ZPW lab

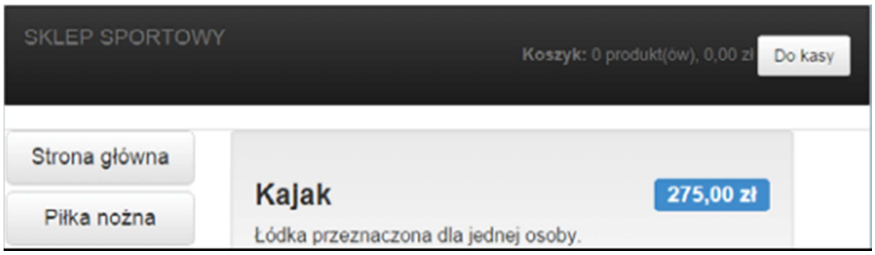
laboratorium 6

Angular kontynuacja

Synchronizacja danych, obsługa strumieni, routing

Nasza sklep internetowy składa się z kilku komponentów komunikujących się ze sobą albo bezpośrednio albo poprzez usługę. Nadszedł czas na rozszerzenie naszej aplikacji.

Zad 1. Niech na stronie głównej pojawi się nowy komponent prezentujący aktualna wartość zamówionych produktów oraz ich ilość. Po dodaniu (przez stworzony na ostatnich zajęciach komponent) produktu do koszyka na stronie głównej powinien się wyświetlić aktualny stan koszyka. Zaimplementuj potrzebne elementy kodu (komponenty, usługę?) która pozwoli na realizacje powyższego.



W celu poeksperymentowania z synchronizacją danych - w aplikacji użyj dwie instancje tego samego komponentu. Przetestuj czy wyświetlają te same dane.

Zad. 2 Nasza aplikacja będzie miała kilka podstron. Jedna z nich jest strona do rejestracji konta przez użytkownika. Druga główny ekran z lista produktów, kolejna to widok koszyka a ostatnią to ekran potwierdzenia zamówienia. Wykorzystując koncepcje routing zrealizuj odpowiednia funkcjonalność. Podstrony nie musza już w tej chwili mieć pełnej funkcjonalności, może to być tylko prosty mock identyfikujący na której podstronie aktualnie jesteś.

Zad3. Czas na kolejny refaktoring usługi dostarczającej dane. W tym celu użyj modułu

In-memory Web API .

Opis użycia – moje wykłady (wykład 4) lub w tutorialu obecnym na stronie Angular.io.

Uzycie tego modułu powoli nam przejść do odczytu danych w formie strumienia.

Odczyt danych proszę zrealizować przy użyciu httpClient .

Zad 4. Praca ze strumieniem danych –observable (użycie biblioteki RxJS) . Informacje o pracy z ta bibiloteka można znaleźć na stronie

https://www.learnrxjs.io/operators/combination/merge.html lub http://reactivex.io/

Proszę na stronie w wydzielonym komponencie wyświetlić następujące informacje: średnia cena produktu z bazy, ilość produktów, maksymalna i minimalna ilość i cena produktu. Wyświetlić tylko wszystkie produkty których cena jest większa od 100zł.

Większość elementów i technik związanych z Angularem został już poruszona na zajęciach lab, wiec przyszedł teraz czas na bardziej dokładną specyfikację projektu Sklep Internetowy.

Celem lab jest poznawania szczegółów implementacyjnych Angular poprzez realizacje przykładowej aplikacji. Proponowanym tematem jest sklep internetowy sprzedający produkty sportowe (lub jakieś inne). Budowana tutaj aplikacja o nazwie SklepSportowy będzie oparta na klasycznym podejściu stosowanym podczas tworzenia sklepów internetowych.

Aplikacja składać się będzie z dwóch głównych modułów:

-- klienckiego

-- administracyjnego

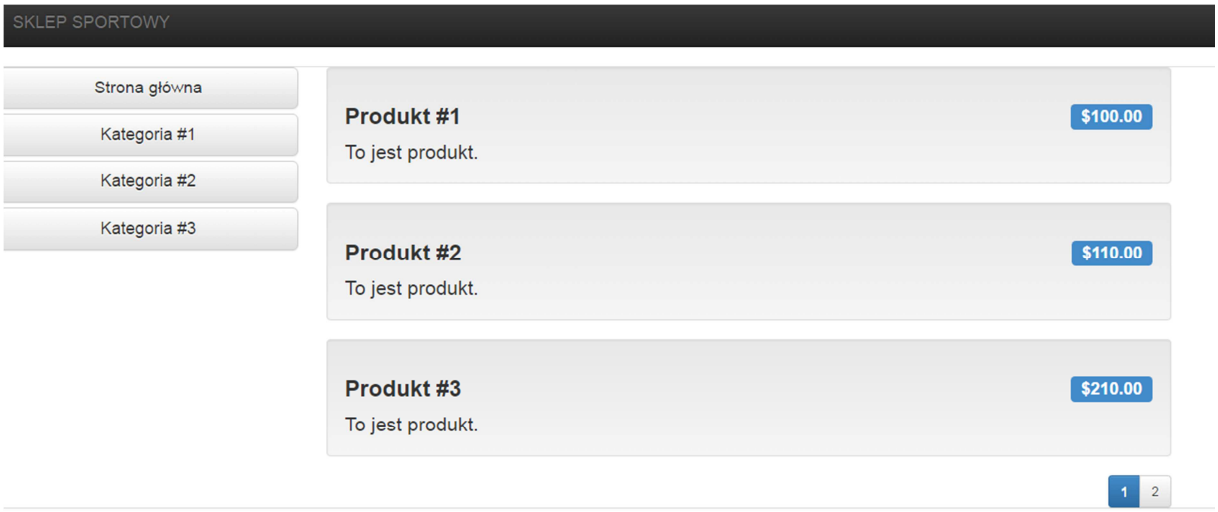
Moduł klienta – prezentuje w efektowny sposób listę dostępnych produktów. Sposób prezentacji pozostawiam Państwu. Aplikacja ma umożliwiać przeglądania produktów (z aktywnym filtrowaniem i wyszukiwaniem zawartości włącznie). Następnie dodawania produktu do koszyka, jego przeglądania oraz finalizacja zakupu.

Poniżej prezentacja typowego scenariusza z przykładowymi ekranami. Proszę się tym nie wzorować – ekrany są trywialne, od Państwa oczekuje bardziej subtelnych rozwiązań.

Szczegóły implementacji:

Podstawowy model danych obejmuje obiekty o nazwie Produkty opisane są za pomocą pól: nazwa, kategoria, ilość produktów, cena jednostkowa, opis oraz link do poglądowego zdjęcia.

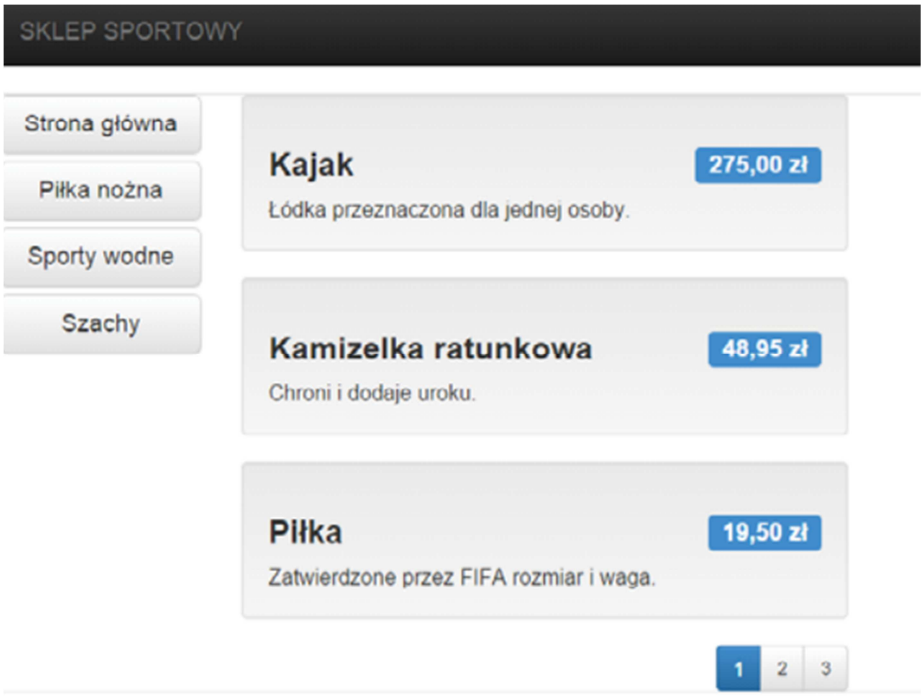
Produkty wyświetlane są w widoku głównym – max 5, 10 lub 20 produktów (wartość którą mogę jako użytkownik wybrać) na ekranie. Gdy jest ich więcej proszę zrobić paginacje. Poglądowy przykład na rysunku poniżej:



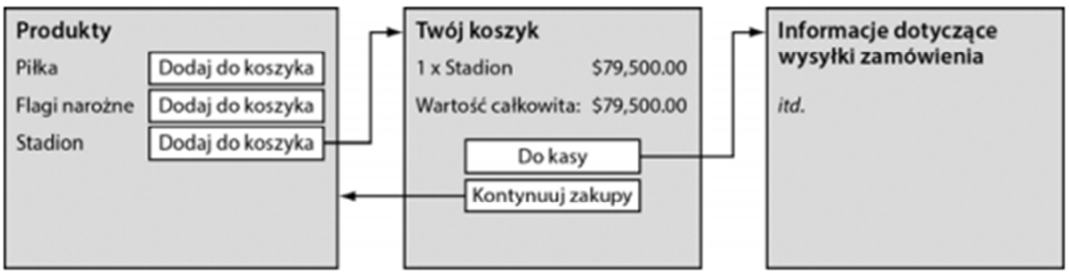
Z lewej strony panelu wyświetlane są kategorie produktów pełniące role filtru do wyświetlania po kategoriach – dostępne wszystkie wartości kategorii. Po wybraniu filtra wyświetlane są tylko produkty z danej kategorii. Wybrany kategoria wyświetlana jest w innym kolorze . Możliwy jest wybór dowolnej ilości kategorii np. 3 na raz.

Można dorobić więcej filtrów (użyj do tego własnego pipe) – np. wyszukujący produkty po fragmencie wprowadzonego tekstu lub cenie (wykorzystaj zakres cenowy – tak aby aplikacja zasugerowała w jakim zakresie są ceny produktów). Używaj podświetlania wybranej/ych kategorii tak aby łatwo ją było zidentyfikować.

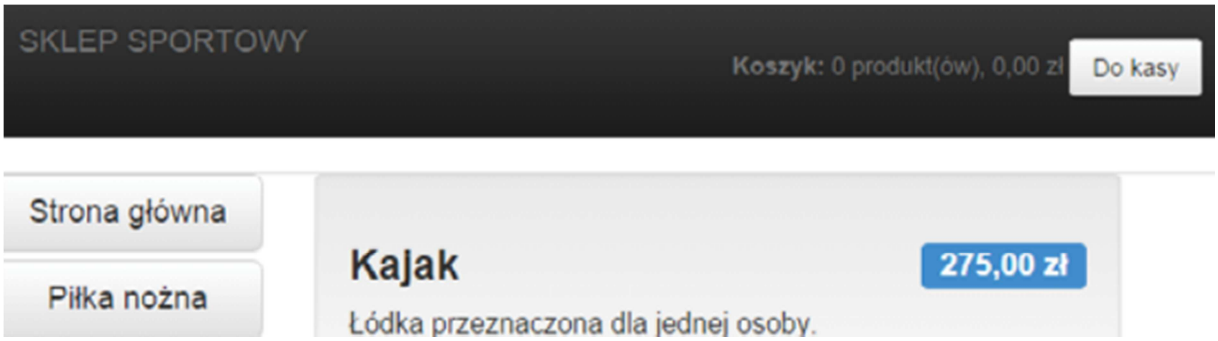
Poniżej przykład aplikacji z konkretnymi danymi:



Rozbudowujemy aplikacje o możliwość zamawiania towaru. Ogólny scenariusz działania wygląda tak jak poniżej:

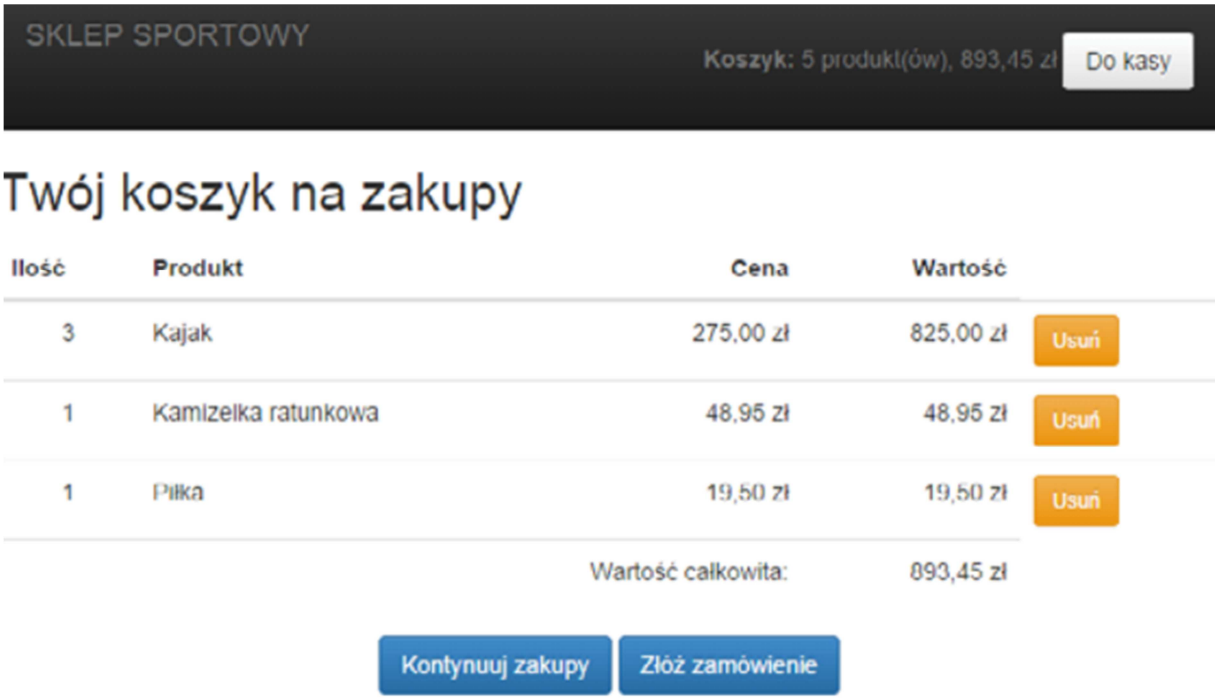


Zaimplementować koszyk zakupów jako usługę. Przy każdym produkcie jest przycisk dodaj do koszyka. Po naciśnięciu odpowiednia wartość pojawia się w koszyku. Pamiętamy o zliczaniu ilość zamówionych produktów tak aby nie można było zamówić więcej produktów niż jest w sklepie.

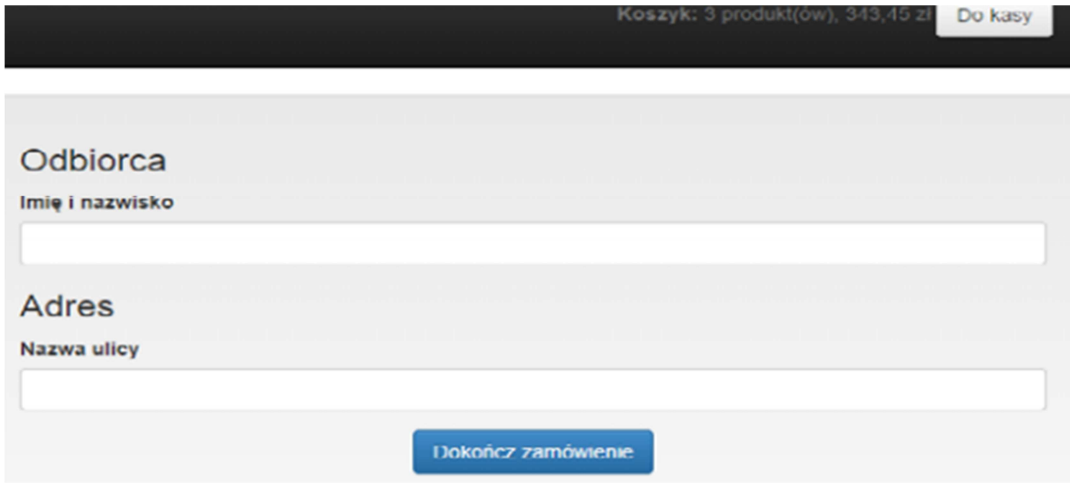


Naciśniecie zakończ zakupy powoduje przejście do widoku koszyka – gdzie jest lista wszystkich wybranych pozycji.

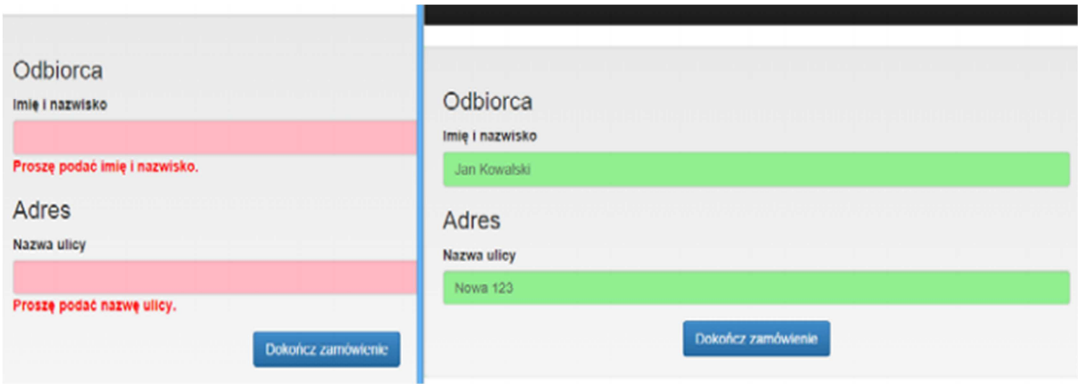
W widoku koszyka oprócz listy wybranych produktów, powinny pojawić się dwa butony: **Kontynuuj zakupy** -> powrót do widoku z produktami oraz **Złóż zamówienie**. Dodatkowo przy każdym produkcie w Koszyku znajduje się przycisk Usuń - usuwający produkt z koszyka. Podgląd koszyka np. tak jak poniżej



Wybranie z widoku Koszyka przycisku Złóż zamówienie skutkuje pojawieniem się formularza z danymi adresowymi -> najprostszy formularz może wyglądać tak jak poniżej:



Zaimplementuj walidacje pol formularza np. tak jak poniżej



Zakończenie procesu składania zamówienia powinno skutkować zapisaniem szczegółów zamówienia i wyświetleniem odpowiedniego komunikatu na stronie.

Moduł administracyjny – umożliwiający zarzadzanie lista produktów, przeglądanie listy zamówień oraz zarządzanie promocjami. Szczegółowa specyfikacja pojawi się na następnych zajęciach.