Ostatnia aktualizacja: 8 paź 2021

Etap 3 Tydzien 1 Dzien 1 - Express wprowadzenie

E3T1D1.pdf
45.8 MB

Express.js Framework albo platforma programistyczna – szkielet do budowy aplikacji. Definiuje on strukturę aplikacji oraz ogólny mechanizm jej działania, a także dostarcza zestaw komponentów i bibliotek ogólnego przeznaczenia do wykonywania określonych zadań. Programista tworzy aplikację, rozbudowując i dostosowując poszczególne komponenty do wymagań realizowanego projektu, tworząc w ten sposób gotową aplikację.
Frameworki bywają niekiedy błędnie zaliczane do bibliotek programistycznych.

Cechy, które wyróżniają je jako samodzielną kategorię oprogramowania, to:

- **odwrócenie sterowania –** w odróżnieniu od aplikacji oraz bibliotek, przepływ sterowania jest narzucany przez framework, a nie przez użytkownika.
- **domyślne zachowanie –** domyślna konfiguracja frameworka musi być użyteczna i dawać sensowny wynik, zamiast być zbiorem pustych operacji do nadpisania przez programistę.
- **rozszerzalność –** poszczególne komponenty frameworka powinny być rozszerzalne przez programistę, jeśli ten chce rozbudować je o niezbędne mu dodatkowe funkcje.
- **zamknięta struktura wewnętrzna –** programista może rozbudowywać framework, ale nie poprzez modyfikację domyślnego kodu.

Źródło https://pl.wikipedia.org/wiki/Framework

Express.js , lub po prostu **Express** , jest back- endowym frameworkiem aplikacji internetowych dla Node.js , wydanym jako darmowe oprogramowanie o otwartym kodzie źródłowym na licencji MIT . Przeznaczony jest do budowania aplikacji internetowych i API . [3] Został nazwany de facto standardowym frameworkiem serwerowym dla Node.js . [4]

Wykorzystanie frameworków przyśpiesza pisanie aplikacji. W praktyce do dużych aplikacji wykorzystuje się najczęściej właśnie frameworki, a nie czysty HTTP Server w Node.js.

Powyższy rysunek przedstawia warstwy tego co się dzieje, kiedy mówimy o backendzie. Express jest ściśle związany z backendem.

Aktualną wersją (już od kilku lat) Express.js jest wersja 4.17.1, natomiast dostępna jest nowsza odsłona frameworka 5.0, która już od dłuższego czasu jest w fazie alpha (czyli bardzo niestabilna – początek drogi).

Zapisz kopię w Evernote

- **framework** serwera webowego dla Node.js.
- **Fast** – Express.js jest szybki
- prawie tak szybki jak sam HTTP Server Node.js (serwer Node'owy jest niskopoziomowy, a serwer postawiony za pomocą Express'a jest warstwę wyżej – taka nadbudówka), wykonanie kodu jest bardzo optymalne, szybkie.
- **Unopinionated** – Express.js nie mówi jak mamy żyć (w tym momencie kłóci się z definicją frameworka), nie jest to wada ani zaleta, jest to ważna cecha. Jest elastyczny.
- **Minimalist** – minimalistyczny, zawiera tylko najważniejsze rzeczy. W praktyce można go rozszerzać o dodatkowe moduły za pomocą paczek, np. npm, ale nie trzeba.

Express.js pozwala nam na wykonanie wielu rzeczy, które znajdziemy w Node.js, ale pozwala to robić szybciej, tzn łatwiej, przyjemniej, krótszy kod. To dzięki dostarczeniu narzędzi pomagających nam pisać aplikacje webowe i mobilne – od strony backendu.

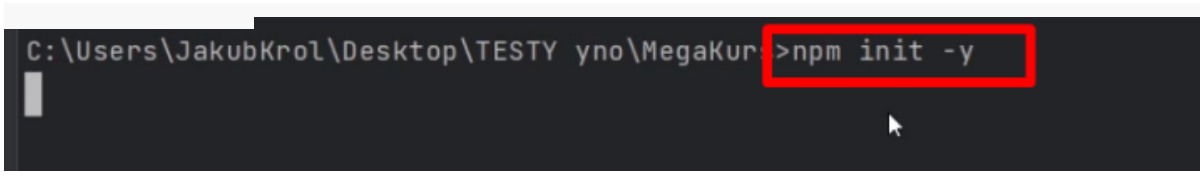
Jest też często wykorzystywany jako podstawa dla innych bardziej rozbudowanych frameworków dla Node.

API's – pisanie ścieżek, obsługa poszczególnych metod http itd jest dużo łatwiejsza i szybsza z użyciem Express.js **Performance** - Express korzysta ze znanych z Node.js koncepcji – m.in. obiektów req i res. Dodane są tylko pewne możliwości. To przekłada się również na szybkie działanie Express.js

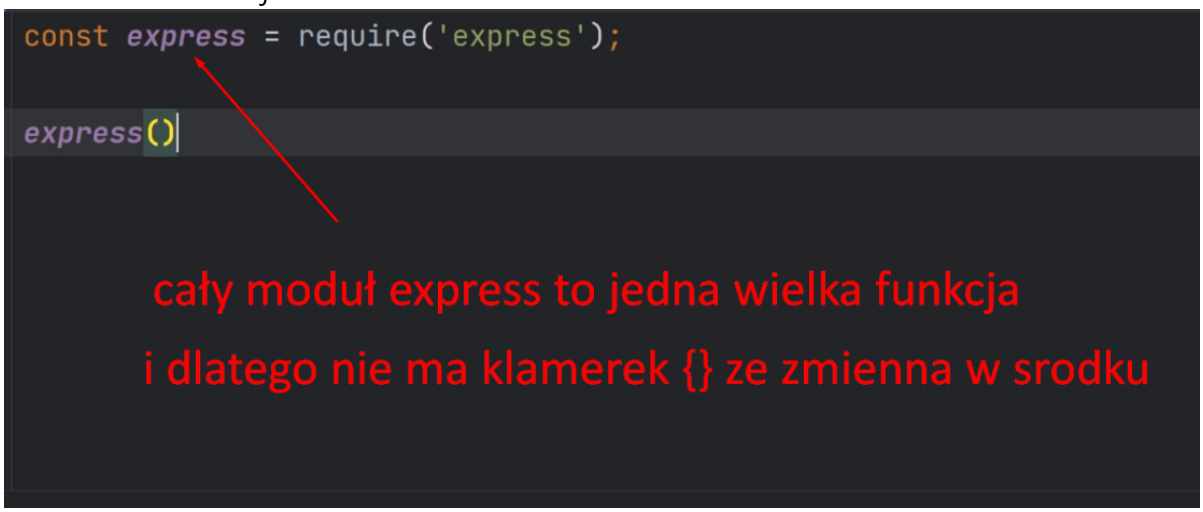
Express.js jest bardzo często wykorzystywany jako baza dla innych, bardziej rozbudowanych frameworków webowych dla Node. Inne przykłady frameworków :

Zapisz kopię w Evernote

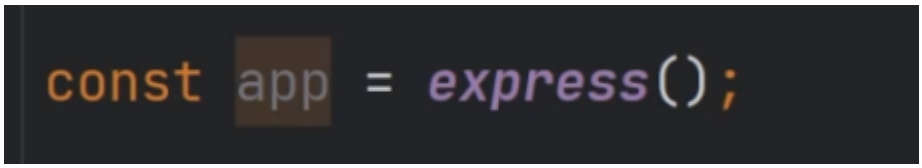
1. Tworzymy nowy projekt.
2. Tworzenie pliku **package.json - npm init -y**



3. Instalacja Express.js - npm i express
4. Zaimportowanie Express – **require('express')** – cały ten moduł jest jedną wielką funkcją dlatego nie ma klamerek tutaj.



5. Stworzenie serwera.



6. Uruchamianie serwera - app.listen (numer_portu); - zazwyczaj robi się to na samym końcu – w tym momencie nasz program oczekuje, „zawiesza się"