

# Testowanie oprogramowania

Michał Kleszczyński, IOiS Zima 2020

## Lab1 – wprowadzenie

### Zadanie 1

#### Opis środowiska testowego

Środowisko testowe będzie składać się z NodeJS w najnowszej dostępnej wersji (15.4.0), oraz przeglądarki Chrome w najnowszej dostępnej wersji (87.0.4280.88)

#### Narzędzia

Podczas wykonywania testów zostaną użyte następujące narzędzia:

- Jasmine – Behavior-Driven-Development framework do testowania kodu JS. Nie jest zależny od żadnego innego frameworka.
- Karma – test runner, środowisko uruchomieniowe dla testów.
- Protractor – framework testowy do testów E2E przeznaczony tylko do testowania aplikacji napisanych z wykorzystaniem Angular. Umożliwia uruchomienie przeglądarki i wykonywanie w niej konkretnych akcji użytkownika

### Zadanie 2

Wykonałbym następujące testy

- Sprawdzenie czy dla pustego inputu wynikowy ciąg znaków to „0”
- Sprawdzenie czy dla inputu z wartością ciąg znaków to wartość + długość wartości
- Sprawdzenie czy dla inputu ze znakami specjalnymi wartość jest poprawna

## Lab2 – testy jednostkowe

### [app/pipes/filter-trips.pipe.ts](#)

1. Sprawdzenie czy pipe jest tworzony bez błędów.
2. Sprawdzenie czy jeżeli jest podana data to następuje filtracja po dacie.
3. Sprawdzenie czy jeżeli nazwa jest podana to następuje filtracja po nazwie.
4. Sprawdzenie czy jeżeli zakres cenowy jest podany to następuje filtracja po zakresie cenowym.
5. Sprawdzenie czy jeżeli zakres ocen jest podany to następuje filtracja po zakresie cenowym.
6. Sprawdzenie czy filtry się łączą.
7. Sprawdzenie czy jeżeli daty nie są zdefiniowane to nie następuje filtracja po dacie.

### [app/services/shopping-cart.service.ts](#)

1. Sprawdzenie czy usługa jest tworzona bez błędów.
2. Sprawdzenie czy usługa inicjuje się z pustymi danymi.
3. Sprawdzenie czy dodanie wycieczki aktualizuje odpowiednio dane.
4. Sprawdzenie czy dodanie istniejącej wycieczki zwiększy licznik dla tej wycieczki.
5. Sprawdzenie czy usunięcie istniejącej wycieczki zmniejszy licznik dla tej wycieczki.
6. Sprawdzenie czy usunięcie wycieczki z licznikiem równym 1 usunie ją.
7. Sprawdzenie czy zapis przy pustej ilości wycieczek nie robi nic.
8. Sprawdzenie czy zapis przy większej ilości wycieczek wykona zapis rezerwacji **n** razy, gdzie **n** to liczba wycieczek (nie suma liczników).

### [app/services/trips.service.ts](#)

1. Sprawdzenie czy usługa jest tworzona bez błędów.
2. Sprawdzenie czy index zwraca kolekcję wycieczek.
3. Sprawdzenie czy show wczytuje wycieczkę o odpowiednim identyfikatorze i zwraca ją.
4. Sprawdzenie czy create dodaje nową wycieczkę (czy metoda jest wywoływana).
5. Sprawdzenie czy update aktualizuje istniejącą wycieczkę.
6. Sprawdzenie czy updateComments aktualizuje tylko komentarze wycieczki.
7. Sprawdzenie czy updateReservations aktualizuje tylko rezerwacje wycieczki.
8. Sprawdzenie czy destroy usuwa wycieczkę.