
DOKUMENTACJA API PROJEKTU

„KOMPUTEROWA ANALIZA DYSKUSJI

NA FORUM”

CONTENTS

1. Wstęp.....	2
2. Uwierzytelnienie	2
3. Format danych.....	2
4. Kody błędów.....	2
5. Endpointy	2
api/user/.....	2
api/login/.....	3
api/file/.....	4
api/file/<int:id>.....	5
api/discussionList/<int:id>.....	6
api/discussions/<int:file_id>/<int:dimension_id>.....	6
api/visualize/<int:discussion_id>.....	7
api/dimension/.....	8
api/dimension/<int:id>.....	10
api/file/dimension/.....	11
api/file/dimension/<int:file_id>.....	12
api/file/dimension/<int:file_id>/<int:dimension_id>.....	13
api/file/category/<int:dimension_id>/<int:category_id>.....	13
api/label/.....	14
api/classifier/<int:id>.....	15
api/argClassifier/<int:id>.....	16
api/export/.....	16

1. WSTĘP

Niniejszy dokument opisuje model REST API backendu projektu Narzędzia do komputerowej analizy dyskusji na forum (NKADF). Definiuje endpointy API, format zapytań oraz odpowiedzi, niezbędne nagłówki oraz kod statusu.

2. UWIERZYTELNIENIE

API projektu NKADF wykorzystuje JWT (JSON Web Token) w celu uwierzytelnienia użytkownika wysyłającego zapytanie. Token uzyskuje się w wyniku logowania. W celu przeprowadzenia operacji CRUD poprzez udostępnione endpointy, niezbędne jest umieszczenie ważnego tokenu w nagłówku zapytania HTTP do API.

3. FORMAT DANYCH

API projektu NKADF przeważnie zwraca odpowiedź w formacie JSON reprezentującą stan zasobów pobranych z bazy danych. Zapytania HTTP, których celem jest edycja zasobów bazy danych, wymagają umieszczenia odpowiednich parametrów w formacie JSON w ciele zapytania.

4. KODY BŁĘDÓW

Wykorzystywane są standardowe kody błędów HTTP.

5. ENDPOINTY

API/USER/

Atrybuty:

- email (string) – adres email nowego użytkownika
- password (string) – hasło nowego użytkownika

Stworzenie nowego użytkownika [POST]:

W celu stworzenia nowego użytkownika należy wysłać JSON-a z wymaganymi atrybutami umieszczonymi w ciele zapytania.

- Zapytanie:

```
{
  "email": "user@test.com",
  "password": "password"
}
```

- Poprawna odpowiedź z serwera:

```
{
  "message": "User created successfully",
  "statusCode": 200
}
```

- Odpowiedź, gdy użytkownik o takim emailu już istnieje:

```
{
  "message": "User already exists",
  "statusCode": 409
}
```

API/LOGIN/

Atrybuty:

- email (string) – adres istniejącego użytkownika
- password(string) – hasło istniejącego użytkownika
- token (string) – token uwierzytelniający użytkownika, uzyskany po pomyślnym zalogowaniu
- exp (int) – czas w postaci UNIX-owej informujący, kiedy token ulega unieważnieniu

Logowanie na istniejące konto użytkownika [POST]:

W celu zalogowania się na istniejące konto użytkownika, należy wysłać zapytanie, umieszczając w ciele zapytania niezbędne parametry w formacie JSON.

- Zapytanie:

```
{
  "email": "user@test.com",
  "password": "password"
}
```

- Poprawna odpowiedź z serwera:

```
{
  "message": "User authenticated successfully",
  "statusCode": 200,
  "token": "token uwierzytelniający",
  "exp": "czas do unieważnienia tokenu"
}
```

- Odpowiedź, gdy podane dane są nieprawidłowe zwracany jest kod 401 oraz komunikat błędu, przykładowo:

```
{
  "message": "Error: User authentication failed",
  "statusCode": 401
}
```

- Odpowiedź, gdy nie udało się spełnić żądania:

```
{
  "message": "Error: Server couldn't process request",
  "statusCode": 406
}
```

- Odpowiedź, gdy token uwierzytelniający jest nieważny:

```
{
  "message": "Error: Token is blacklisted",
  "statusCode": 409
}
```

API/FILE/

Atrybuty:

- title (string) – nazwa identyfikująca przesyłany plik
- id (int) – identyfikator pliku w bazie danych
- file (string) – nazwa pliku w bazie danych
- uploaded_at (datetime) – czas przesłania pliku, dodawane automatycznie do bazy danych

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Przesłanie pliku forum na serwer [POST]:

W celu przesłania wyeksportowanego z platformy Moodle pliku forum w formacie XML, należy załączyć plik jako form-data w kluczu "file". Opcjonalnie można dodać klucz "title".

- Poprawna odpowiedź z serwera:

```
{
  "message": "File uploaded successfully",
  "statusCode": 200
}
```

- Odpowiedź, jeśli nie przesłano pliku:

```
{
  "message": "Error: File is missing",
  "statusCode": 422
}
```

- Odpowiedź, gdy przesłany plik jest niepoprawny:

```
{
  "message": "Error: Server couldn't process request",
  "statusCode": 406
}
```

Wyświetlenie listy plików przesłanych przez użytkownika [GET]:

W celu odebrania informacji o aktualnie przesłanych przez użytkownika plikach należy wysłać zapytanie GET oraz uwzględnić token uwierzytelniający w nagłówku zapytania.

- Odpowiedź:

```
{
  "files": [
    {
      "id": 1,
      "file": "nazwa pliku",
      "uploaded_at": "czas przesłania pliku"
    }
  ]
  "statusCode": 406
}
```

API/FILE/<INT:ID>

Atrybuty:

- id (int) – identyfikator pliku w bazie danych

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Usuwanie pliku z bazy danych [DELETE]:

Aby usunąć plik z bazy danych należy wskazać jego id w adresie URL zapytania oraz umieścić token uwierzytelniający w nagłówku.

- Odpowiedź:

```
{
  "message": "File has been deleted successfully",
  "statusCode": 200
}
```

API/DISCUSSIONLIST/<INT:ID>

Atrybuty:

- id (int) – identyfikator pliku

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Wyświetlenie listy dyskusji [GET]:

Aby wyświetlić listę dyskusji zawartych w pliku, należy wskazać identyfikator pliku w adresie URL zapytania HTTP.

- Odpowiedź:

```
{
  "id": 1,
  "name": "nazwa forum",
  "discussions": [
    {
      "id": 1,
      "title": "tytuł dyskusji"
    }
  ]
}
```

API/DISCUSSIONS/<INT:FILE_ID>/<INT:DIMENSION_ID>

Atrybuty:

- file_id (int) – identyfikator pliku
- dimension_id (int) -> identyfikator wymiaru anotacji
- parent (int) -> identyfikator postu nadrzędnego na platformie Moodle pobrany z przesłanego pliku XML
- author (int) -> identyfikator autora na platformie Moodle pobrany z przesłanego pliku XML

- message (string) -> tablica zawierająca post użytkownika z podziałem na paragrafy
- para_id (int) -> identyfikatory paragrafów w bazie danych
- label (string) -> tablica zawierająca kategorie przypisane do paragrafów w zadanym wymiarze anotacji
- user_updated (boolean) -> informacja, czy anotacja została wygenerowana automatycznie, czy została ustawiona ręcznie przez użytkownika

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Wyświetlenie dyskusji w wymiarze anotacji [GET]:

Aby wyświetlić dyskusję w wybranym wymiarze anotacji, należy wskazać identyfikator pliku oraz identyfikator wymiaru w adresie URL zapytania HTTP. W przypadku, gdy identyfikator wymiaru zostanie ustawiony jako 0, w odpowiedzi zostaną pominięte klucze label oraz user_updated.

- Odpowiedź:

```
{
  "posts": [
    {
      "id": 303,
      "parent": 0,
      "author": 2,
      "message": [
        "Czy tworząc nowy system powinniśmy korzystać z najnowszych technologii dostępnych w branży?", "Odpowiedź uzasadnij."
      ],
      "para_id": [ 2195, 2196 ],
      "label": [ "nieistotna", "nieistotna" ],
      "user_updated": [ false, false ]
    }
  ]
}
```

API/VISUALIZE/<INT:DISCUSSION_ID>

Atrybuty:

- discussion_id (int) – identyfikator dyskusji w bazie danych
- author (int) – identyfikator użytkownika pobrany z pliku XML
- id (int) – identyfikator postu pobrany z pliku XML

- parent (int) – identyfikator postu nadrzędnego pobrany z pliku XML. Post o ID wynoszącym 0 jest pierwszym postem w dyskusji.
- message (string) – treść postu pobrana z pliku XML

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Wyświetlenie dyskusji z hierarchią postów [GET]:

Aby wyświetlić dyskusję z informacją o hierarchii postów celem ich wizualizacji na frontendzie, należy wskazać identyfikator dyskusji w adresie URL zapytania HTTP.

- Odpowiedź:

```
{
  "posts": [
    {
      "author": 2,
      "id": 30,
      "parent": 0,
      "message": {
        "Czy tworząc nowy system powinniśmy korzystać z najnowszych technologii dostępnych w branży?", "Odpowiedź uzasadnij."
      }
    }
  ]
}
```

API/DIMENSION/

Atrybuty:

- id (int) – identyfikator wymiaru w bazie danych
- name (string) – nazwa wymiaru
- categories (string) – lista nazw kategorii wymiaru
- description (string) – opis wymiaru
- categories -> id (int) – identyfikator kategorii w bazie danych

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Wyświetlenie listy wymiarów użytkownika [GET]:

Aby wyświetlić listę wymiarów użytkownika należy wykonać zapytanie HTTP z uwzględnieniem tokenu uwierzytelniającego w nagłówku zapytania.

- Odpowiedź:

```
{
  "data": [
    {
      "id": 1,
      "name": "Polaryzacja opinii",
    },
    {
      "id": 2,
      "name": "Schematy argumentacji"
    }
  ]
}
```

Stworzenie nowego wymiaru użytkownika [POST]:

Aby stworzyć nowy wymiar należy wysłać zapytanie HTTP, zamieszczając niezbędne dane w ciele zapytania w formacie JSON oraz uwzględniając token uwierzytelniający w nagłówku zapytania.

- Zapytanie:

```
{
  "name": "Nazwa wymiaru",
  "categories": ["kategoria1", "kategoria2"],
  "description": "Opis wymiaru"
}
```

- Odpowiedź:

```
{
  "message": "Annotation dimension saved successfully",
  "statusCode": 200
}
```

Aktualizacja istniejącego wymiaru użytkownika [PATCH]:

Aby zaktualizować istniejący wymiar należy wysłać zapytanie HTTP, zamieszczając niezbędne dane w ciele zapytania w formacie JSON oraz uwzględniając token uwierzytelniający w nagłówku zapytania.

- Zapytanie:

```
{
  "id": 1,
  "categories": [{ id: 1, "category": "kategoria1"}, { id: 2, "category": "kategoria2"}],
  "new_categories": ["kategoria3", "kategoria4"]
  "description": "Nowy opis wymiaru",
  "new_name": "Nowa nazwa wymiaru"
}
```

- Odpowiedź:

```
{
  "message": "Dimension's description has been updated successfully",
  "statusCode": 200
}
```

API/DIMENSION/<INT:ID>

Atrybuty:

- id (int) – identyfikator wymiaru w bazie danych
- data -> id (int) – identyfikator kategorii
- data -> category (string) – nazwa kategorii wymiaru
- description (string) – opis wymiaru

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Wyświetlenie szczegółów wymiarów użytkownika [GET]:

Aby wyświetlić szczegóły wymiaru użytkownika należy wykonać zapytanie HTTP z uwzględnieniem tokenu uwierzytelniającego w nagłówku zapytania oraz podając identyfikator wymiaru w adresie URL zapytania.

- Odpowiedź:

```
{
  "data": [
    {
      "id": 1,
      "category": "kategoria1",
    },
    {
      "id": 2,
      "category": "kategoria2"
    }
  ],
  "description": "Opis wymiaru",
  "statusCode": 200
}
```

Usunięcie wymiaru użytkownika [DELETE]:

Aby usunąć wymiar użytkownika należy wykonać zapytanie HTTP z uwzględnieniem tokenu uwierzytelniającego w nagłówku zapytania oraz podając identyfikator wymiaru w adresie URL zapytania.

- Odpowiedź:

```
{
  "message": "Dimension removed successfully",
  "statusCode": 200
}
```

API/FILE/DIMENSION/

Atrybuty:

- dimension_id (int) – identyfikator wymiaru w bazie danych
- file_id (int) – identyfikator pliku w bazie danych

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Przypisanie wymiaru użytkownika do pliku [POST]:

Aby dodać wymiaru użytkownika do pliku należy wykonać zapytanie HTTP z uwzględnieniem tokenu uwierzytelniającego w nagłówku zapytania oraz podając niezbędne dane w formacie JSON w ciele zapytania HTTP.

- Zapytanie:

```
{
  "file_id": 1
  "dimension_id": 3
}
```

- Odpowiedź:

```
{
  "message": "Dimesion successfully added to specified file",
  "statusCode": 200
}
```

API/FILE/DIMENSION/<INT:FILE_ID>

Atrybuty:

- file_id (int) – identyfikator pliku w bazie danych
- data -> id (int) – identyfikator wymiaru w bazie danych
- data ->name (string) – nazwa wymiaru w bazie danych

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Wyświetlenie listy wymiarów przypisanych do pliku [GET]:

Aby wyświetlić listę wymiarów użytkownika przypisanych do wskazanego pliku należy wykonać zapytanie HTTP z uwzględnieniem tokenu uwierzytelniającego w nagłówku zapytania oraz podając identyfikator pliku w adresie zapytania.

- Odpowiedź:

```
{
  "data": [
    {
      "id": 1,
      "name": "Polaryzacja opinii",
    },
  ],
}
```

```
{
  "id": 2,
  "name": "Schematy argumentacji"
},
"statusCode": 200
}
```

API/FILE/DIMENSION/<INT:FILE_ID>/<INT:DIMENSION_ID>

Atrybuty:

- file_id (int) – identyfikator pliku w bazie danych
- dimension_id (int) – identyfikator wymiaru w bazie danych

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Usunięcie wymiaru użytkownika z pliku [DELETE]:

Aby usunąć wymiar użytkownika z pliku należy wskazać identyfikator pliku oraz identyfikator wymiaru w adresie zapytania HTTP oraz uwzględnić token uwierzytelniający w nagłówku zapytania.

- Odpowiedź:

```
{
  "message": "Dimension successfully removed from specified file",
  "statusCode": 200
}
```

API/FILE/CATEGORY/<INT:DIMENSION_ID>/<INT:CATEGORY_ID>

Atrybuty:

- dimension_id (int) – identyfikator pliku w bazie danych
- category_id (int) – identyfikator kategorii w bazie danych

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Usunięcie kategorii ze wskazanego wymiaru użytkownika [DELETE]:

Aby usunąć kategorię ze wskazanego wymiaru należy podać identyfikator wymiaru oraz identyfikator kategorii w adresie zapytania HTTP oraz uwzględnić token uwierzytelniający w nagłówku zapytania.

- Odpowiedź:

```
{
  "message": "Category has been successfully removed",
  "statusCode": 200
}
```

API/LABEL/

Atrybuty:

- id (int) – identyfikator paragrafu w bazie danych
- category (int) – identyfikator kategorii w bazie danych

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Anotowanie paragrafów wybranymi kategoriami [POST]:

Aby otagować paragrafy wybranymi kategoriami zdefiniowanymi przez użytkownika należy umieścić odpowiednie informacje w formacie JSON w ciele zapytania HTTP, uwzględniając token uwierzytelniający w nagłówku.

- Zapytanie:

```
{
  "data": [
    {
      "id": 7,
      "category": 2
    },
    {
      "id": 8,
      "category": 3
    }
  ]
}
```

- Odpowiedź:

```
{
  "message": "Labels have been successfully added",
  "statusCode": 200
}
```

Usuwanie anotacji wybranych paragrafów [DELETE]:

Aby usunąć anotacje należy umieścić odpowiednie informacje w formacie JSON w ciele zapytania HTTP, uwzględniając token uwierzytelniający w nagłówku.

- Zapytanie:

```
{
  "data": [
    {
      "id": 7,
      "category": 2
    },
    {
      "id": 8,
      "category": 3
    }
  ]
}
```

- Odpowiedź:

```
{
  "message": "Labels have been successfully removed",
  "statusCode": 200
}
```

API/CLASSIFIER/<INT:ID>

Atrybuty:

- id (int) – identyfikator dyskusji w bazie danych

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Wygenerowanie i przypisanie anotacji z klasyfikatora polaryzacji opinii [GET]:

Aby wygenerować dla zadanej dyskusji anotacje klasyfikatorem polaryzacji opinii, należy wskazać identyfikator dyskusji w adresie URL zapytania HTTP, uwzględniając token uwierzytelniający w nagłówku.

- Odpowiedź:

```
{
  "message": "Labels created successfully",
  "statusCode": 200
}
```

API/ARGCLASSIFIER/<INT:ID>

Atrybuty:

- id (int) – identyfikator dyskusji w bazie danych

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Wygenerowanie i przypisanie anotacji z klasyfikatora schematów argumentacji [GET]:

Aby wygenerować dla zadanej dyskusji anotacje klasyfikatorem schematów argumentacji, należy wskazać identyfikator dyskusji w adresie URL zapytania HTTP, uwzględniając token uwierzytelniający w nagłówku.

- Odpowiedź:

```
{
  "message": "Labels created successfully",
  "statusCode": 200
}
```

API/EXPORT/

Atrybuty:

- discussion (int) – identyfikator dyskusji w bazie danych
- dimension (int) – identyfikator wymiaru w bazie danych

Wymagane nagłówki:

- Bearer – token uwierzytelniający uzyskany w wyniku logowania

Wyeksportowanie dyskusji wraz z anotacjami w wybranym wymiarze [POST]:

Aby wyeksportować dyskusję w wybranym wymiarze anotacji do pliku .tsv, należy podać niezbędne dane w formacie JSON w ciele zapytania HTTP, uwzględniając token uwierzytelniający w nagłówku. W przypadku, gdy podany identyfikator wymiaru będzie równy 0, zostanie wyeksportowana dyskusja bez anotacji.

- Zapytanie:

```
{  
  "discussion": 2,  
  "dimension": 5  
}
```

- Odpowiedź:

Content-Type: text/tab-separated-values