie ryzyka wystąpienia problemów;
- ułatwienie w ocenieniu zmian w stosunku do pierwotnej wersji.

This is a **sample ASP.NET** Core application *that* demonstrates basic features of ASP.NET Core in an API backend.

allows
intera

! [] (A)

To a M

! [] (A)

****Online Sample:****

* [Live AlbumViewer]

Sample1(<https://samples.west-wind.com/AlbumViewerCore/>)

ASP.NET Core Features

The **ASP.NET Core** Backend demonstrates

- * Creating an API backend service
 - * Isolating business logic and Controller code
 - * Using a base Repository for business logic
 - * Separating code into separate projects
 - * Custom User Authentication

Ready

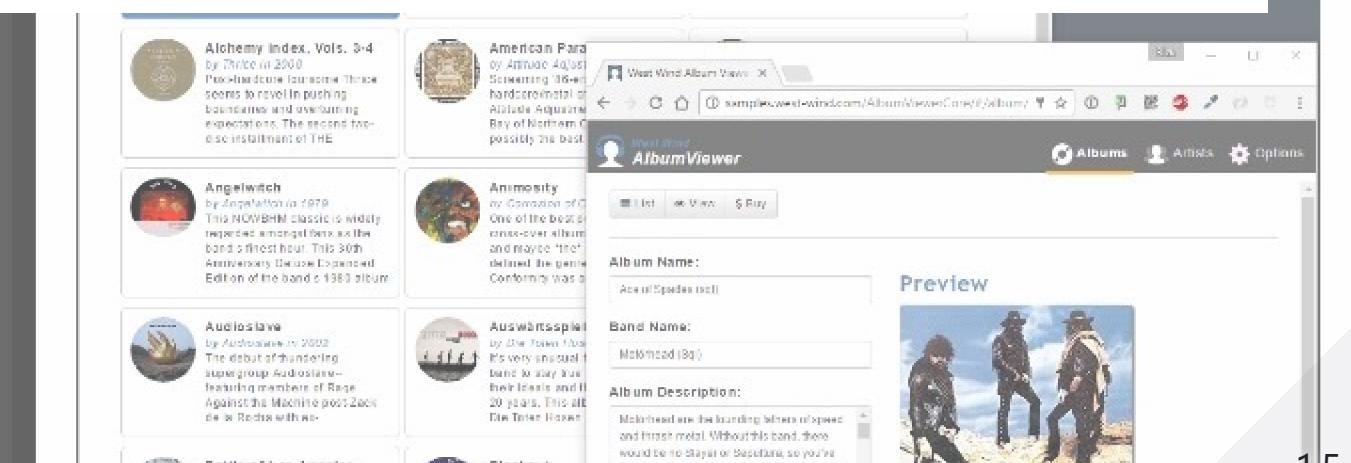
West Wind Album Viewer ASP.NET Core Sample

Sample Angular 2.0 application demonstrating ASP.NET Core API features

This is a **sample ASP.NET** Core application that demonstrates basic features of ASP.NET Core in an API backend application with an Angular 2.0 frontend. The application allows browsing and editing of a list of albums and artists interactively.

Jak to dokumentacja bez Worda? -

Docs as code



841 words, 144 lines

twilight

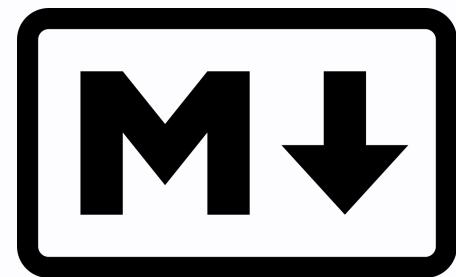
Dharkan

Jaka jest idea "Docs as code"?

- upodobnienie procesu tworzenia dokumentacji do procesu tworzenia kodu aplikacji;
- przyspieszenie tworzenia dokumentacji;
- usprawnienie wersjonowania, zastosowanie systemu kontroli wersji.

Jakie są możliwości?

- narzędzia do tworzenia plików tekstowych, diagramów oraz bardziej zaawansowanych, interaktywnych rozwiązań np. z wykorzystaniem Markdown;
- tworzenie dokumentacji w IDE, np. w Visual Studio Code;
- wykorzystanie rozwiązań z programowania typu if, for, kolekcji.



Visual Studio Code

Co nam to faktycznie daje?

- zwiększenie łatwości i szybkości tworzenia/edykcji dokumentacji w trakcie wykonywania zadań;
- możliwość przechowywania dokumentacji w repo:
 - sprawdzanie dokumentacji w trakcie code review,
 - automatyczna aktualizacja dokumentacji w trakcie wdrożeń;
- dokumentacja żyje razem z kodem i jest poważniej traktowana.



Takie wspaniałe rozwiązanie posiada wady?

- techwriterzy mogą mieć trudności;
- tak jak podczas tworzenia aplikacji, mogą pojawić się różnego rodzaju błędy np. komplikacji;
- początkowa konfiguracja rozwiązania;
- czasem wymagana dodatkowa konfiguracja przy lokalnym uruchamianiu.

```
cking for ffi_raw_call()... no
cking for shlwapi.h... no
cking for ruby/thread.h... yes
cking for rb_thread_blocking_region()...
cking for rb_thread_call_with_gvl()...
cking for rb_thread_call_without_gvl()...
ating extconf.h
ating Makefile

rent directory: /Users/admin/Desktop/Hel
e "DESTDIR=" clean

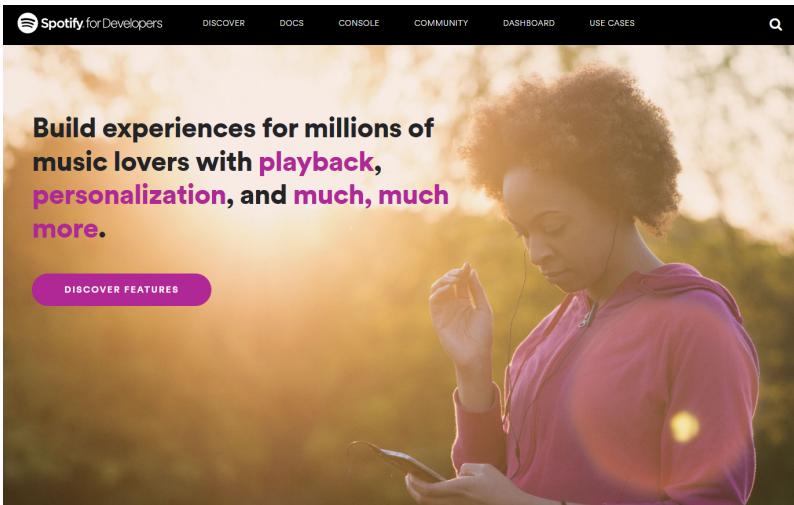
rent directory: /Users/admin/Desktop/Hel
e "DESTDIR="
n/sh: -c: line 0: unexpected EOF while
n/sh: -c: line 1: syntax error: unexpect
e: *** ["/Users/admin/Desktop>Hello] Er
e failed, exit code 2

files will remain installed in /Users/
ception.
ults logged to /Users/admin/Desktop/Hel
ld/Ruby/app/vendor/bundle/extensions/x86

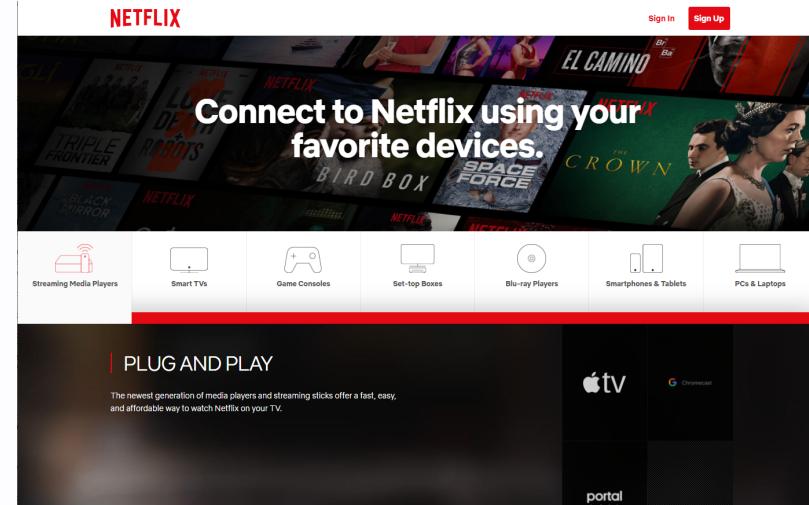
error occurred while installing ffi (1.0.0)
  be sure that `gem install ffi -v '1.9.25
  --no-rdoc --no-ri -- --with-ffi-cflags=-fPIC` succeeds.

Gemfile:
  selenium-webdriver was resolved to 3.141.1
    childprocess was resolved to 0.9.0, wh
      ffi
        run bundle exec spring binstub -
old not find gem 'rails (> 5.2.1)' in a
  -825 'bundle install' to install missing ge
```

Czy ktoś tego używa?



Spotify for developers



Netflix Devices

Jekyll

- zaprojektowany do tworzenia blogów, ale nadaje się świetnie do tworzenia interaktywnej dokumentacji
- domyślnie wspierany przez Github Pages;
- dosyć prosty w konfiguracji i uruchamianiu



Konfiguracja

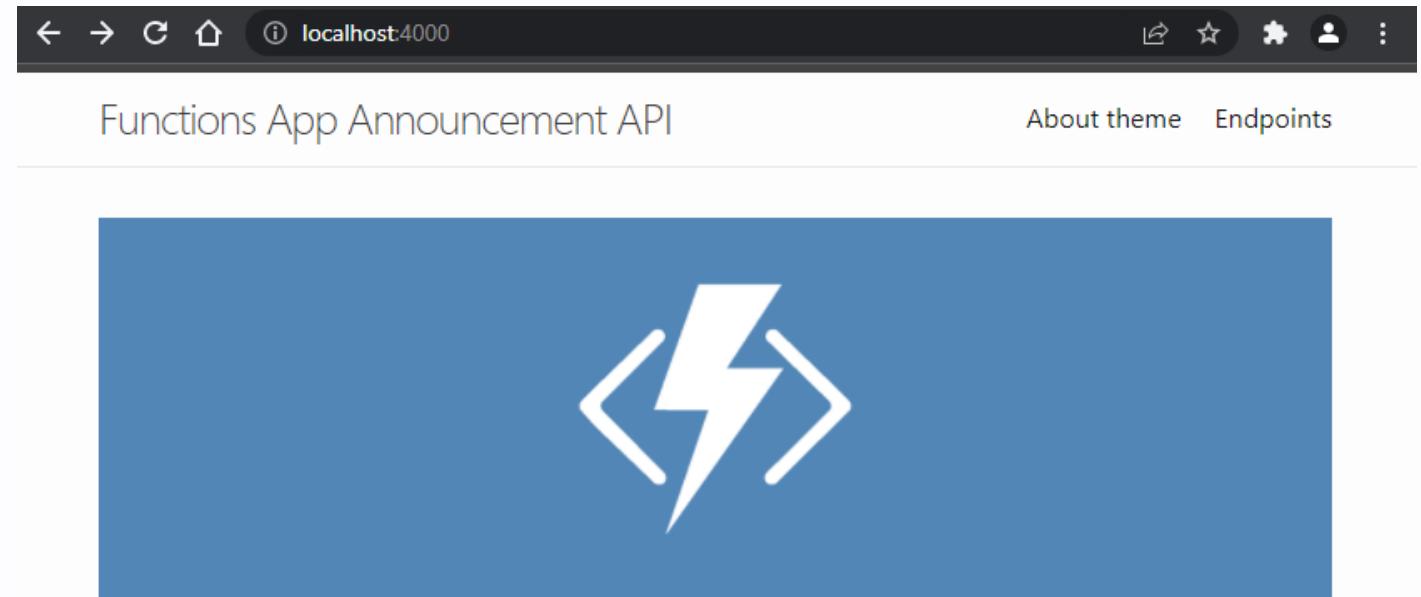
Github

The screenshot shows the GitHub Pages configuration interface. At the top, it says "GitHub Pages" and explains its purpose: "GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository." A green box at the top indicates that the site is published at <https://pawjo.github.io/azure-functions-announcement-api/>. Below this, the "Source" section shows the site is built from the `/docs` folder in the `docs-tests` branch. There are dropdown menus for "Branch" set to `docs-tests` and "Folder" set to `/docs`, along with a "Save" button. The "Theme Chooser" section allows selecting a Jekyll theme, currently using "Minima". A "Change theme" button is available. The "Custom domain" section lets users serve their site from a different domain, with a "Save" and "Remove" button.

Lokalnie

Instalacja Ruby z devkitem, jekyll, bundler, następnie instalacja pakietów z bundle oraz uruchomienie lokalnego serwera.

Jekyll demo



Introduction

This is an example of Azure Function App used as API. It is also sample documentation that was created with "docs as code" approach.

Documentation

- [About theme](#)
- [Endpoints](#)

Demo 2 - prezentacja

File Edit Selection View Go Run Terminal Help jak_zyc_z_dokumentacją.md - Prezentacja o dokumentacji - Visual Studio Code

jak_zyc_z_dokumentacją.md x
jak_zyc_z_dokumentacją.md > ...
! [bg: opacity:0.8](![urazy/docs_as_code_example.jpg)
173
174
175
176 <!-- footer: Jak to dokumentacja bez Worda? - Docs as code -->
177
178 ## Jaka jest idea "Docs as code"?
179
180 - upodobnienie procesu tworzenia dokumentacji do procesu tworzenia kodu aplikacji;
181 - przyspieszenie tworzenia dokumentacji;
182 - usprawnienie wersjonowania, zastosowanie systemu kontroli wersji.
183
184 ---
185 ## Jakie są możliwości?
186
187 - narzędzia do tworzenia plików tekstowych, diagramów oraz bardziej zaawansowanych, interaktywnych rozwiązań np. z wykorzystaniem Markdown;
188 - tworzenie dokumentacji w IDE, np. w Visual Studio Code;
189 - wykorzystanie rozwiązań z programowania typu if, for, kolekcji.
190
191 |||
192 |--|
193 |!(w:28)(Obrazy/markdown.png)||![w:308](Obrazy/vsc.webp)|
194
195
196 ---
197
198 ! [bg right:20% w:250](Obrazy/super_happy.jpg)
199
200 ## Co nam to faktycznie daje?
201
202 - zwiększenie łatwości i szybkości tworzenia/edykcji dokumentacji w trakcie wykonywania zadań;
203 - możliwość przechowywania dokumentacji w repo:
204 * sprawdzanie dokumentacji w trakcie code review;
205 * automatyczna aktualizacja dokumentacji w trakcie wdrożeń;
206 - dokumentacja żyje razem z kodem i jest poważniej traktowana.
207
208 ---
209
210 ! [bg right:25%](Obrazy/ruby_error2.png)
211
212 ## Takie wspaniałe rozwiązanie posiada wady?
213
214 - techwriterzy mogą mieć trudności;
215 - tak jak podczas tworzenia aplikacji, mogą pojawić się różnego rodzaju błędy np. komplikacji;
216 - początkowa konfiguracja rozwiązania;
217 - czasem wymagana dodatkowa konfiguracja przy lokalnym uruchamianiu.
218
219

Prezentacja jak_zyc_z_dokumentacją.md Backend demonstrates:
Creating an API backend service
Isolating business logic and Controller code
Using a base Repository for business logic
Separating concerns in separate projects
Custom User Authentication
Ready

B41 words, 144 lines Twilight Darkhan

Jaka jest idea "Docs as code"?

- upodobnienie procesu tworzenia dokumentacji do procesu tworzenia kodu aplikacji;
- przyspieszenie tworzenia dokumentacji;
- usprawnienie wersjonowania, zastosowanie systemu kontroli wersji.

Jak to dokumentacja bez Worda? - Docs as code

Jakie są możliwości?

- narzędzia do tworzenia plików tekstowych, diagramów oraz bardziej zaawansowanych, interaktywnych rozwiązań np. z wykorzystaniem Markdown;
- tworzenie dokumentacji w IDE, np. w Visual Studio Code;
- wykorzystanie rozwiązań z programowania typu if, for, kolekcji.

Prezentacja została wykonana przy pomocy frameworka Marpit.

Dziękuję za uwagę!

Materiały:

- <https://testerzy.pl/baza-wiedzy/artykuly/jak-poprawnie-tworzyc-dokumentacje>
- <https://wedlugplanu.pl/001/>
- <https://www.writethedocs.org/>
- <https://bulldogjob.pl/articles/779-pisz-dokumentacje-tak-jak-kod>
- <https://docs-as-co.de/>
- <https://jekyllrb.com>
- <https://docs.github.com/en/pages/setting-up-a-github-pages-site-with-jekyll>