

## Spis treści

1. Wstęp .....	3
2. Użytkownicy .....	4
2.1. Funkcje niezalogowanego użytkownika .....	4
2.1.1. Rejestracja.....	4
2.1.2. Logowanie.....	4
2.2. Funkcje użytkownika .....	5
2.2.1. Pobranie szczegółów konta .....	5
2.2.2. Zmiana hasła .....	5
2.2.3. Dezaktywacja konta .....	6
2.3. Funkcje administratora.....	6
2.3.1. Pobranie listy wszystkich użytkowników .....	6
2.3.2. Pobranie listy ról użytkowników .....	7
2.3.3. Reset hasła dowolnego użytkownika .....	8
2.3.4. Edycja użytkownika.....	8
3. Miejsca.....	9
3.1. Funkcje użytkownika .....	9
3.1.1. Pobranie listy wszystkich miejsc (zaakceptowanych i aktywnych).....	9
3.1.2. Pobranie wybranego miejsca .....	9
3.1.3. Dodanie nowego miejsca .....	10
3.1.4. Pobranie listy wszystkich odwiedzonych miejsc .....	10
3.1.5. Pobranie listy wszystkich nieodwiedzonych miejsc .....	11
3.2. Funkcje administratora.....	12
3.2.1. Pobranie listy wszystkich miejsc .....	12
3.2.2. Akceptacja miejsca .....	13
3.2.3. Edycja miejsca .....	14
4. Wizyty.....	15
4.1. Funkcje użytkownika .....	15
4.1.1. Pobranie listy odbytych wizyt.....	15
4.1.2. Pobranie listy nieodbytych wizyt.....	15
4.1.3. Dodanie nowej wizyty .....	16
4.1.4. Usunięcie nieodbytej wizyty .....	17
4.1.5. Oznaczenie wizyty jako „odbyta”.....	18
4.1.6. Wyczyszczenie historii odbytych wizyt.....	18
4.1.7. Usunięcie z historii pojedynczej odbytej wizyty .....	19

5.	Komentarze .....	20
5.1.	Funkcje użytkownika .....	20
5.1.1.	Dodanie komentarza .....	20
5.1.2.	Edycja komentarza .....	21
5.1.3.	Usunięcie komentarza .....	21
5.1.4.	Pobranie komentarza do wizyty .....	22
5.1.5.	Pobranie komentarzy dla wybranego miejsca .....	22
5.2.	Funkcje administratora .....	23
5.2.1.	Ukrycie komentarza .....	23
5.2.2.	Przywrócenie komentarza .....	23
5.2.3.	Pobranie wszystkich komentarzy dla wybranego miejsca (również nieaktywnych) .....	24
6.	Zdjęcia .....	25
6.1.	Uwagi .....	25
6.2.	Funkcje użytkownika .....	25
6.2.1.	Pobranie listy szczegółów zdjęć dla danego miejsca .....	25
6.2.2.	Pobranie danego pliku zdjęcia .....	25
6.2.3.	Dodanie nowego zdjęcia .....	26
6.3.	Funkcje administratora .....	26
6.3.1.	Pobranie listy szczegółów wszystkich niezaakceptowanych zdjęć .....	26
6.3.2.	Akceptacja zdjęcia .....	27
6.3.3.	Usunięcie zdjęcia .....	27

## 1. Wstęp

- 1.1. W dokumentacji znajdziecie kompletny spis postawionych serwisów REST.
- 1.2. Formatem, w którym przekazujemy dane jest JSON. Przykładowo, gdzie w podpunkcie „parametry przekazywane w ciele żądania” wymienione są parametry „username” oraz „password”, JSON powinien być zbudowany w następujący sposób:  
`{"username":wartość parametru username, "password":wartość parametru password}`
- 1.3. Dla każdej odpowiedzi z kodem 409, dołączany jest błąd w formacie JSON:  
`{"message":Treść błędu}`
- 1.4. Do odpowiedzi 403 nie są dołączane żadne informacje o błędach
- 1.5. Do każdego żądania oprócz logowania i rejestracji należy przekazywać nagłówek „Token”, a w nim wartość tokenu, którą otrzymaliśmy podczas logowania.
- 1.6. Jeśli w którymś z punktów nie ma podpunktu „Parametry URL”, „Nagłówki HTML”, „Zawartość”, itp., to znaczy że nie są one w danym żądaniu potrzebne, lub nie są zwracane
- 1.7. Jeśli w podpunkcie „URL”, w adresie znajduje się wartość zawarta w nawiasie klamrowym, oznacza to że jest to parametr, np. gdy jest adres /user/register/{language}, to gdy chcemy wysłać żądanie, i otrzymać ewentualną odpowiedź w języku polskim, strzelamy pod adres /user/register/pl
- 1.8. Aplikacja zwraca kody błędów w języku polskim lub angielskim. Aby dostać odpowiedź w języku polskim, jako parametr „language” należy wysłać „pl”. W przypadku każdej innej wartości parametru, np. „en”, „de”, albo nawet „abcd”, zostanie zwrócony błąd w języku angielskim
- 1.9. Otrzymanie błędu 403, oznacza problem z autoryzacją użytkownika, która może wystąpić w wielu przypadkach, np. wygasły token, próba wywołania funkcji zarezerwowanej dla administratora, mimo posługiwania się tokenem administratora, lub niezgodność ID użytkownika przesyłanego np. jako parametr (w zależności od żądania), a ID wyluskany z tokenu. W przypadku otrzymania błędu 403, należy natychmiast wylogować użytkownika i usunąć wszystkie dane, takie jak ID użytkownika lub token z ciasteczek przeglądarki, bądź innego miejsca gdzie takie dane będą składowane, a następnie zażądać ponownego zalogowania się.
- 1.10. Parametry opisywane są według schematu: (Typ) Nazwa parametru – opis parametru

## 2. Użytkownicy

### 2.1. Funkcje niezalogowanego użytkownika

#### 2.1.1. Rejestracja

- Metoda
  - POST
- URL
  - /user/register/{language}
- Parametry URL
  - (String) language – język, w jakim ma być zwrócony ewentualny błąd
- Parametry przekazywane w ciele żądania
  - (String) username – nazwa użytkownika
  - (String) password – hasło użytkownika
- Odpowiedź powodzenia
  - 200 OK
- Odpowiedzi błędów:
  - 403 Forbidden
    - Powody:
      - Próba zarejestrowania użytkownika z rolą administratora
  - 409 Conflict
    - Powody:
      - Zajęta nazwa użytkownika
      - Podany login jest pusty, bądź składa się z samych spacji
      - Podane hasło jest puste, bądź składa się z samych spacji

#### 2.1.2. Logowanie

- Metoda
  - POST
- URL
  - /user/login
- Parametry przekazywane w ciele żądania
  - (String) username – nazwa użytkownika
  - (String) password – hasło użytkownika
- Odpowiedź powodzenia
  - 200 OK
  - Nagłówki HTML:
    - Token – „klucz”, który należy przekazywać przy każdym żądaniu z wyjątkiem żądania logowania, celem uwierzytelnienia
    - UserId – identyfikator użytkownika w bazie danych. Konieczny do przekazywania z niektórymi żądaniami
    - Admin – wartość, która mówi czy zalogowany użytkownik ma uprawnienia administratora. Możliwe wartości: „true” lub „false” (w formacie String)
- Odpowiedzi błędów
  - 403 Forbidden
    - Powody
      - Niepoprawne dane logowania
    - Uwagi
      - Aplikacja nie zwraca informacji czy to nie znaleziono użytkownika, czy znaleziono użytkownika, a hasło jest błędne. Po prostu jak nie pasuje coś, to wyrzuca błąd i koniec.

## 2.2. Funkcje użytkownika

### 2.2.1. Pobranie szczegółów konta

- Metoda
  - GET
- URL
  - /user/details/{userId}
- Parametry URL
  - (long) userId – ID użytkownika
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Obiekt zawierający szczegóły użytkownika, posiadający następujące parametry:
      - (long) id – ID użytkownika w bazie danych
      - (String) username – nazwa użytkownika
      - (boolean) active – czy konto jest aktywne
      - (long) userRoleId – ID roli użytkownika
- Odpowiedzi błędu
  - 403 Forbidden
    - Powody błędu
      - ID podane jako parametr przekazany jako paramer URL i ID wyłuskane z tokenu nie są takie same

### 2.2.2. Zmiana hasła

- Metoda
  - PUT
- URL
  - /user/changePassword/{language}
- Parametry URL
  - (String) language – język, w jakim ma być zwrócony ewentualny błąd
- Parametry przekazywane w ciele żądania
  - (long) userId – ID użytkownika
  - (String) oldPassword – stare hasło
  - (String) newPassword – nowe hasło
- Odpowiedź powodzenia
  - 200 OK
- Odpowiedzi błędu
  - 409 Conflict
    - Powody błędu
      - Stare hasło jest niepoprawne
      - Nie odnaleziono użytkownika o podanym ID
      - Podane nowe hasło jest puste, bądź składa się z samych spacji
  - 403 Forbidden
    - Powody błędu
      - ID podane jako parametr przekazany w ciele żądania i ID wyłuskane z tokenu nie są takie same

### 2.2.3. Dezaktywacja konta

- Metoda
  - PUT
- URL
  - /user/deactivate/{userId}/{language}
- Parametry URL
  - (long) userId – ID użytkownika
  - (String) language – język, w jakim ma być zwrócony ewentualny błąd
- Odpowiedź powodzenia
  - 200 OK
- Odpowiedzi błędu
  - 409 Conflict
    - Powody błędu
      - Nie odnaleziono użytkownika o podanym ID
  - 403 Forbidden
    - Powody błędu
      - ID podane jako parametr URL i ID wyłuskane z tokenu nie są takie same

## 2.3. Funkcje administratora

### 2.3.1. Pobranie listy wszystkich użytkowników

- Metoda
  - GET
- URL
  - /admin/user/all
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista wszystkich użytkowników, gdzie obiekt użytkownika zawiera następujące parametry:
      - (long) id – ID użytkownika w bazie danych
      - (String) username – nazwa użytkownika
      - (boolean) active – czy konto jest aktywne
      - (long) userRoleId – ID roli użytkownika
  - 403 Forbidden
    - Powody błędu
      - Token który został podany, zawiera dane użytkownika niebędącego administratorem

### 2.3.2. Pobranie listy ról użytkowników

- Metoda
  - GET
- URL
  - /admin/role/all
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista wszystkich ról użytkowników, gdzie obiekt roli zawiera następujące parametry:
      - (long) id – ID roli w bazie danych
      - (String) label – nazwa roli użytkownika
- Odpowiedzi błędu
  - 403 Forbidden
    - Powody błędu
      - Token który został podany, zawiera dane użytkownika niebędącego administratorem

### 2.3.3. Reset hasła dowolnego użytkownika

- Metoda
  - PUT
- URL
  - /admin/user/resetPassword/{language}
- Parametry URL
  - (String) language – język, w jakim ma być zwrócony ewentualny błąd
- Parametry przekazywane w ciele żądania
  - (long) userId – ID użytkownika
  - (String) newPassword – nowe hasło
- Odpowiedź powodzenia
  - 200 OK
- Odpowiedzi błędu
  - 403 Forbidden
    - Powody błędu
      - Token który został podany, zawiera dane użytkownika niebędącego administratorem
  - 409 Conflict
    - Powody błędu
      - Nie znaleziono użytkownika o podanym ID
      - Podane nowe hasło jest puste, bądź składa się z samych spacji

### 2.3.4. Edycja użytkownika

- Metoda
  - PUT
- URL
  - /admin/user/edit/{language}
- Parametry URL
  - (String) language – język, w jakim ma być zwrócony ewentualny błąd
- Parametry przekazywane w ciele żądania
  - (long) id – ID użytkownika w bazie danych
  - (String) username – nazwa użytkownika
  - (boolean) active – czy konto jest aktywne
  - (long) userRoleId – ID roli użytkownika
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista wszystkich użytkowników, gdzie obiekt użytkownika zawiera następujące parametry:
      - (long) id – ID użytkownika w bazie danych
      - (String) username – nazwa użytkownika
      - (boolean) active – czy konto jest aktywne
      - (long) userRoleId – ID roli użytkownika
- Odpowiedzi błędu
  - 403 Forbidden
    - Powody błędu
      - Token który został podany, zawiera dane użytkownika niebędącego administratorem
  - 409 Conflict
    - Powody błędu
      - Nie znaleziono użytkownika o podanym ID
      - Podana nazwa użytkownika jest już zajęta
      - Podana nowa nazwa użytkownika jest pusta, bądź składa się z samych spacji



### 3. Miejsca

#### 3.1. Funkcje użytkownika

##### 3.1.1. Pobranie listy wszystkich miejsc (zaakceptowanych i aktywnych)

- Metoda
  - GET
- URL
  - /place/all
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista wszystkich zaakceptowanych przez administratora miejsc, gdzie obiekt miejsca posiada następujące parametry:
      - (long) id – ID miejsca w bazie danych
      - (String) name – nazwa miejsca
      - (String) address – adres miejsca
      - (String) gps – współrzędne miejsca
      - (String) description – opis miejsca
      - (boolean) accepted – czy miejsce zostało zaakceptowane (w tym przypadku będzie zawsze true)
      - (boolean) active – czy miejsce jest „aktywne”, czyli czy np. dalej jest otwarte (w tym przypadku zawsze będzie true)
      - (long) userId – ID użytkownika, który dodał do bazy dane miejsce
      - (String) username – nazwa użytkownika, który dodał do bazy dane miejsce

##### 3.1.2. Pobranie wybranego miejsca

- Metoda
  - GET
- URL
  - /place/{placeId}/{language}
- Parametry URL
  - (long) placeId – ID miejsca, którego szczegóły mają zostać zwrócone
  - (String) language – język, w jakim ma zostać zwrócony ewentualny błąd
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: obiekt miejsca posiadający następujące parametry:
      - (long) id – ID miejsca w bazie danych
      - (String) name – nazwa miejsca
      - (String) address – adres miejsca
      - (String) gps – współrzędne miejsca
      - (String) description – opis miejsca
      - (boolean) accepted – czy miejsce zostało zaakceptowane (w tym przypadku będzie zawsze true)
      - (boolean) active – czy miejsce jest „aktywne”, czyli czy np. dalej jest otwarte (w tym przypadku zawsze będzie true)
      - (long) userId – ID użytkownika, który dodał do bazy dane miejsce
      - (String) username – nazwa użytkownika, który dodał do bazy dane miejsce
  - 409 Conflict
    - Powody błędu
      - Nie znaleziono miejsca o podanym ID

### 3.1.3. Dodanie nowego miejsca

- Metoda
  - POST
- URL
  - /place/new/{language}
- Parametry URL
  - (String) language – język, w którym zwrócony będzie ewentualny błąd
- Parametry przekazywane w ciele żądania
  - (String) name – nazwa miejsca
  - (String) address – adres miejsca
  - (String) gps – współrzędne miejsca
  - (String) description – opis miejsca (nieobowiązkowe)
  - (long) userId – ID użytkownika, który dodał do bazy dane miejsce
- Odpowiedź powodzenia
  - 200 OK
- Odpowiedź błędu
  - 403 Forbidden
    - Powody błędu
      - ID użytkownika podane jako parametr w ciele funkcji i ID wyłuskane z tokenu nie zgadzają się
  - 409 Conflict
    - Powody błędu
      - Nie odnaleziono użytkownika o podanym ID
      - Miejsce o identycznej nazwie i adresie już istnieje w bazie danych
      - Miejsce o identycznej nazwie i współrzędnych GPS już istnieje w bazie danych

### 3.1.4. Pobranie listy wszystkich odwiedzonych miejsc

- Metoda
  - GET
- URL
  - /place/allVisitedPlaces/{userId}
- Parametry URL
  - (long) userId – ID użytkownika w bazie danych
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista miejsc, dla których zostały zdefiniowane wizyty przez danego użytkownika, oraz zostały one oznaczone jako odbyte. Obiekt miejsca posiada następujące parametry:
      - (long) id – ID miejsca w bazie danych
      - (String) name – nazwa miejsca
      - (String) address – adres miejsca
      - (String) gps – współrzędne miejsca
      - (String) description – opis miejsca
      - (boolean) accepted – czy miejsce zostało zaakceptowane
      - (boolean) active – czy miejsce jest „aktywne”, czyli czy np. dalej jest otwarte
      - (long) userId – ID użytkownika, który dodał do bazy dane miejsce
      - (String) username – nazwa użytkownika, który dodał do bazy dane miejsce
- Odpowiedź błędu
  - 403 Forbidden
    - Przyczyny błędu
      - ID użytkownika podane jako parametr i ID wyłuskane z tokenu nie zgadzają się

### 3.1.5. Pobranie listy wszystkich nieodwiedzonych miejsc

- Metoda
  - GET
- URL
  - /place/allNotVisitedPlaces/{userId}
- Parametry URL
  - (long) userId – ID użytkownika w bazie danych
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista miejsc, dla których zostały zdefiniowane wizyty przez danego użytkownika, oraz zostały one oznaczone jako nieodbyte. Obiekt miejsca posiada następujące parametry:
      - (long) id – ID miejsca w bazie danych
      - (String) name – nazwa miejsca
      - (String) address – adres miejsca
      - (String) gps – współrzędne miejsca
      - (String) description – opis miejsca
      - (boolean) accepted – czy miejsce zostało zaakceptowane
      - (boolean) active – czy miejsce jest „aktywne”, czyli czy np. dalej jest otwarte
      - (long) userId – ID użytkownika, który dodał do bazy dane miejsce
      - (String) username – nazwa użytkownika, który dodał do bazy dane miejsce
- Odpowiedź błędu
  - 403 Forbidden
    - Przyczyny błędu
      - ID użytkownika podane jako parametr i ID wyłuskane z tokenu nie zgadzają się

## 3.2. Funkcje administratora

### 3.2.1. Pobranie listy wszystkich miejsc

- Metoda
  - GET
- URL
  - /admin/place/all
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista wszystkich miejsc, gdzie obiekt miejsca posiada następujące parametry:
      - (long) id – ID miejsca w bazie danych
      - (String) name – nazwa miejsca
      - (String) address – adres miejsca
      - (String) gps – współrzędne miejsca
      - (String) description – opis miejsca
      - (boolean) accepted – czy miejsce zostało zaakceptowane
      - (boolean) active – czy miejsce jest „aktywne”, czyli czy np. dalej jest otwarte
      - (long) userId – ID użytkownika, który dodał do bazy dane miejsce
      - (String) username – nazwa użytkownika, który dodał do bazy dane miejsce
- Odpowiedzi błędu
  - 403 Forbidden
    - Powody błędu
      - Token który został podany, zawiera dane użytkownika niebędącego administratorem

### 3.2.2. Akceptacja miejsca

- Metoda
  - PUT
- URL
  - /admin/place/accept/{placeId}/{language}
- Parametry URL
  - (long) placeId – ID miejsca w bazie danych
  - (String) language – język, w którym ma zostać zwrócony ewentualny błąd
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista wszystkich miejsc, gdzie obiekt miejsca posiada następujące parametry:
      - (long) id – ID miejsca w bazie danych
      - (String) name – nazwa miejsca
      - (String) address – adres miejsca
      - (String) gps – współrzędne miejsca
      - (String) description – opis miejsca
      - (boolean) accepted – czy miejsce zostało zaakceptowane
      - (boolean) active – czy miejsce jest „aktywne”, czyli czy np. dalej jest otwarte
      - (long) userId – ID użytkownika, który dodał do bazy dane miejsce
      - (String) username – nazwa użytkownika, który dodał do bazy dane miejsce
- Odpowiedź błędu
  - 403 Forbidden
    - Powody błędu
      - Token który został podany, zawiera dane użytkownika niebędącego administratorem
  - 409 Conflict
    - Powody błędu
      - Nie znaleziono miejsca o podanym ID

### 3.2.3. Edycja miejsca

- Metoda
  - PUT
- URL
  - /admin/place/edit/{language}
- Parametry URL
  - (String) language – język, w którym zostanie zwrócony ewentualny błąd
- Parametry przekazywane w ciele żądania
  - (long) id – ID miejsca w bazie danych
  - (String) name – nazwa miejsca
  - (String) address – adres miejsca
  - (String) gps – współrzędne miejsca
  - (String) description – opis miejsca
  - (boolean) accepted – czy miejsce zostało zaakceptowane
  - (boolean) active – czy miejsce jest „aktywne”, czyli czy np. dalej jest otwarte
  - (long) userId – ID użytkownika, który dodał do bazy dane miejsce
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista wszystkich miejsc, gdzie obiekt miejsca posiada następujące parametry:
      - (long) id – ID miejsca w bazie danych
      - (String) name – nazwa miejsca
      - (String) address – adres miejsca
      - (String) gps – współrzędne miejsca
      - (String) description – opis miejsca
      - (boolean) accepted – czy miejsce zostało zaakceptowane
      - (boolean) active – czy miejsce jest „aktywne”, czyli czy np. dalej jest otwarte
      - (long) userId – ID użytkownika, który dodał do bazy dane miejsce
      - (String) username – nazwa użytkownika, który dodał do bazy dane miejsce
- Odpowiedź błędu
  - 403 Forbidden
    - Powody błędu
      - Token który został podany, zawiera dane użytkownika niebędącego administratorem
  - 409 Conflict
    - Powody błędu
      - Nie znaleziono miejsca o podanym ID
      - Nie znaleziono użytkownika o podanym ID
      - Miejsce o identycznej nazwie i adresie już istnieje w bazie danych
      - Miejsce o identycznej nazwie i współrzędnych GPS już istnieje w bazie danych

## 4. Wizyty

### 4.1. Funkcje użytkownika

#### 4.1.1. Pobranie listy odbytych wizyt

- Metoda
  - GET
- URL
  - /visit/myVisitedPlaces/{userId}
- Parametry URL
  - (long) userId – ID użytkownika w bazie danych
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista odbytych wizyt, gdzie obiekt wizyty posiada następujące parametry:
      - (long) id – ID wizyty w bazie danych
      - (Timestamp) date – data kiedy odbyła się wizyta (w tym przypadku zawsze będzie null)
      - (boolean) visited – czy wizyta się odbyła (w tym przypadku zawsze false)
      - (boolean) visible – czy wizyta jest widoczna (w tym przypadku zawsze true)
      - (long) userId – ID użytkownika w bazie danych
      - (long) placeId – ID miejsca w bazie danych
- Odpowiedź błędu
  - 403 Forbidden
    - Przyczyny błędu
      - ID użytkownika podane jako parametr i ID wyłuskane z tokenu nie zgadzają się

#### 4.1.2. Pobranie listy nieodbytych wizyt

- Metoda
  - GET
- URL
  - /visit/myNotVisitedPlaces/{userId}
- Parametry URL
  - (long) userId – ID użytkownika w bazie danych
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista nieodbytych wizyt, gdzie obiekt wizyty posiada następujące parametry:
      - (long) id – ID wizyty w bazie danych
      - (Timestamp) date – data kiedy odbyła się wizyta (w tym przypadku zawsze będzie null)
      - (boolean) visited – czy wizyta się odbyła (w tym przypadku zawsze false)
      - (boolean) visible – czy wizyta jest widoczna (w tym przypadku zawsze true)
      - (long) userId – ID użytkownika w bazie danych
      - (long) placeId – ID miejsca w bazie danych
- Odpowiedź błędu
  - 403 Forbidden
    - Przyczyny błędu
      - ID użytkownika podane jako parametr i ID wyłuskane z tokenu nie zgadzają się

#### 4.1.3. Dodanie nowej wizyty

- Metoda
  - POST
- URL
  - /visit/add/{language}
- Parametry URL
  - (String) language – język, w którym ma zostać przekazany ewentualny błąd
- Parametry przekazywane w ciele żądania
  - (long) userId – ID użytkownika w bazie danych
  - (long) placeId – ID miejsca w bazie danych
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista nieodbytych wizyt, gdzie obiekt wizyty posiada następujące parametry:
      - (long) id – ID wizyty w bazie danych
      - (Timestamp) date – data kiedy odbyła się wizyta (w tym przypadku zawsze będzie null)
      - (boolean) visited – czy wizyta się odbyła (w tym przypadku zawsze false)
      - (boolean) visible – czy wizyta jest widoczna (w tym przypadku zawsze true)
      - (long) userId – ID użytkownika w bazie danych
      - (long) placeId – ID miejsca w bazie danych
- Odpowiedź błędu
  - 403 Forbidden
    - Przyczyny błędu
      - ID użytkownika przekazana jako parametr w ciele żądania i ID wyłuskane z tokenu nie zgadzają się
  - 409 Conflict
    - Przyczyny błędu
      - Nie znaleziono miejsca o podanym ID
      - W bazie znaleziono już nieodbyłą wizytę w danym miejscu



#### 4.1.4. Usunięcie nieodbytej wizyty

- Metoda
  - DELETE
- URL
  - /visit/deleteNotVisitedPlace/{visitId}/{userId}/{language}
- Parametry URL
  - (long) visitId – ID wizyty w bazie danych
  - (long) userId – ID użytkownika w bazie danych
  - (String) language – język, w którym ma zostać przekazany ewentualny błąd
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista nieodbytych wizyt, gdzie obiekt wizyty posiada następujące parametry:
      - (long) id – ID wizyty w bazie danych
      - (Timestamp) date – data kiedy odbyła się wizyta (w tym przypadku zawsze będzie null)
      - (boolean) visited – czy wizyta się odbyła (w tym przypadku zawsze false)
      - (boolean) visible – czy wizyta jest widoczna (w tym przypadku zawsze true)
      - (long) userId – ID użytkownika w bazie danych
      - (long) placeId – ID miejsca w bazie danych
- Odpowiedź błędu
  - 403 Forbidden
    - Przyczyny błędu
      - ID użytkownika przekazana jako parametr URL i ID wyłuskane z tokenu nie zgadzają się
  - 409 Conflict
    - Przyczyny błędu
      - Nie znaleziono wizyty o podanym ID
      - Wizyta została już oznaczona jako odbyta

#### 4.1.5. Oznaczenie wizyty jako „odbyta”

- Metoda
  - PUT
- URL
  - /visit/setAsVisited/{language}
- Parametry URL
  - (String) language – język, w którym ma zostać zwrócony ewentualny błąd
- Parametry przekazywane w ciele żądania
  - (Timestamp) date – data, kiedy odbyła się wizyta
  - (long) visitId – ID wizyty w bazie danych
  - (long) userId – ID użytkownika w bazie danych
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista odbytych wizyt, gdzie obiekt wizyty posiada następujące parametry:
      - (long) id – ID wizyty w bazie danych
      - (Timestamp) date – data kiedy odbyła się wizyta (w tym przypadku zawsze będzie null)
      - (boolean) visited – czy wizyta się odbyła (w tym przypadku zawsze false)
      - (boolean) visible – czy wizyta jest widoczna (w tym przypadku zawsze true)
      - (long) userId – ID użytkownika w bazie danych
      - (long) placeId – ID miejsca w bazie danych
- Odpowiedź błędu
  - 403 Forbidden
    - Przyczyny błędu
      - ID użytkownika przekazany jako parametr przekazany w ciele żądania i ID wyłuskane z tokenu nie zgadzają się
      - ID użytkownika przekazany jako parametr przekazany w ciele żądania i ID użytkownika przypisane do wizyty, której ID zostało przekazane w ciele żądania nie zgadzają się
      - ID użytkownika przypisany do wizyty, której ID zostało przekazane w ciele żądania i ID wyłuskane z tokenu nie zgadzają się
  - 409 Conflict
    - Przyczyny błędu
      - Wizyta została już oznaczona jako odbyta
      - Nie odnaleziono wizyty w bazie danych

#### 4.1.6. Wyczyszczenie historii odbytych wizyt

- Metoda
  - PUT
- URL
  - /visit/clearHistory/{userId}
- Parametry URL
  - (long) userId – ID użytkownika w bazie danych
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość
      - Pusta lista odbytych wizyt
- Odpowiedź błędu
  - 403 Forbidden
    - Przyczyny błędu
      - ID użytkownika przekazany jako parametr URL i ID wyłuskane z tokenu nie zgadzają się

#### 4.1.7. Usunięcie z historii pojedynczej odbytej wizyty

- Metoda
  - PUT
- URL
  - /visit/hideVisitedPlace/{visitId}/{userId}/{language}
- Parametry URL
  - (long) visitId – ID wizyty w bazie danych
  - (long) userId – ID użytkownika w bazie danych
  - (String) language – język, w którym będzie zwrócony ewentualny błąd
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista odbytych wizyt, gdzie obiekt wizyty posiada następujące parametry:
      - (long) id – ID wizyty w bazie danych
      - (Timestamp) date – data kiedy odbyła się wizyta (w tym przypadku zawsze będzie null)
      - (boolean) visited – czy wizyta się odbyła (w tym przypadku zawsze false)
      - (boolean) visible – czy wizyta jest widoczna (w tym przypadku zawsze true)
      - (long) userId – ID użytkownika w bazie danych
      - (long) placeId – ID miejsca w bazie danych
- Odpowiedź błędu
  - 403 Forbidden
    - Przyczyny błędu
      - ID użytkownika przekazany jako parametr URL i ID wyłuskane z tokenu nie zgadzają się
      - ID użytkownika przekazany jako parametr URL i ID użytkownika przypisane do wizyty, której ID zostało przekazane w ciele żądania nie zgadzają się
      - ID użytkownika przypisany do wizyty, której ID zostało przekazane jako parametr URL i ID wyłuskane z tokenu nie zgadzają się
  - 409 Conflict
    - Przyczyny błędu
      - Nie odnaleziono wizyty o podanym ID
      - Wizyta podana o podanym ID nie została oznaczona jako odbyta

## 5. Komentarze

### 5.1. Funkcje użytkownika

#### 5.1.1. Dodanie komentarza

- Metoda
  - POST
- URL
  - /comment/add/{userId}/{language}
- Parametry URL
  - (long) userId – ID użytkownika w bazie danych
  - (String) language – język, w którym ma zostać zwrócony ewentualny błąd
- Parametry przekazywane w ciele żądania
  - (String) text – treść komentarza
  - (boolean) recommended – czy komentujący poleca dane miejsce
  - (boolean) active – czy komentarz jest widoczny – ustawiona wartość nie ma znaczenia w tym przypadku, ponieważ i tak zostanie ustawiona na true przed dodaniem do bazy
  - (long) visitId – ID wizyty w bazie danych
- Odpowiedź powodzenia
  - 200 OK
- Odpowiedź błędu
  - 403 Forbidden
    - Przyczyny błędu
      - ID użytkownika podane jako parametr URL i ID użytkownika wyłuskane z tokenu nie zgadzają się
      - ID użytkownika podane jako parametr URL i ID użytkownika przypisanego do wizyty, której ID zostało przekazane w ciele żądania nie zgadzają się
      - ID użytkownika przypisanego do wizyty, której ID zostało przekazane w ciele żądania oraz ID użytkownika wyłuskane z tokenu nie zgadzają się
  - 409 Conflict
    - Nie odnaleziono wizyty o podanym ID
    - Wizyta została już skomentowana

### 5.1.2. Edycja komentarza

- Metoda
  - POST
- URL
  - /comment/edit/{language}
- Parametry URL
  - (String) language – język, w którym ma zostać zwrócony ewentualny błąd
- Parametry przekazywane w ciele żądania
  - (long) id – ID komentarza w bazie danych
  - (String) text – treść komentarza
  - (boolean) recommended – czy komentujący poleca dane miejsce
  - (boolean) active – czy komentarz jest widoczny
  - (long) visitId – ID wizyty w bazie danych, do której przypisany jest komentarz
- Odpowiedź powodzenia
  - 200 OK
- Odpowiedzi błędu
  - 403 Forbidden
    - Przyczyny błędu
      - ID użytkownika wyluskane z tokenu nie zgadza się z ID użytkownika przypisanego do wizyty, której dotyczy komentarz przekazany w ciele żądania
      - ID komentarza przekazanego w ciele żądania nie zgadza się z ID komentarza znajdującego się w bazie danych i przypisanego do wizyty, której ID zostało przekazane w ciele żądania (jako visitID)
  - 409 Conflict
    - Przyczyny błędu
      - Do wizyty, której ID zostało przekazane w ciele żądania nie jest przypisany żaden komentarz

### 5.1.3. Usunięcie komentarza

- Metoda
  - DELETE
- URL
  - /comment/delete/{userId}/{visitId}/{language}
- Parametry URL
  - (long) userId – ID użytkownika w bazie danych, który „jest właścicielem” wizyty o ID podanym jako parametr URL
  - (long) visitId – ID wizyty, do której przypisany jest komentarz, który ma zostać usunięty
  - (String) language – język, w którym ma zostać zwrócony ewentualny błąd
- Odpowiedź powodzenia
  - 200 OK
- Odpowiedź błędu
  - 403 Forbidden
    - ID użytkownika podane jako parametr URL i ID użytkownika wyluskane z tokenu nie zgadzają się
    - ID użytkownika podane jako parametr URL i ID użytkownika przypisanego do wizyty nie zgadzają się
  - 409 Conflict
    - Wizyta o ID podanym jako parametr URL nie została skomentowana
    - Wizyta o ID podanym jako parametr URL nie została znaleziona w bazie danych

#### 5.1.4. Pobranie komentarza do wizyty

- Metoda
  - GET
- URL
  - /comment/findByVisitId/{userId}/{visitId}
- Parametry URL
  - (long) userId – ID użytkownika w bazie danych, który „jest właścicielem” wizyty o ID podanym jako parametr URL
  - (long) visitId – ID wizyty, do której przypisany jest komentarz, który ma zostać usunięty
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Komentarz w postaci:
      - (long) id – ID komentarza w bazie danych
      - (String) text – treść komentarza
      - (boolean) recommended – czy miejsce jest polecane przez komentującego
      - (Timestamp) visitDate – data odwiedzin miejsca
      - (boolean) active – czy komentarz jest widoczny (nie został ukryty przez administratora)
      - (String) username – nazwa użytkownika, który dodał do bazy dany komentarz
      - (long) userId – ID użytkownika, który dodał do bazy dany komentarz
      - (long) visitId – ID wizyty, do której przypisany jest komentarz
  - 200 OK
    - Pusta wiadomość w przypadku, gdy wizyta nie została skomentowana
- Odpowiedź błędu
  - 403 Forbidden
    - ID użytkownika podane jako parametr URL i ID użytkownika wyłuskane z tokenu nie zgadzają się
    - ID użytkownika podane jako parametr URL i ID użytkownika przypisanego do wizyty nie zgadzają się

#### 5.1.5. Pobranie komentarzy dla wybranego miejsca

- Metoda
  - GET
- URL
  - /comment/allByPlaceId/{placeId}
- Parametry URL
  - (long) placeId – ID miejsca w bazie danych
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista wszystkich komentarzy przypisanych do danego miejsca, gdzie obiekt miejsca posiada następujące parametry:
      - (long) id – ID komentarza w bazie danych
      - (String) text – treść komentarza
      - (boolean) recommended – czy miejsce jest polecane przez komentującego
      - (Timestamp) visitDate – data odwiedzin miejsca
      - (boolean) active – czy komentarz jest widoczny (nie został ukryty przez administratora) – w tym przypadku zawsze true
      - (String) username – nazwa użytkownika, który dodał do bazy dany komentarz
      - (long) userId – ID użytkownika, który dodał do bazy dany komentarz
      - (long) visitId – ID wizyty, której dotyczy komentarz

## 5.2. Funkcje administratora

### 5.2.1. Ukrycie komentarza

- Metoda
  - POST
- URL
  - /admin/comment/hide/{visitId}/{language}
- Parametry URL
  - (long) visitId – ID wizyty w bazie danych, do której przypisany jest komentarz
  - (String) language – język, w którym ma zostać zwrócony ewentualny błąd
- Odpowiedź powodzenia
  - 200 OK
- Odpowiedź błędu
  - 403 Forbidden
    - Token który został podany, zawiera dane użytkownika niebędącego administratorem
  - 409 Conflict
    - Nie odnaleziono wizyty o podanym ID
    - Do wizyty o podanym ID nie został przypisany komentarz

### 5.2.2. Przywrócenie komentarza

- Metoda
  - POST
- URL
  - /admin/comment/show/{visitId}/{language}
- Parametry URL
  - (long) visitId – ID wizyty w bazie danych, do której przypisany jest komentarz
  - (String) language – język, w którym ma zostać zwrócony ewentualny błąd
- Odpowiedź powodzenia
  - 200 OK
- Odpowiedź błędu
  - 403 Forbidden
    - Token który został podany, zawiera dane użytkownika niebędącego administratorem
  - 409 Conflict
    - Nie odnaleziono wizyty o podanym ID
    - Do wizyty o podanym ID nie został przypisany komentarz

### 5.2.3. Pobranie wszystkich komentarzy dla wybranego miejsca (również nieaktywnych)

- Metoda
  - GET
- URL
  - /admin/comment/allByPlaceId/{placeId}
- Parametry URL
  - (long) placeId – ID miejsca w bazie danych
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista wszystkich komentarzy przypisanych do danego miejsca, gdzie obiekt miejsca posiada następujące parametry:
      - (long) id – ID komentarza w bazie danych
      - (String) text – treść komentarza
      - (boolean) recommended – czy miejsce jest polecane przez komentującego
      - (Timestamp) visitDate – data odwiedzin miejsca
      - (boolean) active – czy komentarz jest widoczny (nie został ukryty przez administratora)
      - (String) username – nazwa użytkownika, który dodał do bazy dany komentarz
      - (long) userId – ID użytkownika, który dodał do bazy dany komentarz
      - (long) visitId – ID wizyty, której dotyczy komentarz
- Odpowiedzi błędu
  - 403 Forbidden
    - Powody błędu
      - Token który został podany, zawiera dane użytkownika niebędącego administratorem



## 6. Zdjęcia

### 6.1. Uwagi

Celem przyspieszenia działania komunikacji między BackEndem, a FrontEndem, sam plik zdjęcia jest przechowywany w osobnej tabeli, co informacje o tym zdjęciu. Dzięki temu, gdy ktoś wejdzie w zakładkę ze wszystkimi zdjęciami danego miejsca, aplikacja nie będzie musiała wysłać w żądaniu wszystkich zdjęć na raz, a jedynie informacje o tych zdjęciach, a sam plik zostanie wysłany dopiero gdy ktoś będzie chciał obejrzeć to jedno, konkretne zdjęcie. Przez to w tej części dokumentacji będę posługiwał się terminami „plik zdjęcia” oraz „szczegóły zdjęcia”. W przypadku, gdy ktoś będzie wysyłał nowe zdjęcie do dodania, to wtedy będzie wysyłał wszystko jednym JSONem, a backEnd sobie z tym poradzi.

### 6.2. Funkcje użytkownika

#### 6.2.1. Pobranie listy szczegółów zdjęć dla danego miejsca

- Metoda
  - GET
- URL
  - /photo/allByPlaceId/{placeId}
- Parametry URL
  - (long) placeId – ID miejsca w bazie danych
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista szczegółów zdjęć, gdzie obiekt szczegółów zdjęcia posiada atrybuty:
      - (long) id – ID obiektu szczegółów zdjęcia w bazie danych
      - (Timestamp) date – data zrobienia zdjęcia
      - (boolean) accepted – czy zdjęcie zostało zaakceptowane przez administratora
      - (long) userId – ID użytkownika, który przesłał dane zdjęcie
      - (long) placeId – ID miejsca, które przedstawia dane zdjęcie
      - (String) username – nazwa użytkownika, który przesłał dane zdjęcie

#### 6.2.2. Pobranie danego pliku zdjęcia

- Metoda
  - GET
- URL
  - /photo/file/{photoId}/{language}
- Parametry URL
  - (long) photoId – ID obiektu szczegółów zdjęcia, które opisują plik zdjęcia, który chcemy pobrać
  - (String) language – język, w którym ma zostać zwrócony ewentualny błąd
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Obiekt pliku zdjęcia, który zawiera następujące atrybuty
      - (long) id – ID obiektu pliku zdjęcia w bazie danych
      - (byte[]) file – tablica bajtów reprezentująca plik zdjęcia
- Odpowiedź błędu
  - 409 Conflict
    - Przyczyny błędu
      - Nie odnaleziono obiektu szczegółów zdjęcia o podanym ID
      - Do danego obiektu szczegółów zdjęcia o podanym ID nie ma przypisanego obiektu pliku zdjęcia

### 6.2.3. Dodanie nowego zdjęcia

- Metoda
  - POST
- URL
  - /photo/add
- Parametry przekazywane w ciele żądania
  - (byte[]) file – tablica bajtów reprezentująca plik zdjęcia
  - (Timestamp) date – data zrobienia zdjęcia
  - (long) userId – ID użytkownika w bazie danych, który wykonał zdjęcie
  - (long) placeId – ID miejsca w bazie danych, które zdjęcie przedstawia
- Odpowiedź powodzenia
  - 200 OK
- Odpowiedź błędu
  - 403 Forbidden
    - Przyczyny błędu
      - ID użytkownika podane jako parametr przekazany w ciele żądania i ID użytkownika wyłuskane z tokenu nie zgadzają się

## 6.3. Funkcje administratora

### 6.3.1. Pobranie listy szczegółów wszystkich niezaakceptowanych zdjęć

- Metoda
  - GET
- URL
  - /admin/photo/allNotAccepted
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista szczegółów zdjęć niezaakceptowanych przez administratora, gdzie obiekt szczegółów zdjęcia posiada atrybuty:
      - (long) id – ID obiektu szczegółów zdjęcia w bazie danych
      - (Timestamp) date – data zrobienia zdjęcia
      - (boolean) accepted – czy zdjęcie zostało zaakceptowane przez administratora
      - (long) userId – ID użytkownika, który przesłał dane zdjęcie
      - (long) placeId – ID miejsca, które przedstawia dane zdjęcie
      - (String) username – nazwa użytkownika, który przesłał dane zdjęcie
- Odpowiedź błędu
  - 403 Forbidden
    - Przyczyny błędu
      - Token który został podany, zawiera dane użytkownika niebędącego administratorem

### 6.3.2. Akceptacja zdjęcia

- Metoda
  - PUT
- URL
  - /admin/photo/accept/{photoId}/{language}
- Parametry URL
  - (long) photoId – ID obiektu szczegółów zdjęcia w bazie danych
  - (String) language – język, w którym ma zostać zwrócony ewentualny błąd
- Odpowiedź powodzenia
  - 200 OK
    - Zawartość: Lista szczegółów zdjęć niezaakceptowanych przez administratora, gdzie obiekt szczegółów zdjęcia posiada atrybuty:
      - (long) id – ID obiektu szczegółów zdjęcia w bazie danych
      - (Timestamp) date – data zrobienia zdjęcia
      - (boolean) accepted – czy zdjęcie zostało zaakceptowane przez administratora
      - (long) userId – ID użytkownika, który przesłał dane zdjęcie
      - (long) placeId – ID miejsca, które przedstawia dane zdjęcie
      - (String) username – nazwa użytkownika, który przesłał dane zdjęcie
- Odpowiedź błędu
  - 403 Forbidden
    - Przyczyny błędu
      - Token który został podany, zawiera dane użytkownika niebędącego administratorem
  - 409 Conflict
    - Przyczyny błędu
      - Nie odnaleziono obiektu szczegółów zdjęcia w bazie danych

### 6.3.3. Usunięcie zdjęcia

- Metoda
  - DELETE
- URL
  - /admin/photo/delete/{photoId}/{language}
- Parametry URL
  - (long) photoId – ID obiektu szczegółów zdjęcia w bazie danych
  - (String) language – język, w którym ma zostać zwrócony ewentualny błąd
- Odpowiedź powodzenia
  - 200 OK
- Odpowiedź błędu
  - 403 Forbidden
    - Przyczyny błędu
      - Token który został podany, zawiera dane użytkownika niebędącego administratorem
  - 409 Conflict
    - Przyczyny błędu
      - Nie odnaleziono obiektu szczegółów zdjęcia w bazie danych