

Name: Arthur M. Artugue

Year and section: IV-BCSAD

Assignment 1 - REST API

CREATE / POST

The screenshot displays a REST client interface with a dark theme. At the top, there are tabs for 'Overview', 'RESTful API Basics #blueprint', 'POST Get data', and 'POST Post data'. The 'POST Get data' tab is active. Below the tabs, the URL bar shows 'POST' and '({dev_url}) /api/v0/users'. A 'Send' button is to the right. Below the URL bar, there are tabs for 'Params', 'Authorization', 'Headers (10)', 'Body', 'Scripts', and 'Settings'. The 'Body' tab is active, showing a JSON body:

```
1 {
2   "first_name": "Arthur",
3   "last_name": "Artugue",
4   "email": "arthur.gmail.com"
5 }
```

 Below the body, there are tabs for 'Body', 'Cookies', 'Headers (5)', and 'Test Results (1/1)'. The 'Test Results' tab is active, showing a green status bar with '200 OK', '85 ms', and '236 B'. Below the status bar, there are tabs for 'JSON', 'Preview', and 'Visualize'. The 'JSON' tab is active, showing the same JSON body as above.

Overview RESTful API Basics #blueprint POST Get data POST Post data + No environment

RESTful API Basics #blueprint / Get data Save Share

POST ({dev_url}) /api/v0/users Send

Params Authorization Headers (10) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "first_name": "Arthur",
3   "last_name": "Artugue",
4   "email": "arthur.gmail.com"
5 }
```

Body Cookies Headers (5) Test Results (1/1) 200 OK 85 ms 236 B Save Response

{ JSON Preview Visualize

```
1 {
2   "first_name": "Arthur",
3   "last_name": "Artugue",
4   "email": "arthur.gmail.com"
5 }
```

READ / GET

- Retrieve all users

The screenshot shows a REST client interface with a GET request to `[[dev_url]] /api/v0/users`. The response is a 200 OK status with a response time of 77 ms and a body size of 312 B. The response body is a JSON array containing two user objects.

```
1 [
2   {
3     "first_name": "Arthur",
4     "last_name": "Artugue",
5     "email": "arthur.gmail.com"
6   },
7   {
8     "first_name": "Leila",
9     "last_name": "Montemayor",
10    "email": "leila@gmail.com"
11  }
12 ]
```

- Retrieve specific user by ID

The screenshot shows a REST client interface with a GET request to `[[dev_url]] /api/v0/users/1`. The response is a 200 OK status with a response time of 12 ms and a body size of 236 B. The response body is a JSON object representing a single user.

```
1 {
2   "first_name": "Arthur",
3   "last_name": "Artugue",
4   "email": "arthur.gmail.com"
5 }
```

UPDATE / PUT

- Updating user information (email)

The screenshot shows the REST Client interface with a PUT request to `PUT {{dev_url}} /api/v0/users/1`. The request body is a JSON object: `{ "first_name": "Arthur", "last_name": "Artugue", "email": "arthur@gmail.com" }`. The response is a `200 OK` status with a response time of 17 ms and a body size of 236 B. The response body is also a JSON object: `{ "first_name": "Arthur", "last_name": "Artugue", "email": "arthur@gmail.com" }`.

- Updated user info retrieved using GET protocol

The screenshot shows the REST Client interface with a GET request to `GET {{dev_url}} /api/v0/users`. The response is a `200 OK` status with a response time of 7 ms and a body size of 312 B. The response body is a JSON array of two user objects: `[{ "first_name": "Arthur", "last_name": "Artugue", "email": "arthur@gmail.com" }, { "first_name": "Leila", "last_name": "Montemayor", "email": "leila@gmail.com" }]`.

DELETE / DELETE

- Deleting user

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

- Method:** DELETE
- URL:** `{{dev_url}} /api/v0/users/1`
- Body:** A JSON object representing a user:

```
1 {
2   "first_name": "Arthur",
3   "last_name": "Artugue",
4   "email": "arthur@gmail.com"
5 }
```
- Response:** 200 OK, 13 ms, 123 B
- Body Tab:** Raw

- All users retrieved, and the deleted user is now not in the database

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

- Method:** GET
- URL:** `{{dev_url}} /api/v0/users`
- Body:** A JSON array containing one user object:

```
1 [
2   {
3     "first_name": "Leila",
4     "last_name": "Montemayor",
5     "email": "leila@gmail.com"
6   }
7 ]
```
- Response:** 200 OK, 5 ms, 239 B
- Body Tab:** JSON