

Sprawozdanie 2 z Web 2.0

Angular 7.0

W drugim sprawozdaniu trzeba wykazać się podstawową znajomością frameworka Angular.

Sprawozdania przysyłamy na adres: rperlinski.pcz@gmail.com

Termin: do końca semestru, jak sprawdzę do ostatnich laboratoriów to wpis na ostatnich laboratoriach.

Załączamy:

- sprawozdanie w pliku PDF,
- spakowany katalog z projektem, tak jak zawsze: spakowany w paczkę, github, GoogleDrive, ...

Tytuł: Web 2.0 - Nazwisko - Spr 2

Załączniki: Nazwisko.pdf, Nazwisko.zip

Aplikacja nie musi wyjątkowo dobrze wyglądać (nie oceniam wyglądu), ale wszystko powinno działać. **Spakowany kod oddajemy bez katalogu node_modules!** W Angular 7 to około 250MB. Cała aplikacja ma być gotowa do działania po wykonaniu polecenia: **npm install**.

Chyba najprościej zrobić sobie porządnie laboratorium 10 z własną tablicą obiektów innych niż studenci. Zaczynamy od liczby punktów 1.5. Więcej zrobione, większa ocena. Zadanie 0 (sprawozdanie w pliku PDF) i zdania 1 i 2 to minimum na zaliczenie tego sprawozdania – na ocenę 3.0 (jeśli to minimum będzie porządnie zrobione).

Zawartość sprawozdania (2 punkty obowiązkowe, inne według uznania):

1. Opis aplikacji

Jeden, dwa akapity opisu aplikacji – czego dotyczy projekt i na jaką ocenę. Dodać zrzut ekranu całej aplikacji (widoku głównego komponentu).

2. Elementy konstrukcyjne aplikacji

Opis, w kolejnych podpunktach, najważniejszych elementów konstrukcyjnych aplikacji potrzebnych na daną ocenę. Jakie są: dane w tablicach, komponenty, wykorzystane dyrektywy, usługi, rodzaje wykorzystywanego wiązania danych (po jednym przykładzie każdego rodzaju).

Nie musi być to wyczerpujący opis z wchodzeniem w szczegóły ale przydałoby się najważniejsze elementy przedstawić, żeby wiedział ile jest zrobione bez potrzeby pieczołowitego szukania tego w kodzie.

Każdy student ma na początku ocenę (liczbę punktów) z tego sprawozdania 1.5. Za zaimplementowanie w aplikacji i wykazanie w sprawozdaniu każdego z poniższych elementów są dodatkowe punkty do oceny ze sprawozdania.

| Zad. | Punkty | Funkcjonalność w aplikacji |
|------|--------|--|
| 0 | ± 0.5 | Opis sprawozdania – plik PDF Zależnie od jakości można dostać + 0.5 do oceny, albo mniej, a nawet jakiś minus, jeśli ktoś to zrobi bardzo źle... |

| | | |
|---|-------|---|
| 1 | + 0.4 | <p>Spersonalizowane „Hello World”</p> <p>Jeden moduł zawierający jeden komponent z wyświetlaniem zmienionej treści niż domyślne „Hello World” czyli jakiś swój obrazek, nagłówek, kilka zdań treści, może jakieś menu...</p> |
| 2 | + 0.6 | <p>Jeden komponent z danymi i ich edycją</p> <p>Utworzenie w pliku tablicy z obiektami (przynajmniej 5 elementów), które będziemy prezentować w komponencie, np. książki, samochody, kraje, zwierzęta w zoo, rzeki w Polsce, mikroprocesory... Prezentacja w głównym komponencie tej listy z możliwością klikania i edycji danych wybranego obiektu.</p> |
| 3 | + 0.4 | <p>Dwa komponenty zależne, nadrzędny i podrzędny</p> <p>Drugi komponent podrzędny do tego głównego, który wyświetla szczegóły wybranego obiektu i pozwala na edycje danych wybranego obiektu.</p> |
| 4 | + 0.6 | <p>Własna usługa, wykorzystana zamiast prostego wczytania danych z tablicy</p> <p>Tworzymy własną usługę, którą wykorzystujemy w komponencie głównym zamiast danych obiektów umieszczonych w tablicy. Usługa powinna działać poprzez wstrzykiwanie zależności. Trzeba też użyć ngOnInit do wczytania danych z usługi.</p> |
| 5 | + 0.4 | <p>Własne metody w komponentach</p> <p>Trzeba stworzyć i użyć przynajmniej dwóch metod w którymś ze swoich komponentów, które będą miały chociaż kilka linijek kodu, np. średnia ocen.</p> |
| 6 | + 0.5 | <p>Dwa dodatkowe komponenty</p> <p>Utworzenie dwóch dodatkowych komponentów wzajemnie powiązanych i robiących coś sensownego, np. Prezentacja danych tablicowych umieszczonych wewnątrz każdego z naszych obiektów, np. ocen z różnych przedmiotów studentów.</p> |
| 7 | + 0.5 | <p>Wykorzystanie trasowania</p> <p>Udostępnienie różnych adresów URL w aplikacji, które są związane z różnymi komponentami. Powinny być przynajmniej 3 adresy URL dla 3 różnych komponentów. Zrobienie komponentu nadrzędnego i podrzędnego, przekazujących sobie dane poprzez parametr w adresie URL to co innego niż zawartość zadania 3.</p> |
| 8 | + 0.5 | <p>Wykorzystanie usługi HTTP</p> <p>Trzeba wykorzystać usługę HTTP w Angular z jakimś działającym API. Powinny być przynajmniej 3 operacje z podstawowego CRUD.</p> |
| 9 | + 0.4 | <p>Aplikacja działająca online</p> <p>Zrobienie aplikacji działającej online podnosi trochę ocenę. Trzeba mieć dostęp do jakiegoś własnego (darmowego lub nie) serwera plików.</p> |