

## Hito 4 - pruebas Postman

NPM tests:

```
Ran all test suites.
root@KHYRON:/home/khyron/Desafio_Latam/hitos/hito_4# npm test

> Hito 4 - macro-photography-api@1.0.0 test
> NODE_ENV=test jest --forceExit --detectOpenHandles

PASS tests/photos.test.js
  Macro Photography API
    GET /api/photos
      ✓ should return all photos (100 ms)
    GET /api/photos/:id
      ✓ should return a specific photo (13 ms)
      ✓ should return 404 for non-existent photo (11 ms)
    POST /api/photos
      ✓ should create a new photo (27 ms)
      ✓ should return 400 for invalid data (14 ms)
    PUT /api/photos/:id
      ✓ should update a photo (10 ms)
      ✓ should return 404 for non-existent photo (11 ms)
    DELETE /api/photos/:id
      ✓ should delete a photo (11 ms)
      ✓ should return 404 for non-existent photo (12 ms)

Test Suites: 1 passed, 1 total
Tests:       9 passed, 9 total
Snapshots:   0 total
Time:        1.545 s
Ran all test suites.
```

Listar todos los registros:

The screenshot shows the Postman interface with a GET request to `http://localhost:3000/api/photos`. The response is a JSON array of two photo objects. The first object has `id: 1`, `title: "Dewdrops on Spiderweb"`, `photographer: "Jane Doe"`, `camera_model: "Canon EOS 5D Mark IV"`, and various null values for other fields. The second object has `id: 2`, `title: "Butterfly Wings"`, `photographer: "Carlos López"`, `camera_model: "Nikon D850"`, and various null values for other fields.

```
1 {
2   "status": "success",
3   "payload": [
4     {
5       "id": 1,
6       "title": "Dewdrops on Spiderweb",
7       "photographer": "Jane Doe",
8       "camera_model": "Canon EOS 5D Mark IV",
9       "lens_model": null,
10      "aperture": null,
11      "shutter_speed": null,
12      "iso": null,
13      "focal_length": null,
14      "magnification": null,
15      "subject": null,
16      "location": null,
17      "date_taken": null,
18      "description": null,
19      "tags": [
20        "nature",
21        "macro"
22      ]
23    },
24    {
25      "id": 2,
26      "title": "Butterfly Wings",
27      "photographer": "Carlos López",
28      "camera_model": "Nikon D850",
29      "lens_model": null,
30      "aperture": null,
31      "shutter_speed": null,
32      "iso": null,
```

## Crear nuevo registro:

http://localhost:3000/api/photos

**POST** http://localhost:3000/api/photos

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☐ Text

```
1 {
2   "title": "Butterfly Wings",
3   "photographer": "Carlos López",
4   "camera_model": "Nikon D850",
5   "aperture": "f/8",
6   "iso": 400,
7   "tags": ["insect", "nature"]
8 }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview JSON

```
1 {
2   "status": "success",
3   "payload": {
4     "id": 3,
5     "title": "Butterfly Wings",
6     "photographer": "Carlos López",
7     "camera_model": "Nikon D850",
8     "aperture": "f/8",
9     "iso": 400,
10    "tags": [
11      "insect",
12      "nature"
13    ],
14    "lens_model": null,
15    "shutter_speed": null,
16    "focal_length": null,
17    "magnification": null,
18    "subject": null,
19    "location": null,
20    "date_taken": null,
21    "description": null
22  }
23 }
```

Error al enviar registro incompleto:

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

- URL:** `http://localhost:3000/api/photos`
- Method:** `POST`
- Body Type:** `raw` (selected)
- Request Body (JSON):**

```
1 {
2   "photographer": "Carlos López",
3   "camera_model": "Nikon D850",
4   "aperture": "f/8",
5   "iso": 400,
6   "tags": ["insect", "nature"]
7 }
```
- Response Body (JSON):**

```
1 {
2   "status": "error",
3   "message": "Title and photographer are required"
4 }
```

Obtener datos registro 1:

http://localhost:3000/api/photos/1

GET

http://localhost:3000/api/photos/1

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

BodyCookiesHeaders (7)Test Results

Pretty

Raw

Preview

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

"status": "success",

"payload": {

"id": 1,

"title": "Dewdrops on Spiderweb",

"photographer": "Jane Doe",

"camera\_model": "Canon EOS 5D Mark IV",

"lens\_model": null,

"aperture": null,

"shutter\_speed": null,

"iso": null,

"focal\_length": null,

"magnification": null,

"subject": null,

"location": null,

"date\_taken": null,

"description": null,

"tags": [

"nature",

"macro"

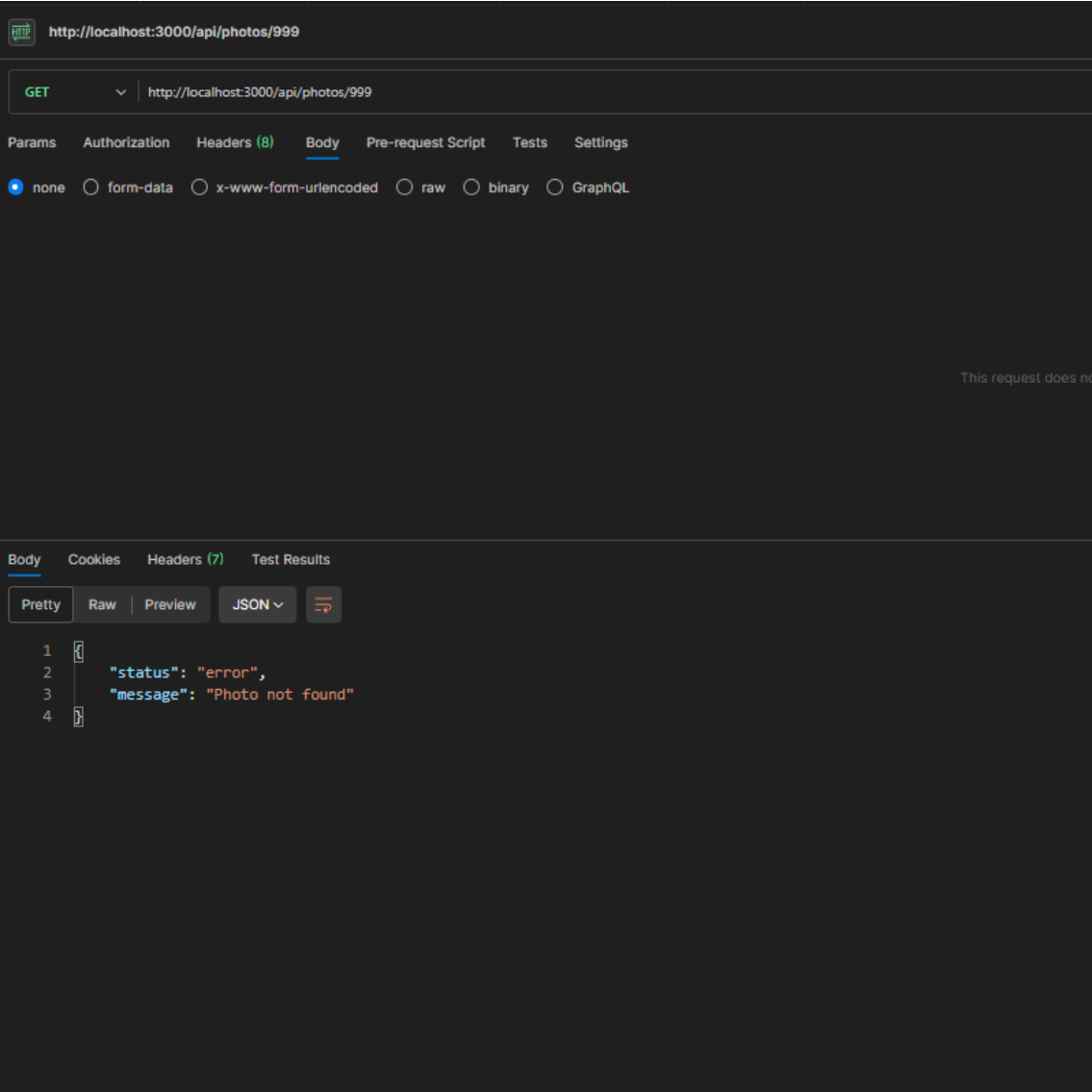
]

}

}

1

Error consulta registro inexistente:



Actualizar registro:

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

- URL:** `http://localhost:3000/api/photos/1`
- Method:** `PUT`
- Body Type:** `raw` (selected)
- Request Body:**

```
1 {  
2   "title": "Updated Spiderweb",  
3   "description": "Macro photo update test"  
4 }
```
- Response Body:**

```
1 {  
2   "status": "success",  
3   "payload": {  
4     "id": 1,  
5     "title": "Updated Spiderweb",  
6     "photographer": "Jane Doe",  
7     "camera_model": "Canon EOS 5D Mark IV",  
8     "lens_model": null,  
9     "aperture": null,  
10    "shutter_speed": null,  
11    "iso": null,  
12    "focal_length": null,  
13    "magnification": null,  
14    "subject": null,  
15    "location": null,  
16    "date_taken": null,  
17    "description": "Macro photo update test",  
18    "tags": [  
19      "nature",  
20      "macro"  
21    ]  
22  }
```

Error actualizar registro inexistente:

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

- URL:** `http://localhost:3000/api/photos/999`
- Method:** `PUT`
- Body (Request):**

```
1 {  
2   "title": "Updated Spiderweb",  
3   "description": "Macro photo update test"  
4 }
```
- Body (Response):**

```
1 {  
2   "status": "error",  
3   "message": "Photo not found"  
4 }
```

Eliminación registro:

