

Japankitty - API

Joanna Janiszewska Piotr Halama Karolina Mucha

24 stycznia 2018

Spis treści

1	Słowem wstępu	2
2	Słowa i kursy	3
2.1	Lista kursów	3
2.2	Słownictwo	3
2.3	Kanji	3
2.4	Definicje	3
2.5	Wiadomość dnia	3
3	Obsługa użytkowników	4
3.1	Rejestracja	4
3.2	Logowanie	4
3.3	Wylogowanie	5
3.4	Informacje o użytkowniku	5
3.5	Edycja danych użytkownika	5
3.6	Pobieranie danych użytkowników	6
3.7	Edycja statusu użytkownika	7
4	Teksty	7
4.1	Pobieranie listy tekstów	7
4.2	Dodawanie tekstu	8
4.3	Edycja tekstu	9
4.4	Usuwanie tekstu	9

1 Słowem wstępu

Niniejsza dokumentacja została stworzona, aby umożliwić sprawne i wygodne korzystanie z API serwisu Japankitty.

2 Słowa i kursy

Niniejsza część dokumentacji poświęcona jest uzyskiwaniu listy kursów oraz rekordów powiązanych z danym kursem z tabel Vocabulary, Kanji oraz Definitions.

2.1 Lista kursów

Lista kursów jest dostępna pod punktem końcowym

```
/api/courses/
```

Zwracana jest tablica JSON zawierająca obiekty z polami id, file, name oraz type. Id to numer kursu w bazie, file to krótka nazwa używana w innych żądaniach, name jest nazwą przyjazną użytkownikowi, a type definiuje jakiego typu jest to kurs.

2.2 Słownictwo

Lista słów jest dostępna pod punktem końcowym

```
/api/vocabulary/<nazwa kursu>
```

Zwracana jest tablica JSON zawierająca obiekty z polami id, kana, kanji oraz polish.

2.3 Kanji

Lista słów jest dostępna pod punktem końcowym

```
/api/kanji/<nazwa kursu>
```

Zwracana jest tablica JSON zawierająca obiekty z polami id, kanji, kunyomi oraz onyomi.

2.4 Definicje

Lista słów jest dostępna pod punktem końcowym

```
/api/definitions/<nazwa kursu>
```

Zwracana jest tablica JSON zawierająca obiekty z polami id, word oraz definition.

2.5 Wiadomość dnia

Wiadomość dnia jest dostępna pod punktem końcowym

```
/api/motd
```

Zwracany jest tekst wiadomości dnia.

3 Obsługa użytkowników

3.1 Rejestracja

Rejestracja użytkowników dostępna jest pod punktem końcowym

`/api/register/`

Rejestracja użytkowników wymaga ustawienia nagłówków “Accept: application/json”, “Content-Type: application/json” oraz wysłania żądania JSON w następującym formacie:

```
{
    "name": "username",
    "email": "user_email",
    "password": "pass"
}
```

Zwracany jest JSON w następującym formacie:

```
{
    "success": {
        "token": "oauth_token",
        "name": "username",
        "role": "role"
    }
}
```

Gdzie token to token autoryzacji OAuth używany m.in. dla żądań wylogowania czy pozyskania informacji o użytkowniku.

3.2 Logowanie

Logowanie dostępne jest pod punktem końcowym

`/api/login/`

Logowanie użytkowników wymaga ustawienia nagłówków “Accept: application/json”, “Content-Type: application/json” oraz wysłania żądania JSON w następującym formacie:

```
{
    "email": "user_email",
    "password": "pass"
}
```

Zwracany jest JSON w następującym formacie:

```
{
    "success": {
        "token": "oauth_token",
        "role": "role"
    }
}
```

Gdzie token to token autoryzacji OAuth używany m.in. dla żądań wylogowania czy pozyskania informacji o użytkowniku.

3.3 Wylogowanie

Wylogowanie dostępne jest pod punktem końcowym

`/api/logout/`

Wylogowanie użytkowników wymaga ustawienia nagłówków “Accept: application/json”, “Authorization: Bearer oauth_token”, gdzie `oauth_token` to token pozyskany z API logowania lub rejestracji użytkownika. Przykładowy nagłówek Authorization może mieć postać

`Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh(...)tdrVCHAujAc`

Zwracany jest JSON w następującym formacie:

```
{
    "success": "success"
}
```

Użytkownik jest wylogowywany ze wszystkich aktywnych sesji jednocześnie.

3.4 Informacje o użytkowniku

Informacje o użytkowniku dostępne są pod punktem końcowym

`/api/get-details/`

Pozyskanie informacji o użytkowniku wymaga ustawienia nagłówków “Accept: application/json”, “Authorization: Bearer oauth_token”, gdzie `oauth_token` to token pozyskany z API logowania lub rejestracji użytkownika. Przykładowy nagłówek Authorization może mieć postać

`Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh(...)tdrVCHAujAc`

Zwracany jest JSON w następującym formacie (dane przykładowe):

```
{
  "success": {
    "id": 3,
    "name": "username",
    "email": "user_email",
    "created_at": "2017-11-14 14:05:08",
    "updated_at": "2017-11-14 14:05:08",
    "sex": "male",
    "surname": "surname",
    "birthday": "1970-01-01",
    "role": "user"
  }
}
```

3.5 Edycja danych użytkownika

Edycja danych użytkownika dostępna jest pod punktem końcowym

`/api/edit-account/`

Edycja danych użytkownika wymaga ustawienia nagłówków "Accept: application/json", "Content-Type: application/json", "Authorization: Bearer oauth_token", gdzie oauth_token to token pozyskany z API logowania lub rejestracji użytkownika. Przykładowy nagłówek Authorization może mieć postać

Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh (...) tdrVCHAujAc

Należy także wysłać JSON z następującymi polami (każde z pól jest opcjonalne, z wyjątkiem hasła: przy zmianie hasła muszą zostać wysłane jednocześnie pola old_password oraz new_password):

```
{
  "new_name": "new_username",
  "new_email": "new_user_email",
  "old_password": "old_pass",
  "new_password": "new_pass",
  "new_sex": "new_sex",
  "new_surname": "new_surname",
  "new_birthday": "new_birthday"
}
```

Zwracany jest JSON w następującej formie:

```
{
  "success": "success"
}
```

3.6 Pobieranie danych użytkowników

Pobieranie danych użytkowników dostępne jest pod punktem końcowym

/api/users/

Pobieranie danych użytkowników wymaga ustawienia nagłówków "Accept: application/json", "Authorization: Bearer oauth_token", gdzie oauth_token to token pozyskany z API logowania lub rejestracji użytkownika. Przykładowy nagłówek Authorization może mieć postać

Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh (...) tdrVCHAujAc

Użytkownik wysyłający żądanie musi także posiadać rolę admin. Zwracany jest JSON w następującej formie:

```
{
  "success": [
    {
      "id": 3,
      "name": "testowyJa",
      "email": "new2@halamix2.pl",
      "created_at": "2017-11-14 14:05:08",
      "updated_at": "2018-01-06 12:47:25",
      "sex": "unknown",
      "surname": "missing",
      "birthday": "1970-01-01",
      "role": "user"
    }
  ]
}
```

```
    },  
    {  
      "id": 6,  
      "name": "testowyJa2",  
      "email": "lol2@halex2.pl",  
      "created_at": "2018-01-06 12:11:25",  
      "updated_at": "2018-01-06 12:11:25",  
      "sex": "unknown",  
      "surname": "missing",  
      "birthday": "1970-01-01",  
      "role": "user"  
    }  
  ]  
}
```

3.7 Edycja statusu użytkownika

Edycja statusu użytkownika dostępna jest pod punktem końcowym

`/api/change-user-status/`

Edycja statusu użytkownika wymaga ustawienia nagłówków “Accept: application/json”, “Content-Type: application/json”, “Authorization: Bearer oauth_token”, gdzie oauth_token to token pozyskany z API logowania lub rejestracji użytkownika. Przykładowy nagłówek Authorization może mieć postać

Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbnVudCI6IjE2MjE1MjE2NTUwIiwiaWF0IjoiMTY2MjE1MjE2NTUwIn0=

Użytkownik wysyłający żądanie musi także posiadać rolę **admin**.

Należy także wysłać JSON z następującymi polami:

```
{  
  "id": "id",  
  "status": "status"  
}
```

Zwracany jest JSON w następującym formacie:

```
{  
  "success": "success"  
}
```

Użytkownik, którego status został ustawiony na **blocked** jest wylogowywany ze wszystkich sesji i nie ma możliwości ponownego zalogowania.

4 Teksty

4.1 Pobieranie listy tekstów

Pobieranie listy tekstów dostępne jest pod punktem końcowym

`/api/texts/`

Pobieranie listy tekstów wymaga ustawienia nagłówków "Accept: application/json", "Authorization: Bearer oauth_token", gdzie oauth_token to token pozyskany z API logowania lub rejestracji użytkownika. przykładowy nagłówek Authorization może mieć postać

Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh (...) tdrVCHAujAc

Zwracany jest JSON w następującej formie:

```
{
  "success": [
    {
      "id": 3,
      "name": "testowyJa",
      "email": "new2@halamix2.pl",
      "created_at": "2017-11-14 14:05:08",
      "updated_at": "2018-01-06 12:47:25",
      "sex": "unknown",
      "surname": "missing",
      "birthday": "1970-01-01",
      "role": "user"
    },
    {
      "id": 6,
      "name": "testowyJa2",
      "email": "lol2@halamix2.pl",
      "created_at": "2018-01-06 12:11:25",
      "updated_at": "2018-01-06 12:11:25",
      "sex": "unknown",
      "surname": "missing",
      "birthday": "1970-01-01",
      "role": "user"
    }
  ]
}
```

4.2 Dodawanie tekstu

Dodawanie tekstu dostępne jest pod punktem końcowym

/api/users/

Dodawanie tekstu wymaga ustawienia nagłówków "Accept: application/json", "Content-Type: application/json", "Authorization: Bearer oauth_token", gdzie oauth_token to token pozyskany z API logowania lub rejestracji użytkownika. przykładowy nagłówek Authorization może mieć postać

Authorization: Bearer eyJ0eXAiOiJKV1QiLCJh (...) tdrVCHAujAc

Użytkownik wysyłający żądanie musi także posiadać rolę admin.

Należy także wysłać JSON z następującymi polami:

```
{
  "offerDescription": "offer Description",
```

Zwracany jest JSON w następującym formacie:

```
{
  "success": "success"
}
```

4.3 Edycja tekstu

Edycja tekstu dostępne jest pod punktem końcowym

```
/api/users/
```

Edycja tekstu wymaga ustawienia nagłówków “Accept: application/json“, “Content-Type: application/json“, “Authorization: Bearer oauth_token“, gdzie oauth_token to token pozyskany z API logowania lub rejestracji użytkownika. przykładowy nagłówek Authorization może mieć postać

Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbnVudCI6IiJ9.eyJ1b290eXAiOiJKV1QiLCJhbnVudCI6IiJ9.tdrVCHAuJAc

Użytkownik wysyłający żądanie musi także posiadać rolę `admin`.

Należy także wysłać JSON z następującymi polami:

```
{
    "id": "id",
    "offerDescription": "offer Description",
    "offerOption1": "offerOption",
    "offerOption2": "offerOption",
    "offerOption3": "offerOption",
    "contactDescription": "contactDescription",
    "aboutCourse": "aboutCourse"
}
```

Zwracany jest JSON w następującym formacie:

```
{
  "success": "success"
}
```

4.4 Usuwanie tekstu

Edycja tekstu dostępne jest pod punktem końcowym

```
/api/users/
```

Usuwanie tekstu wymaga ustawienia nagłówków “Accept: application/json“, “Content-Type: application/json“, “Authorization: Bearer oauth_token“, gdzie oauth_token to token pozyskany z API logowania lub rejestracji użytkownika. przykładowy nagłówek Authorization może mieć postać

