API Documentation for Scheduling System

1. Overview

This document provides the API endpoints for the Scheduling System, detailing the available operations for managing email logs, employees, tasks, notifications, and user authentication.

2. API Endpoints

2.1 Email Log Controller

Retrieve All Email Logs

Endpoint: GET /api/v1/email-logs

Postman Example:

Method: GET

URL: http://localhost:8080/api/v1/email-logs

Response:


```
[
  {
    "id": 1,
    "userId": 123,
    "taskId": 456,
    "email": "example@example.com"
  }
]
```


Retrieve Email Logs by User ID

Endpoint: GET /api/v1/email-logs/by-user

Postman Example:

Method: GET

URL: http://localhost:8080/api/v1/email-logs/by-user?userId=123

Response:

```
[
  {
    "id": 1,
    "userId": 123,
    "taskId": 456,
    "email": "example@example.com",
    "status": "SENT"
  }
]
```


Retrieve Email Logs by Task ID

Endpoint: GET /api/v1/email-logs/by-task

Postman Example:

Method: GET

URL: http://localhost:8080/api/v1/email-logs/by-task?taskId=456

Response:


```
[
  {
    "id": 1,
    "userId": 123,
    "taskId": 456,
    "email": "example@example.com"
  }
]
```

2.2 Employee Controller

Save Employee

Endpoint: POST /api/v1/employee/saveEmployee

Postman Example:

Method: POST

URL: http://localhost:8080/api/v1/employee/saveEmployee

Body:

```
{
   "name": "John Doe",
   "empNumber": "EMP001",
   "email": "john.doe@example.com"
}
```

Response:

```
{
   "code": "00",
   "message": "Success",
   "content": {
      "name": "John Doe",
      "empNumber": "EMP001",
      "email": "john.doe@example.com"
   }
}
```

Update Employee

Endpoint: PUT /api/v1/employee/updateEmployee

Postman Example:

Method: PUT

URL: http://localhost:8080/api/v1/employee/updateEmployee

Body:

```
{
   "empID": 1,
```

```
  "name": "John Doe Updated",
  "empNumber": "EMP001",
  "email": "john.doe.updated@example.com"
}
```


Response:


```
{
  "code": "00",
  "message": "Success",
  "content": {
    "empID": 1,
    "name": "John Doe Updated",
    "empNumber": "EMP001",
    "email": "john.doe.updated@example.com"
  }
}
```


Get All Employees

Endpoint: GET /api/v1/employee/getAllEmployees

Postman Example:

Method: GET

URL: http://localhost:8080/api/v1/employee/getAllEmployees

Response:


```
[
  {
    "empID": 1,
    "name": "John Doe",
    "empNumber": "EMP001",
    "email": "john.doe@example.com"
  }
]
```

2.3 Mail Controller

Send Email

Endpoint: POST /api/v1/mail/sendemail

Postman Example:

Method: POST

URL: http://localhost:8080/api/v1/mail/sendemail

Body:

```
{
   "to": "recipient@example.com",
   "subject": "Test Email",
   "body": "This is a test email."
}
```

Response:

```
{
   "message": "Email sent successfully."
}
```

2.4 Task Controller

Create Task

Endpoint: POST /api/tasks

Postman Example:

Method: POST

URL: http://localhost:8080/api/tasks

Body:

```
{
   "title": "New Task",
   "description": "Task description",
   "userId": 123
}
```

Response:

```
{
  "message": "Task created successfully."
}
```

2.5 User Controller

Register User

Endpoint: POST /api/auth/register

Postman Example:

Method: POST

URL: http://localhost:8080/api/auth/register

Body:

```
{
  "username": "testuser",
  "password": "password123"
}
```

Response:

```
{
  "message": "User registered successfully."
}
```

3. Error Handling

Standard error responses should include a message and an appropriate HTTP status code.

4. Conclusion

This API documentation provides a comprehensive overview of the available endpoints in the Scheduling System, enabling developers to integrate and utilize the functionalities effectively.