

# Swagger spec for REST API

# Table of Contents

- Overview ..... 1
  - Version information ..... 1
  - Contact information ..... 1
  - License information ..... 1
  - Tags ..... 1
- Paths ..... 1
  - get all departments ..... 1
  - update department ..... 2
  - create new department ..... 2
  - get all employees with information about average salary ..... 3
  - get department by id ..... 4
  - delete department by id ..... 4
  - get employee’s list from department ..... 5
  - get all employees ..... 6
  - update employee ..... 6
  - create employee ..... 7
  - get filtered list of employees ..... 8
  - get employee by id ..... 9
  - delete employee by id ..... 9
- Definitions ..... 10
  - Department ..... 10
  - DepartmentWithAverageSalary ..... 10
  - Employee ..... 10

# Overview

This is a sample spec for Department REST API

## Version information

Version: v1

## Contact information

Contact: Igor Contact Email: @mrigor

## License information

License: test task for EPAM License URL: <http://www.testcompany.com/licenses/LICENSE-2.0.html>

Terms of service: <http://www.testcompany.com/terms-and-conditions>

## Tags

- departments: Endpoint for Department specific operations
- employees: Endpoint for Employee specific operations

## Paths

### get all departments

GET /rest/departments

### Description

get all departments

### Responses

HTTP Code	Description	Schema
200	successful operation	<a href="#">Department</a>

## Produces

- application/json

## Tags

- departments

# update department

PUT /rest/departments

## Description

update department

## Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	body	new department	false	<a href="#">Department</a>	

## Responses

HTTP Code	Description	Schema
default	successful operation	No Content

## Consumes

- application/json

## Tags

- departments

# create new department

POST /rest/departments

## Description

create new department

## Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	body	new department	false	Department	

## Responses

HTTP Code	Description	Schema
200	successful operation	Department

## Consumes

- application/json

## Produces

- application/json

## Tags

- departments

# get all employees with information about average salary

```
GET /rest/departments/withAvgSalary
```

## Description

get department's list with information about average salary by each department. If department don't have any employees, then AVG salary will be = 0

## Responses

HTTP Code	Description	Schema
200	successful operation	DepartmentWithAverageSalary array

Produces

- application/json

Tags

- departments

get department by id

GET /rest/departments/{id}

Description

get department by id

Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	id	identifier of department	true	integer (int32)	

Responses

HTTP Code	Description	Schema
200	successful operation	<a href="#">Employee</a>
404	department with such identifier doesn't exists	No Content

Produces

- application/json

Tags

- departments

delete department by id

DELETE /rest/departments/{id}

## Description

delete department by id

## Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	id	identifier of department	true	integer (int32)	

## Responses

HTTP Code	Description	Schema
404	department with such identifier doesn't exists	No Content

## Tags

- departments

# get employee's list from department

```
GET /rest/departments/{id}/employeeess
```

## Description

get employee from department

## Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	id	identifier of department	true	integer (int32)	

## Responses

HTTP Code	Description	Schema
200	successful operation	<a href="#">Employee</a> array
404	department with such identifier doesn't exists	No Content

## Produces

- application/json

## Tags

- departments

# get all employees

GET /rest/employees

## Description

get all employees

## Responses

HTTP Code	Description	Schema
200	successful operation	<a href="#">Department</a> array

## Produces

- application/json

## Tags

- employees

# update employee

PUT /rest/employees

## Description

update employee

## Parameters



Type	Name	Description	Required	Schema	Default
BodyParameter	body	new employee	false	<a href="#">Employee</a>	

## Responses

HTTP Code	Description	Schema
default	successful operation	No Content

## Consumes

- application/json

## Tags

- employees

# create employee

POST /rest/employees

## Description

create employee

## Parameters

Type	Name	Description	Required	Schema	Default
BodyParameter	body		false	<a href="#">Employee</a>	

## Responses

HTTP Code	Description	Schema
200	successful operation	<a href="#">Employee</a>

## Consumes

- application/json

Produces

- application/json

Tags

- employees

get filtered list of employees

GET /rest/employees/filtered

Description

get filtered list of employees by params

Parameters

Type	Name	Description	Required	Schema	Default
QueryParameter	from	from	false	string (date)	
QueryParameter	to		false	string (date)	
QueryParameter	departmentid		false	integer (int32)	

Responses

HTTP Code	Description	Schema
200	successful operation	<a href="#">Department</a> array

Produces

- application/json

Tags

- employees

# get employee by id

GET /rest/employees/{id}

## Description

get employee by id

## Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	id	identifier of employee	true	integer (int32)	

## Responses

HTTP Code	Description	Schema
200	successful operation	<a href="#">Employee</a>
404	employee with such identifier doesn't exists	No Content

## Produces

- application/json

## Tags

- employees

# delete employee by id

DELETE /rest/employees/{id}

## Description

delete employee by id

## Parameters

Type	Name	Description	Required	Schema	Default
PathParameter	id	identifier of employee	true	integer (int32)	

## Responses

HTTP Code	Description	Schema
200	successful operation	<a href="#">Employee</a>
404	employee with such identifier doesn't exists	No Content

## Tags

- employees

# Definitions

## Department

Name	Description	Required	Schema	Default
id		false	integer (int32)	
name		false	string	

## DepartmentWithAverageSalary

Name	Description	Required	Schema	Default
id		false	integer (int32)	
name		false	string	
averageSalary		false	integer (int32)	

## Employee

Name	Description	Required	Schema	Default
id		false	integer (int32)	
fullName		false	string	

Name	Description	Required	Schema	Default
birthDay		false	string (date)	
salary		false	integer (int32)	
departmentId		false	integer (int32)	