

Testowanie oprogramowania

Nazwa aplikacji: ShoppingListApp

Opis: Aplikacja napisana w języku Typescript przy użyciu frameworka Angular. Do autentykacji użytkowników oraz persystencji danych korzysta z narzędzia Firebase.

1 Test jednostkowe oraz integracyjne

Poniższe testy, zakładają wcześniejsze stworzenie testowanego elementu wraz z potrzebnymi zależnościami, bądź ich substytutami tzw. mockami. Pozostałe warunki opisano w kolumnie "warunki początkowe"

1.1 Testy serwisów

AuthService

- Korzysta z Firebase do autentykacji użytkowników

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody pobierającej dane ze źródła |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody Tworzącej nowe konto użytkownika |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|--|
| Warunki początkowe | Istniejące konto użytkownika |
| Procedura | Wywołanie metody logującej użytkownika |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|------|
| Warunki początkowe | brak |
|--------------------|------|

| | |
|-----------------|--|
| Procedura | Wywołanie metody wylogowującej użytkownika |
| Warunki końcowe | Metoda zostaje wywołana |

ListsService

- Korzysta z Firebase do persystencji danych - zapisywania list zakupów.

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody pobierającej dane ze źródła |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | brak |
| Procedura | Sprawdzenie czy UID użytkownika jest takie samo jak w authService |
| Warunki końcowe | Wartości UID są takie same |

ListService

- Korzysta z Firebase do persystencji danych - zapisywania produktów w liście zakupów.

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody pobierającej dane ze źródła |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | brak |
| Procedura | Sprawdzenie czy UID użytkownika jest takie samo jak w authService |
| Warunki końcowe | Wartości UID są takie same |

1.2 Testy routingu

AuthGuard

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody pobierającej dane ze źródła |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|--|
| Warunki początkowe | Użytkownik nie jest zalogowany |
| Procedura | Umożliwienie użytkownikowi przejście do zasobów dostępnych dla zalogowanych użytkowników |
| Warunki końcowe | Zwrócona jest wartość fałszywa |

1.3 Testy komponentów

LoginComponent

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody pobierającej dane ze źródła |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | Formularz jest nieprawidłowo wypełniony |
| Procedura | Zablokowanie przycisku submit, gdy formularz jest nieprawidłowy |
| Warunki końcowe | Przycisk zostaje zablokowany |

| | |
|--------------------|---|
| Warunki początkowe | Usługa nie łąduje się |
| Procedura | Zablokowanie wyświetlania komunikatu ładowania, gdy usługa nie łąduje się |
| Warunki końcowe | Komunikat nie pojawia się |

| | |
|--------------------|---|
| Warunki początkowe | brak |
| Procedura | Sprawdzenie opisu pola tekstowego Email |
| Warunki końcowe | Opis pola to Email |

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Sprawdzenie opisu pola tekstowego Password |
| Warunki końcowe | Opis pola to Password |

| | |
|--------------------|---|
| Warunki początkowe | Hasło jest prawidłowe |
| Procedura | Sprawdzenie czy nie wyświetla się komunikat o błędnym hasle |
| Warunki końcowe | Komunikat nie wyświetla się |

| | |
|--------------------|---|
| Warunki początkowe | Email jest prawidłowy |
| Procedura | Sprawdzenie czy nie wyświetla się komunikat o błędnym adresie email |
| Warunki końcowe | Komunikat nie wyświetla się |

| | |
|--------------------|---|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody isAuth z serwisu AuthService |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody loading z serwisu AuthService |
| Warunki końcowe | Metoda zostaje wywołana |

RegisterComponent

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody pobierającej dane ze źródła |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | Formularz jest nieprawidłowo wypełniony |
| Procedura | Zablokowanie przycisku submit, gdy formularz jest nieprawidłowy |
| Warunki końcowe | Przycisk zostaje zablokowany |

| | |
|--------------------|---|
| Warunki początkowe | Usługa nie ładuje się |
| Procedura | Zablokowanie wyświetlania komunikatu ładowania, gdy usługa nie ładuje się |
| Warunki końcowe | Komunikat nie pojawia się |

| | |
|--------------------|---|
| Warunki początkowe | brak |
| Procedura | Sprawdzenie opisu pola tekstowego Email |
| Warunki końcowe | Opis pola to Email |

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Sprawdzenie opisu pola tekstowego Password |
| Warunki końcowe | Opis pola to Password |

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Sprawdzenie opisu pola tekstowego Confirm Password |
| Warunki końcowe | Opis pola to Confirm Password |

| | |
|--------------------|---|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody isAuth z serwisu AuthService |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody loading z serwisu AuthService |
| Warunki końcowe | Metoda zostaje wywołana |

ListsComponent

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody pobierającej dane ze źródła |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody addList z serwisu ListsService |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|--|
| Warunki początkowe | Id listy |
| Procedura | Wywołanie metody updateListStatus z serwisu ListsService |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | Id listy |
| Procedura | Wywołanie metody editList w komponencie |

| | |
|-----------------|-------------------------|
| Warunki końcowe | Metoda zostaje wywołana |
|-----------------|-------------------------|

| | |
|--------------------|--|
| Warunki początkowe | Id Listy |
| Procedura | Wywołanie metody deleteList z serwisu ListsService |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|--|
| Warunki początkowe | Id Listy |
| Procedura | Wywołanie metody updateListFields z serwisu ListsService |
| Warunki końcowe | Metoda zostaje wywołana |

ListComponent

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody pobierającej dane ze źródła |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody addList z serwisu ListService |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | Id listy |
| Procedura | Wywołanie metody updateListItemStatus z serwisu ListService |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|----------|
| Warunki początkowe | Id listy |
|--------------------|----------|

| | |
|-----------------|---|
| Procedura | Wywołanie metody editListItem w komponencie |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | Id Listy |
| Procedura | Wywołanie metody deleteListItem z serwisu ListService |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | Id Listy |
| Procedura | Wywołanie metody updateListItemFields z serwisu ListService |
| Warunki końcowe | Metoda zostaje wywołana |

ListsEditComponent

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody pobierającej dane ze źródła |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody edytującej listę po kliknięciu w przycisk save |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | Ustawienie wartości w formularzu |
| Procedura | Wywołanie metody sprawdzającej poprawność wartości w formularzu |
| Warunki końcowe | Metoda zostaje wywołana |

ListEditComponent

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody pobierającej dane ze źródła |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody edytującej produkt z listy po kliknięciu w przycisk save |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | Ustawienie wartości w formularzu |
| Procedura | Wywołanie metody sprawdzającej poprawność wartości w formularzu |
| Warunki końcowe | Metoda zostaje wywołana |

AppComponent

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody pobierającej dane ze źródła |
| Warunki końcowe | Metoda zostaje wywołana, aplikacja zostaje utworzona |

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Wywołanie metody isAuthenticated z serwisu AuthService |
| Warunki końcowe | Metoda zostaje wywołana |

| | |
|--------------------|---|
| Warunki początkowe | Użytkownik jest zalogowany |
| Procedura | Wywołanie metody logout z serwisu AuthService gdy przycisk logout zostaje kliknięty |

| | |
|-----------------|-------------------------|
| Warunki końcowe | Metoda zostaje wywołana |
|-----------------|-------------------------|

| | |
|--------------------|--|
| Warunki początkowe | Użytkownik nie jest zalogowany |
| Procedura | Wywołanie metody isAuth z serwisu AuthService i sprawdzenie czy zwraca prawidłowe dane |
| Warunki końcowe | Metoda zwraca wartość false |

| | |
|--------------------|--|
| Warunki początkowe | Użytkownik nie jest zalogowany |
| Procedura | Sprawdzenie czy przycisk do logowania występuje na stronie |
| Warunki końcowe | Przycisk występuje na stronie |

| | |
|--------------------|--|
| Warunki początkowe | Użytkownik nie jest zalogowany |
| Procedura | Sprawdzenie czy przycisk do rejestracji występuje na stronie |
| Warunki końcowe | Przycisk występuje na stronie |

| | |
|--------------------|--|
| Warunki początkowe | Użytkownik nie jest zalogowany |
| Procedura | Sprawdzenie czy przycisk do logowania ma prawidłową wartość tekstową |
| Warunki końcowe | Opis pola to Login |

| | |
|--------------------|--|
| Warunki początkowe | Użytkownik nie jest zalogowany |
| Procedura | Sprawdzenie czy przycisk do rejestracji ma prawidłową wartość tekstową |
| Warunki końcowe | Opis pola to Register |

| | |
|--------------------|--|
| Warunki początkowe | Użytkownik nie jest zalogowany |
| Procedura | Sprawdzenie czy przycisk do wylogowywania występuje na stronie |

| | |
|-----------------|-----------------------------------|
| Warunki końcowe | Przycisk nie występuje na stronie |
|-----------------|-----------------------------------|

| | |
|--------------------|--|
| Warunki początkowe | brak |
| Procedura | Sprawdzenie czy przycisk do rejestracji ma prawidłową wartość tekstową |
| Warunki końcowe | Opis pola to Register |

| | |
|--------------------|---|
| Warunki początkowe | brak |
| Procedura | Aplikcja powiniea mieć odpowiedni tekst ShoppingListApp |
| Warunki końcowe | Tekst to ShoppingListApp |

| | |
|--------------------|---|
| Warunki początkowe | brak |
| Procedura | Aplikcja powiniea mieć odpowiedni tytuł shoppingListApp |
| Warunki końcowe | Tytuł to shoppingListApp |

Pokrycie kodu testami

```
===== Coverage summary =====
Statements : 69.23% ( 99/143 )
Branches   : 66.66% ( 4/6 )
Functions  : 60.6% ( 40/66 )
Lines      : 69.11% ( 94/136 )
=====
```

Pokrycie kodu testami wynosi około 69%

Uruchamianie testów jednostkowych i integracyjnych

- Posiadanie instalacji node.js + npm kompatybilnej z bibliotekami z package.json
- Posiadanie lokalnej kopii kodu
- Posiadanie Angular CLI dla lokalnej instalacji node.js

Procedura testowania

- Uruchomienie komendy npm install w katalogu projektu

- Uruchomienie komendy ng test w katalogu projektu
- W celu wygenerowania raportu z pokrycia testami, uruchomienie ng test --code-coverage

2. Testy systemowe

- aplikacja jest uruchamiana lokalnie na komputerze użytkownika np.: <http://localhost:4200/>
- w aplikacji istnieje konto testowe, można również utworzyć kolejne
- testy zostały zaimplementowane w języku TypeScript przy użyciu frameworka Cypress

Przypadki testowe

Test1: Wyświetlenie strony głównej
 Test2: Logowanie użytkownika
 Test3: Wylogowanie użytkownika
 Test4: Dodanie nowej listy zakupów
 Test5: Edycja istniejącej listy zakupów
 Test6: Otworzenie istniejącej listy zakupów
 Test7: Usunięcie istniejącej listy zakupów
 Test8: Dodanie produktów do istniejącej listy zakupów
 Test9: Rejestracja użytkownika z różnymi hasłami

2.1.1 Test1:

| | |
|-----------------|--|
| Nazwa | Wyświetlenie strony głównej |
| Opis | Sprawdzenie czy jest możliwość wyświetlenia strony głównej wraz z paskiem nawigacyjnym |
| Warunki Wstępne | Użytkownik nie jest zalogowany |
| Typ testu | Pozytywny |

| l.p | Krok | Oczekiwany rezultat | Rezultat [1 lub 0] |
|-----|--|---|--------------------|
| 1 | Uruchom aplikacje http://localhost:4200/ | Aplikacja zostaje uruchomiona | 1 |
| 2 | Sprawdź czy widoczny jest przycisk [Login] oraz [Register] | Przycisk [Login] jest widoczny Przycisk [Register] jest widoczny | 1 |

2.1.2 Test2:

| | |
|-----------------|---|
| Nazwa | Logowanie użytkownika |
| Opis | Sprawdzenie czy jest możliwość zalogowania się istniejącego użytkownika |
| Warunki Wstępne | Użytkownik nie jest zalogowany/ Użytkownik jest zarejestrowany |
| Typ testu | Pozytywny |

| l.p | Krok | Oczekiwany rezultat | Rezultat [1 lub 0] |
|-----|--|--|--------------------|
| 1 | Uruchom aplikacje http://localhost:4200/ | Aplikacja zostaje uruchomiona | 1 |
| 2 | Naciśnij przycisk [Login] w menu nawigacyjnym | Użytkownik zostaje przekierowany do formularza logowania | 1 |
| 3 | Wpisz email w pole [Email] | Możliwość wpisania emaila | 1 |
| 4 | Wpisz hasło w pole [Password] | Możliwość wpisania hasła | 1 |
| 5 | Naciśnij przycisk [Submit] | Użytkownik zostaje przekierowany do listy zakupów | 1 |

2.1.3 Test3:

| | |
|-----------------|--|
| Nazwa | Wylogowanie użytkownika |
| Opis | Sprawdzenie czy jest możliwość wylogowania się z aplikacji |
| Warunki Wstępne | Użytkownik nie jest zalogowany/ Użytkownik jest zarejestrowany |
| Typ testu | Pozytywny |

| l.p | Krok | Oczekiwany rezultat | Rezultat [1 lub 0] |
|-----|--|--|--------------------|
| 1 | Uruchom aplikacje http://localhost:4200/ | Aplikacja zostaje uruchomiona | 1 |
| 2 | Wykonaj kroki podane w Test2 | | 1 |
| 3 | Naciśnij przycisk [Logout] w menu nawigacyjnym | Użytkownik zostaje wylogowany i przekierowany na stronę główną | 1 |

2.1.4 Test4:

| | |
|-----------------|--|
| Nazwa | Dodanie nowej listy zakupów |
| Opis | Sprawdzenie czy jest możliwość dodania nowej listy zakupów |
| Warunki Wstępne | Użytkownik nie jest zalogowany/ Użytkownik jest zarejestrowany |
| Typ testu | Pozytywny |

| l.p | Krok | Oczekiwany rezultat | Rezultat [1 lub 0] |
|-----|--|-------------------------------|--------------------|
| 1 | Uruchom aplikacje http://localhost:4200/ | Aplikacja zostaje uruchomiona | 1 |
| 2 | Wykonaj kroki podane w Test2 | | 1 |

| | | | |
|---|---|---------------------------|---|
| 3 | Wprowadź tytuł listy zakupów w polu [Title] | Możliwość wpisania tytułu | 1 |
| 4 | Wprowadz date listy zakupow w polu [dd/mm/yy] | Mozliwosc wybrania daty | 1 |
| 5 | Dodaj nowa liste naciskając zielony przycisk z plusem [AddButton] | Nowa lista zostaje dodana | 1 |

2.1.5 Test5:

| | |
|-----------------|--|
| Nazwa | Edycja istniejącej listy zakupów |
| Opis | Sprawdzenie czy jest możliwość edytowania istniejącej listy zakupów |
| Warunki Wstępne | Użytkownik nie jest zalogowany/ Użytkownik jest zarejestrowany/ w aplikacji jest utworzona lista zakupów |
| Typ testu | Pozytywny |

| l.p | Krok | Oczekiwany rezultat | Rezultat [1 lub 0] |
|-----|---|--|--------------------|
| 1 | Uruchom aplikacje http://localhost:4200/ | Aplikacja zostaje uruchomiona | 1 |
| 2 | Wykonaj kroki podane w Test2 | | 1 |
| 3 | Wybierz listę zakupów i naciśnij przycisk długopisu [EditButton] | Otwarty zostaje formularz edycji listy | 1 |
| 4 | Wprowadź nowy tytuł listy zakupów w polu [Title] | Możliwość wpisania nowego tytułu | 1 |
| 5 | Wprowadź nową datę listy zakupów w polu [dd/mm/yy] | Możliwość wybrania nowej daty | 1 |
| 6 | Zatwierdź zmiany naciskając przycisk [Save] | Lista zostaje wyedytowana | a |
| | | | |

2.1.6 Test6:

| | |
|-----------------|--|
| Nazwa | Otworzenie istniejącej listy zakupów |
| Opis | Sprawdzenie czy istnieje możliwość otwarcia istniejącej listy zakupów |
| Warunki Wstępne | Użytkownik nie jest zalogowany/ Użytkownik jest zarejestrowany/ w aplikacji jest utworzona lista zakupów |

| | |
|-----------|-----------|
| Typ testu | Pozytywny |
|-----------|-----------|

| I.p | Krok | Oczekiwany rezultat | Rezultat [1 lub 0] |
|-----|--|---------------------------------|--------------------|
| 1 | Uruchom aplikacje http://localhost:4200/ | Aplikacja zostaje uruchomiona | 1 |
| 2 | Wykonaj kroki podane w Test2 | | 1 |
| 3 | Wybierz listę zakupów i naciśnij przycisk strzałki [ArrowButton] | Otwarty zostaje formularz listy | 1 |
| | | | |

2.1.7 Test7:

| | |
|-----------------|--|
| Nazwa | Usuniecie istniejącej listy zakupów |
| Opis | Sprawdzenie czy jest możliwość usunięcia istniejącej listy zakupów |
| Warunki Wstępne | Użytkownik nie jest zalogowany/ Użytkownik jest zarejestrowany/ w aplikacji jest utworzona lista zakupów |
| Typ testu | Pozytywny |

| I.p | Krok | Oczekiwany rezultat | Rezultat [1 lub 0] |
|-----|--|-------------------------------|--------------------|
| 1 | Uruchom aplikacje http://localhost:4200/ | Aplikacja zostaje uruchomiona | 1 |
| 2 | Wykonaj kroki podane w Test2 | | 1 |
| 3 | Wybierz listę zakupów i naciśnij przycisk kosza [RubishBinButton] | Lista zostaje usunięta | 1 |
| | | | |

2.1.8 Test8:

| | |
|-----------------|--|
| Nazwa | Logowanie użytkownika |
| Opis | Dodanie produktów do istniejącej listy zakupów |
| Warunki Wstępne | Użytkownik nie jest zalogowany/ Użytkownik jest zarejestrowany/ w aplikacji jest utworzona lista zakupów |
| Typ testu | Pozytywny |

| I.p | Krok | Oczekiwany rezultat | Rezultat [1 lub 0] |
|-----|--|-------------------------------|--------------------|
| 1 | Uruchom aplikacje http://localhost:4200/ | Aplikacja zostaje uruchomiona | 1 |

| | | | |
|---|---|--|---|
| 2 | Wykonaj kroki podane w Test2 | | 1 |
| 3 | Wybierz listę zakupów i naciśnij przycisk długopisu [ArrowButton] | Otwarty zostaje formularz listy | 1 |
| 4 | Wprowadź nazwę produktu w polu [Product] | Możliwość wpisania nazwy produktu | 1 |
| 5 | Wprowadź ilość w polu [Quantity] | Możliwość wprowadzenia ilości | 1 |
| 6 | Wybierz jednostkę miary z listy rozwijalnej w 3 kolumnie | Możliwość wybrania jednostki miary z listy rozwijalnej | 1 |
| 7 | Dodaj nowy produkt naciskając zielony przycisk z plusem [AddButton] | Nowy produkt zostaje dodany do listy | 1 |
| 8 | Naciśnij przycisk [Go Back] | Użytkownik zostaje przekierowany do swoich list | 1 |

2.1.9 Test9:

| | |
|-----------------|--|
| Nazwa | Rejestracja użytkownika z różnymi hasłami |
| Opis | Sprawdzenie czy jest rejestracji nowego użytkownika podając różne hasła w formularzu |
| Warunki Wstępne | Użytkownik nie jest zalogowany/ Użytkownik nie jest zarejestrowany |
| Typ testu | Negatywny |

| l.p | Krok | Oczekiwany rezultat | Rezultat [1 lub 0] |
|-----|---|--|--------------------|
| 1 | Uruchom aplikacje http://localhost:4200/ | Aplikacja zostaje uruchomiona | 1 |
| 2 | Naciśnij przycisk [Register] w menu nawigacyjnym | Użytkownik zostaje przekierowany do formularza rejestracji | 1 |
| 3 | Wpisz email w pole [Email] | Możliwość wpisania email | 1 |
| 4 | Wpisz hasło w pole [Password] | Możliwość wpisania hasła | 1 |
| 5 | Wpisz inne hasło w pole [Confirm Password] | Możliwość wpisania innego hasła | 1 |

| | | | |
|---|----------------------------|---------------------------------------|---|
| 6 | Naciśnij przycisk [Submit] | Użytkownik nie zostaje zarejestrowany | 0 |
|---|----------------------------|---------------------------------------|---|

2.2 Uruchamianie testów systemowych

Do uruchomienia testów systemowych potrzebujemy skonfigurowanego środowiska w tym celu należy:

1. Pobrać projekt z testami Cypress
2. Uruchomić projekt w IDE np. VisualStudio Code
3. Skonfigurować narzędzie Cypress w terminalu IDE
 - `npm init`
 - `npm cypress install --save-dev`
 - `npm typescript install -save-dev`
4. Uruchomić Cypress
 - `npx cypress open`
5. Uruchomić testy za pośrednictwem UI

3 Testy wydajnościowe

3.1 Użyte narzędzia i technologie

Do przeprowadzenia testów użyto narzędzia Locust. Locust to łatwe w użyciu, skryptowe i skalowalne narzędzie do testowania wydajności. Zachowanie użytkowników jest definiowane w języku Python. Nie ma potrzeby używania żadnego interfejsu do konfiguracji testów. Sam proces testowy można obserwować w czasie rzeczywistym w udostępnionym do tego GUI bądź w terminalu. Locust umożliwia również pobranie raportu z testów w formacie html. Dzięki tym rozwiązaniom Locust jest nieskończenie rozszerzalny oraz bardzo przyjazny dla użytkownika.

W celu przeprowadzenia testów w języku Python została zaimplementowana klasa testowa `MyReqRes(HttpUser)` zawierająca metody odpowiedzialne za testowanie poszczególnych requestów podzielone na zadania.

3.1.1 Scenariusz Testowy:


- Utworzenie listy
- Pobranie pojedynczego dokumentu

- Pobranie całej listy
- Usunięcie pojedynczego elementu

Wyniki testów:

Users: 100

Spawn rate: 10/s


LOCUST

HOST
<https://firestore.googleapis.com>

STATUS
READY
 0 users

Start new load test


Number of users (peak concurrency)

Spawn rate (users started/second)

Host (e.g. <http://www.example.com>)

[Start swarming](#)

Statystyka:



LOCUST


HOST
<https://firestore.googleapis.com>

STATUS
RUNNING
 100 users
[Edit](#)

RPS
99

FAILURES
0%





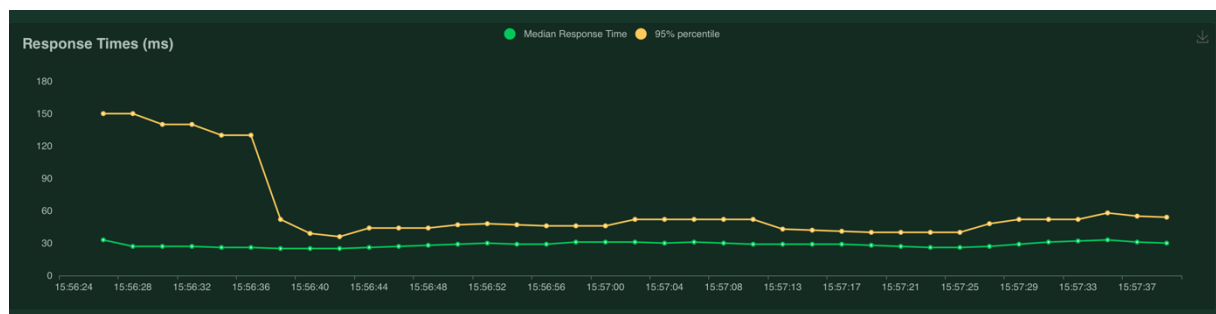
[Statistics](#)
[Charts](#)
[Failures](#)
[Exceptions](#)
[Current ratio](#)
[Download Data](#)

| Type | Name | # Requests | # Fails | Median (ms) | 90%ile (ms) | 99%ile (ms) | Average (ms) | Min (ms) | Max (ms) | Average size (bytes) | Current RPS | Current Failures/s |
|-------------------|--|-------------|----------|-------------|-------------|-------------|--------------|-----------|------------|----------------------|-------------|--------------------|
| GET | /v1/projects/shoppinglist-app8/databases/(default)/documents/lists | 414 | 0 | 30 | 49 | 170 | 40 | 21 | 309 | 10679 | 23.7 | 0 |
| POST | /v1/projects/shoppinglist-app8/databases/(default)/documents/lists | 397 | 0 | 28 | 46 | 160 | 37 | 21 | 229 | 419 | 25.6 | 0 |
| GET | /v1/projects/shoppinglist-app8/databases/(default)/documents/lists/6pEcXys1ZHUUAUjwxc6s | 437 | 0 | 24 | 38 | 140 | 31 | 17 | 251 | 443 | 26 | 0 |
| DELETE | /v1/projects/shoppinglist-app8/databases/(default)/documents/lists/M8MOIX05uaXB.Jsz9IKFM | 402 | 0 | 23 | 37 | 140 | 30 | 17 | 197 | 3 | 23.7 | 0 |
| Aggregated | | 1650 | 0 | 26 | 44 | 150 | 34 | 17 | 309 | 2898 | 99 | 0 |

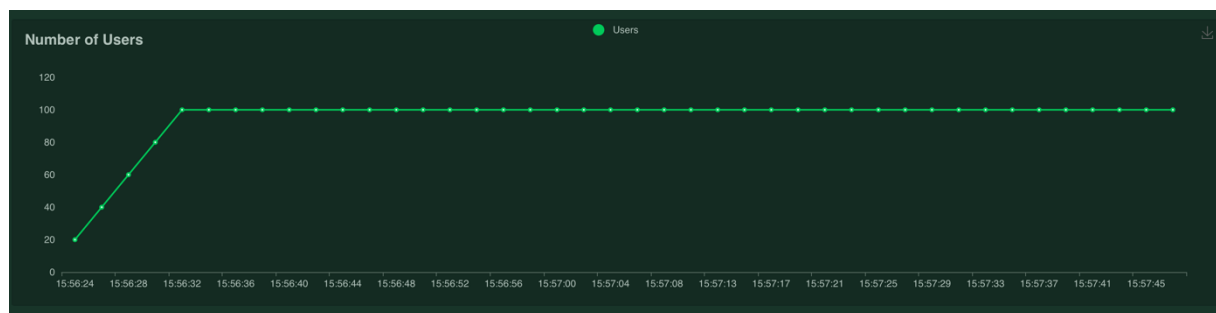
Ilość requestów na sekundę:



Czasy odpowiedzi:



Liczba użytkowników w czasie:



Wnioski z testów

Z otrzymanych wyników możemy zaobserwować, że API Firebase utrzymuje się na jednolitym poziomie po osiągnięciu ustalonej liczby użytkowników. Spadek czasu odpowiedzi po uzyskaniu nominalnej wartości użytkowników może świadczyć o optymalizacji zasobów przy wzmożonym ruchu. Po ustabilizowaniu czasu odpowiedzi pozostają na optymalnym poziomie.

Artefakty i reprodukcja wyników

Artefaktami przeprowadzonych testów są 2 pliki. Znajdują się one kolejno w folderze load oraz reports w katalogu z projektem. W celu reprodukcji wyników, należy mieć zainstalowaną bibliotekę Locust w IDE obsługującym język Python.

1. locustPerformanceTests.py

Jest to plik zawierający kod odpowiadający za konfigurację testów. W celu uruchomienia testów:

```
pip install locust
```

```
locust -f locustPerformanceTests.py
```

2. report_locust.html

Raport zawierający wszystkie informacje z testów w formacie html

Wymagania co do środowiska testowego

- komputer z systemem Windows, OsX, Linux oraz przeglądarka internetowa

4 Test bezpieczeństwa

4.1 Użyte narzędzia:

W celu wykonania testów bezpieczeństwa użyto narzędzia OWASP Zed Attack Proxy (ZAP). Po zakończeniu testów został wygenerowany raport postaci pliku html.

4.1 Procedura testowania:

Ustawienie proxy w ZAP jak również w Postman na ten sam port w localhost. Wysłanie wszystkich testowanych requestow tak, żeby zostały przejęte przez program ZAP. Następnie uruchomienie procedury Active scan jak również uruchomienie Spider oraz Ajax Spider.

4.3 Scenariusz Testowy:

- Utworzenie listy
- Pobranie pojedynczego dokumentu
- Pobranie całej listy
- Usunięcie pojedynczego elementu

4.4 Wyniki testów:

Alerty:

| Alert type | Risk | Count |
|--|---------------|------------------|
| SQL Injection | High | 1 (20.0%) |
| Content Security Policy (CSP) Header Not Set | Medium | 10 (200.0%) |
| Strict-Transport-Security Header Not Set | Low | 19 (380.0%) |
| Re-examine Cache-control Directives | Informational | 5 (100.0%) |
| User Agent Fuzzer | Informational | 57 (1,140.0%) |
| Total | | 5 |

Podsumowanie:

| | | Confidence | | | | |
|------|---------------|-------------------|--------------|--------------|--------------|--------------|
| | | User Confirmed | High | Medium | Low | Total |
| Risk | High | 0 (0.0%) | 0 (0.0%) | 1 (20.0%) | 0 (0.0%) | 1 (20.0%) |
| | Medium | 0 (0.0%) | 1 (20.0%) | 0 (0.0%) | 0 (0.0%) | 1 (20.0%) |
| | Low | 0 (0.0%) | 1 (20.0%) | 0 (0.0%) | 0 (0.0%) | 1 (20.0%) |
| | Informational | 0 (0.0%) | 0 (0.0%) | 1 (20.0%) | 1 (20.0%) | 2 (40.0%) |
| | Total | 0 (0.0%) | 2 (40.0%) | 2 (40.0%) | 1 (20.0%) | 5 (100%) |

4.5 Interpretacja wyników:

- High

Alert:

SQL Injection

SQL Injection - Metoda ataku komputerowego wykorzystująca lukę w zabezpieczeniach aplikacji polegającą na nieodpowiednim filtrowaniu lub niedostatecznym typowaniu danych użytkownika, które to dane są później wykorzystywane przy wykonaniu zapytań do bazy danych

Wyniki strony zostały pomyślnie zmanipulowane przy użyciu warunków boolean [default AND 1=1 --] oraz [default AND 1=2 --]. Zwrócono dane dla oryginalnego parametru. Luka została wykryta dzięki udanemu ograniczeniu pierwotnie zwracanych danych, poprzez manipulację parametrem.

Rozwiązanie:

Nie należy ufać danym wprowadzonym po stronie klienta, nawet jeśli istnieje walidacja po stronie klienta. Ogólnie rzecz biorąc, należy sprawdzić typ wszystkich danych po stronie serwera.

- Medium

Alert:

Content Security Policy (CSP) Header Not Set

Content Security Policy (CSP) - dodatkowa warstwa bezpieczeństwa, która pomaga wykryć i złagodzić pewne typy ataków, w tym ataki Cross Site Scripting (XSS) i ataki typu data injection. Ataki te są wykorzystywane do wszystkiego, od kradzieży danych do zniszczenia witryny lub dystrybucji złośliwego oprogramowania. CSP zapewnia zestaw standardowych nagłówków HTTP, które pozwalają właścicielom stron internetowych zadeklarować zatwierdzone źródła treści, które przeglądarki powinny mieć prawo załadować na danej stronie - uwzględnione typy to JavaScript, CSS, ramki HTML, czcionki, obrazy i obiekty osadzone, takie jak aplety Java, ActiveX, pliki audio i wideo.

Rozwiązanie:

Należy upewnić się, że serwer internetowy, serwer aplikacji, load balancer. jest skonfigurowany tak, aby ustawić nagłówek Content-Security-Policy, aby uzyskać optymalną obsługę przeglądarki.

- Low

Alert:

Strict-Transport-Security Header

HTTP Strict Transport Security (HSTS) to mechanizm polityki bezpieczeństwa sieciowego, za pomocą którego serwer sieciowy deklaruje, że zgodne agenty użytkownika (takie jak

przeglądarka internetowa) mają wchodzić w interakcje z nim przy użyciu wyłącznie bezpiecznych połączeń HTTPS (tj. HTTP z warstwą TLS/SSL). HSTS jest protokołem IETF Standards Track i jest określony w RFC 6797.

Rozwiązanie:

Należy upewnić się, że serwer internetowy, serwer aplikacji, load balancer. jest skonfigurowany tak, aby egzekwować Strict-Transport-Security.

- Informational

Alert:

Re-examine Cache-control Directives

Nagłówek cache-control nie został prawidłowo ustawiony lub brakuje go, co pozwala przeglądarce i proxy na buforowanie zawartości. W przypadku statycznych zasobów, takich jak pliki css, js lub obrazy, może to być zamierzone, jednak zasoby powinny być sprawdzone, aby upewnić się, że żadne wrażliwe treści nie będą buforowane.

Rozwiązanie:

Dla bezpieczeństwa treści, należy upewnić się, że nagłówek cache-control HTTP jest ustawiony na "no-cache, no-store, must-revalidate". Jeśli zasób powinien być buforowany należy rozważyć ustawienie dyrektyw "public, max-age, immutable".

4.6 Artefakty i reprodukcja wyników

Artefaktem z przeprowadzonych testów jest 1 pliki. Plik znajduje się w folderze reports w katalogu z projektem. W celu reprodukcji wyników, należy mieć zainstalowane oraz skonfigurowane narzędzia ZAP oraz Postman (lub podobne) i wykonać skanowanie w programie.

1. report_zap.html

Jest to plik zawierający raport w postaci pliku html wygenerowany przy użyciu narzędzia ZAP.

Wymagania co do środowiska testowego

- komputer z systemem Windows, OsX, Linux oraz zainstalowane narzędzie ZAP oraz Postman (lub podobne)