

Report: Design of API Endpoints and Request/Response Formats

Pavan Sekhar Naidu

March 25, 2024

1 Introduction

This report provides an overview of the design and implementation of the API endpoints for our project. It outlines the rationale behind the design choices and details the request and response formats for each endpoint.

2 Design of API Endpoints

2.1 User Authentication

The `/api/auth/` endpoint is responsible for user authentication. It accepts POST requests with the user's username and password in the request body. Upon successful authentication, it returns an authentication token.

- **Endpoint:** `/api/auth/`
- **Method:** POST
- **Purpose:** This endpoint handles user authentication by validating the provided username and password.
- **Request Format:**

```
{
  "username": "string",
  "password": "string"
}
```

- **Response Format:**

– Success (HTTP 200 OK):

```
{
  "token": "string"
}
```

– Error (HTTP 400 Bad Request):

```
{
  "error": "string"
}
```

2.2 CRUD Operations for Assignments

The `/api/assignments/` endpoint provides CRUD (Create, Read, Update, Delete) operations for managing assignments.

- **Endpoint:** `/api/assignments/`

- **Methods:**

- **GET:** This method retrieves a list of all assignments available in the system.
- **POST:** This method creates a new assignment in the system. It requires the following parameters in the request body:
 - * **title:** A string representing the title of the assignment.
 - * **description:** A string containing the description or instructions for the assignment.
 - * **due_date:** The due date for the assignment in the format `YYYY-MM-DD`.
 - * **teacher:** The ID of the teacher who created the assignment.

- **Request Format (POST):**

```
{
  "title": "string",
  "description": "string",
  "due_date": "YYYY-MM-DD",
  "teacher": "integer"
}
```

- **Response Format:**

- **Success (HTTP 200 OK or HTTP 201 Created):** In case of success, the server responds with the details of the newly created assignment. The response contains the following fields:
 - * **id:** An integer representing the unique identifier of the assignment.
 - * **title:** A string representing the title of the assignment.
 - * **description:** A string containing the description or instructions for the assignment.
 - * **due_date:** The due date for the assignment in the format `YYYY-MM-DD`.
 - * **teacher:** The ID of the teacher who created the assignment.
- **Error (HTTP 400 Bad Request):** In case of an error, such as invalid request parameters, the server responds with an error message in the following format:

```
{
  "error": "string"
}
```

3 Conclusion

The design of our API endpoints aims to provide a clear and consistent interface for interacting with our system. By following RESTful principles and standardizing request and response formats, we ensure interoperability and ease of use for client applications.