API Documentation

Introduction

- The ling_classifier API endpoint is designed to accept text, process it by parsing into sentences and words, and then store this processed linguistic data in the system. It provides a structured way to submit and analyze textual content, returning an identifier for the processed text entry along with associated linguistic data.

Add Text

1. **Description**: This endpoint allows the client to add a new text entry to the system. It requires several parameters to create a text entry and returns a success message along with the ID of the newly created text.
2. **Endpoint**: /add_text
3. **Method**: POST

Request

- **Headers**: *Content-Type: application/json*

Body:

```
{
  "text": "Example text",
  "text_name": "Example Name",
  "text_source": "Example Source",
  "text_lang": "en",
  "text_genre": "Example Genre"
}
```

- **Parameters**:
  - text (string, required): The content of the text.
  - text_name (string, required): The name or title of the text.
  - text_source (string, required): The source or origin of the text.
  - text_lang (string, required): The language of the text.
  - text_genre (string, required): The genre or category of the text.

Response

- **Success**:
  - **Status Code**: 200 OK

Body:

```
{
  "msg": "success",
  "text_id": 1
}
```

  - msg (string): A message indicating the success of the operation.
  - text_id (integer): The ID of the newly created text entry.
- **Error**:
  - **Status Code**: 400 Bad Request

Body:

```
{
  "msg": "Missing parameters"
}
```

  - msg (string): An error message indicating that one or more required parameters are missing.

Example Requests

Successful Request:

```
POST /add_text
Content-Type: application/json
{
  "text": "Example text",
  "text_name": "Example Name",
  "text_source": "Example Source",
  "text_lang": "en",
  "text_genre": "Example Genre"
}
```

Successful Response:

```
{
  "msg": "success",
  "text_id": 1
}
```

- **Request with Missing Parameters**:

```
POST /add_text
Content-Type: application/json

{
  "text": "Example text",
  "text_name": "Example Name",
  "text_source": "Example Source",
  "text_lang": "en"
}
```

- **Error Response**:

```
{
  "msg": "Missing parameters"
}
```

Notes:

- Ensure that all required fields are provided in the request body.
- The text_id returned in the success response is generated by the server and can be used for future reference or operations related to this text entry.
- Logging is performed on the server side for successful operations, and this information can be found in the application logs.

Get Text

4. **Description**: This endpoint allows the user to retrieve a text using a text id.
5. **Endpoint**: /get_text/<int:text_id>
6. **Method**: GET

Request

- **Headers**: *Content-Type: application/json*

Body:

```
GET /get_text/123
Content-Type: application/json
```

- **Parameters**:
  - text_id (int, required): The id assigned to the text

Response

- **Success**:
  - **Status Code**: 200 OK

Body:

```
{ "text": "This is the body of the text requested." }
```

    - Text (string): the body of the text requested
- **Error**:
    - **Status Code**: 400 Bad Request

Body:

```
{
 "msg": "Text not found"
}
```

    - msg (string): An error message indicating that the requested text was not found

Example Requests

Successful Request:

```
GET /get_text/123
Content-Type: application/json
```

Successful Response:

```
{ "text": "This is the body of the text requested." }
```

- **Request with Non-Existent Text**:

```
GET /get_text/85834291
Content-Type: application/json
```

- **Error Response**:

```
{
  {"msg": "Text not found"}
}
```

Notes:

- Ensure that the text_id is entered and that it exists.
- This endpoint is intended for retrieving text entries based on their unique identifiers. If the text ID does not exist, a *400 Bad Request* response is returned.

Get All Texts

Description: This endpoint allows the user to retrieve all existing texts.

Endpoint: /get_all_texts

Method: GET

Request

- **Headers**: Content-Type: application/json
- **Body:**
  GET /add_words_
  Content-Type: application/json

  The request does not include a body. Simply perform the GET request.

- **Parameters**: none

Response

Success:

Status Code: 200 OK

Body:

```
[ { "text_id": 1, "text": "This is the body of the first text." },
{ "text_id": 2, "text": "This is the body of the second text." } ]
```

- text_id (integer): The unique identifier of the text.
- text (string): The content of the text.

Error:

- Status Code: 500 Internal Server Error
- Body:

```
{
 "msg": "Unable to retrieve texts."
}
```

- msg (string): An error message indicating that the system was unable to retrieve texts.

Example Requests

- **Successful Request:**

```
GET /get_all_texts Content-Type: application/json
Content-Type: application/json
```

- **Successful Response:**

```
[ { "text_id": 1, "text": "This is the body of the first text." },
{ "text_id": 2, "text": "This is the body of the second text." } ]
```

Notes:

The response is a JSON array containing objects for each text entry. Each object includes the text_id and text content.