Hackathon- Node JS

Time Duration: 2 hours

- Read the problem statement thoroughly
- Must follow coding standards
- To upload in ION, upload zip file with name CT/DTReferenceno.zip
- Make sure your project is free from compilation errors.
- Please refer Test Runner App Instructions document for steps to use Test Runner App.
- For any technical support on the TestRunnerApp please connect on the Webex details provide in support section of this document between 11 AM to 1:30 PM during the Hackathon.

- Develop a Node JS Web application using Express Framework to perform Employee related operations.

The services that can be performed are

- List all employees
- List employee by id
- Add Employee
- Update Employee
- Delete Employee

Prerequisites- Below modules are required to be installed and used in application.

If not, please install them using below command.

- **Express module**

Install Express using below command

```
npm install  express
```

- **CORS module**

Install CORS using below command

```
npm install  cors
```

There is an input file employees.json provided in the question which needs to be read and written based on the operations.

The API's to be implemented in this application are given in the below table. Refer the result verification section to know the exact input and output payload format.

Assume the random number generated by the TestRunner is 1234.

API Endpoint	HTTP Method	Input and Output
/1234listEmployees	GET	Fetches the list of all employees from the employees.json file
/1234listEmployeeById/{id}	GET	Fetches the employee details for a given id in the request from the employees.json file
/1234addEmployee/{id}	POST	Method takes an employee payload as input and adds the employee to the employees data in the employees.json. Method returns the list of all employees from the employees.json file
/1234updateEmployee/{id}	PUT	Method takes an employee payload as input and updates the employee to the employees data in the employees.json whose id matches the given id in the request. Method returns the list of all employees from the employees.json file
/1234deleteEmployee/{id}	DELETE	Deletes an employee of given id from the employees data in the employees.json Method returns the list of all employees from the employees.json file.

Result Verification

Below is the list of URLs which can be hit from POSTMAN API after running the above application.

Assuming the port number : 3004 and randomNumber as 1234

Application URLS and payloads – For testing it as a standalone application

- **List all Employees - GET**

URL : http://localhost:3004/1234listEmployees

Output

```
{"employee1":{"name":"Adam","department":"HR","id":1},"employee2":
{"name":"Steve","department":"IT","id":2},"employee3":
{"name":"Ramesh","department":"Admin","id":3}}
```

- **List employee by id (for example 2) – GET**

URL : http://localhost:3004/1234listEmployeeById/2

Output

```
{"name":"Steve","department":"IT","id":2}
```

- **Add Employee with id (for example 4)– POST**

URL : http://localhost:3004/1234addEmployee/4

Input Payload

```
{"name":"Mohit","department":"IT","id":4}
```

Output

```
{"employee1":{"name":"Adam","department":"HR","id":1},"employee2":
{"name":"Steve","department":"IT","id":2},"employee3":
{"name":"Ramesh","department":"Admin","id":3},"employee4":
{"name":"Mohit","department":"IT","id":4}}
```

- **Updating Employee for an id (for example 3)– PUT**

URL: http://localhost:3004/1234updateEmployee/3

Input Payload

```
{"name":"Ramesh","department":"Finance","id":3}
```

Output

```
{"employee1":{"name":"Adam","department":"HR","id":1},"employee2":
{"name":"Steve","department":"IT","id":2},"employee3":
{"name":"Ramesh","department":"Finance","id":3},"employee4":
{"name":"Mohit","department":"IT","id":4}}
```

- **Deleting Employee with id (for example 4)- DELETE**

URL : http://localhost:3004/1234deleteEmployee/4

Output

```
{"employee1":{"name":"Adam","department":"HR","id":1},"employee2":
{"name":"Steve","department":"IT","id":2},"employee3":
{"name":"Ramesh","department":"Finance","id":3}}
```

Support:

URL: https://tcs2.webex.com

Session Number:

1701451800

Password:

Xpr123