

Employee Manager API Documentation

The Employee Manager API provides a comprehensive set of endpoints to manage employee and department data. This documentation covers the usage of each endpoint, request/response formats, and authentication/authorization requirements.

Base URL

<http://localhost:4200/>

Account information

The API implements two accounts:

- Admin (user: admin, pass: 1357\$)
- User (user: user, pass: 1234\$)

The Admin account is of role 'Administrator' and has access to all endpoints.

The User account is of role 'User' and has access only to the Employee endpoints.

Authentication and Authorization

The API uses token-based authentication with role-based access control to secure the endpoints. Each user must include an access token in the **Authorization** header of their HTTP requests using the Bearer Token scheme:

Authorization: Bearer <access_token>

The access token can be obtained by authenticating through the **/account/login** endpoint using a valid username and password.

Endpoints

Account

- **POST /account/login**

This endpoint allows users to authenticate and obtain an access token.

Request:

- Headers:
 - **Content-Type: application/json**
- Body:
 - { "username": "<username>", "password": "<password>" }

Response:

- Status: 200 OK
 - Body: Returns a token and the authenticated user.
 - { "status": true, "msg": "<access_token >", "value": "<authenticated_user>" }

- Status: 200 OK
 - Body: Invalid credentials.
 - { "status": true, "msg": "Invalid credentials!" }
- Status: 200 OK
 - Body: Non-existent user.
 - { "status": true, "msg": "User does not exist!" }
- Status: 400 Bad Request
 - Body: Error message indicating invalid input or missing required fields.
 - { "status": false, "msg": "Invalid model!" }
- Status 500: Internal Server Error
 - Body: Error message indicating a server-side issue.
 - { "status": false, "msg": "Could not login." }.
- **POST /account/refresh-token**

Refreshes an expired token.

Request:

- Headers:
 - **Authorization: Bearer <access_token>**

Responses:

- Status: 200 OK
 - Body: Returns the newly generated token.
 - { "status": false, "msg": "Ok", "value": "<access_token>" }.
- Status: 500 Not Found
 - Body: Error message indicating the token was invalid.
 - { "status": false, "msg": "Invalid token!" }.

Employees

- **GET /employees**

Retrieves a list of employees.

Request:

- Headers:
 - **Authorization: Bearer <access_token>**

Responses:

- Status: 200 OK
 - Body: List of employees.
 - { "status": true, "msg": "Ok", "value": [{ "employeeId": 1, "employeeName": "John Doe", "departmentId": 1, "salary": 5000, "dateJoined": "2022-01-01" }, ...] }
- Status: 401 Unauthorized
 - Body: Unauthorized access error message.
- Status: 500 Internal Server Error
 - Body: Error message indicating a server-side issue.
 - { "status": false, "msg": "An error occurred!" }.

- **POST /employees**

Adds a new employee.

Request:

- Headers:
 - **Authorization: Bearer <access_token>**
 - **Content-Type: application/json**
- Body:
 - { "employeeName": "John Doe", "departmentId": 1, "salary": 5000, "dateJoined": "2022-01-01" }

Responses:

- Status: 201 Created
 - Body: Returns the newly created employee.
 - { "status": true, "msg": "Employee created successfully", "value": { "employeeId": 1, "employeeName": "John Doe", "departmentId": 1, "salary": 5000, "dateJoined": "2022-01-01" } }
- Status: 400 Bad Request
 - Body: Error message indicating invalid input or missing required fields.
 - { "status": false, "msg": "Invalid model!" }
- Status: 401 Unauthorized
 - Body: Unauthorized access error message.

- Status: 500 Internal Server Error
 - Body: Error message indicating a server-side issue.
 - { "status": false, "msg": "Could not add employee!" }

- **PUT /employees/{employeeId}**

Updates an existing employee.

Request:

- Headers:
 - **Authorization: Bearer <access_token>**
 - **Content-Type: application/json**
- Body:
 - { "employeeId": 1, "employeeName": "John Doe", "departmentId": 1, "salary": 6000, "dateJoined": "2022-01-01" }

Responses:

- Status: 200 OK
 - Body: Returns the updated employee.
 - { "status": true, "msg": "Employee updated successfully", "value": { "employeeId": 1, "employeeName": "John Doe", "departmentId": 1, "salary": 6000, "dateJoined": "2022-01-01" } }
- Status: 400 Bad Request
 - Body: Error message indicating invalid input or missing required fields.
 - { "status": false, "msg": "Invalid model!" }
- Status: 401 Unauthorized
 - Body: Unauthorized access error message.
- Status: 500 Not Found
 - Body: Error message indicating the employee was not found.
 - { "status": false, "msg": "Employee does not exist!" }
- Status: 500 Internal Server Error
 - Body: Error message indicating a server-side issue.
 - { "status": false, "msg": "Could not update employee!" }

- **DELETE /employees/{employeeId}**

Deletes an employee.

Request:

- Headers:
 - **Authorization: Bearer <access_token>**

Responses:

- Status: 200 OK
 - Body: Returns a Boolean positive.
 - { "status": "true", "msg": "Ok", "value": "true" }
- Status: 401 Unauthorized
 - Body: Unauthorized access error message.
- Status: 500 Not Found
 - Body: Error message indicating the employee was not found.
 - { "status": false, "msg": "Employee does not exist!" }
- Status: 500 Internal Server Error
 - Body: Error message indicating a server-side issue.
 - { "status": false, "msg": "Could not delete employee!" }

Departments

- **GET /departments**

Retrieves a list of departments.

Request:

- Headers:
 - **Authorization: Bearer <access_token>**

Responses:

- Status: 200 OK
- Body: List of departments.
 - { "status": true, "msg": "Ok", "value": [{ "departmentId": 1, "departmentName": "IT" }, ...] }
- Status: 401 Unauthorized
 - Body: Unauthorized access error message.

- Status: 500 Internal Server Error
 - Body: Error message indicating a server-side issue.
 - { "status": false, "msg": "An error occurred!" }.

- **POST /departments**

Adds a new department.

Request:

- Headers:
 - **Authorization: Bearer <access_token>**
 - **Content-Type: application/json**
- Body:
 - { "departmentName": "Sales" }

Responses:

- Status: 201 Created
 - Body: Returns newly created department.
 - { "status": true, "msg": "Ok", "value": { "departmentId": 1, "departmentName": "Sales" } }
- Status: 400 Bad Request
 - Body: Error message indicating invalid input or missing required fields.
 - { "status": false, "msg": "Invalid model!" }
- Status: 401 Unauthorized
 - Body: Unauthorized access error message.
- Status: 500 Internal Server Error
 - Body: Error message indicating a server-side issue.
 - { "status": false, "msg": "An error occurred!" }.

- **PUT /departments/{departmentId}**

Updates an existing department.

Request:

- Headers:
 - **Authorization: Bearer <access_token>**

- **Content-Type: application/json**
- Body:
 - { "departmentId": "1", "departmentName": "Sales" }

Responses:

- Status: 200 OK
 - Body: Returns the update department.
 - { "status": true, "msg": "Ok", "value": { "departmentId": 1, "departmentName": "Sales" } }
- Status: 400 Bad Request
 - Body: Error message indicating invalid input or missing required fields.
 - { "status": false, "msg": "Invalid model!" }
- Status: 401 Unauthorized
 - Body: Unauthorized access error message.
- Status: 500 Not Found
 - Body: Error message indicating the department was not found.
 - { "status": false, "msg": "Department does not exist!" }
- Status: 500 Internal Server Error
 - Body: Error message indicating a server-side issue.
 - { "status": false, "msg": "An error occurred!" }.

• **DELETE /departments/{departmentId}**

Deletes a department.

Request:

- Headers:
 - **Authorization: Bearer <access_token>**

Responses:

- Status: 200 OK
 - Body: Returns a Boolean positive.
 - { "status": "true", "msg": "Ok", "value": "true" }

- Status: 401 Unauthorized
 - Body: Unauthorized access error message.
- Status: 500 Not Found
 - Body: Error message indicating the department was not found.
 - { "status": false, "msg": "Department does not exist!" }
- Status: 500 Internal Server Error
 - Body: Error message indicating a server-side issue.
 - { "status": false, "msg": "An error occurred!" }.