API Chaining

Example 1: Students API

Step1: Send Post request (create student) then collect "id" from response.

http://localhost:3000/students

Request body

```
{
    "name": "Scott",
    "location": "France",
    "phone": "123456",
    "courses": [
        "C",
        "C++"
    ]
}
```

Tests

Include below script in *Tests* to collect id as environment variable.

```
//Capturing id from response
var jsonData = JSON.parse(responseBody);
pm.environment.set("id",jsonData.id);
```

**Step2: Send Get request (display student) using for "id" captured from previous request.**

http://localhost:3000/students/**{{id}}**

Step3: Send Delete request (delete student) using for "id" captured from previous request.

http://localhost:3000/students/**{{id}}**

Example 2: GoRest API (Users)

https://gorest.co.in/

- Do not post your personal data like name, email, phone, photo etc...
- For paged results parameter "page" should be passed in url ex: GET `/public/v2/users?page=1`
- Request methods PUT, POST, PATCH, DELETE needs access token, which needs to be passed with "Authorization" header as Bearer token.
- API Versions `/public-api/*`, `/public/v1/*` and `/public/v2/*`
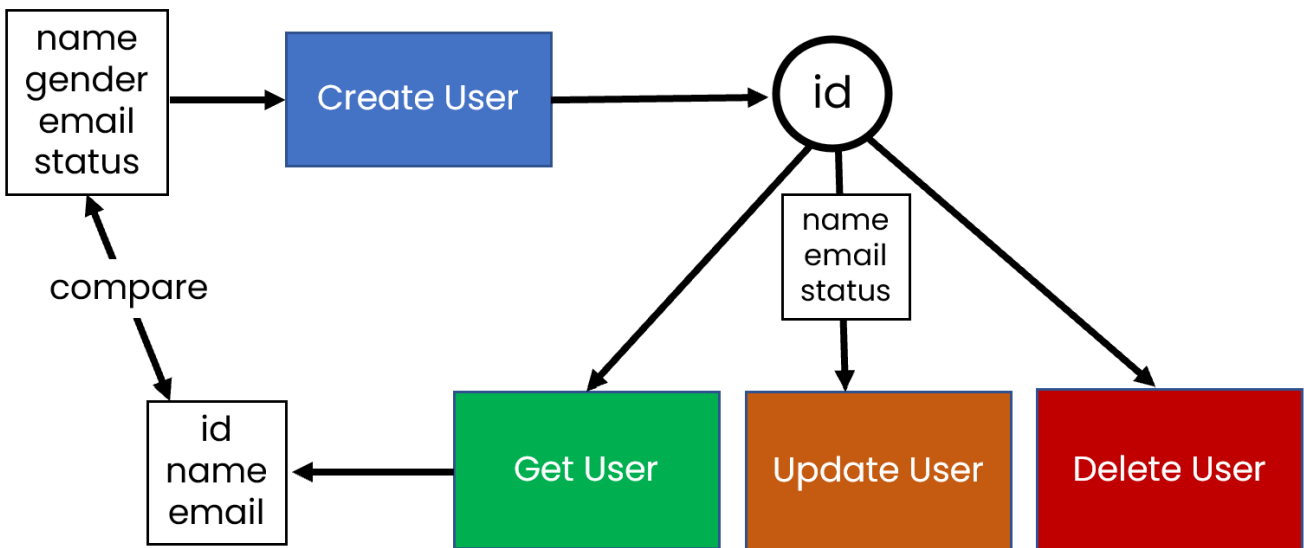- Get your access token
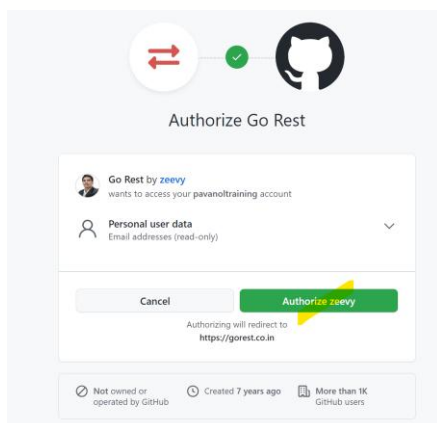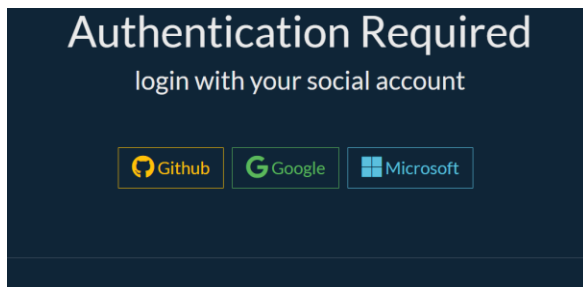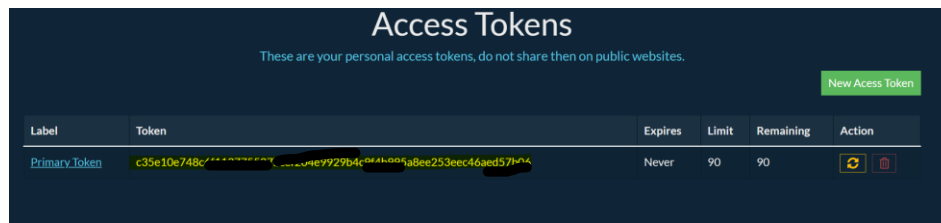
Access Token :

c35e1█████████████████bcef204e9929b4c9f4b995a8ee25
3eec46aed57b06

https://gorest.co.in

POST	/public/v2/users	Create a new user
GET	/public/v2/users/23	Get user details
PUT|PATCH	/public/v2/users/23	Update user details
DELETE	/public/v2/users/23	Delete user

Create User (Post)

{{url}}/public/v2/users

Request Body

```
{
    "name": "{{name_env}}",
    "gender": "male",
    "email": "{{email_env}}",
    "status": "inactive"
}
```

Pre-Request script

```
var random=Math.random().toString(36).substring(2);

var useremail="Jim"+random+"@gmail.com";
var username="Jim"+random;
```

```
//console.log(useremail);    // print to know value
//console.log(username);     // print to know value

pm.environment.set("email_env",useremail);
pm.environment.set("name_env",username)
```

Tests
```
//Capturing id from response & Set as envirnment var
var jsonData = JSON.parse(responseBody);
pm.environment.set("userid_env",jsonData.id);
```

Get User details (Get)

{{url}}/public/v2/users/{{userid_env}}

Tests
```
//Validating JSON fields in Response
pm.test("values of fields in response", () => {
    var jsonData = pm.response.json();

    pm.expect(jsonData.id).to.eql(pm.environment
.get("userid_env"));
    pm.expect(jsonData.email).to.eql(pm.environm
ent.get("email_env"));
    pm.expect(jsonData.name).to.eql(pm.environme
nt.get("name_env"));
    });
```

Update User details (Put)

{{url}}/public/v2/users/{{userid_env}}

Request Body

```
{
    "name": "{{name_env}}",
    "email": "{{email_env}}",
    "status": "active"
}
```

Pre-Request script

```
var random=Math.random().toString(36).substring(2);

var useremail="Jim"+random+"@gmail.com";
var username="Jim"+random;

pm.environment.set("email_env",useremail);
pm.environment.set("name_env",username)
```

Delete user (Delete)

{{url}}/public/v2/users/{{userid_env}}