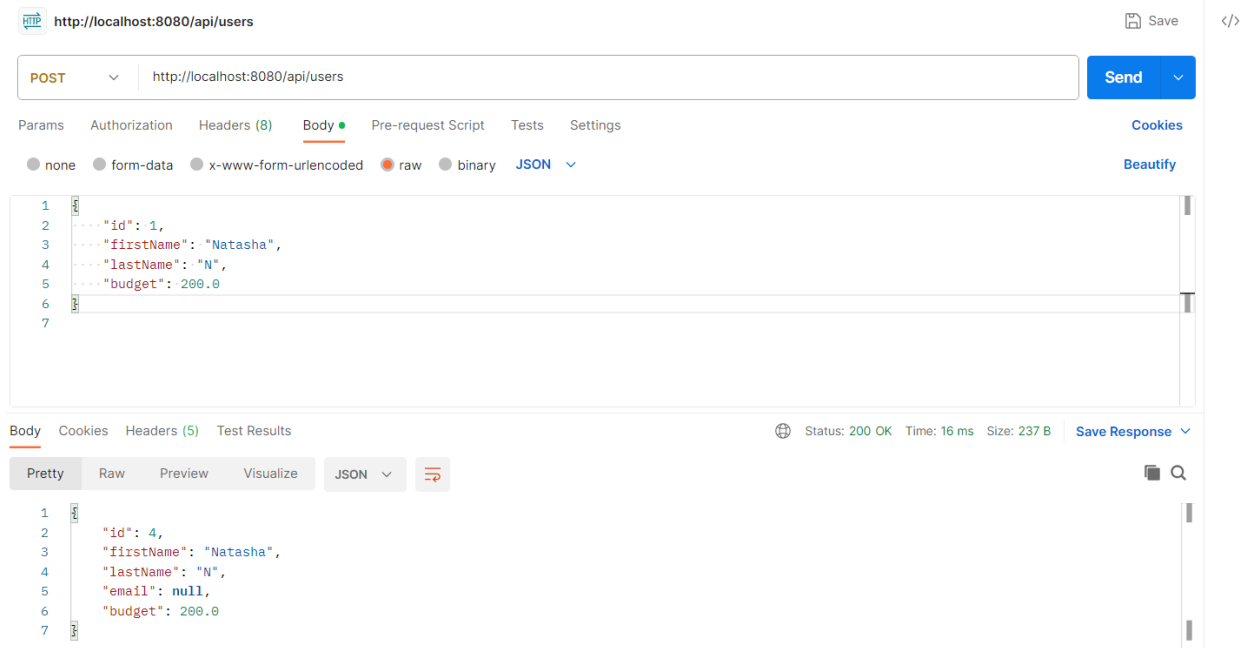


Manually Tested via Postman

- URL: <http://localhost:8080/api/users>

Method: **POST**



Body:

```
{
  "id": 1,
  "firstName": "Natasha",
  "lastName": "N",
  "budget": 200.0
}
```

- URL: <http://localhost:8080/api/users>

Method: **POST**

Body:

```
{
  "id": 1,
  "firstName": "John",
  "lastName": "Doe",
  "budget": 500.0
}
```

POST http://localhost:8080/api/ + ...

http://localhost:8080/api/users Save

POST http://localhost:8080/api/users Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

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

```
1 {
2   "id": 1,
3   "firstName": "John",
4   "lastName": "Doe",
5   "budget": 500.0
6 }
7
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 647 ms Size: 236 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 3,
3   "firstName": "John",
4   "lastName": "Doe",
5   "email": null,
6   "budget": 500.0
7 }
```

URL: <http://localhost:8080/api/users>

Method: GET

GET http://localhost:8080/api/ + ...

http://localhost:8080/api/users Save

GET http://localhost:8080/api/users Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Beautify

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

1

Body Cookies Headers (5) Test Results 200 OK 13 ms 486 B Save Response

Pretty Raw Preview Visualize JSON

```
11 "firstName": "John",
12 "lastName": "Doe",
13 "email": null,
14 "budget": 500.0
15 },
16 {
17   "id": 4,
18   "firstName": "Natasha",
19   "lastName": "N",
20   "email": null,
21   "budget": 200.0
22 },
23 {
24   "id": 5,
25   "firstName": "Natasha1",
26   "lastName": "Net2",
27   "email": "nn@mail.com",
28   "budget": 200.0
29 }
30 }
```

URL address: <http://localhost:8080/api/pets>

Method: **POST**

Body:

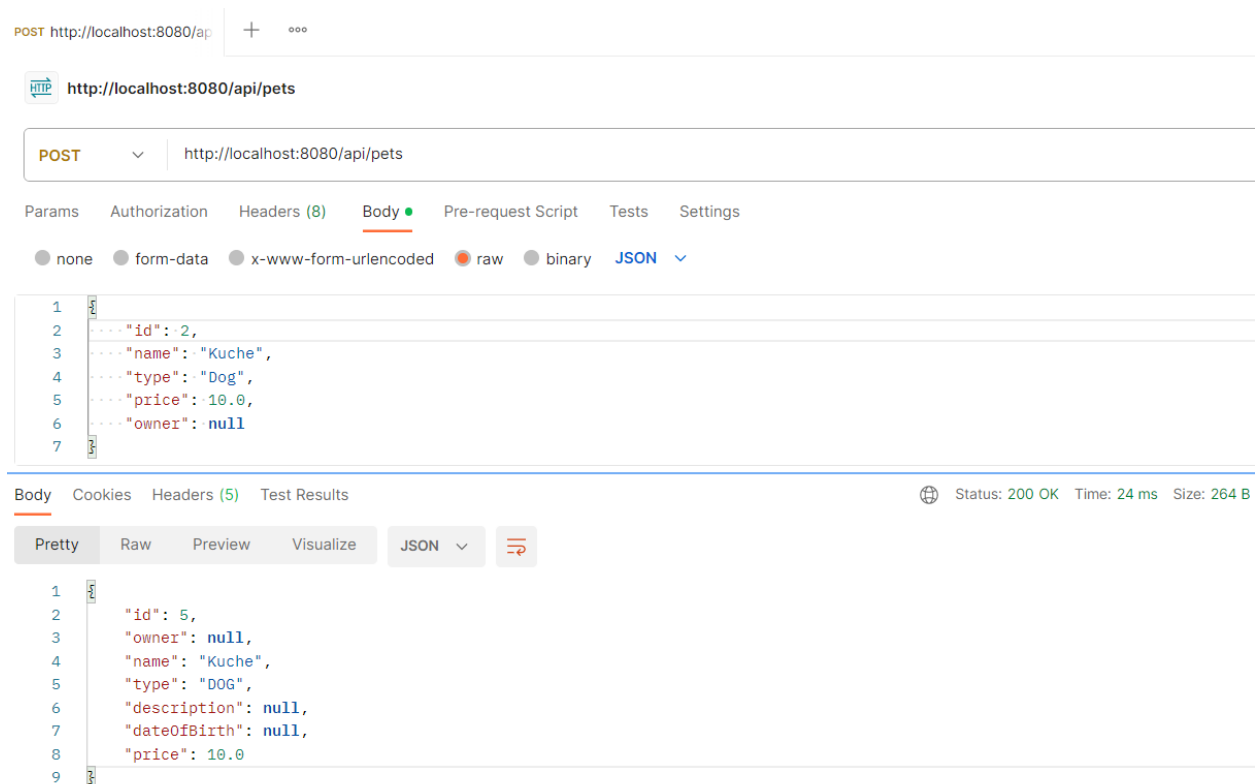
```
{
  "id": 1,
  "name": "Mache",
  "type": "CAT",
  "price": 12.0,
  "owner": null
}
```

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

- Request:**
 - Method: **POST**
 - URL: **http://localhost:8080/api/pets**
 - Body (JSON):

```
1 {
2   "id": 1,
3   "name": "Mache",
4   "type": "CAT",
5   "price": 12.0,
6   "owner": null
7 }
8
```
- Response:**
 - Status: **200 OK**
 - Time: **174 ms**
 - Size: **264 B**
 - Body (JSON):

```
1 {
2   "id": 4,
3   "owner": null,
4   "name": "Mache",
5   "type": "CAT",
6   "description": null,
7   "dateOfBirth": null,
8   "price": 12.0
9 }
```



URL address: <http://localhost:8080/api/pets>

Method: POST

```
{
  "id": 2,
  "name": "Kuche",
  "type": "Dog",
  "price": 10.0,
  "owner": null
}
```

URL address: <http://localhost:8080/api/pets>

Method: GET

HTTP http://localhost:8080/api/pets Save </>

GET http://localhost:8080/api/pets Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies Beautify

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

1

Body Cookies Headers (5) Test Results Status: 200 OK Time: 25 ms Size: 574 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 5,
4     "owner": null,
5     "name": "Kuche",
6     "type": "DOG",
7     "description": null,
8     "dateOfBirth": null,
9     "price": 10.0
10  },
11  {
12    "id": 4,
13    "owner": null,
14    "name": "Mache",
15    "type": "CAT",
16    "description": null,
17    "dateOfBirth": null,
18    "price": 12.0
```

HTTP http://localhost:8080/api/buy Save

POST http://localhost:8080/api/buy Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies Beautify

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

1

Body Cookies Headers (5) Test Results Status: 200 OK Time: 1777 ms Size: 197 B Save Response

Pretty Raw Preview Visualize **Text**

```
1 The pets are bought successfully!
```

The owner_id in Pets database table should be empty (that means that the Pet is not bought and it don't have owner):

public.pet/airport/postgres@PostgreSQL 14

Query Query History

```

1 SELECT * FROM public.pet
2 ORDER BY id ASC

```

Data Output Messages Notifications

	id [PK] bigint	date_of_birth date	description character varying (255)	name character varying (255)	price double precision	type character varying (255)	owner_id bigint
1	1	2000-01-01	1	Fluffy	2	CAT	[null]
2	2	[null]	[null]	R	3	CAT	[null]

Preview of “Owner” database table before execution of buy method:

public.owner/airport/postgres@PostgreSQL 14

Query Query History

```

1 SELECT * FROM public.owner
2 ORDER BY id ASC

```

Data Output Messages Notifications

	id [PK] bigint	budget double precision	email character varying (255)	first_name character varying (255)	last_name character varying (255)
1	1	1000	john@net.com	John	Doe
2	2	25	natasha@mail.com	Natasha	Netkovska

After the execution of method:

POST http://localhost:8080/api/buy

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

1 The pets are bought successfully!

Status: 200 OK Time: 39 ms Size: 197 B Save Response

In pgAdmin:

Query Query History

```
1 SELECT * FROM public.owner
2 ORDER BY id ASC
```

Data Output Messages Notifications

	id [PK] bigint	budget double precision	email character varying (255)	first_name character varying (255)	last_name character varying (255)
1	1	998	john@net.com	John	Doe
2	2	25	natasha@mail.com	Natasha	Netkovska

URL Address: <http://localhost:8080/api/buyHistory>

Method: GET


HTTP <http://localhost:8080/api/buyHistory>


GET <http://localhost:8080/api/buyHistory>

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary JSON ▾

1 5

Body Cookies Headers (5) Test Results  Status: 200 OK

Pretty Raw Preview Visualize JSON ▾ 

```
1 {
2   {
3     "dateOfExecution": "2024-04-03T16:23:20.931854",
4     "successfulBuys": 1,
5     "failedBuys": 0
6   },
7   {
8     "dateOfExecution": "2024-04-03T16:23:20.9329596",
9     "successfulBuys": 1,
10    "failedBuys": 1
11  },
12  {
13    "dateOfExecution": "2024-04-03T16:23:20.9659872",
14    "successfulBuys": 1,
15    "failedBuys": 2
16  },
17  {
18    "dateOfExecution": "2024-04-03T16:23:20.9659872",
19    "successfulBuys": 1,
20    "failedBuys": 3
21  }
```

Solved by: Natasha Netkovska