

Project Info

Name: Summary App

Server Runtime: NodeJS

Libraries:

express: ExpressJS framework

mongodb: MongoDB library

bcrypt: For storing user password in a secure manner in DB

jsonwebtoken: Generate and validate auth token

Database Engine: MongoDB

Credentials:

Port: 27017

Username: admin

Password: secretpassword

Database Name: summary_app

Collections:

Collection 1: employee

Description: Stores employee info

Object Schema:

name: string

salary: number

currency: string

department: string

on_contract: boolean

sub_department: string

Collection 2: user

Description: Stores user account

Object Schema:

username: string

password: string

Project Directory Structure:

-SummaryApp

- Dockerfile
- docker-compose.yml.
- script.py
- readme.pdf
- SummaryApp.postman_collection.json

app

- app.js
- package.json
- controller**
 - employee.js
 - user.js
- model**
 - model.js
- utils
 - auth.js
 - db.js
 - validation.js

Root directory

- Docker File
- Docker Compose file
- Script to populate data in DB.
- Readme file for project info
- Postman collection import file

NodeJS app directory

- Main file
- Package.json file for package installation

Controller files directory

- Employee controller file
- User controler file

Model files directory

- Model file for employee and user
- Util directory for util files
- Auth file for authentication logic
- Database connection setup file
- Request payload and other validations logic

Command to run application:

Step #1:

Open terminal in project's root directory

Step #2:

Start program using docker command

Command:

docker-compose up

App logs to observe

[INFO] Server started at 80 port

[INFO] Connection with database successfully established

Server started successful

DB connection success

Steps #3:

Execute python script to populate data in database

Command:

python3 script.py

Script logs to observe

Employee creation success...

Scrip completed

Employee APIs

#1 Fetch all employees

Method: **GET**

Url: **http://localhost:8080/api/employee**

Header: Authorization: {{token}}

Success Response: {

```
"success": true,
"data": [
  {
    "_id": "63cd72fe9ca53b43a22cc1b5",
    "name": "Abhishek",
    "salary": 145000,
    "currency": "USD",
    "department": "Engineering",
    "sub_department": "Platform"
  },
  {
    "_id": "63cd72fe9ca53b43a22cc1b6",
    "name": "Anurag",
    "salary": 90000,
    "currency": "USD",
    "department": "Banking",
    "on_contract": true,
    "sub_department": "Loan"
  }
],
"count": 2
}
```

#2 Fetch an employee by id

Method: **GET**

Url: **http://localhost:8080/api/employee/{{employee_id}}**

Header: Authorization: {{token}}

Success Response: {

```
"success": true,
"data": [
  {
    "_id": "63cd72fe9ca53b43a22cc1b5",
    "name": "Abhishek",
    "salary": 145000,
    "currency": "USD",
    "department": "Engineering",
    "sub_department": "Platform"
  }
],
```

```
        "count": 1
    }
}
```

#3 Add a new employee

Method: **POST**

Url: **http://localhost:8080/api/employee**

Header: Authorization: {{token}}

Body: {

```
    "name": "Fedrick",
    "salary": 100,
    "currency": "USD",
    "department": "Engineering",
    "sub_department": "Platform"
```

}

Success Response: {

```
    "success": true,
    "data": {
        "acknowledged": true,
        "insertedId": "63cd745f9ca53b43a22cc1be"
    }
}
```

#4 Delete an employee by id

Method: **DELETE**

Url: **http://localhost:8080/api/employee/{{employee_id}}**

Header: Authorization: {{token}}

Success Response: {

```
    "success": true,
    "data": {
        "acknowledged": true,
        "deletedCount": 1
    }
}
```

#5 Fetch employee where on_contract=true

Method: GET

Url: **http://localhost:8080/api/employee?on_contract=true**

Note: on_contract query string is supported in aggregation APIs also

Header: Authorization: {{token}}

Success Response: {

```
    "success": true,
    "data": [
      {
        "_id": "63cd72fe9ca53b43a22cc1b6",
        "name": "Anurag",
        "salary": 90000,
        "currency": "USD",
        "department": "Banking",
        "on_contract": true,
        "sub_department": "Loan"
      }
    ],
    "count": 1,
    "filter": {
      "on_contract": true
    }
  }
```

#6 Fetch max(), min(), mean() salary in company

Method: GET

Url: **http://localhost:8080/api/employee?salary=max**

Header: Authorization: {{token}}

Success Response: {

```
  {
    "success": true,
    "data": [
      {
        "_id": null,
        "salary": 200000000
      }
    ],
    "count": 1,
    "aggregation": "max"
  }
```

#7 Fetch max(), min(), mean() salary in a department

Method: **GET**

Url: **http://localhost:8080/api/employee?salary=max&department=Engineering**

Header: Authorization: {{token}}

Success Response:

```
{
  "success": true,
  "data": [
    {
      "_id": null,
      "salary": 200000000
    }
  ],
  "count": 1,
  "filter": {
    "department": "Engineering"
  },
  "aggregation": "max"
}
```

#8 Fetch max(), min(), mean() salary in a sub_department

Method: **GET**

Url: **http://localhost:8080/api/employee?salary=max&department=Engineering&sub_department=**

Platform

Header: Authorization: {{token}}

Success Response:

```
{
  "success": true,
  "data": [
    {
      "_id": null,
      "salary": 200000000
    }
  ],
  "count": 1,
  "filter": {
    "department": "Engineering",
    "sub_department": "Platform"
  },
  "aggregation": "max"
}
```

#9 Fetch all departments

Method: **GET**

Url: **http://localhost:8080/api/employee/department/**

Header: Authorization: {{token}}

Success Response:

```
{
  "success": true,
  "data": [
    "Administration",
    "Banking",
    "Engineering",
    "Operations"
  ],
  "count": 4
}
```

#10 Fetch employees of a department

Method: **GET**

Url: **http://localhost:8080/api/employee/department/Engineering**

Header: Authorization: {{token}}

Success Response:

```
{
  "success": true,
  "data": [
    {
      "_id": "63cd72fe9ca53b43a22cc1b5",
      "name": "Abhishek",
      "salary": 145000,
      "currency": "USD",
      "department": "Engineering",
      "sub_department": "Platform"
    },
    {
      "_id": "63cd72fe9ca53b43a22cc1b7",
      "name": "Himani",
      "salary": 240000,
      "currency": "USD",
      "department": "Engineering",
      "sub_department": "Platform"
    },
    {
      "_id": "63cd72fe9ca53b43a22cc1b9",
      "name": "Ragini",
      "salary": 30,
      "currency": "USD",
      "department": "Engineering",
      "sub_department": "Platform"
    },
    {
      "_id": "63cd72fe9ca53b43a22cc1bd",
      "name": "Anupam",
      "salary": 200000000,
      "currency": "INR",
      "department": "Engineering",
      "sub_department": "Platform"
    }
  ],
  "count": 4,
}
```



```
        "filter": {
            "department": "Engineering"
        }
    }
}
```

#11 Fetch employees of a sub_department

Method: GET

Url: http://localhost:8080/api/employee/department/Engineering?sub_department=Platform

Header: Authorization: {{token}}

Success Response:

```
{
  "success": true,
  "data": [
    {
      "_id": "63cd72fe9ca53b43a22cc1b5",
      "name": "Abhishek",
      "salary": 145000,
      "currency": "USD",
      "department": "Engineering",
      "sub_department": "Platform"
    },
    {
      "_id": "63cd72fe9ca53b43a22cc1b7",
      "name": "Himani",
      "salary": 240000,
      "currency": "USD",
      "department": "Engineering",
      "sub_department": "Platform"
    }
  ],
  "count": 2,
  "filter": {
    "department": "Engineering",
    "sub_department": "Platform"
  }
}
```

#12 Fetch max(), min(), mean() salary by department

Method: **GET**

Url: **http://localhost:8080/api/employee/department/Engineering?salary=max**

Header: Authorization: {{token}}

Success Response:

```
{
  "success": true,
  "data": [
    {
      "_id": "Banking",
      "salary": 90000
    },
    {
      "_id": "Administration",
      "salary": 30
    },
    {
      "_id": "Operations",
      "salary": 70000
    },
    {
      "_id": "Engineering",
      "salary": 200000000
    }
  ],
  "count": 4,
  "filter": {
    "department": "Engineering"
  },
  "aggregation": "max"
}
```

#13 Fetch max(), min(), mean() salary by sub_department

Method: GET

Url: http://localhost:8080/api/employee/department/Engineering?sub_department=Platform&salary=max

Header: Authorization: {{token}}

Success Response:

```
{
  "success": true,
  "data": [
    {
      "_id": "CustomerOnboarding",
      "salary": 70000
    },
    {
      "_id": "Platform",
      "salary": 200000000
    },
    {
      "_id": "Agriculture",
      "salary": 30
    },
    {
      "_id": "Loan",
      "salary": 90000
    }
  ],
  "count": 4,
  "filter": {
    "department": "Engineering"
  },
  "aggregation": "max"
}
```

User APIs

#1 Fetch all users

Method: **GET**

Url: **http://localhost:8080/api/user**

Header: Authorization: {{token}}

Success Response:

```
{
  "success": true,
  "data": [
    {
      "_id": "63cd72fe9ca53b43a22cc1b4",
      "username": "administrator"
    }
  ],
  "count": 1
}
```

#2 Create a new user

Method: **POST**

Url: **http://localhost:8080/api/user**

Body:

```
{
  "username": "nikhilgautam",
  "password": "12345"
}
```

Success Response:

```
{
  "success": true,
  "data": {
    "acknowledged": true,
    "insertedId": "63cd7fd79ca53b43a22cc1bf"
  }
}
```

#3 Fetch an user by id

Method: **GET**

Url: **http://localhost:8080/api/user/{{user_id}}**

Success Response:

```
{
  "success": true,
  "data": [
    {
      "_id": "63cd72fe9ca53b43a22cc1b4",
      "username": "administrator"
    }
  ],
}
```

#4 User login

Url: <http://localhost:8080/api/user/login>

Success Response:

#4 Delete an user

Url: **http://localhost:8080/api/user/{{user_id}}**

```
{
  "success": true,
  "data": {
    "acknowledged": true,
    "deletedCount": 1
  }
}
```