

ENSF-381: Full Stack Web Development Laboratory Lab04

GitHub Link:

<https://github.com/corelogic-bit/ENSF381-lab04.git>

Group Information:

Name 1 (UCID#): Noor Haj Yousef (30206296)

Name 2 (UCID#): Himaal Ishaq (30249404)

Exercise 2:

Screenshot from Step 9:

The screenshot shows the 'NEW RESOURCE' dialog in Mockapi.io. The background interface includes a sidebar with 'Projects / Full Stack' and 'API endpoint' section showing 'https://@a1d6k' and a 'New resource' button. The main area has 'all' and 'Reset all' buttons.

NEW RESOURCE

Resource name
Enter meaningful resource name, it will be used to generate API endpoints.
users_api

Schema
Define Resource schema, it will be used to generate mock data.

id Object ID

first_name Faker.js name.firstName

user_group Number

Object template Requires subscription [See plans](#)
To define more complex structure for your data, use JSON template. You can reference Faker.js function using `$function_name`.

```
{
  "username": "$internet.userName",
  "knownIp": ["$internet.ip", "$internet.ipv6"],
  "profile": {
    "firstName": "$name.firstName",
    "lastName": "$name.lastName",
    "statisticData": [100, 200, 300]
  }
}
```

Endpoints Requires subscription [See plans](#)
Enable/disable endpoints and customize response JSON.
By default, Mockapi will return either a list of items or a single item, depending on the request. To define more complex response, provide JSON template. You can reference Faker.js methods using `$function_name`.

- `$mockData` - data stored in DB.
- `$count` - number of records stored in DB.

`$mockData` EXAMPLE

```
{
  "requestId": "$datatype.uuid",
  "items": "$mockData",
  "count": "$count",
  "anyKey": "anyValue"
}
```

☒ GET /users_api \$mockData

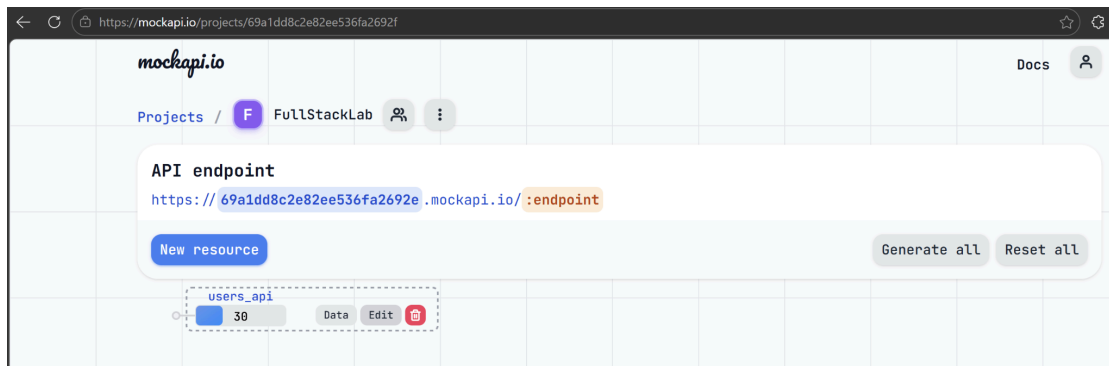
☒ GET /users_api/:id \$mockData

☒ POST /users_api \$mockData

☒ PUT /users_api/:id \$mockData

☒ DELETE /users_api/:id \$mockData

Screenshot from Step 12:

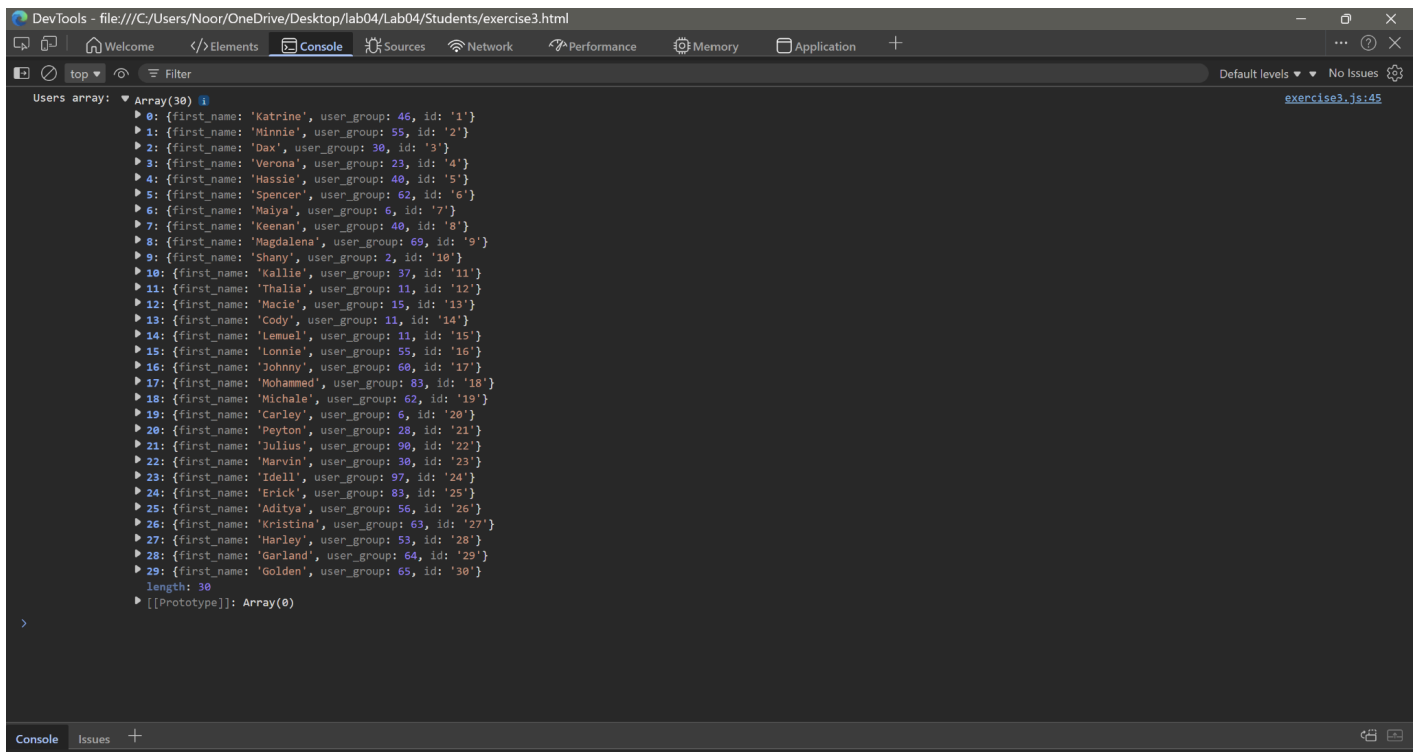


Screenshot from Step 14:



Exercise 3:

Screenshot A:



Screenshot B:

User Directory

View users and delete a user by ID from the API.

Controls

Delete by ID

e.g. 12

Delete

Sort by Group

Sort by ID

Grid / List

User Boxes

Katrina

first_name: Katrina
user_group: 46
id: 1

Minnie

first_name: Minnie
user_group: 55
id: 2

Dax

first_name: Dax
user_group: 30
id: 3

Verona

first_name: Verona
user_group: 23
id: 4

Haasle

first_name: Haasle
user_group: 40
id: 5

Spencer

first_name: Spencer
user_group: 62
id: 6

Malya

first_name: Malya
user_group: 6
id: 7

Keenan

first_name: Keenan
user_group: 40
id: 8

Magdalena

first_name: Magdalena
user_group: 69
id: 9

Shany

first_name: Shany
user_group: 2
id: 10

Kallie

first_name: Kallie
user_group: 37
id: 11

Thalia

first_name: Thalia
user_group: 11
id: 12

Macle

first_name: Macle
user_group: 15
id: 13

Cody

first_name: Cody
user_group: 11
id: 14

Lemuel

first_name: Lemuel
user_group: 11
id: 15

Lonnie

first_name: Lonnie
user_group: 55
id: 16

Johnny

first_name: Johnny
user_group: 60
id: 17

Mohammed

first_name: Mohammed
user_group: 83
id: 18

Michale

first_name: Michale
user_group: 62
id: 19

Carley

first_name: Carley
user_group: 6
id: 20

Peyton

first_name: Peyton
user_group: 28
id: 21

Julius

first_name: Julius
user_group: 90
id: 22

Marvin

first_name: Marvin
user_group: 30
id: 23

Idell

first_name: Idell
user_group: 97
id: 24

Erick

first_name: Erick
user_group: 83
id: 25

Aditya

first_name: Aditya
user_group: 56
id: 26

Kristina

first_name: Kristina
user_group: 63
id: 27

Harley

first_name: Harley
user_group: 53
id: 28

Garland

first_name: Garland
user_group: 64
id: 29

Golden

first_name: Golden
user_group: 65
id: 30