**Job Management API Documentation**

**Introduction**

The Job Management API provides endpoints for managing jobs, including adding, updating, and retrieving job details. This API allows users to perform CRUD operations on job entities.

**Endpoints**

**Jobs**

GET /api/v1/jobs

**Description**: Retrieves a list of all jobs.

**Status Code**: 200 OK

**Body**: Array of job objects containing job details.

POST /api/v1/jobs

**Description**: Adds a new job to the system.

**Request Body**: AddJobDto object containing job details.

**Status Code**: 201 Created

**Body**: Job object representing the newly added job.

PUT /api/v1/jobs/{id}

**Description**: Updates an existing job.

**Parameters**:

**id:** The ID of the job to update.

**Request Body:** UpdateJobDto object containing updated job details.

**Status Code:** 200 OK

GET /api/v1/jobs/{id}

**Description:** Retrieves details of a specific job.

**Parameters**:

**id:** The ID of the job to retrieve.

**Response:**

**Status Code:** 200 OK

**Body:** Job object representing the requested job.

POST /api/v1/jobs/List

**Description**: Retrieves a paginated list of jobs based on search criteria.

**Request Body:** JobListRequest object containing search parameters.

**Response:**

**Status Code:** 200 OK

**Body:** Object containing total count of matching jobs and paginated list of job objects.

**Request and Response Formats**

**AddJobDto:**

title: string

description: string

locationId: int

departmentId: int

closingDate: DateTime

**UpdateJobDto:**

title: string

description: string

locationId: int

departmentId: int

closingDate: DateTime

**JobListRequest:**

q: string (optional)

locationId: int (optional)

departmentId: int (optional)

pageNo: int

pageSize: int

**Job:**

id: int

code: string

title: string

description: string

location: Location object

department: Department object

postedDate: DateTime

closingDate: DateTime

**Here's a brief explanation of each endpoint in your DepartmentController:**

**GetallDepartments [HttpGet]:** Retrieves all departments from the database and returns them as a list.

**GetDepartmentById [HttpGet("{id}")]:** Retrieves a specific department by its ID. If the department with the specified ID is not found, it returns a 404 Not Found status code.

**AddDepartment [HttpPost]:** Adds a new department to the database based on the data provided in the request body (AddDepartmentDto). It then returns the newly created department along with a 201 Created status code.

**UpdateDepartment [HttpPut("{id}")]:** Updates an existing department identified by its ID. It first checks if the department exists, and if so, updates its title with the data provided in the request body (UpdateDepartmentDto). It then returns a 200 OK status code.

**Here's a brief explanation of each endpoint in your locationController:**

**GetAllLocations [HttpGet]:** Retrieves all locations from the database and returns them as a list.

**GetLocationById [HttpGet("{id}")]:** Retrieves a specific location by its ID. If the location with the specified ID is not found, it returns a 404 Not Found status code.

**AddLocation [HttpPost]:** Adds a new location to the database based on the data provided in the request body (AddLocationDto). It then returns the newly created location along with a 201 Created status code.

**UpdateLocation [HttpPut("{id}")]:** Updates an existing location identified by its ID. It first checks if the location exists, and if so, updates its details with the data provided in the request body (UpdateLocationDto). It then returns a 200 OK status code.

**Error Handling**

The API returns appropriate HTTP status codes along with error messages in case of errors.

Status Code 500 is returned for internal server errors.

SNAPSHOTS



      