

Name: Arthur M. Artugue

Year and section: IV-BCSAD

Assignment 1 - REST API

CREATE / POST

The screenshot shows the Postman application interface. At the top, there are tabs for 'Overview', 'RESTful API Basics #blueprint', 'POST Get data' (which is active), and 'POST Post data'. Below the tabs, the URL is set to `HTTP RESTful API Basics #blueprint / Get data`. The method is selected as 'POST'. The 'Body' tab is active, showing a raw JSON payload:

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

Below the body, the response details are shown: '200 OK' status, '85 ms' duration, '236 B' size, and a 'Save Response' button.

At the bottom, there is a preview of the JSON response body, which matches the posted data:

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

READ / GET

- Retrieve all users

The screenshot shows the Postman interface with a successful API call. The URL is `{{dev_url}} /api/v0/users`. 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 the Postman interface with a successful API call. The URL is `{{dev_url}} /api/v0/users/1`. 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 Postman interface with a PUT request to update user information. The URL is `{{dev_url}}/api/v0/users/1`. The request body is a JSON object:

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

The response status is 200 OK, with a response body identical to the request body.

- Updated user info retrieved using GET protocol

The screenshot shows the Postman interface with a GET request to retrieve user information. The URL is `{{dev_url}}/api/v0/users`. The request body is a JSON object:

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

The response status is 200 OK, with a response body containing a list of users, including the updated one.

```
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 ]
```

DELETE / DELETE

- Deleting user

The screenshot shows the Postman interface with a DELETE request to {{dev_url}} /api/v0/users/1. The request body contains the following JSON:

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

The response status is 200 OK, indicating the user was successfully deleted.

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

The screenshot shows the Postman interface with a GET request to {{dev_url}} /api/v0/users. The request body contains the following JSON:

```
1 [
2   {
3     "first_name": "Arthur",
4     "last_name": "Artugue",
5     "email": "arthur@gmail.com"
6   }
7 ]
```

The response status is 200 OK, showing the remaining users in the database.