

API Documentation

Base URL

http://localhost:8080/api

Department Endpoints:

Method	Endpoint	Description
GET	/departments	Get all departments
GET	/departments/{id}	Get department by ID
POST	/departments	Create new department
PUT	/departments/{id}	Update department
DELETE	/departments/{id}	Delete department

Department Request Body:

```
{ "name": "Engineering", "location": "New York", "budget": 500000.00 }
```

Employee Endpoints

Method	Endpoint	Description
GET	/employees	Get all employees
GET	/employees?departmentId={id}	Filter employees by department
GET	/employees/{id}	Get employee by ID
POST	/employees	Create new employee
PUT	/employees/{id}	Update employee
DELETE	/employees/{id}	Delete employee

Employee Request Body:

```
{ "name": "John Doe", "email": "john.doe@example.com", "phone": "1234567890", "hireDate": "2024-01-15", "salary": 75000.00, "departmentId": 1 }
```

Project Endpoints

Method	Endpoint	Description
GET	/projects	Get all projects
GET	/projects?departmentId={id}	Filter projects by department
GET	/projects/{id}	Get project by ID
POST	/projects	Create new project
PUT	/projects/{id}	Update project
DELETE	/projects/{id}	Delete project

Project Request Body:

```
{"name": "Mobile App Development", "description": "Develop a new mobile application",  
"startDate": "2024-01-01", "endDate": "2024-06-30", "departmentId": 1}
```

Employee-Project Assignment Endpoints

Method	Endpoint	Description
POST	/employee-projects	Assign employee to project
GET	/employee-projects/employee/{employeeId}	Get all projects for an employee
GET	/employee-projects/project/{projectId}	Get all employees on a project
PUT	/employee-projects/{id}/role?role={newRole}	Update employee role
DELETE	/employee-projects/{id}	Remove assignment

Assignment Request Body:

```
{ "employeeId": 1, "projectId": 1, "role": "Developer"}
```

Response Status Codes

- **200 OK:** Successful GET, PUT requests
- **201 Created:** Successful POST request
- **204 No Content:** Successful DELETE request
- **400 Bad Request:** Validation error or business rule violation
- **404 Not Found:** Resource not found
- **500 Internal Server Error:** Server error

Error Response Format

```
{  
  "status": 400,  
  "message": "Error description",  
  "timestamp": "2024-01-15T10:30:00"  
}
```

Validation Rules

Department

- Name: Required, 2-100 characters
- Location: Required, 2-200 characters
- Budget: Required, must be ≥ 0

Employee

- Name: Required, 2-100 characters
- Email: Required, valid email format, unique
- Phone: Required, 10-15 digits
- Hire Date: Required, cannot be future date
- Salary: Required, must be ≥ 0
- Department: Optional

Project

- Name: Required, 2-100 characters
- Description: Optional, max 500 characters
- Start Date: Required
- End Date: Required
- Department: Optional

Employee-Project Assignment

- Employee ID: Required
- Project ID: Required
- Role: Required, 2-50 characters

Business Rules

1. **Department Deletion:** Cannot delete a department that has assigned employees
2. **Email Uniqueness:** Each employee must have a unique email address
3. **Assignment Uniqueness:** An employee cannot be assigned to the same project twice
4. **Cascade Deletion:**
 - Deleting an employee removes all their project assignments
 - Deleting a project removes all employee assignments to that project

Testing the Application

Manual Testing Steps

Start both backend and frontend servers

Navigate to <http://localhost:4200>

Create a Department:

- Go to Departments tab
- Click "Add Department"
- Fill in the form and save

Create Employees:

- Go to Employees tab
- Click "Add Employee"
- Assign to the created department

Create Projects:

- Go to Projects tab
- Click "Add Project"
- Assign to the department

Assign Employees to Projects:

- Go to Assignments tab
- Select view mode
- Click "Assign Employee to Project"
- Select employee, project, and role