

# RESTful API

*baseUrl: <https://fakerestapi.azurewebsites.net/index.html>*

## Test Plan

### Unit test (Positive Cases)

For each request:

- Assert response status code is 200.
- Assert response body is as expected by a contract test.
- Assert request execution time is less than 200ms .

1. GET /api/v1/Activities (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** No)
2. POST /api/v1/Activities (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** Yes)
3. GET /api/v1/Activities/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** No)
4. PUT /api/v1/Activities/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** Yes)
5. DEL /api/v1/Activities/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** No)
6. GET /api/v1/Authors (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** No)
7. POST /api/v1/Authors (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** Yes)
8. GET /api/v1/Authors/authors/books/:idBook (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** No)
9. GET /api/v1/Authors/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** No)
10. PUT /api/v1/Authors/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** Yes)
11. DEL /api/v1/Authors/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** No)
12. GET /api/v1/Books (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** No)
13. POST /api/v1/Books (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** Yes)
14. GET /api/v1/Books/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** No)
15. PUT /api/v1/Books/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** Yes)
16. DEL /api/v1/Books/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** No)
17. GET /api/v1/CoverPhotos (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** No)
18. POST /api/v1/CoverPhotos (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** Yes)
19. GET /api/v1/CoverPhotos/books/covers/:idBook (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** No)
20. GET /api/v1/CoverPhotos/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** No)

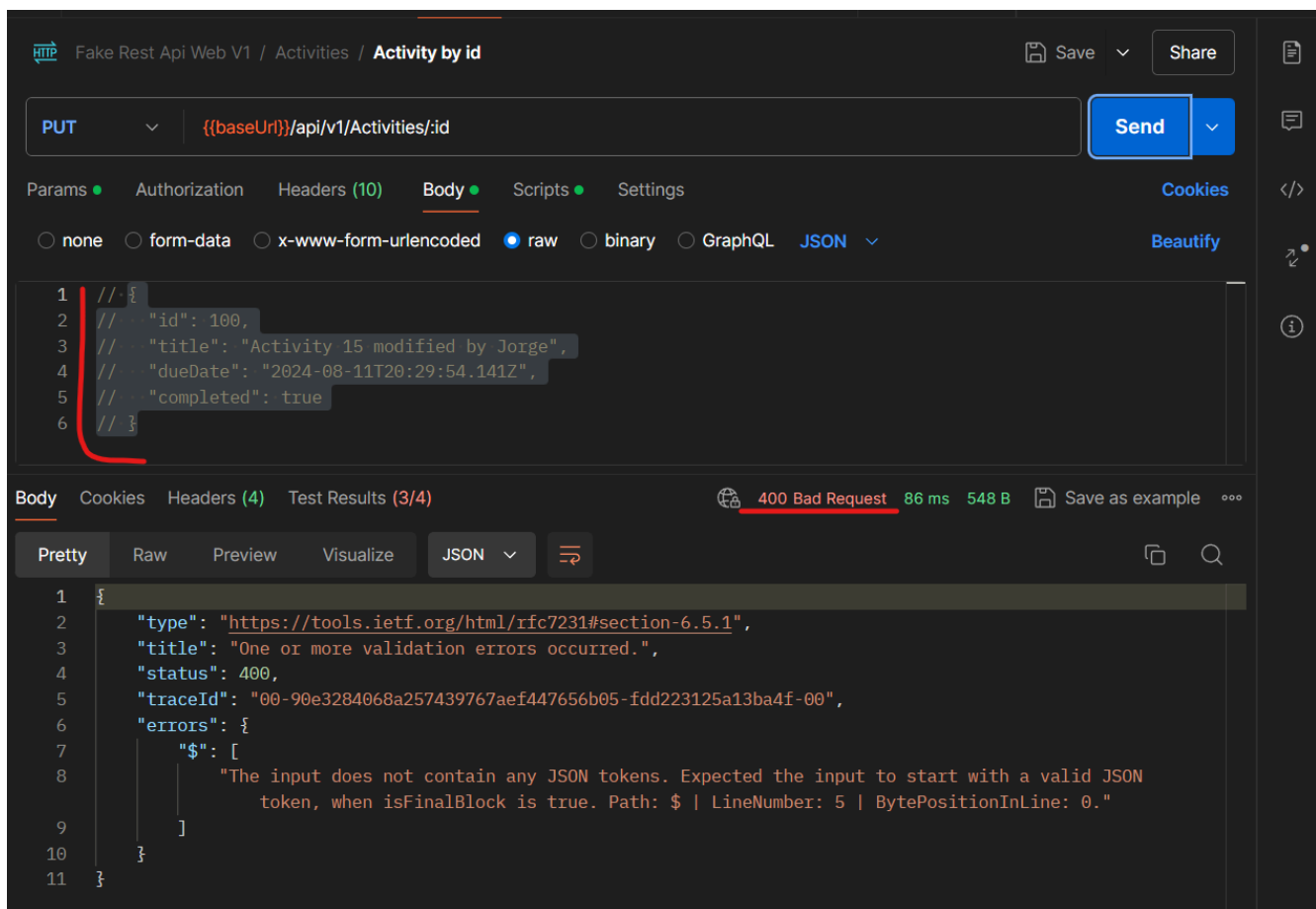
21. PUT /api/v1/CoverPhotos/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** Yes)
22. DEL /api/v1/CoverPhotos/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** No)
23. GET /api/v1/Users (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** No)
24. POST /api/v1/Users(**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** Yes)
25. GET /api/v1/Users/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** No)
26. PUT /api/v1/Users/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** Yes)
27. DEL /api/v1/Users/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** No)

**Unit test (Negative Cases):** Same APIs requests like above but just sending the request without parameter or body for example

For each request:

- Assert response status code is 4XX.
- Assert request execution time is less than 200ms .

Example 1: PUT /api/v1/Activities/:id (**Param:** Yes, **Authorization:** No, **Header:** Yes, **Body:** **NO**)



The screenshot shows a REST client interface for a PUT request to `/api/v1/Activities/:id`. The request body is set to raw JSON, but it contains a syntax error: a trailing comma at the end of the last object property. The response is a 400 Bad Request with a detailed error message in the JSON body.

```

1 // {
2 //   "id": 100,
3 //   "title": "Activity 15 modified by Jorge",
4 //   "dueDate": "2024-08-11T20:29:54.141Z",
5 //   "completed": true
6 // }

```

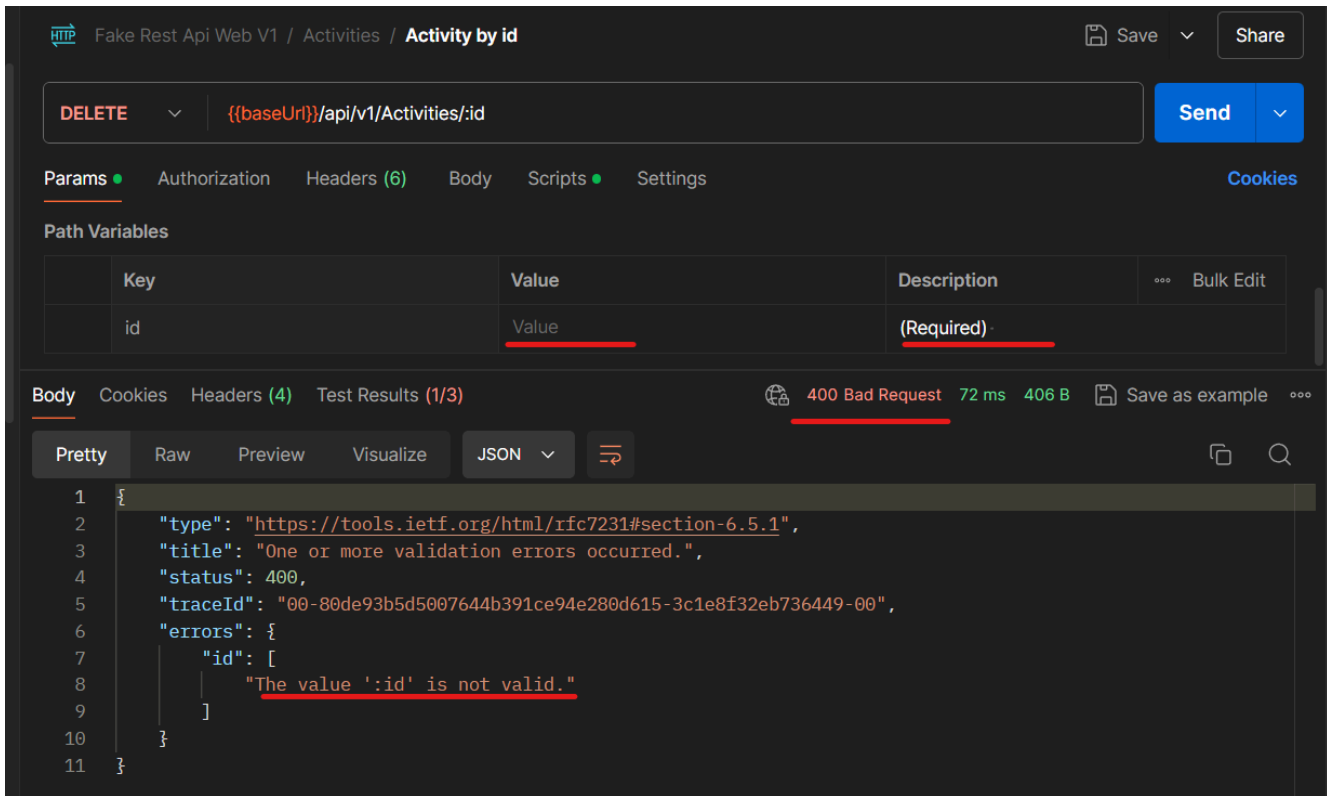
**Body** Cookies Headers (4) Test Results (3/4) 400 Bad Request 86 ms 548 B Save as example

```

1 {
2   "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
3   "title": "One or more validation errors occurred.",
4   "status": 400,
5   "traceId": "00-90e3284068a257439767aef447656b05-fdd223125a13ba4f-00",
6   "errors": {
7     "$": [
8       "The input does not contain any JSON tokens. Expected the input to start with a valid JSON token, when isFinalBlock is true. Path: $ | LineNumber: 5 | BytePositionInLine: 0."
9     ]
10  }
11 }

```

Example 2: DEL /api/v1/Activities/:id (Param: NO, Authorization: No, Header: Yes, Body: No)



The screenshot shows the Fake REST API client interface. The top bar indicates the API is 'Fake Rest Api Web V1' and the current endpoint is 'Activities / Activity by id'. The method is set to 'DELETE' and the URL is '{{baseUrl}}/api/v1/Activities/:id'. The 'Send' button is visible.

Below the URL bar, the 'Params' tab is selected. It shows a table for 'Path Variables':

| Key | Value | Description |
|-----|-------|-------------|
| id  | Value | (Required)  |

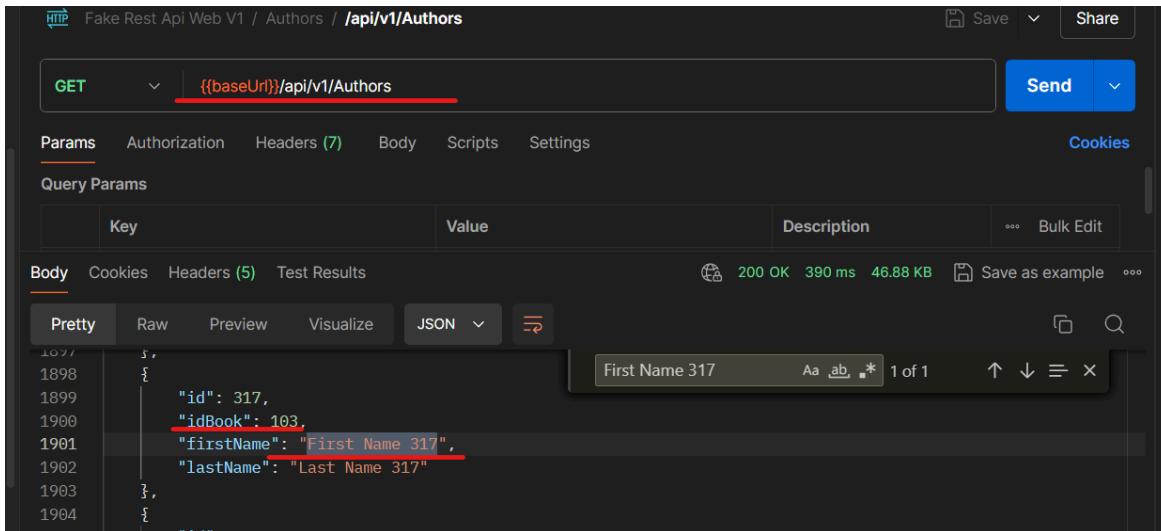
The 'Body' tab is also visible, showing the response details: '400 Bad Request', '72 ms', and '406 B'. The response body is displayed in JSON format:

```
1 {
2   "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
3   "title": "One or more validation errors occurred.",
4   "status": 400,
5   "traceId": "00-80de93b5d5007644b391ce94e280d615-3c1e8f32eb736449-00",
6   "errors": {
7     "id": [
8       "The value ':id' is not valid."
9     ]
10  }
11 }
```

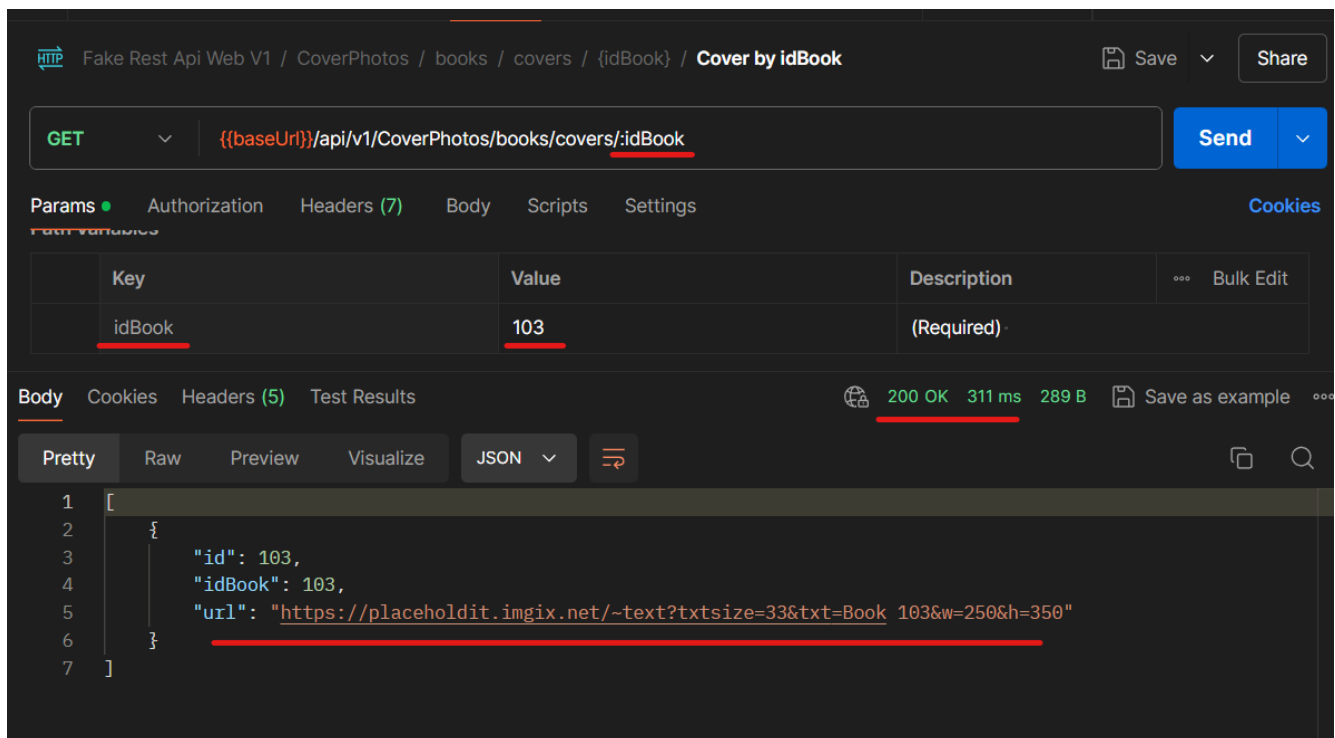
## Integration test

Example: Get the cover photo of the book which author first name is “First Name 317”

- GET request to `/api/v1/Authors` (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** No)
- Search for “First Name 317”



- GET `/api/v1/CoverPhotos/books/covers/:idBook` (**Param:** 103, **Authorization:** No, **Header:** Yes, **Body:** No)



## E2E test

*Scenario: If not existing, add a “MeDirect Assessment” book authored by Jorge and record it as an activity.*

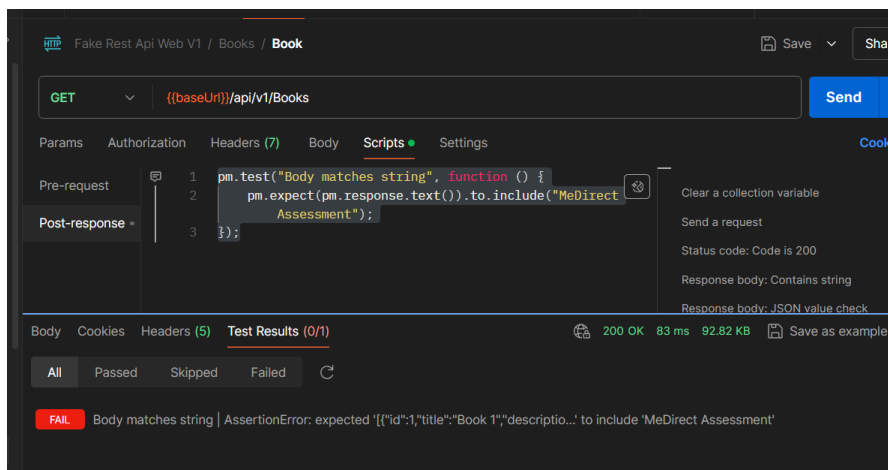
- GET /api/v1/Books (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** No) and search for the book name with a script
- POST /api/v1/Books (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** Yes)
- GET /api/v1/Authors/authors/books:idBook (**Param:** idBook=777, **Authorization:** No, **Header:** Yes, **Body:** No) and get the id
- PUT /api/v1/Books:id (**Param:** idBook=777, **Authorization:** No, **Header:** Yes, **Body:** yes)
- POST /api/v1/Activities (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** Yes)

## Test Execution

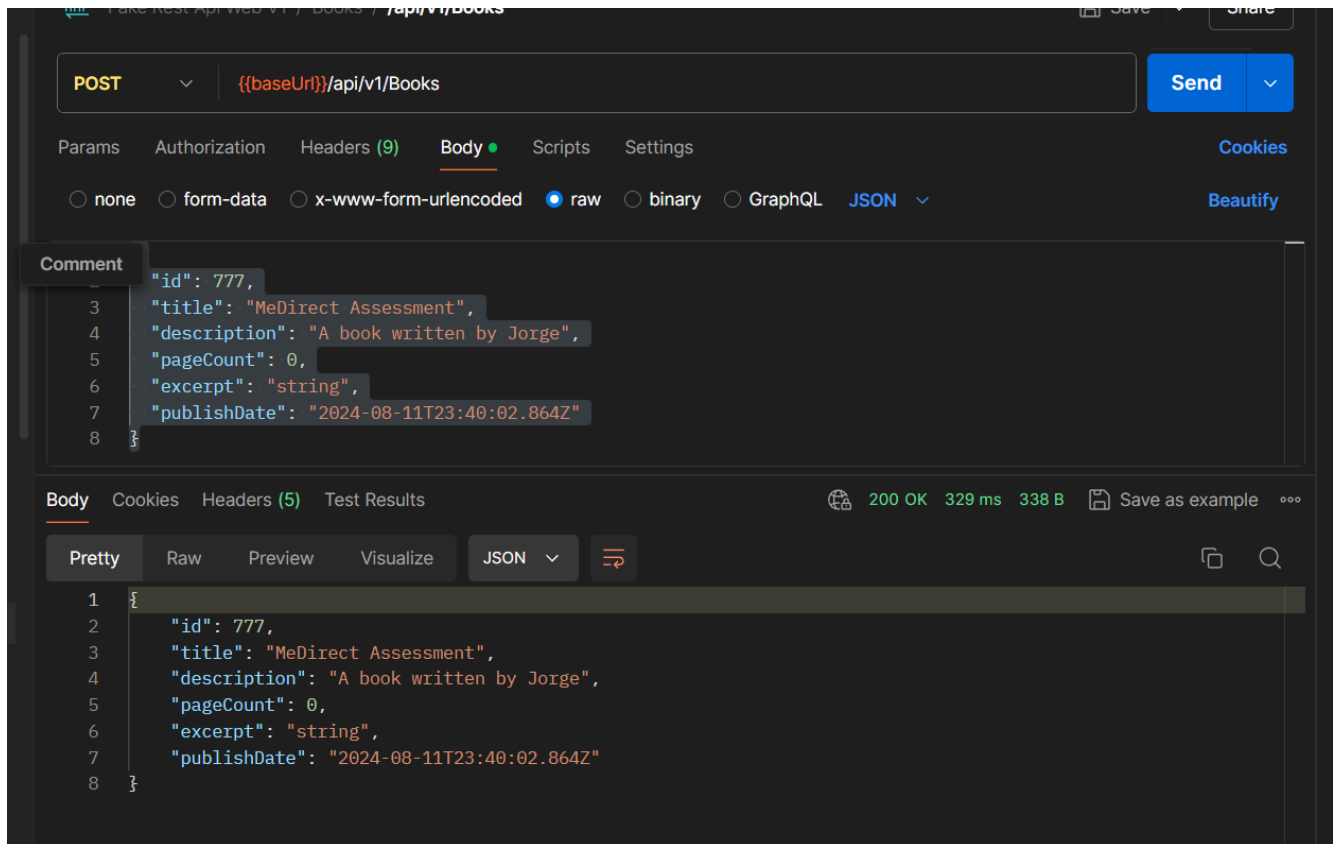
E2E:

*Scenario: If not existing, add a “MeDirect Assessment” book authored by Jorge and record it as an activity.*

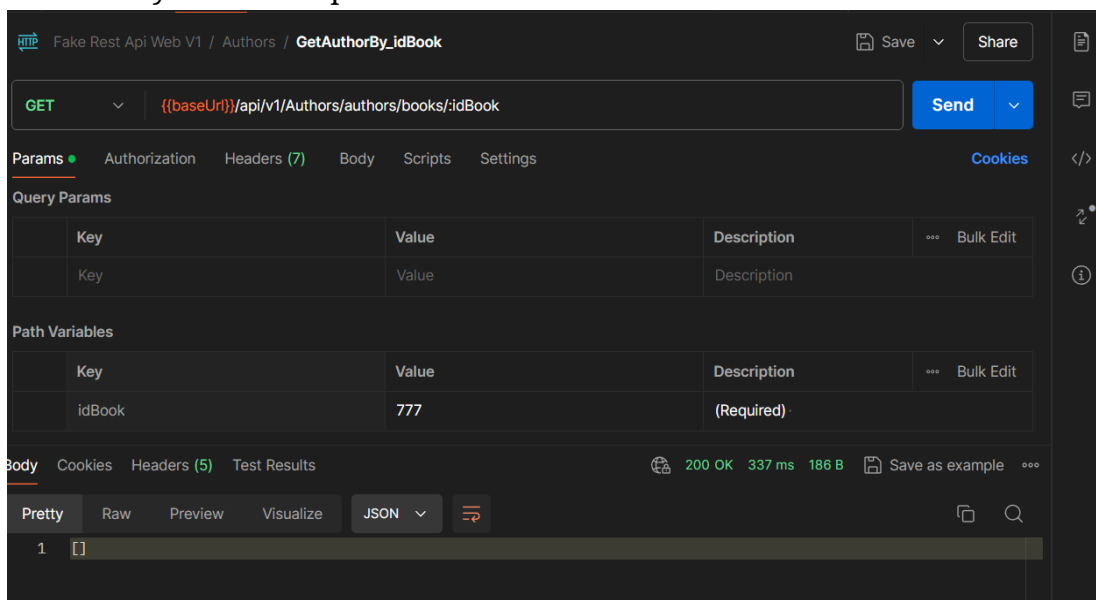
- GET /api/v1/Books (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** No) and search for the book name with the script **Passed**



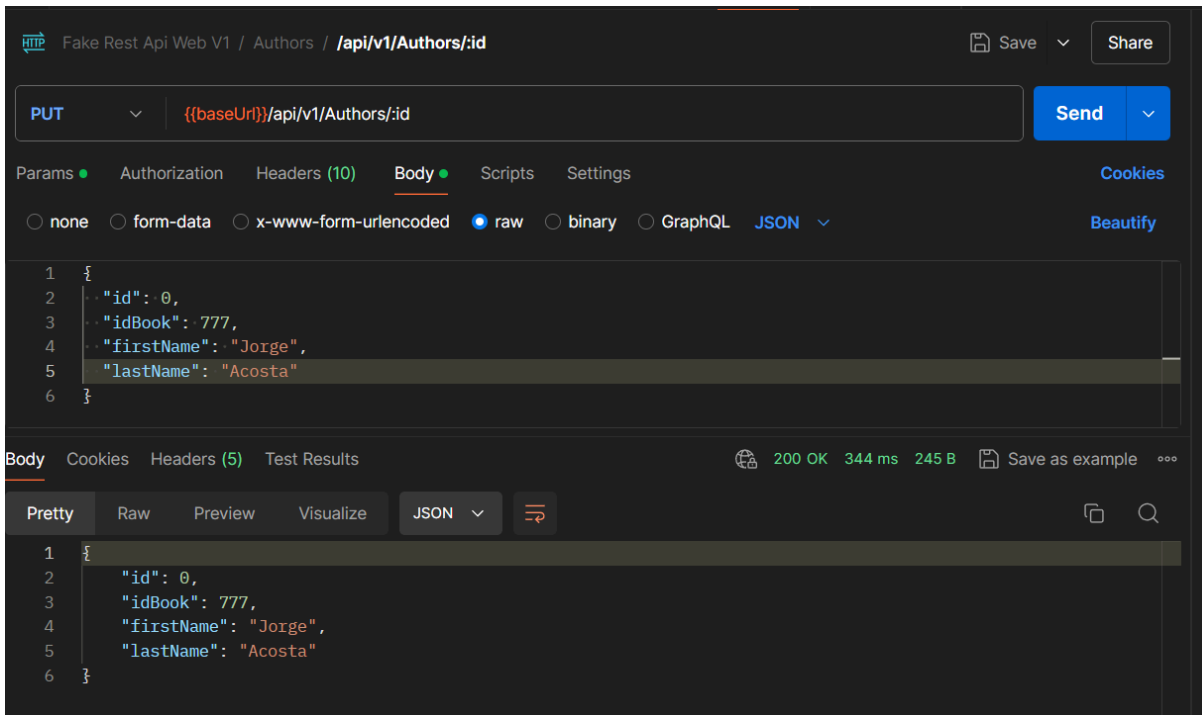
- POST /api/v1/Books (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** Yes) with body: Passed



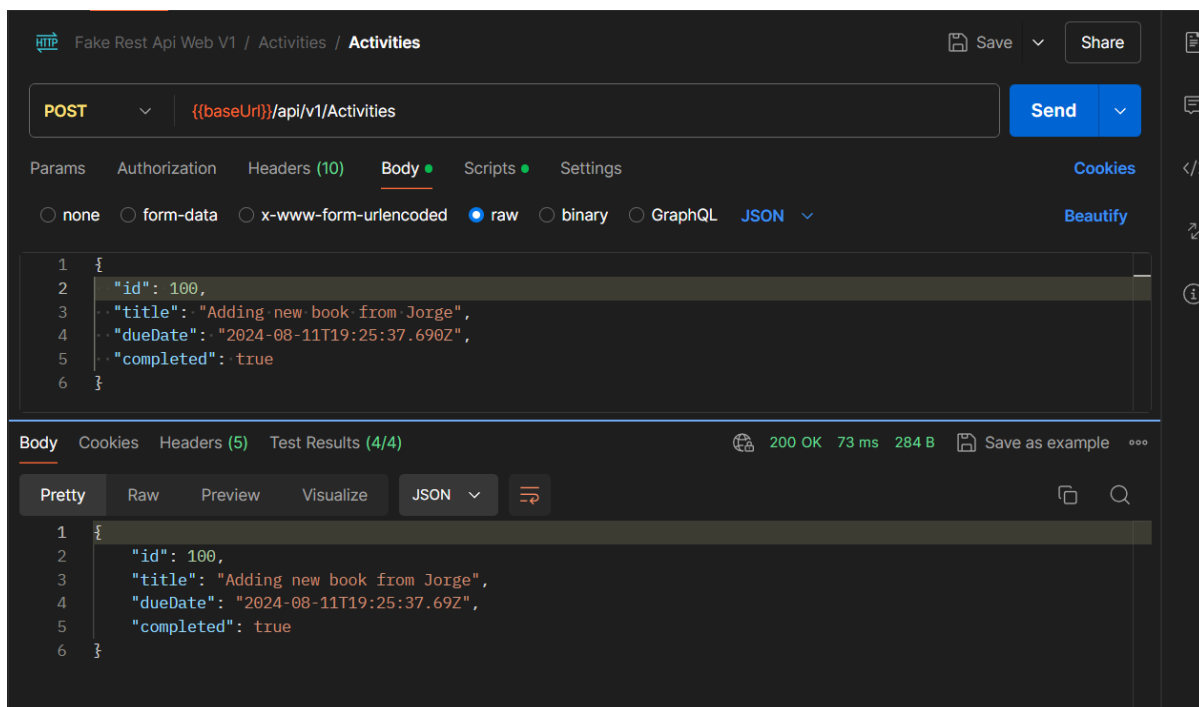
- GET /api/v1/Authors/authors/books:idBook (**Param:** idBook=777, **Authorization:** No, **Header:** Yes, **Body:** No) and get the id **Failed** due to: Response code is 200 but response body is not as expected as idBook 777 does not exist

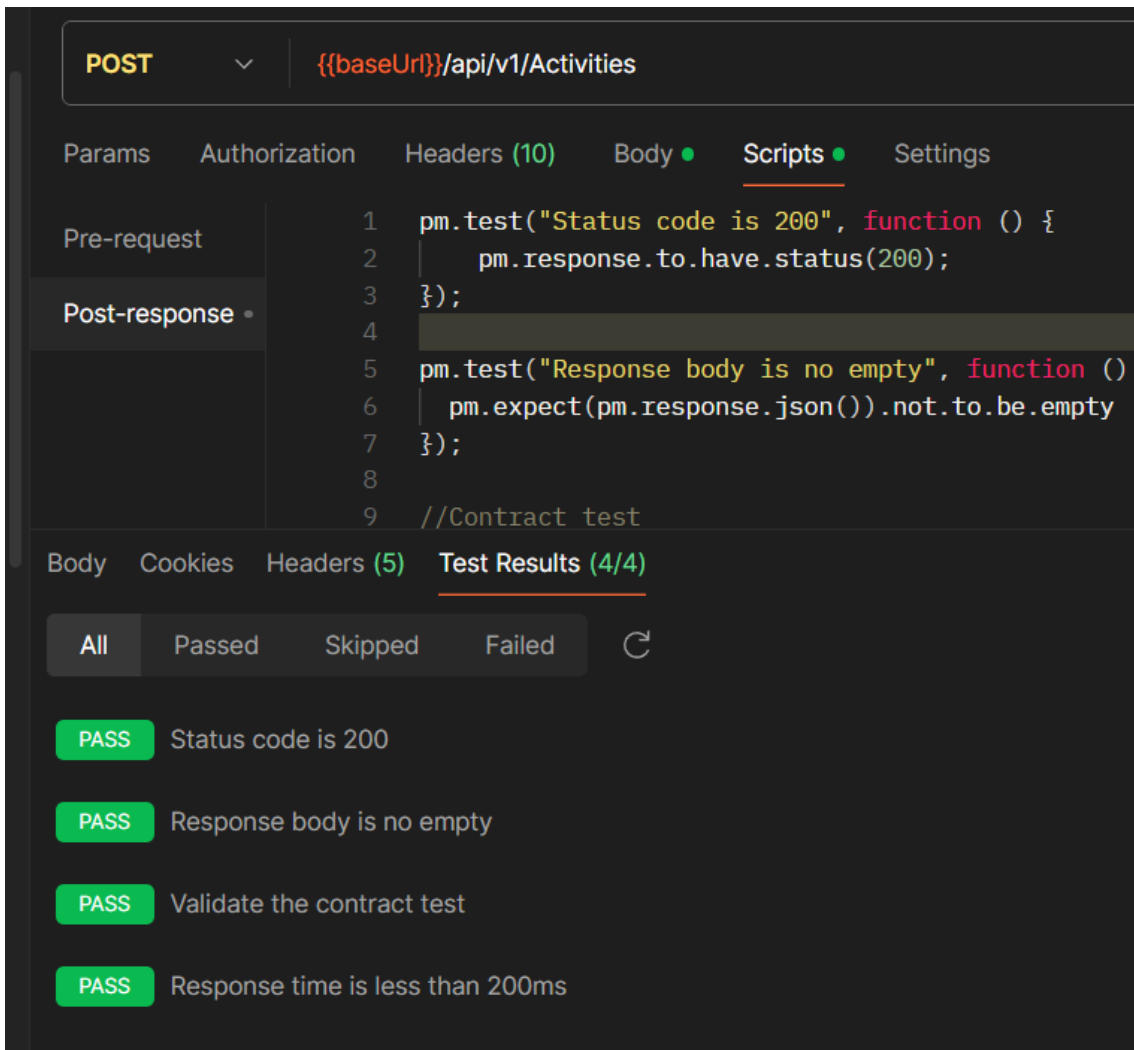


- PUT /api/v1/Books:id (**Param:** idBook=777, **Authorization:** No, **Header:** Yes, **Body:** yes) **Blocked** due to previous step failed as id is not an real value



- POST /api/v1/Activities (**Param:** No, **Authorization:** No, **Header:** Yes, **Body:** Yes) **Blocked** due to previous step failed





## Postman Collection

Check [https://github.com/jorgeacosta19/QA\\_test\\_MeD](https://github.com/jorgeacosta19/QA_test_MeD)

## Improvements (Based on the above tests)

Create the postman collection at an early stage which allows unit testing during development status.

Adding multiple parameters to filter the APIs request will simplify the Integration and E2E as not all will depend on idBook, for example, allow to GET book by title.

Adding Author as part of the response body of GET book API will also help integration between the



*APIS.*