EmployeeDetailsWebApi

This is a ASP.NET Core 3.1 WebApi REST API application. This application is using an entityframework core, services layer and REST Controller to expose the REST API call. Also application is using the dependency injection from ASP.net core.

Application is using in-memory database provided by ASP.net core.

Application is available at following GitHub repository.

https://github.com/dshubhangid/EmployeeDetailsWebApi.git

Prerequisite

```
ASP.net core 3.1 or more
```

Deployment

```
1. Clone the above repository to local machine.

2. Open application with Visual Studio 2019

3. Run the application as default IISExpress.

Applicaiton is running on fixed port number "4001" or "4000".

On local development application is deployed and accessible on below URL.

https://localhost:4001/api/employee

Base URL: `https://localhost:4001/`

Controller route: `api/employee`
```

REST API

Application expose following REST API to access the resouce (Employee).

```
1. Get list of all resource (GET)
2. Get resource by id (GET)
3. Create resource (POST)
4. Update resource (PUT)
5. Delete resource (DELETE)
```

Application Overview

The is application developed to maintain the Employee details. WebApi provides rest api to create, update, delete and list all the employess.

REST Api

1. **Get list of all available Employees**

```
Method: `GET`
URL: `https://localhost:4001/api/employee`
```

2. **Get employee by id**

```
Method: `GET`
URL: `https://localhost:4001/api/employee/{employeeId}`

For example:

`https://localhost:4001/api/employee/1`
```

3. **Create new Employee**

```
Method: `POST`
URL: `https://localhost:4001/api/employee`
Body as Json:

{
    "FirstName": "Peter",
    "LastName": "Parker",
    "Organization": "Spider Man",
    "Position": "Actor",
    "Address": "xyz street 15",
    "City": "London",
    "Gender": "Male",
    "Salary": 3000,
    "Department": "Movies"
}
```

4. **Update existing Employee**

```
Method: `PUT`

URL: `https://localhost:4001/api/employee/{employeeId}`
(for example: https://localhost:4001/api/employee/1)

Body as Json:

{
    "FirstName": "Peter",
    "LastName": "Parker",
    "Organization": "Spider Man 22",
    "Position": "Actor",
    "Address": "xyz street 15",
    "City": "New York",
    "Gender": "Male",
    "Salary": 5000,
    "Department": "Movies"
}
```

5. **Delete existing Employee**

```
Method: `DELETE`
URL: `https://localhost:4001/api/employee/{employeeId}`

(for example: https://localhost:4001/api/employee/1)
```