

Ejercicios prácticos con API

API

Se han creado las peticiones en la aplicación de postman, con ayuda de la información de la [API](#) de openproject, donde especifica que se debe poner para recibir las peticiones que queremos.

He añadido la API key generada en la parte de authorization, el type es el Basic auth, aquí en username Apikey y en Password la API key generada en nuestro OpenProject.

PARTE 1 - CRUD básico de proyectos.

1.1 Lista todos los proyectos.

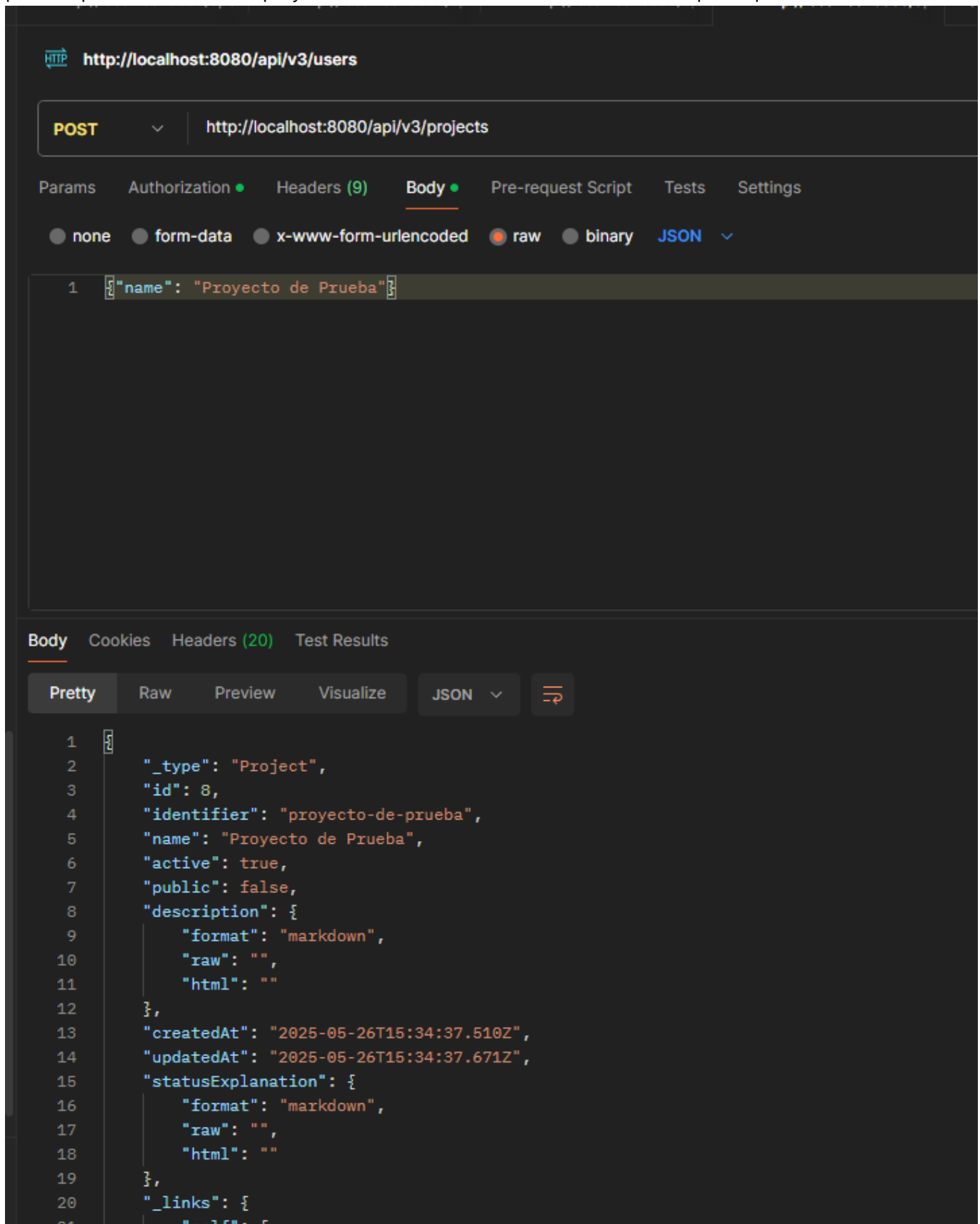
Con "GET" y la ruta indicada en la api, hemos hecho la petición, como resultado nos ha dado un JSON con todos los proyectos de nuestro OpenProject.

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/api/v3/projects`. The response is displayed in the 'Body' tab, formatted as JSON. The JSON structure is as follows:

```
1 {
2   "_type": "Collection",
3   "total": 6,
4   "count": 6,
5   "pageSize": 20,
6   "offset": 1,
7   "_embedded": {
8     "elements": [
9       {
10        "_type": "Project",
11        "id": 7,
12        "identifier": "proyecto-2-dot-1",
13        "name": "Proyecto 2.1",
14        "active": true,
15        "public": false,
16        "description": {
17          "format": "markdown",
18          "raw": "",
19          "html": ""
20        },
21        "createdAt": "2025-05-22T14:26:59.537Z",
22        "updatedAt": "2025-05-22T14:26:59.648Z",
23        "statusExplanation": {
24          "format": "markdown",
25          "raw": "",
26          "html": ""
27        }
28      }
29    ]
30  }
```

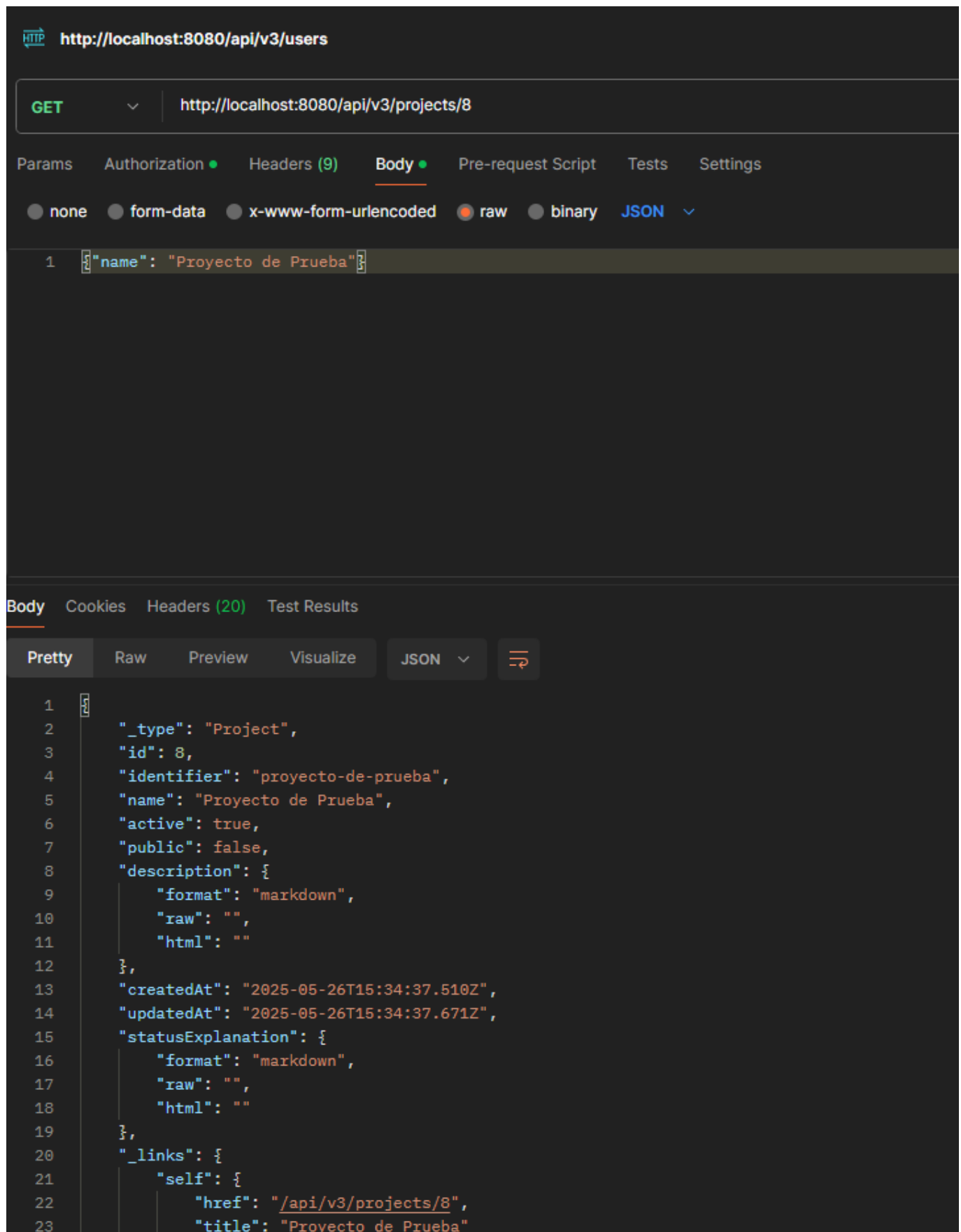
1.2 Crea un proyecto llamado "Proyecto de Prueba".

Para este ejemplo utilizo el metodo POST, y en el body le indicaremos en formato JSON "name":"Proyecto de prueba" para crear un nuevo proyecto. Podriamos utilizar todo el formato completo que nos da en la API.



1.3 Obtén los detalles del proyecto que acabas de crear.

Con el metodo GET y la misma ruta pero añadiendo /8 que es el id del proyecto, obtendremos en formato JSON la información del proyecto. En la captura se ve el body, pero es de la consulta anterior.



1.4 Cambia el nombre del proyecto creado en el punto anterior a "Proyecto Editado".

Para editar utilizaremos el metodo PATCH, y como hemos hecho al crearlo, solo editando el nombre, cambiaremos el nombre al proyecto.

Como podemos ver en la imagen, el identifiier, continua siendo proyectodeprueba, ya que al crearlo se puso

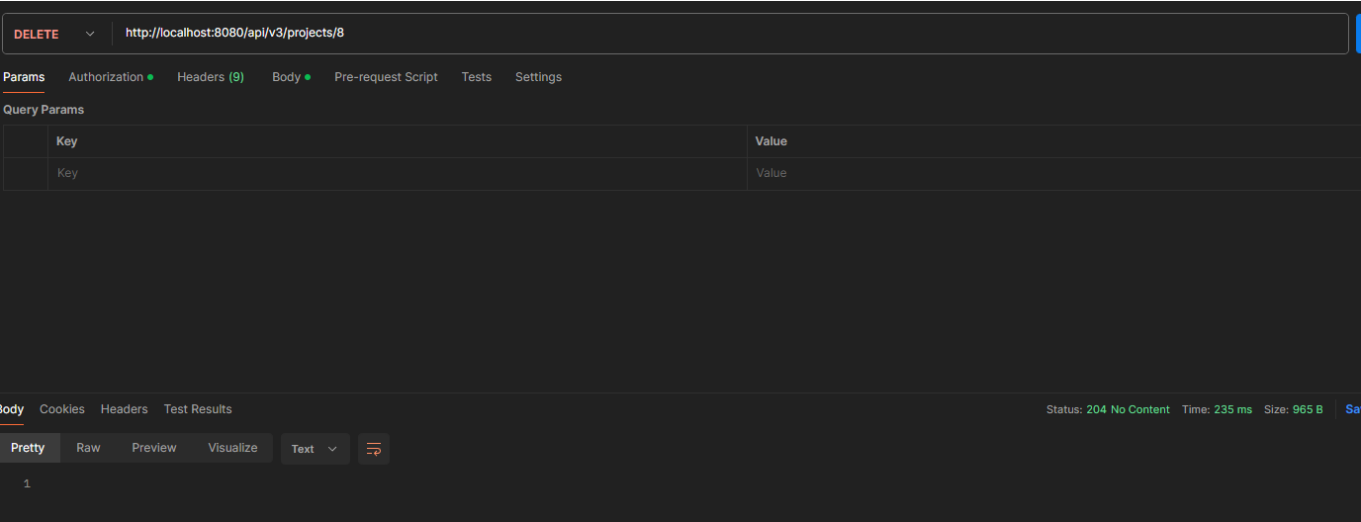
este automáticamente, y al editarlo no estamos editando este campo.

The screenshot shows a REST client interface. At the top, the URL bar displays `http://localhost:8080/api/v3/users`. Below it, the method is set to `PATCH` and the URL is `http://localhost:8080/api/v3/projects/8`. The `Body` tab is selected, showing a JSON payload: `{ "name": "Proyeto Editado" }`. The bottom section shows the response body in the `Pretty` view, displaying a JSON object with the following structure:

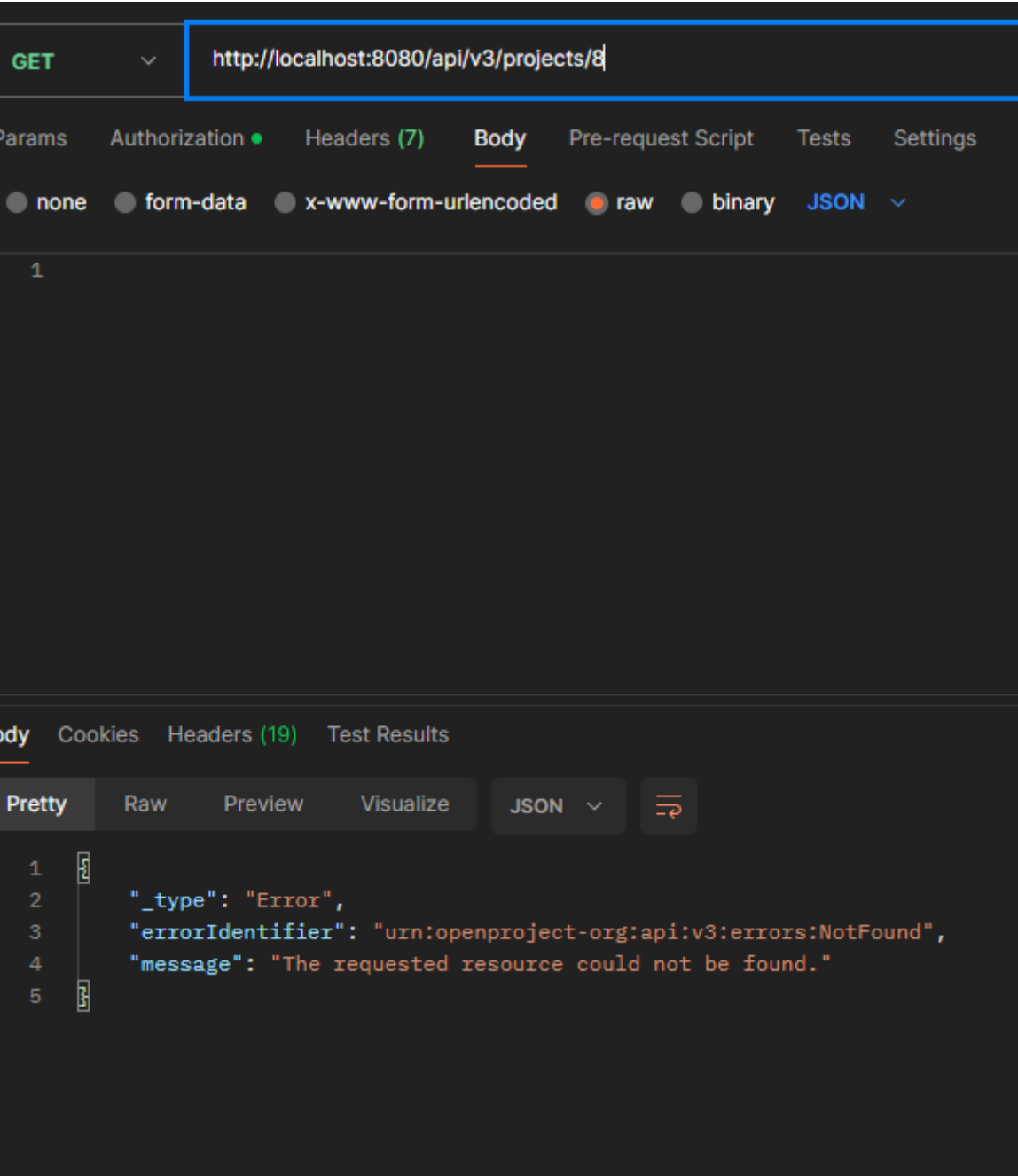
```
1 {
2   "_type": "Project",
3   "id": 8,
4   "identifier": "proyecto-de-prueba",
5   "name": "Proyeto Editado",
6   "active": true,
7   "public": false,
8   "description": {
9     "format": "markdown",
10    "raw": "",
11    "html": ""
12  },
13   "createdAt": "2025-05-26T15:34:37.510Z",
14   "updatedAt": "2025-05-26T15:46:01.163Z",
15   "statusExplanation": {
16     "format": "markdown",
17     "raw": "",
18     "html": ""
19   },
20   "_links": {
21     "self": {
```

1.5 Elimina el proyecto creado.

Para eliminar he utilizado el metodo DELETE. como ruta pondremos el id del proyecto a eliminar. En el status podemos ver 204 . Segun la api al devolvernos esta respuesta : Devuelto si el proyecto se eliminó con éxito.



Como podemos ver el proyecto ya no existe.



EXTRA - Lista todos los usuarios de Open Project

Con "GET" y la ruta indicada en la api, hemos hecho la petición, como resultado nos ha dado un JSON con todos los usuarios de nuestro OpenProject.

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

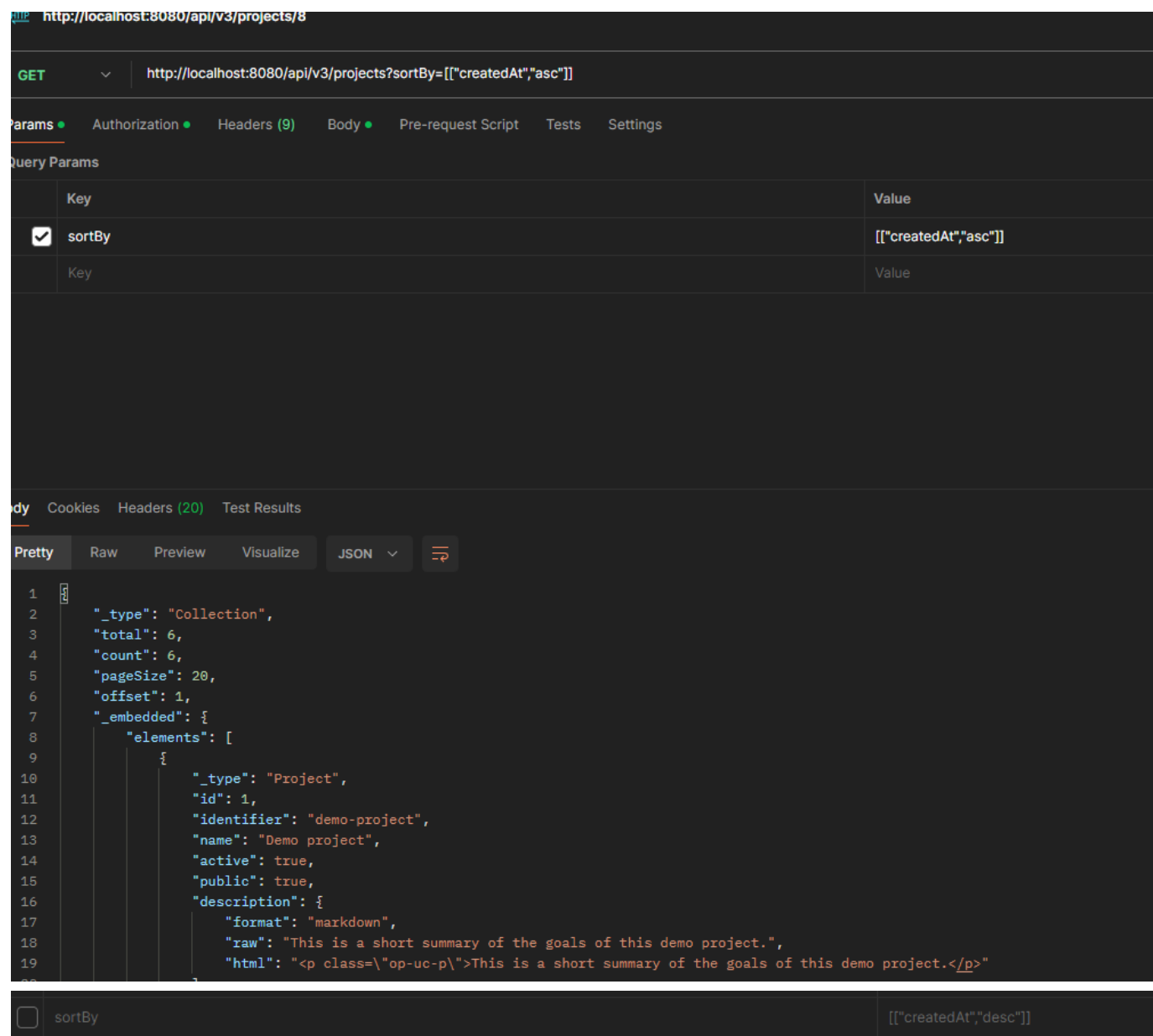
- URL:** `http://localhost:8080/api/v3/users`
- Method:** `GET`
- Params:** Authorization, Headers (7), Body, Pre-request Script, Tests, Settings
- Query Params:** A table with 2 columns: Key, Value. The first row contains the text "Key" in both columns.
- Body:** Pretty, Raw, Preview, Visualize, JSON (selected), and a copy icon.
- JSON Response:**

```
1 {
2   "_type": "Collection",
3   "total": 6,
4   "count": 6,
5   "pageSize": 20,
6   "offset": 1,
7   "_embedded": {
8     "elements": [
9       {
10        "_type": "User",
11        "id": 9,
12        "name": "Noe clase",
13        "createdAt": "2025-05-22T14:32:21.626Z",
14        "updatedAt": "2025-05-22T14:32:21.626Z",
15        "login": "clase3@pruebas.com",
16        "admin": false,
17        "firstName": "Noe",
18        "lastName": "clase",
19        "email": "clase3@pruebas.com",
20        "avatar": "http://gravatar.com/avatar/e98d5c6b91a8a6f7d8d2680538df0e02?default=404&secure=false",
21        "status": "invited",
22        "identityUrl": null,
23        "language": "ca",
24        "_links": {
25          "self": "http://localhost:8080/api/v3/users/9",
26          "collection": "http://localhost:8080/api/v3/users"
27        }
28      }
29    ]
30  }
31 }
```

PARTE 2 - Consultas Ordenadas

2.1 Lista los proyectos ordenados por fecha de creación (de viejo a nuevo)

Para esta consulta se ha utilizado el GET, como queremos que esten ordenados, hemos indicado el `sortBy`, donde los parametros de dentro son el campo que queremos que aplique, en este caso, el orden. Esto se puede hacer directamente en la url que se lo agrega al GET o añadiendo en Params los parametros.



2.2 Lista los proyectos ordenados por fecha de edición (de nuevo a viejo)

He intentado hacerlo con el filtro updatedAt, pero no me deja. La otra forma que he econtrado es con el lastest_activity_at.

[http://localhost:8080/api/v3/projects?filters=\[{"name":{"operator":"=", "values":\["Proyecto 1"\]}\]}\]](http://localhost:8080/api/v3/projects?filters=[{)

GET

http://localhost:8080/api/v3/projects?sortBy=[["updatedAt","desc"]]

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Query Params

	Key	Value
<input checked="" type="checkbox"/>	sortBy	[["updatedAt","desc"]]
	Key	Value

Body

Cookies

Headers (18)

Test Results

Status: 400 Bad Request

Pretty

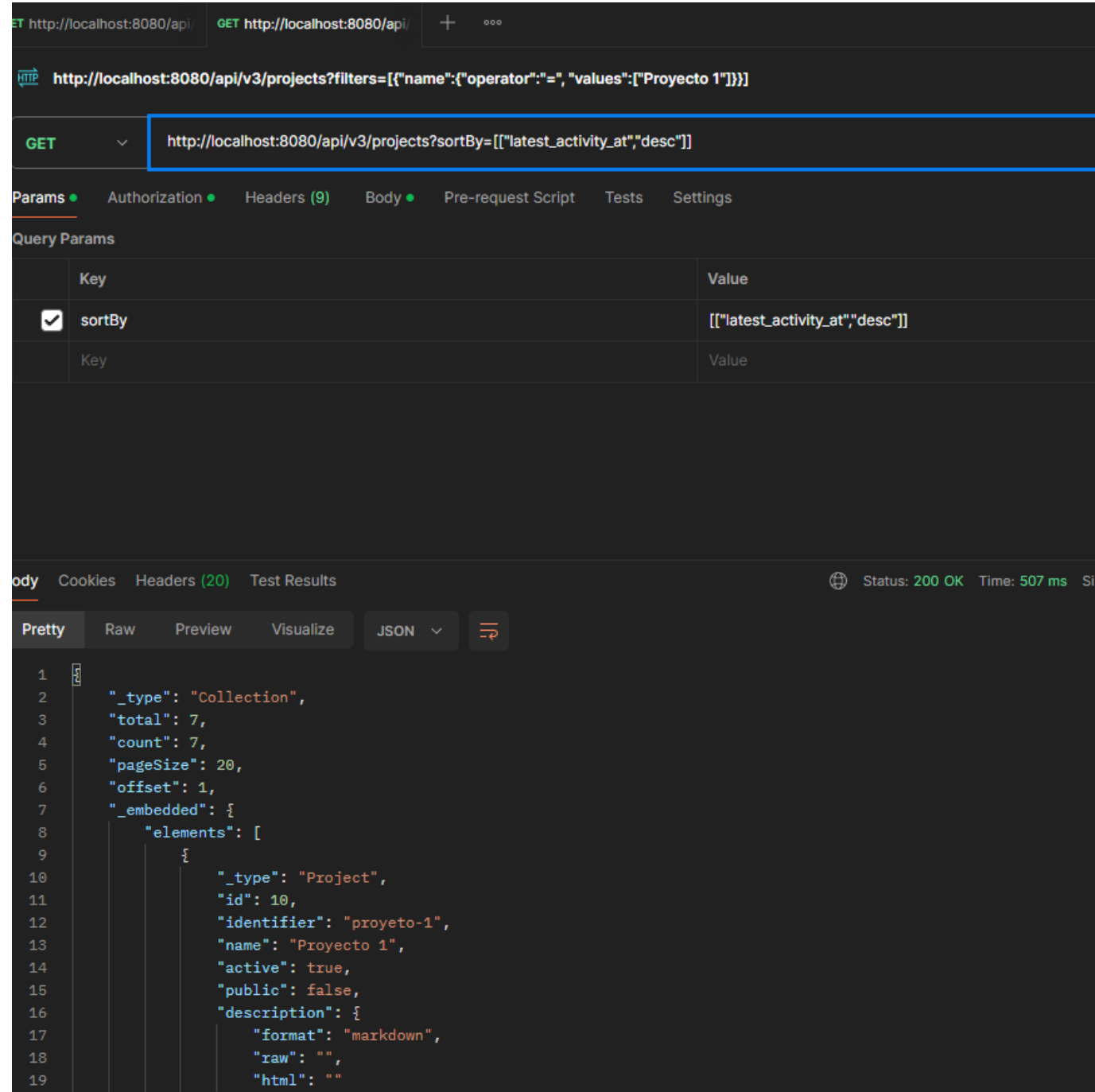
Raw

Preview

Visualize

JSON

```
1 {
2   "_type": "Error",
3   "errorIdentifier": "urn:openproject-org:api:v3:errors:InvalidQuery",
4   "message": "Orders Updated on does not exist."
5 }
```



2.3 Lista los proyectos ordenados por orden alfabético

Para esta petición, cambiaremos el createdAt por el name ya que nos interesa el nombre para poder ordenarlo alfabeticamente. Demo project es el primero, ya que los demas empizan como Proyecto XXXX que

irian despues.

http://localhost:8080/api/v3/projects/8

GET

http://localhost:8080/api/v3/projects?sortBy=[{"name","asc"}]

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Query Params

	Key	Value
<input checked="" type="checkbox"/>	sortBy	[{"name","asc"}]
	Key	Value

Body

Cookies

Headers (20)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

"_type": "Collection",

"total": 6,

"count": 6,

"pageSize": 20,

"offset": 1,

"_embedded": {

"elements": [

{

"_type": "Project",

"id": 1,

"identifier": "demo-project",

"name": "Demo project",

"active": true,

"public": true,

"description": {

"format": "markdown",

"raw": "This is a short summary of the goals of this demo project.",

"html": "<p class=\"op-uc-p\">This is a short summary of the goals of this demo project.</p>"

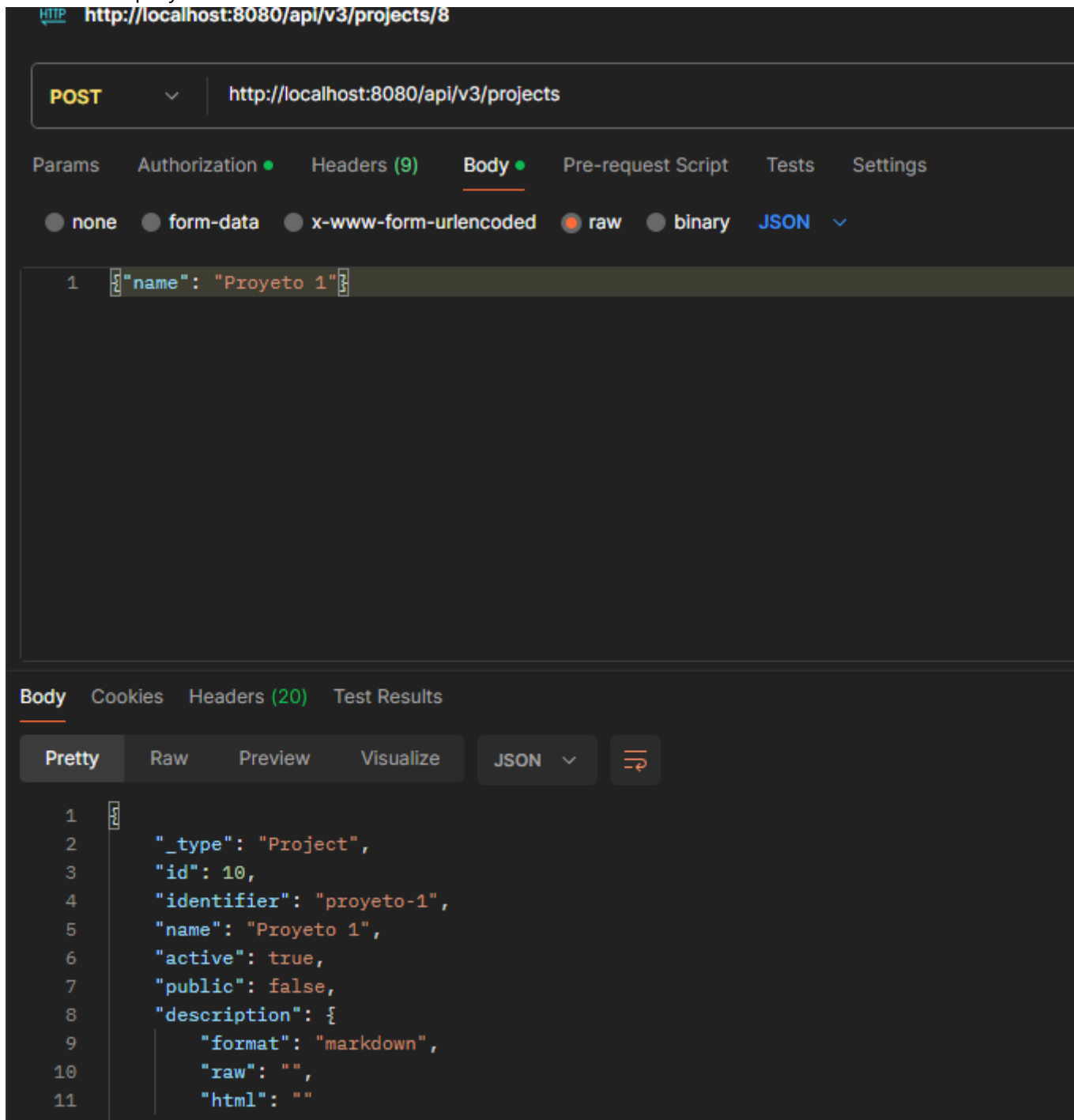
},

"createdAt": "2025-05-21T15:50:54.144Z",

PARTE 3 – Consultas con Filtros

3.1 Crea un proyecto llamado "Proyecto 1".

He creado el proyecto con el metodo POST.



3.2 Lista todos los proyectos llamados "Proyecto 1".

Para esta practica, he buscado como funcionan los filtros, tras una breve pelea, he visto como funciona.

Sintaxis de filtro

La sintaxis del filtro es un objeto JSON que se puede pasar como un parámetro GET a los puntos finales, ya que una cadena JSON codificada por URL se parece a lo siguiente:

```
[
  { "<filter name>": { "operator": "<operator>", "values": [ <value>, ... ] } },
  // ...
]
```

Ejemplo

Se añadiría en los parametros como:

- Key: filters
- Value: iriamos cambiando los parametros, el filtro seria name, ya que queremos saber por nombre, el operador, en este caso igual a, y el valor seria "Proyecto 1".

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/api/v3/projects?filters=[{"name":{"operator":"=", "values":["Proyecto 1"]}]]`. The request is successful (200 OK). The response body is a JSON object representing a collection of projects.

Query Params

Key	Value
sortBy	[["createdAt","desc"]]
filters	[{"name":{"operator":"=", "values":["Proyecto 1"]}]]

Body

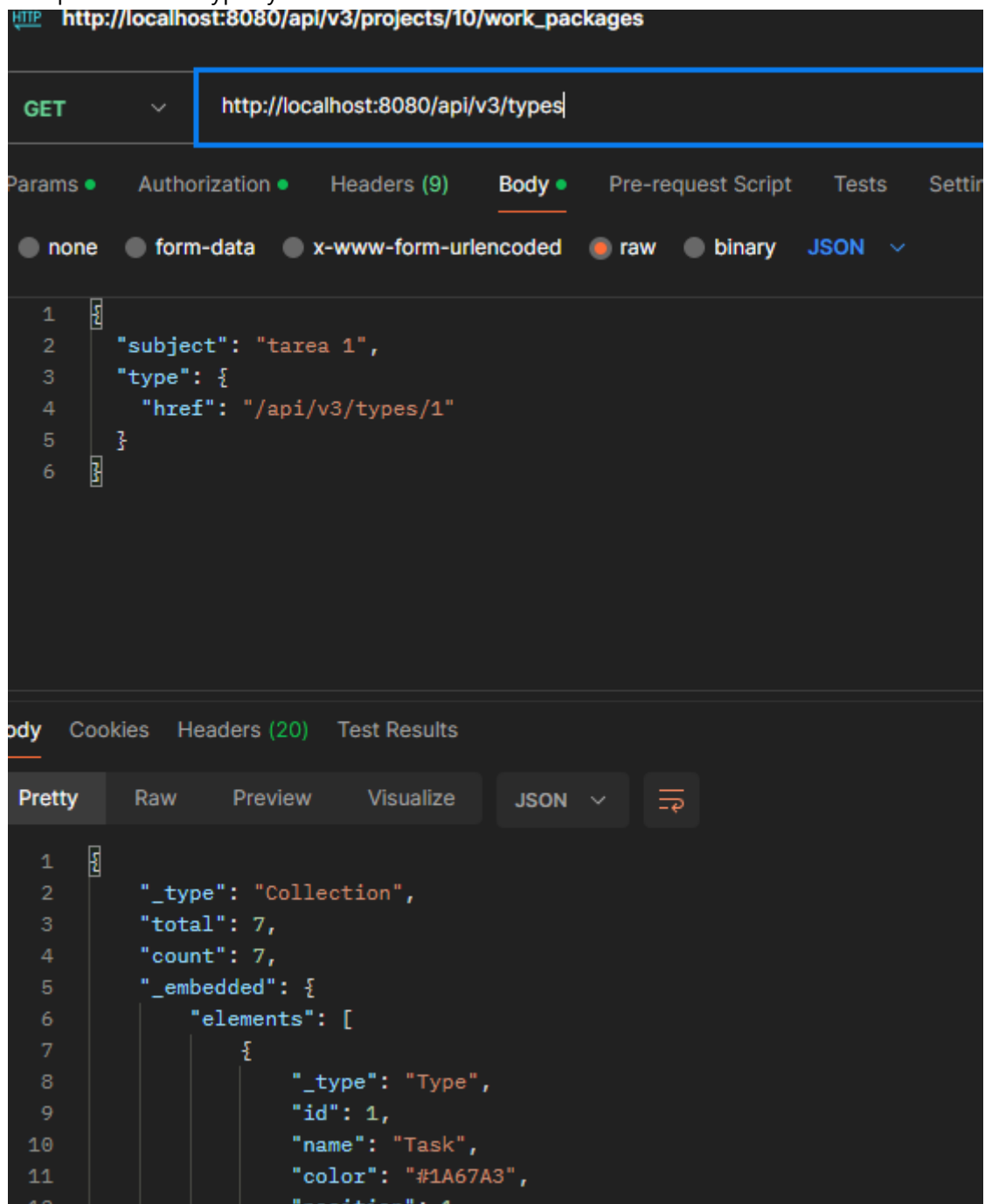
```

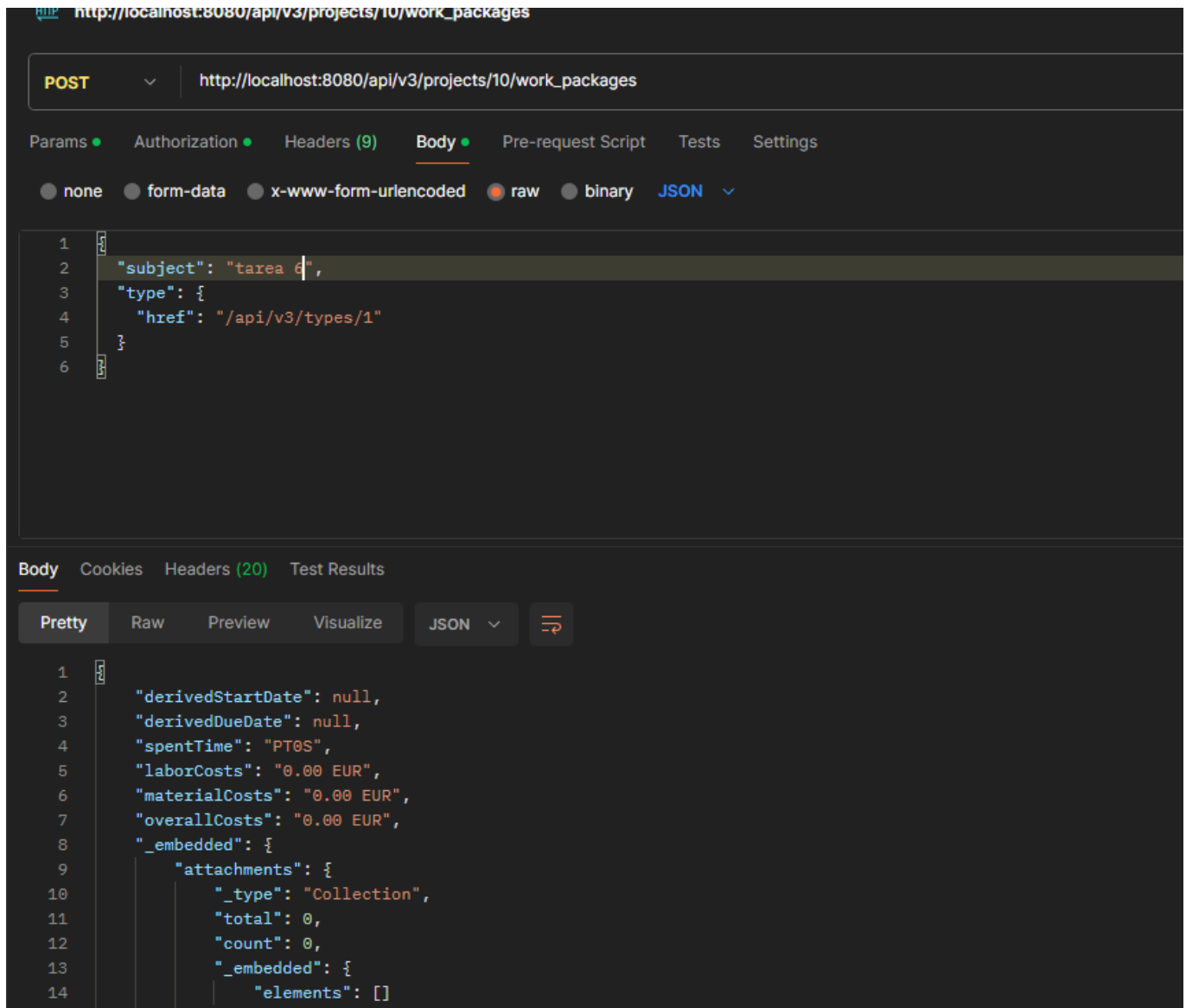
1  {
2    "_type": "Collection",
3    "total": 1,
4    "count": 1,
5    "pageSize": 20,
6    "offset": 1,
7    "_embedded": {
8      "elements": [
9        {
10         "_type": "Project",
11         "id": 4,
12         "identifier": "proyecto-1",
13         "name": "Proyecto 1",
14         "active": true,
15         "public": false,
16         "description": {
17           "format": "markdown",
18           "raw": "",
19           "html": ""
20         },
21         "createdAt": "2025-05-22T14:25:53.932Z",
22         "updatedAt": "2025-05-22T14:25:54.118Z",
23         "statusExplanation": {
24           "format": "markdown",
25           "raw": "",
26           "html": ""
27         }
28       }
29     }
30   }
  
```

3.3 Crea un par de tareas

Para crear tareas pondremos el id del proyecto, con el metodo POST y work_package. En el body crearemos la tarea indicandole subject como el nombre de la tarea y el type que en este caso sera 1. Para esto he hecho un

GET para obtener types y el 1 son Task.





3.4 Lista todas las tareas activas

Con el metodo GET, y el endpoint work_packages.

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/api/v3/projects/10/work_packages`. The request body is a JSON object:

```
{
  "subject": "tarea 6",
  "type": {
    "href": "/api/v3/types/1"
  }
}
```


The response is displayed in the 'Body' tab, showing a JSON object with the following structure:

```
{
  "_type": "WorkPackageCollection",
  "total": 39,
  "count": 20,
  "pageSize": 20,
  "offset": 1,
  "_embedded": {
    "elements": [
      {
        "derivedStartDate": "2025-05-19",
        "derivedDueDate": "2025-06-02",
        "_type": "WorkPackage",
        "id": 2,
        "lockVersion": 0,
        "subject": "Organize open source conference",
        "description": {
          "format": "markdown",
          "raw": "",
          "html": ""
        },
        "scheduleManually": false,
        "startDate": "2025-05-19",
        "dueDate": "2025-06-02",

```

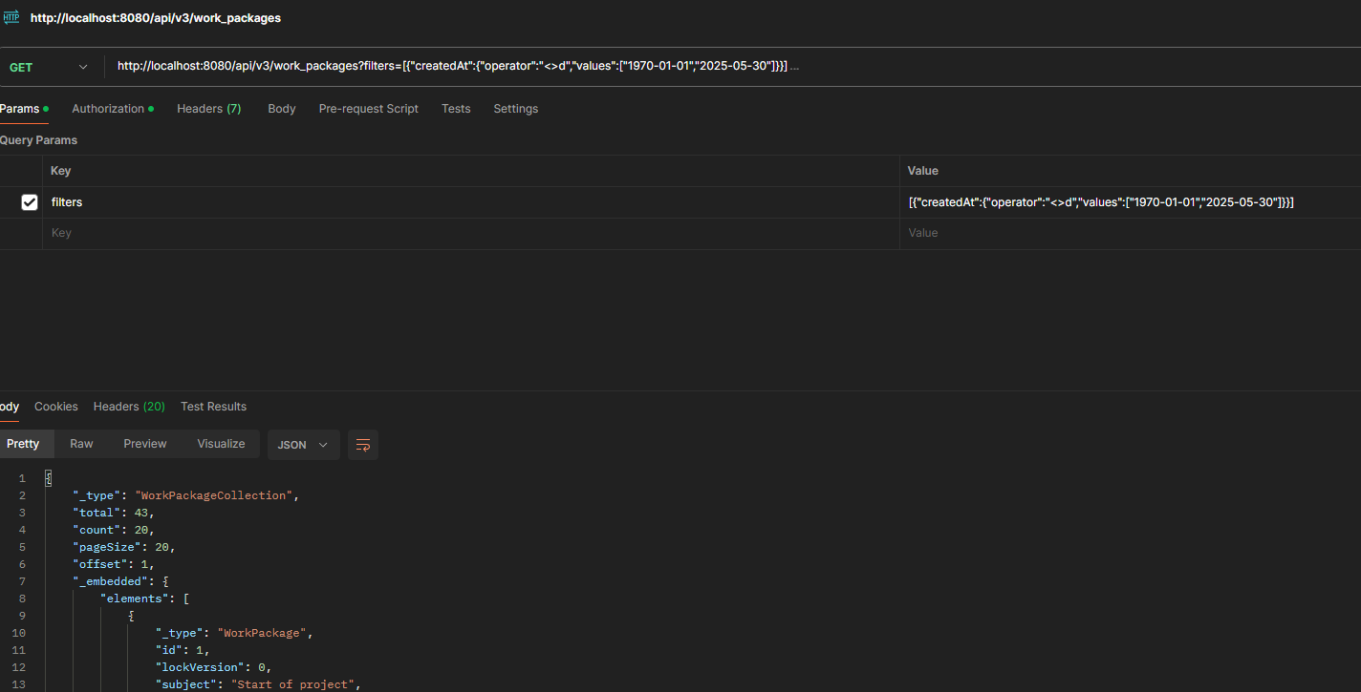
3.5 Lista todas las tareas creadas desde antes del 30 de mayo de 2025.

Para esta peticion volveremos a utilizar los filtros. En este caso un rango de fechas.

	are between the two given dates.	2 ISO8601 date/datetimes
---	----------------------------------	-----------------------------

Para

el filtro he utilizado un rango de fechas ya que no he ecnontrado el operador menor que una fecha que funcionase.



3.6 Lista todas las tareas inactivas creadas después del 20 de mayo

The screenshot displays a REST client interface. At the top, a green status bar shows a successful GET request to `http://localhost:8080/api/v3/work_packages`. Below this, the main area shows a GET request to `http://localhost:8080/api/v3/statuses`. The request is configured with no parameters, authorization, headers, or body. The response is shown in the 'Body' tab, formatted as JSON. The JSON response is an array of two objects. The first object represents a 'Confirmed' status, and the second object represents a 'To be scheduled' status. The 'To be scheduled' status includes various attributes like `_type`, `id`, `name`, `isClosed`, `color`, `isDefault`, `isReadOnly`, `excludedFromTotals`, `defaultDoneRatio`, `position`, and `_links`.

```
GET http://localhost:8080/api/v3/work_packages

GET http://localhost:8080/api/v3/statuses

Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings

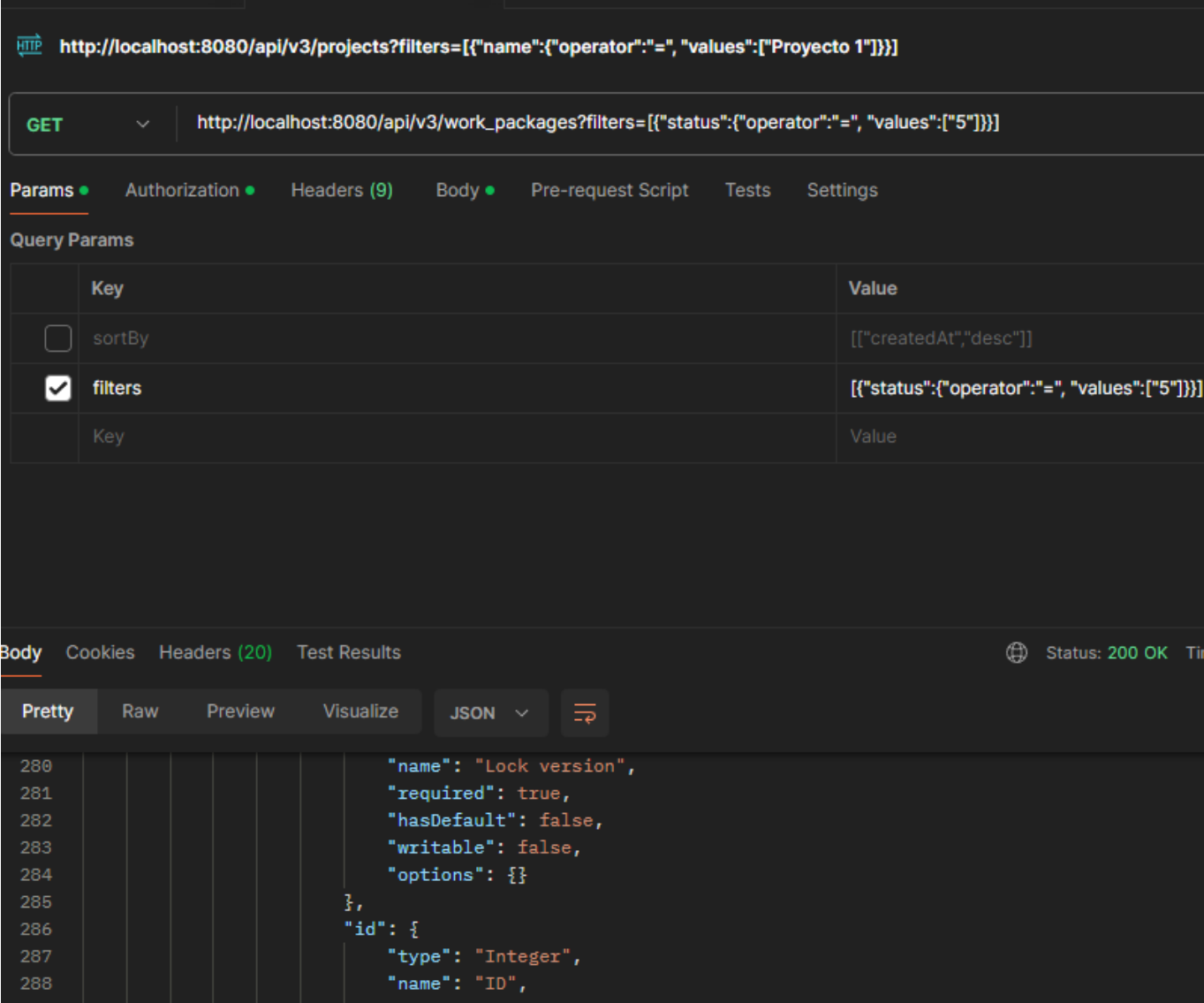
Query Params

Key
Key

Body
Cookies
Headers (20)
Test Results

Pretty
Raw
Preview
Visualize
JSON

[{"_type": "Status", "id": 4, "name": "Confirmed", "isClosed": true, "color": "#FFD700", "isDefault": true, "isReadOnly": true, "excludedFromTotals": false, "defaultDoneRatio": 100, "position": 1, "_links": {"self": {"href": "/api/v3/statuses/4", "title": "Confirmed"}, "work_packages": {"href": "/api/v3/work_packages?status=4", "title": "Work Packages"}}, "href": "/api/v3/statuses/4", "title": "Confirmed"}, {"_type": "Status", "id": 5, "name": "To be scheduled", "isClosed": false, "color": "#FFEC99", "isDefault": false, "isReadOnly": false, "excludedFromTotals": false, "defaultDoneRatio": 20, "position": 5, "_links": {"self": {"href": "/api/v3/statuses/5", "title": "To be scheduled"}, "work_packages": {"href": "/api/v3/work_packages?status=5", "title": "Work Packages"}}, "href": "/api/v3/statuses/5", "title": "To be scheduled"}]
```



BBDD

PARTE 1 - CRUD basico de proyectos.

Para esta parte he utilizado el pgAdmin.

1.1 Lista todos los proyectos.

pgAdmin 4

Object Tools Edit View Window Help

Welcome x openproject/postgres@openproject* x

openproject/postgres@openproject

No limit

Query Query History

1 select * from projects;
2

Data Output Messages Notifications

Showing row:

	id [PK] bigint	name character varying	description text	public boolean	parent_id bigint	created_at timestamp with time zone	updated_at timestamp with time zone	identifier character varying	lft integer	rgt integer	active boolean
1	1	Demo project	This is a short summary of the goals of this demo project.	true	[null]	2025-05-21 15:50:54.144453+00	2025-05-21 15:50:54.144453+00	demo-project	1	2	true
2	2	Scrum project	This is a short summary of the goals of this demo Scrum project.	true	[null]	2025-05-21 15:50:57.538447+00	2025-05-26 17:11:17.687546+00	your-scrum-project	13	14	true
3	4	Proyecto 1		false	[null]	2025-05-22 14:25:53.932302+00	2025-05-26 17:30:56.915199+00	proyecto-1	5	6	true
4	5	Proyecto 2		false	[null]	2025-05-22 14:26:16.754099+00	2025-05-26 17:30:56.915199+00	proyecto-2	7	10	true
5	6	Proyecto 3		false	[null]	2025-05-22 14:26:33.53892+00	2025-05-26 17:30:56.915199+00	proyecto-3	11	12	true
6	7	Proyecto 2.1		false	5	2025-05-22 14:26:59.537329+00	2025-05-26 17:30:56.915199+00	proyecto-2-dot-1	8	9	true
7	10	Proyecto 1	[null]	false	[null]	2025-05-26 17:09:07.279221+00	2025-05-26 17:30:56.915199+00	proyecto-1	3	4	true

1.2 Crea un proyecto llamado "Proyecto de Prueba".

Desde la tabla de proyectos, se puede crear la query para insertar un proyecto. He ido rellenando, en algunas que he podido dejar null lo he dejado, en json final he tenido que poner un objeto vacio.

Query Query History

1 INSERT INTO public.projects(
2 id, name, description, public, parent_id, created_at, updated_at, identifier, lft, rgt, active, templated, status_code, status_expl
3 VALUES (11, 'Proyecto de Prueba', 'Creado para prueba', true, null, current_timestamp, current_timestamp, 'proyecto-prueba', null, 1
4 select * from projects;|

Data Output Messages Notifications

Showing row:

	id [PK] bigint	name character varying	description text	public boolean	parent_id bigint	created_at timestamp with time zone	updated_at timestamp with time zone	identifier character varying	lft integer	rgt integer	active boolean
1	1	Demo project	This is a short summary of the goals of this demo project.	true	[null]	2025-05-21 15:50:54.144453+00	2025-05-21 15:50:54.144453+00	demo-project	1	2	true
2	2	Scrum project	This is a short summary of the goals of this demo Scrum project.	true	[null]	2025-05-21 15:50:57.538447+00	2025-05-26 17:11:17.687546+00	your-scrum-project	13	14	true
3	4	Proyecto 1		false	[null]	2025-05-22 14:25:53.932302+00	2025-05-26 17:30:56.915199+00	proyecto-1	5	6	true
4	5	Proyecto 2		false	[null]	2025-05-22 14:26:16.754099+00	2025-05-26 17:30:56.915199+00	proyecto-2	7	10	true
5	6	Proyecto 3		false	[null]	2025-05-22 14:26:33.53892+00	2025-05-26 17:30:56.915199+00	proyecto-3	11	12	true
6	7	Proyecto 2.1		false	5	2025-05-22 14:26:59.537329+00	2025-05-26 17:30:56.915199+00	proyecto-2-dot-1	8	9	true
7	10	Proyecto 1	[null]	false	[null]	2025-05-26 17:09:07.279221+00	2025-05-26 17:30:56.915199+00	proyecto-1	3	4	true
8	11	Proyecto de Prueba	Creado para prueba	true	[null]	2025-05-28 07:29:44.716894+00	2025-05-28 07:29:44.716894+00	proyecto-prueba	1	2	true

Query History

```
select * from projects where id=11;
```

Data Output Messages Notifications

id [PK] bigint	name character varying	description text	public boolean	parent_id bigint	created_at timestamp with time zone	updated_at timestamp with time zone	identifier character varying	lft integer	rgt integer	active boolean	templated boolean	status_code integer	status_explanation text	settings jsonb
11	Proyecto de Prueba	Creado para prueba	true	[null]	2025-05-28 07:29:44.716894+00	2025-05-28 07:29:44.716894+00	proyecto-prueba	[null]	[null]	true	false	[null]	[null]	{}

Showing rows: 1 to 1 Page No: 1 of 1

1.5 Elimina el proyecto creado.

welcome x openproject/postgres@openproject x

openproject/postgres@openproject

No limit

Query Query History

1 DELETE FROM projects where id =11;

2 select * from projects;|

Data Output Messages Notifications

SQL

	id [PK] bigint	name character varying	description text	public boolean	parent_id bigint	created_at timestamp with time zone	updated_at timestamp with t
1	1	Demo project	This is a short summary of the goals of this demo project.	true	[null]	2025-05-21 15:50:54.144453+00	2025-05-21 15:5
2	2	Scrum project	This is a short summary of the goals of this demo Scrum project.	true	[null]	2025-05-21 15:50:57.538447+00	2025-05-26 17:1
3	4	Proyecto 1		false	[null]	2025-05-22 14:25:53.932302+00	2025-05-26 17:3
4	5	Proyecto 2		false	[null]	2025-05-22 14:26:16.754099+00	2025-05-26 17:3
5	6	Proyecto 3		false	[null]	2025-05-22 14:26:33.53892+00	2025-05-26 17:3
6	7	Proyecto 2.1		false	5	2025-05-22 14:26:59.537329+00	2025-05-26 17:3
7	10	Proyecto 1	[null]	false	[null]	2025-05-26 17:09:07.279221+00	2025-05-26 17:3

EXTRA - Lista todos los usuarios de Open Project

Welcome x openproject/postgres@openproject x

openproject/postgres@openproject

No limit

Query Query History

```
1 select * from users;
```

Data Output Messages Notifications

	ID [PK] bigint	login character varying (256)	firstname character varying	lastname character varying	mail character varying	admin boolean	status integer	last_login_on timestamp with time zone	language character varying (5)	ldap_auth_source_id bigint
1	1			System		true	1	[null]		[null]
2	2			Anonymous		false	1	[null]		[null]
3	3					false	3	[null]		[null]
4	5	angel@pruebas.com	Angel	Subordinado	angel@pruebas.com	false	4	[null]	es	[null]
5	6	adrian@pruebas.com	Adrian	Senior	adrian@pruebas.com	true	4	[null]	es	[null]
6	7	clase@prueba.com	Ivan	clase	clase@prueba.com	false	4	[null]	pl	[null]
7	8	clase2@prueba.com	Dyron	clase	clase2@prueba.com	false	4	[null]	zh-CN	[null]
8	9	clase3@pruebas.com	Noe	clase	clase3@pruebas.com	false	4	[null]	ca	[null]
9	11			Juniors		false	1	[null]		[null]
10	10			Seniors		false	1	[null]		[null]
11	4	admin	Jordi	Admin	admin@example.net	true	1	2025-05-23 08:46:55.031127+00	en	[null]

PARTE 2 - Consultas Ordenadas

2.1 Lista los proyectos ordenados por fecha de creación (de viejo a nuevo)

[illegible]

Tambien serviria sin el ASC porque por defecto lo pondria ordenado.

2.2 Lista los proyectos ordenados por fecha de edición (de nuevo a viejo)

Welcome x openproject/postgres@openproject x

openproject/postgres@openproject

Query Query History

```
1 select * from projects order by updated_at desc;
```

Data Output Messages Notifications

	id [PK] bigint	name character varying	description text	public boolean	parent_id bigint	created_at timestamp with time zone	updated_at timestamp with time zone	identifier character varying	lft integer	rght integer
1	5	Proyecto 2		false	[null]	2025-05-22 14:26:16.754099+00	2025-05-26 17:30:56.915199+00	proyecto-2	7	
2	7	Proyecto 2.1		false	5	2025-05-22 14:26:59.537329+00	2025-05-26 17:30:56.915199+00	proyecto-2-dot-1	8	
3	10	Proyecto 1	[null]	false	[null]	2025-05-26 17:09:07.279221+00	2025-05-26 17:30:56.915199+00	proyeto-1	3	
4	4	Proyecto 1		false	[null]	2025-05-22 14:25:53.932302+00	2025-05-26 17:30:56.915199+00	proyecto-1	5	
5	6	Proyecto 3		false	[null]	2025-05-22 14:26:33.53892+00	2025-05-26 17:30:56.915199+00	proyecto-3	11	
6	2	Scrum project	This is a short summary of the goals of this demo Scrum project.	true	[null]	2025-05-21 15:50:57.538447+00	2025-05-26 17:11:17.687546+00	your-scrum-project	13	
7	1	Demo project	This is a short summary of the goals of this demo project.	true	[null]	2025-05-21 15:50:54.144453+00	2025-05-21 15:50:54.144453+00	demo-project	1	

Total rows: 7 Query complete 00:00:00.086

2.3 Lista los proyectos ordenados por orden alfabético

3.1 Crea un proyecto llamado "Proyecto 1".

26 / 31

3.2 Lista todos los proyectos llamados "Proyecto 1".

Query Query History

```
select * from projects where name= 'Proyecto 1';
```


Data Output Messages Notifications

	id [PK] bigint	name character varying	description text	public boolean	parent_id bigint	created_at timestamp with time zone	updated_at timestamp with time zone	identifier character varying	lft integer	rgt integer	active boolean	templated boolean	status_code integer	status_exp text
	4	Proyecto 1		false	[null]	2025-05-22 14:25:53.932302+00	2025-05-26 17:30:56.915199+00	proyecto-1	5	6	true	false	[null]	
	10	Proyecto 1	[null]	false	[null]	2025-05-26 17:09:07.279221+00	2025-05-26 17:30:56.915199+00	proyecto-1	3	4	true	false	[null]	[null]
	12	Proyecto 1	[null]	true	[null]	2025-05-28 08:06:35.443258+00	2025-05-28 08:06:35.443258+00	proyecto-1-sql	[null]	[null]	true	false	[null]	[null]

Showing rows: 1 to 3 Page No: 1

3.3 Crea un par de tareas

[Query](#) [Query History](#)

[Data Output](#) [Messages](#) [Notifications](#)

Total rows: 47 Query complete 00:00:00.101

Segun los statues que habiamos visto antes en Postman, el 7 era en progreso asique he considerado que estos son los activos.

Output

Messages

Notifications

📄

▼

📄

▼

🗑️

📁

📥

📶

SQL

Showing rows: 1 to 6

Page No: 1

id [PK] bigint	type_id bigint	project_id bigint	subject character varying	description text	due_date date	category_id bigint	status_id bigint	assigned_to_id bigint	priority_id bigint	version_id bigint	author_id bigint
4	1	1	Send invitation to speakers		2025-05-19	[null]	7	4	8	[null]	
3	1	1	Set date and location of conference		2025-05-22	[null]	7	4	8	[null]	
2	3	1	Organize open source conference		2025-06-02	[null]	7	4	8	[null]	
20	1	2	Create wireframes for new landing page	[null]	2025-06-16	[null]	7	4	8		3
17	6	2	Newsletter registration form	[null]	[null]	[null]	7	4	8		2
31	3	2	Develop v1.0	[null]	2025-06-05	[null]	7	4	8	[null]	

En este caso no hace falta indicar un rango de fechas como he hecho en Postman porque aqui si se puede poner el < y la fecha.

```
1
2 select * from work_packages where created_at < '2025-05-30';
```

[Data Output](#) [Messages](#) [Notifications](#)

SQL

Showing rows: 1 to 47

Page No: 1

	id [PK] bigint	type_id bigint	project_id bigint	subject character varying	description text	due_date date	category_id bigint	status_id bigint	assigned bigint
1	1	2	1	Start of project	[null]	2025-05-18	[null]	12	
2	37	1	10	tarea 1	[null]	[null]	[null]	1	
3	38	1	10	tarea 1	[null]	[null]	[null]	1	
4	4	1	1	Send invitation to speakers		2025-05-19	[null]	7	
5	5	1	1	Contact sponsoring partners		2025-05-20	[null]	1	
6	6	1	1	Create sponsorship brochure and hand-outs		2025-05-22	[null]	1	
7	3	1	1	Set date and location of conference		2025-05-22	[null]	7	
8	7	1	1	Invite attendees to conference		2025-05-23	[null]	1	
9	8	1	1	Setup conference website		2025-06-02	[null]	1	
10	2	3	1	Organize open source conference		2025-06-02	[null]	7	
11	9	2	1	Conference		2025-05-22	[null]	1	

Total rows: 47 Query complete 00:00:00.102

✓ Successfully run. Total query runtime: 1

30 / 31

openproject/postgres@openproject

No limit

E

Query Query History

1
2

```
select * from work_packages where created_at > '2025-05-20' and status_id=5;
```

Data Output Messages Notifications

SQL

Showing rows: 1 to 1

Page No: 1

	id [PK] bigint	type_id bigint	project_id bigint	subject character varying	description text	due_date date	category_id bigint	status_id bigint	assigned_to_id bigint	priority_id bigint	version_id bigint	author_id bigint	lock_version integer
1	10	3	1	Follow-up tasks		2025-06-19	[null]	5	4	8	[null]	4	