# Scenariusz testowy – Rejestracja

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Nazwa | Opis | Typ | Czynności przygotowawcze | Czynności końcowe |
| 2 | Rejestracja | Sprawdzenie poprawności działania endpointów do operacji na komentarzach | Testy jednostkowe | * Uruchomić lokalnie backend oraz serwer bazy danych * Uruchomić postmana | - |

## Przypadki testowe – Rejestracja

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Nazwa | Warunki wstępne | Kroki wykonania | Oczekiwany rezultat |
| 1 | Próba zarejestrowania, jeśli nie podano wszystkich wymaganych danych. | * Nie uzupełniono wszyskich wymaganych danych. | * Przy uzupełnianiu danych pozostawić jedno miejsce z wymaganymi danymi puste * Wysłać żądanie POST na dopowiedni endpoint odpowiadający za rejestrację | Użytkownik nie został zarejestrowany do systemu, wysłanie odpowiedzi informującej o nieudanym logowaniu |
| 2 | Próba zarejestrowania użytkownika wykorzystując adres email przypisany do aktywnego konta. | * Poprawne uzupełnienie pól rejestracji bez adresu email. * Podanie emailu przypisanego do aktywnego konta | * Wysłać żądanie POST na dopowiedni endpoint odpowiadający za rejestrację | Użytkownik nie został zarejestrowany do systemu, wysłanie odpowiedzi o statusie informującym o wykorzystaniu tego adresu email przez innego użytkownika |
| 3 | Próba zarejestrowania użytkownika wykorzystując adres numer telefonu przypisany do aktywnego konta. | * Poprawne uzupełnienie pól rejestracji bez numeru telefonu. * Podanie numeru telefonu przypisanego do aktynego konta | * Wysłać żądanie POST na dopowiedni endpoint odpowiadający za rejestrację | Użytkownik nie został zarejestrowany do systemu, wysłanie odpowiedzi o statusie informującym o wykorzystaniu tego numeru telefonu przez innego użytkownika |
| 4 | Próba zarejestrowania użytkownika przy podaniu poprawnych danych. | * Uzupełnione wszystkie wymagane pola | * Wysłać żądanie POST na dopowiedni endpoint odpowiadający za rejestrację | Utworzenie konta użytkownika |

# Scenariusz testowy – Logowanie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Nazwa | Opis | Typ | Czynności przygotowawcze | Czynności końcowe |
| 1 | Logowanie | Sprawdzenie poprawności logowania użytkownika. | Testy jednostkowe | * Uruchomić lokalnie backend oraz serwer bazy danych * Uruchomić postmana | - |

## Przypadki testowe – Logowanie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Nazwa | Warunki wstępne | Kroki wykonania | Oczekiwany rezultat |
| 1 | Próba zalogowania przy wpisaniu poprawnego loginu i hasła | Istnieje użytkownik zarejestrowany w systemie | * Wprowadzenie poprawnego loginu i hasła * Wysłanie żądania POST | Użytkownik został zalogowany o systemu |
| 2 | Próba zalogowania przy wpisaniu niepoprawnego loginu i poprawnego hasła | Istnieje użytkownik z podanym hasłem z w systemie | * Wprowadzenie niepoprawnego loginu i poprawnego hasła * Wysłanie żądania POST | Użytkownik nie został zalogowany do systemu, wysłanie odpowiedzi informującej o nieudanym logowaniu |
| 3 | Próba zalogowania przy wpisaniu niepoprawnego hasła i poprawnego loginu | Istnieje użytkownik z podanym loginem z w systemie | * Wprowadzenie niepoprawnego hasła i poprawnego loginu * Wysłanie żądania POST | Użytkownik nie został zalogowany do systemu, wysłanie odpowiedzi informującej o nieudanym logowaniu |

# Scenariusz testowy – Wylogowanie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Nazwa | Opis | Typ | Czynności przygotowawcze | Czynności końcowe |
| 3 | Wylogowanie | Sprawdzenie poprawności działania endpointów do operacji wylogowania | Testy jednostkowe | * Uruchomić lokalnie backend oraz serwer bazy danych * Uruchomić postmana | - |

## Przypadki testowe – Wylogowanie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Nazwa | Warunki wstępne | Kroki wykonania | Oczekiwany rezultat |
| 1 | Sprawdzenie poprawności procesu wylogowania użytkownika z aktywnej sesji | * Sesja musi być aktywna * Użytkownik musi być zalogowany | * Wysłanie żądania GET na endpoint odpowiadający za wylogowanie | Zwrócono informację o poprawnym wylogowaniu |
| 2 | Próba wylogowania użytkownika z nieaktywnej sesji | * Sesja musi być nieaktywna * Użytkownik musi być zalogowany | * Wysłanie żądania GET na endpoint odpowiadający za wylogowanie | Odpowiedź informująca o potrzebie odświeżenia strony |

# Scenariusz testowy – Zmiana danych użytkownika

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Nazwa | Opis | Typ | Czynności przygotowawcze | Czynności końcowe |
| 4 | Zmiana danych użytkownika | Sprawdzenie poprawności działania endpointów do operacji na danych z profilu użytkownika | Testy jednostkowe | * Uruchomić lokalnie backend oraz serwer bazy danych * Uruchomić postmana | - |

## Przypadki testowe – Zmiana danych użytkownika

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Nazwa | Warunki wstępne | Kroki wykonania | Oczekiwany rezultat |
| 1 | Sprawdzanie poprawności procesu zmiany danych profilu użytkownika | * Sesja musi być aktywna * Użytkownik musi być zalogowany * Muszą zostać wprowadzone poprawne dane | * Wysłanie żądania PUT na endpoint odpowiadający za zmianę danych profilu użytkownika | Zaktualizowanie bazy danych i zwrócenie odpowiedzi informującej o poprawnym zaktualizowaniu profilu |
| 2 | Próba wprowadzenia niepoprawnych danych do profilu użytkownika | * Sesja musi być aktywna * Użytkownik musi być zalogowany * Muszą zostać wprowadzone niepoprawne dane | * Wysłanie żądania PUT na endpoint odpowiadający za zmianę danych profilu użytkownika | Zwróconie odpowiedzi informującej o niepoprawności wprowadzonych danych |
| 3 | Próba zapisania profilu użytkownika bez wprowadzenia wymaganych danych | * Sesja musi być aktywna * Użytkownik musi być zalogowany * Przynajmniej jedno pole z wymaganymi danymi musi pozostać puste | * Wysłanie żądania PUT na endpoint odpowiadający za zmianę danych profilu użytkownika | Zwróconie odpowiedzi informującej o braku wymaganych danych |
|  |  |  |  |  |

# Scenariusz testowy – operacje na postach

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Nazwa | Opis | Typ | Czynności przygotowawcze | Czynności końcowe |
| 1 | Operacje na postach | Sprawdzenie poprawności działania endpointów do operacji na postach | Testy jednostkowe | * Uruchomić lokalnie backend oraz serwer bazy danych * Uruchomić postmana | - |

## Przypadki testowe – operacje na postach

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Nazwa | Warunki wstępne | Kroki wykonania | Oczekiwany rezultat |
| 1 | Usuwanie postu | * Zautoryzowano użytkownika * Istnieje post, który można usunąć | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie postu | Post został usunięty z bazy danych wraz ze wszystkimi powiązanymi z nim komenatrzami, wyświetleniami oraz głosami  Został zwrócony kod 204 – No Content |
| 2 | Modyfikacja postu | * Zautoryzowano użytkownika * Istnieje post, który można zmodyfikować | * Stworzyć zmodyfikowaną wersję postu * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację postu | Post został pomyślnie zmodyfikowany, zmiany zostały zachowane w bazie danych,  Został zwrócony kod 204 – No Content |
| 3 | Dodawanie postu | * Zautoryzowano użytkownika | * Uzupełnić dane potrzebne do stworzenia postu * Wysłać żądanie POST na endpoint odpowiadający za dodanie postu | Post został pomyślnie dodany do bazy danych.  Został zwrócony kod 201 – Created |
| 4 | Zwrócenie postu | * Istnieje post, który można zwrócić | * Wysłać żądanie GET na endpoint odpowiadający za zwrócenie postu | Post został pomyślnie zwrócony  Został zwrócony kod 200 – Ok |
| 5 | Próba usunięcia postu który nie istnieje | * Zautoryzowano użytkownika | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie postu, podając id nieistniejącego postu | Został zwrócony kod 404– Not Found |
| 6 | Próba zmodyfikowania postu który nie istnieje | - | * Stworzyć zmodyfikowaną wersję postu * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację postu, podając id postu który nie istnieje | Został zwrócony kod 404 – Not Found |
| 7 | Próba dodania postu bez treści | * Zautoryzowano użytkownika * Istnieje użytkownik, lub grupa do której może należeć post | * Uzupełnić dane potrzebne do stworzenia postu, z pominięciem treści * Wysłać żądanie POST na endpoint odpowiadający za dodanie postu | Został zwrócony kod 400 – Bad Request |
| 8 | Próba zmodyfikowania postu wysyłając pustą treść postu | * Zautoryzowano użytkownika * Istnieje użytkownik, lub grupa do której może należeć post | * Uzupełnić dane potrzebne do stworzenia postu, z pominięciem treści * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację postu | Został zwrócony kod 400 – Bad Request |
| 9 | Próba usunięcia postu nie będąc autoryzowanym | - | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie postu | Został zwrócony kod 401 – Unauthorized |
| 10 | Próba dodania postu nie będąc autoryzowanym | - | * Uzupełnić dane potrzebne do stworzenia postu * Wysłać żądanie POST na endpoint odpowiadający za dodanie postu | Został zwrócony kod 401 – Unauthorized |
| 11 | Próba modyfikacji postu nie będąc autoryzowanym | - | * Stworzyć zmodyfikowaną wersję postu * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację postu | Został zwrócony kod 401 – Unauthorized |
| 12 | Próba modyfikacji postu nie będąc uprawnionym | * Zautoryzowano użytkownika | * Stworzyć zmodyfikowaną wersję postu * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację postu który nie należy do użytkownika | Został zwrócony kod 403 – Forbidden |
| 13 | Próba usunięcia postu nie będąc uprawnionym | * Zautoryzowano użytkownika | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie postu który nie należy do użytkownika, bądź nie jest administratorem w grupie w której jest post | Został zwrócony kod 403 – Forbidden |
| 14 | Próba dodania postu nie będąc uprawnionym | * Zautoryzowano użytkownika | * Uzupełnić dane potrzebne do stworzenia postu * Wysłać żądanie POST na endpoint odpowiadający za dodanie postu w grupie zamkniętej do której użytkownik nie należy | Został zwrócony kod 403 – Forbidden |
| 14 | Oddanie głosu na post | * Zautoryzowano użytkownika * Istnieje post na który można oddać głos | * Uzupełnić dane potrzebne do stworzenia głosu * Wysłać żądanie POST na endpoint odpowiadający za dodanie głosu | Został zwrócony kod 200 – Ok |
| 15 | Próba oddania głosu na post więcej niż jeden raz | * Zautoryzowano użytkownika | * Uzupełnić dane potrzebne do stworzenia głosu * Dwukrotnie wysłać żądanie POST na endpoint odpowiadający za oddanie głosu | Został zwrócony kod 400 – Bad Request |
| 16 | Cofnięcie oddania głosu na post | * Zautoryzowano użytkownika * Istnieje post na który użytkownik oddał głos | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie głosu | Został zwrócony kod 204 – No Content |
| 17 | Próba cofnięcia głosu na post na który się nie głosowało | * Zautoryzowano użytkownika | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie głosu, na post od id na który nie oddało się głosu | Został zwrócony kod 400 – Bad Request |
| 18 | Próba oddania głosu nie będąc autoryzowanym | - | * Uzupełnić dane potrzebne do stworzenia głosu * Wysłać żądanie POST na endpoint odpowiadający za dodanie głosu | Został zwrócony kod 401 – Unauthorized |
| 19 | Próba oddania głosu nie będąc uprawnionym | - | * Uzupełnić dane potrzebne do stworzenia głosu * Wysłać żądanie POST na endpoint odpowiadający za dodanie głosu, podając id postu w grupie zamkniętej, do której użytkownik nie należy | Został zwrócony kod 403 – Forbidden |

# Scenariusz testowy – operacje na komentarzach

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Nazwa | Opis | Typ | Czynności przygotowawcze | Czynności końcowe |
| 2 | Operacje na komentarzach | Sprawdzenie poprawności działania endpointów do operacji na komentarzach | Testy jednostkowe | * Uruchomić lokalnie backend oraz serwer bazy danych * Uruchomić postmana | - |

## Przypadki testowe – operacje na komentarzach

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Nazwa | Warunki wstępne | Kroki wykonania | Oczekiwany rezultat |
| 1 | Usuwanie komentarza | * Zautoryzowano użytkownika * Istnieje komentarz, który można usunąć | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie komentarza | Komentarz został usunięty z bazy danych wraz ze wszystkimi oddanymi na niego głosami  Został zwrócony kod 204 – No Content |
| 2 | Modyfikacja komentarza | * Zautoryzowano użytkownika * Istnieje komentarz, który można zmodyfikować | * Stworzyć zmodyfikowaną wersję komentarza * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację komentarza | Komentarz został pomyślnie zmodyfikowany, zmiany zostały zachowane w bazie danych,  Został zwrócony kod 204 – No Content |
| 3 | Dodawanie komentarza | * Zautoryzowano użytkownika * Istnieje post do którego można dodać komentarz | * Uzupełnić dane potrzebne do stworzenia komentarza * Wysłać żądanie POST na endpoint odpowiadający za dodanie komentarza | Komentarz został pomyślnie dodany do bazy danych.  Został zwrócony kod 201 – Created |
| 4 | Zwrócenie komentarza | * Istnieje komentarz, który można zwrócić | * Wysłać żądanie GET na endpoint odpowiadający za zwrócenie komentarza | Komentarz został pomyślnie zwrócony  Został zwrócony kod 200 – Ok |
| 5 | Oddanie głosu na post | * Zautoryzowano użytkownika * Istnieje komentarz na który można oddać głos | * Uzupełnić dane potrzebne do stworzenia głosu * Wysłać żądanie POST na endpoint odpowiadający za dodanie głosu | Został zwrócony kod 200 – Ok |
| 5 | Próba usunięcia komentarza który nie istnieje | * Zautoryzowano użytkownika | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie komentarza który nie istnieje | Został zwrócony kod 404 – Not Found |
| 6 | Próba modyfikacji komentarza który nie istnieje | * Zautoryzowano użytkownika | * Stworzyć zmodyfikowaną wersję komentarza * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację komentarza podając id nieistniejącego komentarza | Został zwrócony kod 404 – Not Found |
| 7 | Próba dodania komentarza bez podania id postu | * Zautoryzowano użytkownika | * Uzupełnić dane potrzebne do stworzenia komentarza bez podania id postu * Wysłać żądanie POST na endpoint odpowiadający za dodanie komentarza | Został zwrócony kod 400 – Bad Request |
| 8 | Próba dodania komentarza bez dodania treści | * Zautoryzowano użytkownika * Istnieje post do którego można dodać komentarz | * Uzupełnić dane potrzebne do stworzenia komentarza bez treści komentarza * Wysłać żądanie POST na endpoint odpowiadający za dodanie komentarza | Został zwrócony kod 400 – Bad Request |
| 9 | Próba modyfikacji komentarza bez podania id postu | * Zautoryzowano użytkownika | * Stworzyć zmodyfikowaną wersję komentarza bez podania id postu * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację komentarza | Został zwrócony kod 400 – Bad Request |
| 10 | Próba modyfikacji komentarza bez podania treści | * Zautoryzowano użytkownika | * Stworzyć zmodyfikowaną wersję komentarza bez podania treści * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację komentarza | Został zwrócony kod 400 – Bad Request |
| 11 | Próba usunięcia komentarza nie będąc autoryzowanym | - | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie komentarza | Został zwrócony kod 401 – Unauthorized |
| 12 | Próba dodania komentarza nie będąc autoryzowanym | - | * Uzupełnić dane potrzebne do stworzenia komentarza * Wysłać żądanie POST na endpoint odpowiadający za dodanie komentarza | Został zwrócony kod 401 – Unauthorized |
| 13 | Próba modyfikacji komentarza nie będąc autoryzowanym | - | * Stworzyć zmodyfikowaną wersję komentarza * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację komentarza | Został zwrócony kod 401 – Unauthorized |
| 14 | Próba dodania komentarza nie będąc uprawnionym | * Zautoryzowano użytkownika | * Uzupełnić dane potrzebne do stworzenia komentarza * Wysłać żądanie POST na endpoint odpowiadający za dodanie komentarza w grupie zamkniętej do której użytkownik nie należy | Został zwrócony kod 403 – Forbidden |
| 15 | Próba usunięcia komentarza nie będąc uprawnionym | * Zautoryzowano użytkownika | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie komentarza podając id komentarza który nie należy do użytkownika, bądź nie jest administratorem w grupie w której jest komentarz | Został zwrócony kod 403 – Forbidden |
| 16 | Próba modyfikacji komentarza nie będąc uprawnionym | * Zautoryzowano użytkownika | * Stworzyć zmodyfikowaną wersję komentarza * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację komentarza podając id komentarza który nie należy do użytkownika | Został zwrócony kod 403 – Forbidden |
| 17 | Próba oddania głosu na komentarz więcej niż jeden raz | * Zautoryzowano użytkownika | * Uzupełnić dane potrzebne do stworzenia głosu * Dwukrotnie wysłać żądanie POST na endpoint odpowiadający za oddanie głosu | Został zwrócony kod 400 – Bad Request |
| 18 | Cofnięcie oddania głosu na komentarz | * Zautoryzowano użytkownika * Istnieje post na który użytkownik oddał głos | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie głosu | Został zwrócony kod 204 – No Content |
| 19 | Próba cofnięcia głosu na komentarz na który się nie głosowało | * Zautoryzowano użytkownika | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie głosu, na post o id na który nie oddało się głosu | Został zwrócony kod 400 – Bad Request |
| 20 | Próba oddania głosu nie będąc autoryzowanym | - | * Uzupełnić dane potrzebne do stworzenia głosu * Wysłać żądanie POST na endpoint odpowiadający za dodanie głosu | Został zwrócony kod 401 – Unauthorized |

# Scenariusz testowy – operacje na konwersacjach/wiadomościach

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Nazwa | Opis | Typ | Czynności przygotowawcze | Czynności końcowe |
| 3 | Operacje na konwersacjach | Sprawdzenie poprawności działania endpointów do operacji na konwersacjach / wiadomościach | Testy jednostkowe | * Uruchomić lokalnie backend oraz serwer bazy danych * Uruchomić postmana | - |

## Przypadki testowe – operacje na konwersacjach/wiadomościach

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Nazwa | Warunki wstępne | Kroki wykonania | Oczekiwany rezultat |
| 1 | Dodanie wiadomości | * Zautoryzowano użytkownika * Istnieje konwersacja,do której dodać wiadomość | * Uzupełnić dane potrzebne do stworzenia wiadomości * Wysłać żądanie POST na endpoint odpowiadający za dodanie wiadomości | Wiadomość został pomyślnie dodana do bazy danych.  Został zwrócony kod 201 – Created |
| 2 | Usunięcie wiadomości | * Zautoryzowano użytkownika | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie wiadomości | Wiadomość została usunięta z bazy danych.  Został zwrócony kod 204 – No Content |
| 3 | Modyfikacja wiadomości | * Zautoryzowano użytkownika * Istnieje wiadomość którą możemy zmodyfikować | * Stworzyć zmodyfikowaną wersję wiadomości * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację wiadomości | Wiadomość został pomyślnie zmodyfikowana, zmiany zostały zachowane w bazie danych  Został zwrócony kod 204 – No Content |
| 4 | Zwrócenie wiadomości | * Zautoryzowano użytkownika | * Wysłać żądanie GET na endpoint odpowiadający za zwrócenie wiadomości | Wiadomości zostały pomyślnie zwrócone  Został zwrócony kod 200 – Ok |
| 5 | Zwrócenie wiadomości używając stronicowania | * Zautoryzowano użytkownika | * Wysłać żądanie GET na endpoint odpowiadający za zwrócenie wiadomości, podając parametr page oraz pageSize | Stronicowane wiadomości zostały pomyślnie zwrócone  Został zwrócony kod 200 – Ok |
| 6 | Próba zwrócenia wiadomości używając stronicowania oraz podając nieprawidłowe parametry | * Zautoryzowano użytkownika | * Wysłać żądanie GET na endpoint odpowiadający za zwrócenie wiadomości, podając parametr page oraz pageSize o niewłaściwych wartościach (np. nieliczbowe) | Został zwrócony kod 400 – Bad Request |
| 7 | Próba dodania wiadomości bez treści | * Zautoryzowano użytkownika | * Uzupełnić dane potrzebne do stworzenia wiadomości, pomijając treść wiadomości * Wysłać żądanie POST na endpoint odpowiadający za dodanie wiadomości | Został zwrócony kod 400 – Bad Request |
| 8 | Próba dodania wiadomości bez podania id uczestnika konwersacji | * Zautoryzowano użytkownika | * Uzupełnić dane potrzebne do stworzenia wiadomości, pomijając id uczestnika konwersacji * Wysłać żądanie POST na endpoint odpowiadający za dodanie wiadomości | Został zwrócony kod 400 – Bad Request |
| 9 | Próba modyfikacji wiadomości bez podania treści | * Zautoryzowano użytkownika | * Stworzyć zmodyfikowaną wersję wiadomości, pomijając treść wiadomości * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację wiadomości | Został zwrócony kod 400 – Bad Request |
| 10 | Próba modyfikacji wiadomości bez id uczestnika konwersacji | * Zautoryzowano użytkownika | * Stworzyć zmodyfikowaną wersję wiadomości, pomijając id uczestnika konwersacji * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację wiadomości | Został zwrócony kod 400 – Bad Request |
| 11 | Próba dodania wiadomości nie będąc autoryzowanym | - | * Uzupełnić dane potrzebne do stworzenia wiadomości * Wysłać żądanie PUSH na endpoint odpowiadający za dodanie wiadomości | Został zwrócony kod 401 – Unauthorized |
| 12 | Próba usunięcia wiadomości nie będąc autoryzowanym | - | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie wiadomości | Został zwrócony kod 401 – Unauthorized |
| 13 | Próba modyfikacji wiadomości nie będąc autoryzowanym | - | * Stworzyć zmodyfikowaną wersję wiadomości * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację wiadomości | Został zwrócony kod 401 – Unauthorized |
| 14 | Próba dodania wiadomości nie będąc uprawnionym | * Zautoryzowano użytkownika * Istnieje konwersacja do której możemy dodać wiadomość | * Uzupełnić dane potrzebne do stworzenia wiadomości * Wysłać żądanie POST na endpoint odpowiadający za dodanie wiadomości, w konwersacji grupy do której użytkownik nie należy | Został zwrócony kod 403 – Forbidden |
| 15 | Próba usunięcia wiadomości nie będąc uprawnionym | * Zautoryzowano użytkownika * Istnieje wiadomość którą możemy usunąć | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie wiadomości, podając id wiadomości które nie należy do zalogowanego użytkownika, bądź nie jest on właścicielem konwersacji grupowej do której wiadomość należy | Został zwrócony kod 403 – Forbidden |
| 16 | Próba modyfikacji wiadomości nie będąc uprawnionym | * Zautoryzowano użytkownika * Istnieje wiadomość którą możemy zmodyfikować | * Stworzyć zmodyfikowaną wersję wiadomości * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację wiadomości, podając id wiadomości która nie należy do użytkownika | Został zwrócony kod 403 – Forbidden |
| 17 | Próba usunięcia konwersacji nie będąc autoryzowanym | - | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie konwersacji | Został zwrócony kod 401 – Unauthorized |
| 18 | Próba usunięcia wiadomości która nie istnieje | * Zautoryzowano użytkownika | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie wiadomości, podając id wiadomości która nie istnieje | Został zwrócony kod 404 – Not Found |
| 19 | Próba modyfikacji wiadomości która nie istnieje | * Zautoryzowano użytkownika | * Stworzyć zmodyfikowaną wersję wiadomości * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację wiadomości, podając id wiadomości która nie istnieje | Został zwrócony kod 404 – Not Found |
| 20 | Próba dodania wiadomości do konwersacji która nie istnieje | * Zautoryzowano użytkownika | * Uzupełnić dane potrzebne do stworzenia wiadomości * Wysłać żądanie POST na endpoint odpowiadający za dodanie wiadomości, podając id konwersacji która nie istnieje | Został zwrócony kod 404 – Not Found |

# Scenariusz testowy – Blokowanie użytkownika

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Nazwa | Opis | Typ | Czynności przygotowawcze | Czynności końcowe |
| 1 | Blokowanie użytkownika | Sprawdzenie funkcjonalności blokowania użytkownika | Testy jednostkowe | * Uruchomić lokalnie backend oraz serwer bazy danych * Uruchomić postmana * phpMyAdmin | Zatrzymanie lokalnego serwera i bazy danych |

## Przypadki testowe – Blokowanie użytkownika

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Nazwa | Warunki wstępne | Kroki wykonania | Oczekiwany rezultat |
| 1 | Próba zablokownika użytkownika który nie jest znajomym | W bazie danych istnieje 2 użytkowników którzy nie są znajomymi | * Wysłanie żądania POST na endpoint blokujący danego użytkownika. | Użytkownik został zablokowany, odpowiedź OK |
| 2 | Próba zablokowania użytkownika będącego znajomym | W bazie danych istnieje 2 użytkowników którzy są znajomymi | * Wysłanie żądania POST na endpoint blokujący danego użytkownika. * Ponowne wysłanie POST na endpoint blokujący danego użytkownika | Wysłano odpowiedź potwierdzającą zablokowanie użytkownika.  Po potwierdzeniu użytkowynik został zablokowany  Odpowiedź OK |

# Scenariusz testowy – Zaproszenia do znajomych

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Nazwa | Opis | Typ | Czynności przygotowawcze | Czynności końcowe |
| 2 | Zaproszenia do znajomych | Sprawdzenie poprawności działania endpointów do operacji na zaproszeniach do znajomych | Testy integracyjne | * Uruchomić lokalnie backend oraz serwer bazy danych * Uruchomić postmana * Uruchomić phpMyAdmin | Zatrzymanie lokalnego serwera i bazy danych |

## Przypadki testowe – zaproszenia do znajomych

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Nazwa | Warunki wstępne | Kroki wykonania | Oczekiwany rezultat |
| 1 | Zaproszenie użytkownika który nie jest znajomym | * Istnieje 2 użytkowników którzy nie są znajomymi | * Wysłać żądanie GET na endpoit odpowiadający za zaproszenia do znajomych * Sprawdzenie w phpMyAdmin czy żądanie zostało wysłane | Zaproszenia do znajomych zostało wysłane |
| 2 | Zaproszenie użytkownika który już jest znajomym | * Istnieje 2 użytkowników którzy są znajomymi | * Wysłać żądanie GET na endpoit odpowiadający za zaproszenia do znajomych | Zaproszenia do znajomych nie zostało wysłane i została wysłana odpowiedź informująca o braku możliwości wysłania żądania |
| 3 | Zaproszenie użytkownika który nie jest znajomym i ma zablokowaną opcję zapraszania do znajomych | * Istnieje 2 użytkowników. * Zapraszany ma ustawioną flagę w bazie danych blokującą możliwość zapraszania do znajomych * Użytkownicy nie są znajomymi | * Wysłać żądanie GET na endpoit odpowiadający za zaproszenia do znajomych | Zaproszenia do znajomych nie zostało wysłane i została wysłana odpowiedź informująca o braku możliwości wysłania żądania |
| 4 | Zaproszenie użytkownika który jest znajomym i ma zablokowaną opcję zapraszania do znajomych | * Istnieje 2 użytkowników. * Zapraszany ma ustawioną flagę w bazie danych blokującą możliwość zapraszania do znajomych * Użytkownicy są znajomymi | * Wysłać żądanie GET na endpoit odpowiadający za zaproszenia do znajomych | Zaproszenia do znajomych nie zostało wysłane i została wysłana odpowiedź informująca o braku możliwości wysłania żądania |

# Scenariusz testowy – Żądania dołączenia do grup

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Nazwa | Opis | Typ | Czynności przygotowawcze | Czynności końcowe |
| 3 | Żądania dołączenia do grup | Sprawdzenie poprawności działania endpointów do operacji wylogowania | Testy jednostkowe | * Uruchomić lokalnie backend oraz serwer bazy danych * Uruchomić postmana | Zatrzymanie lokalnego serwera i bazy danych |

## Przypadki testowe – Żądania dołączenia do grup

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Nazwa | Warunki wstępne | Kroki wykonania | Oczekiwany rezultat |
| 1 | Sprawdzenie poprawności dołączenia do grup otwartych | * W systemie musi istnieć otwarta grupa | * Wysłanie żądania GET na endpoint odpowiadający za żądania dołączenia do grup | Żądanie zostało wysłane, odpowiedź OK serwera |
| 2 | Sprawdzenie poprawności dołączenia do grup zamkniętych | * W systemie musi istnieć zamknięta grupa | * Wysłanie żądania GET na endpoint odpowiadający za żądania dołączenia do grup | Żądanie nie zostało wysłane, odpowiedź kod błędu |
| 3 | Sprawdzenie poprawności dołączenie do grup otwartch poprzez zaproszenie | * W systemie musi istnieć dwóch użytkowników * Zapraszający musi należeć do otwartej grupy. * Zapraszany nie może być członkiem tej grupy | * Wysłanie żądania GET na endpoint odpowiadający za żądania dołączenia do grup | Zaproszenie zostało wysłane, odpowiedź OK serwera |
| 4 | Sprawdzenie poprawności dołączenie do grup zamkniętych poprzez zaproszenie | * W systemie musi istnieć dwóch użytkowników. * Zapraszający musi należeć do zamkniętej grupy.   Zapraszany nie może być członkiem tej grupy | * Wysłanie żądania GET na endpoint odpowiadający za żądania dołączenia do grup | Zaproszenie zostało wysłane, odpowiedź OK serwera |
| 5 | Sprawdzenie poprawności dołączenia do grupy członka należącego do danej grupy | * W systemie musi istnieć użytkownik należący do istniejącej grupy. | * Wysłanie żądania GET na endpoint odpowiadający za żądania dołączenia do grup | Żądanie nie zostało wysłane, odpowiedź kod błędu |

# Scenariusz testowy – operacje na grupach

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Nazwa | Opis | Typ | Czynności przygotowawcze | Czynności końcowe |
| 6 | Operacje na grupie | Sprawdzenie poprawności działania endpointów do operacji na grupach | Testy jednostkowe | * Uruchomić lokalnie backend oraz serwer bazy danych * Uruchomić postmana | - |

## Przypadki testowe – operacje na grupach

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Nazwa | Warunki wstępne | Kroki wykonania | Oczekiwany rezultat |
| 1 | Dodawanie grupy | * Zautoryzowano użytkownika | * Uzupełnić dane potrzebne do stworzenia grupy * Wysłać żądanie POST na endpoint odpowiadający za dodawanie grupy | Grupa został pomyślnie dodany do bazy danych.  Został zwrócony kod 201 – Created |
| 2 | Zwrócenie grupy |  | * Wysłać żądanie GET na endpoint odpowiadający za zwrócenie grupy | Grupa został pomyślnie zwrócona  Został zwrócony kod 200 – Ok |
| 3 | Modyfikacja grupy | * Zautoryzowano użytkownika | * Uzupełnić dane potrzebne do modyfikacji grupy * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację grupy | Grupa został pomyślnie zmodyfikowana, zmiany zostały zachowane w bazie danych,  Został zwrócony kod 204 – No Content |
| 4 | Usunięcie grupy | * Zautoryzowano użytkownika | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie grupy | Grupa została usunięta z bazy danych wraz ze wszystkimi powiązanymi z nim postami, komentarzami, wyświetleniami itd.  Został zwrócony kod 204 – No Content |
| 5 | Próba usuniecia grupy nie będąc uprawnionym | * Zautoryzowano użytkownika | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie grupy | Został zwrócony kod 403 – Forbidden |
| 6 | Próba modyfikacji grupy nie będąc uprawnionym | * Zautoryzowano użytkownika | * Uzupełnić dane potrzebne do modyfikacji grupy * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację grupy | Został zwrócony kod 403 – Forbidden |
| 7 | Próba dodania grupy nie będąc autoryzowanym | - | * Uzupełnić dane potrzebne do stworzenia grupy * Wysłać żądanie POST na endpoint odpowiadający za dodawanie grupy | Został zwrócony kod 401 – Unauthorized |
| 8 | Próba usunięcia grupy nie będąc autoryzowanym | - | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie grupy | Został zwrócony kod 401 – Unauthorized |
| 9 | Próba modyfikacji grupy nie będąc autoryzowanym | - | * Uzupełnić dane potrzebne do modyfikacji grupy * Wysłać żądanie PUT na endpoint odpowiadający za modyfikację grupy | Został zwrócony kod 401 – Unauthorized |
| 10 | Dodanie żądania dołączenia do grupy | * Zautoryzowano użytkownika | * Wysłać żądanie POST na endpoint odpowiadający za dodawanie żądania dołączenia do grupy | Żądanie dołączenia zostało pomyślnie dodane do bazy danych.  Został zwrócony kod 201 – Created |
| 11 | Usunięcie żądania dołączenia do grupy | * Zautoryzowano użytkownika | * Wysłać żądanie DELETE na endpoint odpowiadający za dodawanie żądania dołączenia do grupy | Żądanie dołączenia zostało pomyślnie usunięte z bazy danych.  Został zwrócony kod 204 – No Content |
| 12 | Odrzucenie żądania dołączenia do grupy | * Zautoryzowano użytkownika | * Wysłać żądanie GET na endpoint odpowiadający za odrzucenie żądania dołączenia do grupy | Żądanie dołączenia zostało pomyślnie usunięte z bazy danych.  Został zwrócony kod 200 – Ok |
| 13 | Dodanie użytkownika do grupy | * Zautoryzowano użytkownika | * Wysłać żądanie GET na endpoint odpowiadający za przyjęcie żądania dołączenia do grupy | Użytkownik został pomyślnie dodany do grupy, dodano nową instancję group participant do bazy.  Został zwrócony kod 200 – Ok |
| 14 | Usunięcie użytkownika z grupy | * Zautoryzowano użytkownika | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie użytkownika z grupy | Uczestnik grupy został pomyślnie usunięty z bazy danych.  Został zwrócony kod 204 – No Content |
| 15 | Próba dodania żądania do grupy nie będąc autoryzowanym | - | * Wysłać żądanie GET na endpoint odpowiadający za przyjęcie żądania dołączenia do grupy | Został zwrócony kod 401 – Unauthorized |
| 16 | Próba odrzucenia żądania do grupy nie będąc autoryzowanym | - | * Wysłać żądanie GET na endpoint odpowiadający za odrzucenie żądania dołączenia do grupy | Został zwrócony kod 401 – Unauthorized |
| 17 | Próba dodania użytkownika do grupy nie będąc autoryzowanym | - | * Wysłać żądanie GET na endpoint odpowiadający za przyjęcie żądania dołączenia do grupy | Został zwrócony kod 401 – Unauthorized |
| 18 | Próba dodania użytkownika do grupy nie będąc uprawnionym | * Zautoryzowano użytkownika | * Wysłać żądanie GET na endpoint odpowiadający za przyjęcie żądania dołączenia do grupy | Został zwrócony kod 403 – Forbidden |
| 19 | Próba usunięcia użytkownika z grupy nie będąc autoryzowanym | - | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie użytkownika z grupy | Został zwrócony kod 401 – Unauthorized |
| 20 | Próba usunięcia użytkownika z grupy nie będąc uprawnionym | * Zautoryzowano użytkownika | * Wysłać żądanie DELETE na endpoint odpowiadający za usunięcie użytkownika z grupy | Został zwrócony kod 403 – Forbidden |

# Scenariusz testowy – wyszukiwanie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Nazwa | Opis | Typ | Czynności przygotowawcze | Czynności końcowe |
| 7 | Wyszukiwanie | Sprawdzenie poprawności działania endpointów do wyszukiwania | Testy jednostkowe | * Uruchomić lokalnie backend oraz serwer bazy danych * Uruchomić postmana | - |

## Przypadki testowe – wyszukiwanie

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Nazwa | Warunki wstępne | Kroki wykonania | Oczekiwany rezultat |
| 1 | Wyszukiwanie grupy | * Istnieje grupa którą można wyszukać | * Wysłać żądanie GET na endpoint odpowiadający za wyszukanie w parametrze podając nazwę grupy bądź jej część | Grupa o nazwie odpowiadającej bądź mającej w sobie podany parametr została pomyślnie zwrócona w search result  Został zwrócony kod 200 – Ok |
| 2 | Wyszukiwanie użytkowników | * Istnieje użytkownik którego można wyszukać | * Wysłać żądanie GET na endpoint odpowiadający za wyszukanie w parametrze podając imię oraz nazwisko użytkownika | Grupa o nazwie odpowiadającej bądź mającej w sobie podany parametr została pomyślnie zwrócona w search result  Został zwrócony kod 200 – Ok |