

Express.js Assignment

Guidelines:

- i. **This assignment is mandatory for everyone**
- ii. **There will only be a single attempt for each exam and no deadline extension in case of assignments**
- iii. **Any case of unfair means or plagiarism would lead to debarring in final placements without any further consideration.**
- iv. The assignment solutions should be uploaded on Github and links to Github repositories should be shared with the coach for code review. Make sure to add appropriate comments in code wherever possible.

Problem 1:

Build an API to create and retrieve college student details

The screenshot displays a REST client interface with the following details:

- Request Method:** POST (indicated by a red arrow labeled "Request Method")
- Request URL:** http://localhost:3000/student/add (indicated by a red arrow labeled "Request URL")
- Request Body (JSON):**

```
1 {
2   "studentFirstName": "John",
3   "collegeName": "IIT",
4   "location": "Mumbai"
5 }
```

(indicated by a red arrow labeled "Request JSON")
- Response Body (JSON):**

```
1 {
2   "result": "Success"
3 }
```

(indicated by a red arrow labeled "Request Result")

Request URL:- `http://localhost:3000/student/add`

Request Method:- POST

Request JSON:-

```
{
  "studentFirstName": "John",
  "collegeName": "IIT",
  "location": "Mumbai"
}
```

Response JSON:-

```
{
  "result": "Success"
}
```

Objective:-

1. Build an API which accepts the above parameters as a POST request using express.js
2. Use Node.js fs module to store the received details JSON in a file. You can refer to the link below for the process:-

<https://www.geeksforgeeks.org/how-to-read-and-write-json-file-using-node-js/>

3. Build a second API/route with which the stored details are sent back to client as shown below:-

The screenshot displays a REST client interface with the following components:

- Request Tab:** Shows a GET request to `http://localhost:3000/student/getDetails`. A red arrow points to the `GET` method with the label "Request Type". Another red arrow points to the URL with the label "Get the details stored using previous API from this URL".
- Query Params Table:** A table with 3 columns: KEY, VALUE, and DESCRIPTION. The first row contains the text "Key", "Value", and "Description".
- Body Tab:** Shows the response body in JSON format. A red arrow points to the response with the label "Sample response".
- Status Bar:** Indicates "Status: 200 OK" and "Time: 38 ms".

| KEY | VALUE | DESCRIPTION |
|-----|-------|-------------|
| Key | Value | Description |

```
{
  "studentFirstName": "John",
  "collegeName": "IIT",
  "location": "Mumbai"
}
```

Request URL:- `http://localhost:3000/student/getDetails`

Request Method:- GET

Response JSON:-

```
{
  "studentFirstName": "John",
  "collegeName": "IIT",
  "location": "Mumbai"
}
```

Learnings:

How to make APIs using express.js.

Problem 2:

Create an API using express.js and deploy it to Heroku

Students List Example

GET https://ancient-garden-29926.herokuapp.com/student/studentsList

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

| KEY | VALUE | DESCRIPTION |
|-----|-------|-------------|
| Key | Value | Description |

Request Method

Should be a heroku app URL

Body Cookies Headers (8) Test Results Status: 200 OK Time: 253 ms

Pretty Raw Preview Visualize JSON

```
1 {
2   "results": [
3     "Rajesh",
4     "Ramesh",
5     "Sayali",
6     "Sanjana"
7   ]
8 }
```

Response JSON

Request URL:- https://[Heroku-Base-URL]/student/studentsList

Request Method:- GET

Response:-

```
{
  "results": [
    "Rajesh",
    "Ramesh",
    "Sayali",
    "Sanjana"
  ]
}
```

Objective:-

1. Create an API using express.js on your machine which return the below response:-

```
{
  "results": [
    "Rajesh",
    "Ramesh",
    "Sayali",
    "Sanjana"
  ]
}
```

2. Create an account on Heroku:-
<https://www.heroku.com/>
3. Install Heroku CLI on your machine
<https://devcenter.heroku.com/articles/heroku-cli>
4. Deploy your API to Heroku using below guide:
<https://devcenter.heroku.com/articles/deploying-nodejs>
5. Share the Heroku URL and github repo of the project

Learnings

How to deploy an API on a server.