API Documentation

Account Management

This document provides the technical specification for the API endpoints related to creating and managing user accounts, as defined in account_route.py.

1. Add Account

Creates a new user account and links it to an existing employee profile.

- **Endpoint**: POST /account/add
- **Description**: Creates a new user account with a username and a securely hashed password. This action is restricted to administrative users.
- **Authentication**: **Manager Role Required**. The request must include a valid access token for a user with the 'Manager' role in the Authorization header.

Request Body

Field	Type	Required	Description
id	String	Yes	A unique identifier for the new account (e.g., "ACC004").
employee_id	String	Yes	The ID of the employee to link this account to (e.g., "EM004").
username	String	Yes	The user's unique username.

password	String	Yes	he user's plain-text password.

Example Request:

```
{
    "id": "ACC004",
    "employee_id": "EM004",
    "username": "NewUser",
    "password": "a_strong_password"
}
```

Responses

- **201 Created (Success)**
  - **Description:** Returned when the account is created successfully.
  - **Body:**

```
{
        “id”: “ACC004”,
        “employee_id”: “EM004”,
        “username”: “NewUser”
}
```

- **400 Bad Request (Invalid Input / Constraint Violation)**
  - **Description:** Returned if required fields are missing, the data is malformed, or if the *username* or *employee_id* already exists.
  - **Body**

```
{
        “message”: “Violate database constraint”
}
```

- **403 Forbidden (Unauthorised)**
    - **Description:** Returned if the user making the request does not have the ‘Manager’ role.
    - **Body:**

```
{
        “message”: “Access denied for <user_role>”
}
```

2. Get All Accounts

Retrieves a list of all user accounts in the system.

- **Endpoint**: GET /account/
- **Description**: Fetches a complete list of all account records. This action is restricted to administrative users.
- **Authentication**: **Manager Role Required**. The request must include a valid access token for a user with the 'Manager' role in the Authorization header.

Request Body

(None)

Responses

- **200 OK (Success)**
    - **Description:** Returned with a list of all accounts.
    - **Body:**

```
[
        {
                “id”: “ACC001”
                “employee_id”: “EM001”
                “username”: “UserOne”
        },
        {
                "id": "ACC002"
                "employee_id": "EM002"
                "username": "UserTwo"
        }
]
```

- **403 Forbidden (Unauthorised)**
  - **Description:** Returned if the user making the request does not have the ‘Manager’ role.
  - **Body**

```
{
        "message": "Access denied for <user_role>"
}
```

3. Get Account by ID

Retrieves the employee ID associated with a single account.

- **Endpoint**: GET /account/search
- **Description**: Looks up a single account by its unique ID and returns the linked employee's ID.
- **Authentication**: **Manager Role Required**. The request must include a valid access token for a user with the 'Manager' role in the Authorization header.

Query Parameters

Parameter	Type	Required	Description
id	String	Yes	The unique ID of the account to search for.

Example Request URL: /account/search?id=ACC001

Responses

- **200 OK (Success)**
    - **Description:** Returned when the account is found successfully.
    - **Body:**

      ```
      {
              “Employee ID”: “EM001”
      }
      ```

- **400 Bad Request (Missing Parameter)**
    - **Description:** Returned if the *id* query parameter is not included in the request.
    - **Body:**

      ```
      {
              “message”: “Missing contract ID in query params”
      }
      ```

- **404 Not Found (Not Found)**
    - **Description:** Returned if no account with the specified ID exists.
    - **Body:**

      ```
      {
              “message”: “No account found”
      }
      ```