

API Documentation

User Registration

Endpoint: POST /register

Description: Registers a new user and handles the user registration process.

Request Body:

```
{  
  "username": "<username>",  
  "password": "<password>"  
}
```

Response: Returns the access token, success message, and user data.

Status Codes: Returns 201 on success, 400 if required fields are not provided or if the username already exists.

User Login

Endpoint: POST /login

Description: Handles the login functionality for a user.

Request Body:

```
{  
  "username": "<username>",  
  "password": "<password>"  
}
```

Response: Returns a success response with the generated JWT token and a message indicating successful login, along with the serialized user data.

Status Codes: Returns 200 on success, 400 if required fields are not provided or if the password is invalid.

Get Devices

Endpoint: GET /devices

Description: Retrieves a list of devices associated with a specific user.

Response: Returns a list of serialized device data.

Status Codes: Returns 200 on success.

Submit Data

Endpoint: POST /data/submit

Description: Submit data to a device.

Request Body:

```
{
  "device_id": "<device_id>",
  "data": "<data>"
}
```

Response: Returns the HTTP response object.

Status Codes: Returns 201 on success, 400 if required fields are not provided, 401 if the user is not authorized to submit data to the device.

Add Device

Endpoint: POST /devices

Description: Add a new device.

Request Body:

```
{
  "name": "<device_name>",
  "location": "<device_location>"
}
```

Response: Returns the HTTP response object.

Status Codes: Returns 201 on success, 400 if required fields are not provided.

Update Device

Endpoint: PUT /devices/{device_id}

Description: Update a device with the provided device_id.

Request Body:

```
{
  "name": "<new_device_name>",
  "location": "<new_device_location>"
}
```

Response: Returns the HTTP response containing the updated device data or an error message.

Status Codes: Returns 200 on success, 400 if required fields are not provided, 401 if the user is not authorized to update the device.

Delete Device

Endpoint: DELETE /devices/{device_id}

Description: Deletes a device with the provided device_id.

Status Codes: Returns 200 on success, 404 if the device does not exist.

Get All Devices

Endpoint: GET /devices

Description: Retrieves all devices.

Response: Returns a list of serialized device data.

Status Codes: Returns 200 on success.

Get User

Endpoint: GET /users/{user_id}

Description: Retrieves a specific user by their ID.

Response: Returns the serialized data of the retrieved user.

Status Codes: Returns 200 on success, 404 if the user does not exist.

Update Device

Endpoint: PUT /devices/{device_id}

Description: Update a device with the provided data.

Request Body:

```
{
  "name": "<new_device_name>",
  "location": "<new_device_location>",
  "user": "<user>"
}
```

Response: Returns the HTTP response containing the updated device data or an error message.

Status Codes: Returns 200 on success, 400 if required fields are not provided, 404 if the device does not exist.

Get All Users

Endpoint: GET /users

Description: Retrieves all users.

Response: Returns a list of serialized user data.

Status Codes: Returns 200 on success.

Manage User Roles

Endpoint: PUT /users/{user_id}/roles

Description: Manage the roles of a user.

Request Body:

```
{  
  "role": "<new_role>"  
}
```

Response: Returns the HTTP response containing the updated user data or an error message.

Status Codes: Returns 200 on success, 400 if required fields are not provided, 404 if the user does not exist.

Delete User

Endpoint: DELETE /users/{user_id}

Description: Deletes a user with the given user_id.

Status Codes: Returns 200 on success, 404 if the user does not exist.