

Laboratorium 2

działanie warstw pośrednich w Express.js

Pomoce online:

<https://expressjs.com/en/guide/using-middleware.html> – warstwy pośrednie w Express.js

<https://expressjs.com/en/guide/writing-middleware.html> – pisanie własnej warstwy pośredniej w Express.js

<http://www.murvinlai.com/req-and-res-in-nodejs.html> – obiekty req i res w Express.js

Proszę sobie przejrzeć i uruchomić trzy proste programiki: [app1.js](#), [app2.js](#) i [app3.js](#).
Widok działania aplikacji app2 poniżej:



KONIEC

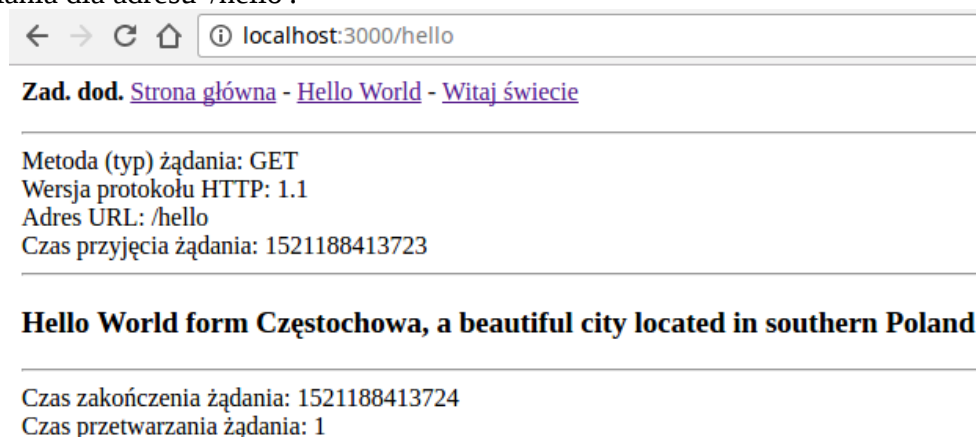
Te trzy programiki są podobne ale trochę się różnią. Prezentują możliwości wykorzystania warstw pośrednich w Express.js. Po przeanalizowaniu kodu i jego zrozumieniu proszę następnie napisać aplikację, która wykorzysta zdobytą wiedzę. Aplikacja powinna obsługiwać trzy adresy URL:

- '/' - strona główna
- '/hello' – jakiś tekst po angielsku
- '/witaj' – jakiś tekst po polsku

Treść odpowiedzi dla każdego adresu powinna być generowana przez trzy albo cztery funkcje z warstw pośrednich.

- 1) Funkcja zapisująca w odpowiedzi do klienta widok menu głównego na górze strony, powinna działać dla wszystkich adresów URL.
- 2) Dla wszystkich adresów poza stroną główną powinny się wyświetlać informacje o żądaniu klienta: typ żądania, adres URL, wersja protokołu HTTP i powinien być zapamiętany czas otrzymania żądania.
- 3) Każdy adres powinien mieć własną unikalną treść, tak zwyczajnie jak to jest w Internecie :)
- 4) Wszystkie adresy powinny wyświetlać na koniec czas zakończenia żądania, czas jego przetwarzania i może coś na koniec np. słowo KONIEC :)

Przykład działania dla adresu '/hello':



KONIEC

Mamy więc trzy (dla strony głównej) lub cztery (dla pozostałych adresów URL) funkcje odpowiedzialne za generowanie odpowiedzi dla klienta (menu, info o żądaniu, treść właściwa strony, info o żądaniu na koniec).

Uwagi:

- `res.send()` wysyła nagłówek odpowiedzi razem z jego treścią, nic już po `send()` nie dopiszemy,
- `res.write()` dopisuje coś do ciała odpowiedzi dla klienta, można tak dopisywać wiele razy,
- `res.end()` zakańcza wysyłanie odpowiedzi do klienta, po `end()` nic już do niej nie dopiszemy,
- `res.header()` wysyła nagłówek odpowiedzi, można wysłać tylko raz, nie można wysłać nagłówka jeśli wcześniej wysłane było już ciało np. metodą `res.write()`,
- `next()` jest konieczne w każdej funkcji, która nie zakańcza przetwarzania żądania czyli w tych, które nie mają np. `res.send()` czy `res.end()`.