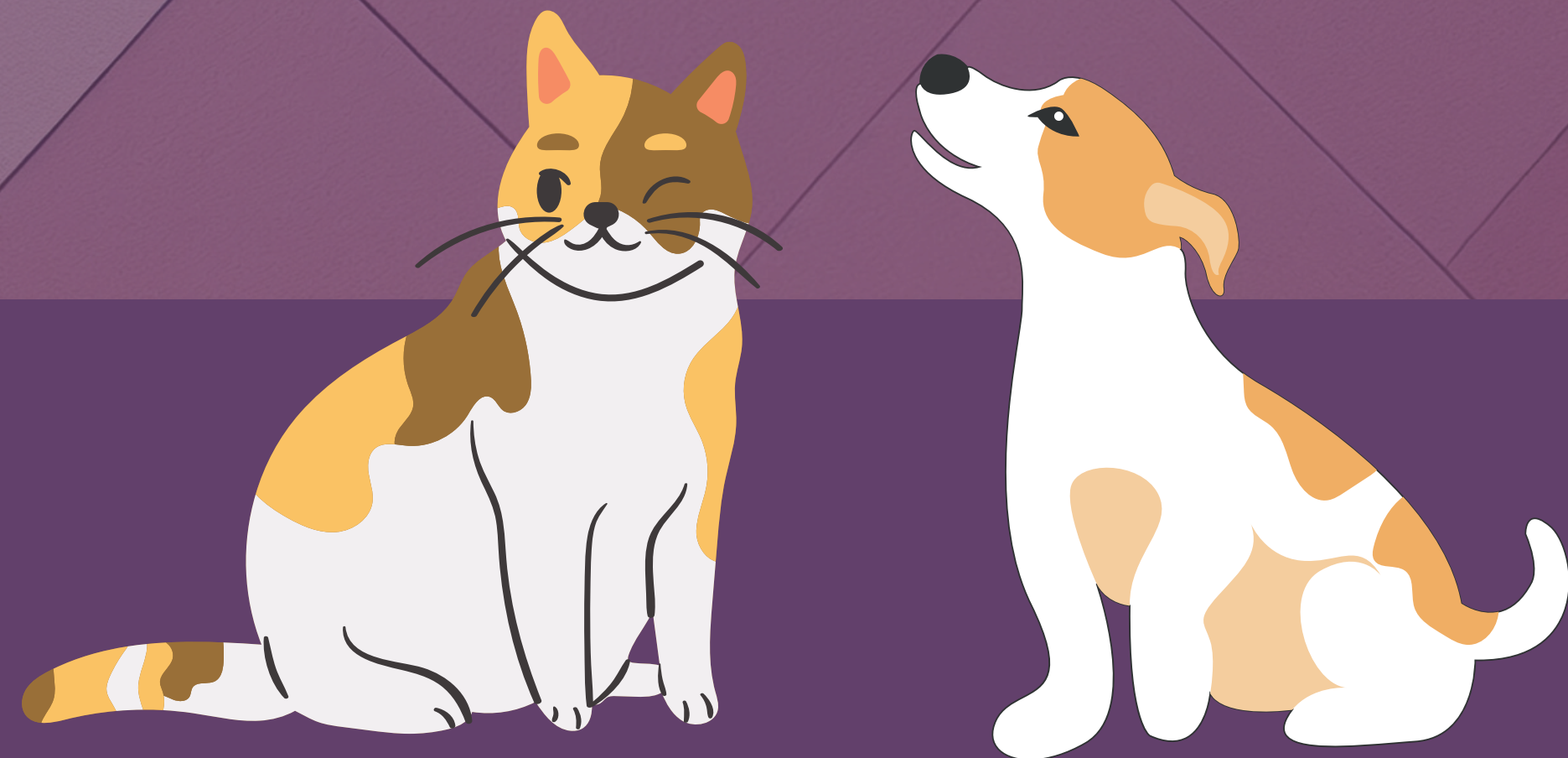


SWAGGER PET STORE ACTIVITY

Presented by
RAHUL CHOWDARY
POOJA M
SHIVANANDA
PRAMOD



Overview of the Swagger Petstore API

The Swagger Petstore API is a sample implementation of a RESTful server that simulates a pet store management system. It provides functionality for adding, updating, and searching for pets in a pet store, managing user accounts, and placing orders. This API serves as an example of how to describe RESTful APIs using Swagger (OpenAPI) specifications.

Key Features:

- Pet Management
- Store Operations
- User Functions

SWAGGER PET STORE API



BaseURL & Configuration

The base URL is the initial part of the address for all the requests in an API. It is the consistent part of the API's internet address and typically includes the protocol (HTTP/HTTPS), the hostname, and the path.

How to Configure the Base URL in Your Application:

- Set the Base URL in the API Client
- Environmental Configuration

Example from Swagger Petstore API:

Base URL: `https://petstore.swagger.io/v2`

Protocol: `https`

Hostname: `petstore.swagger.io`

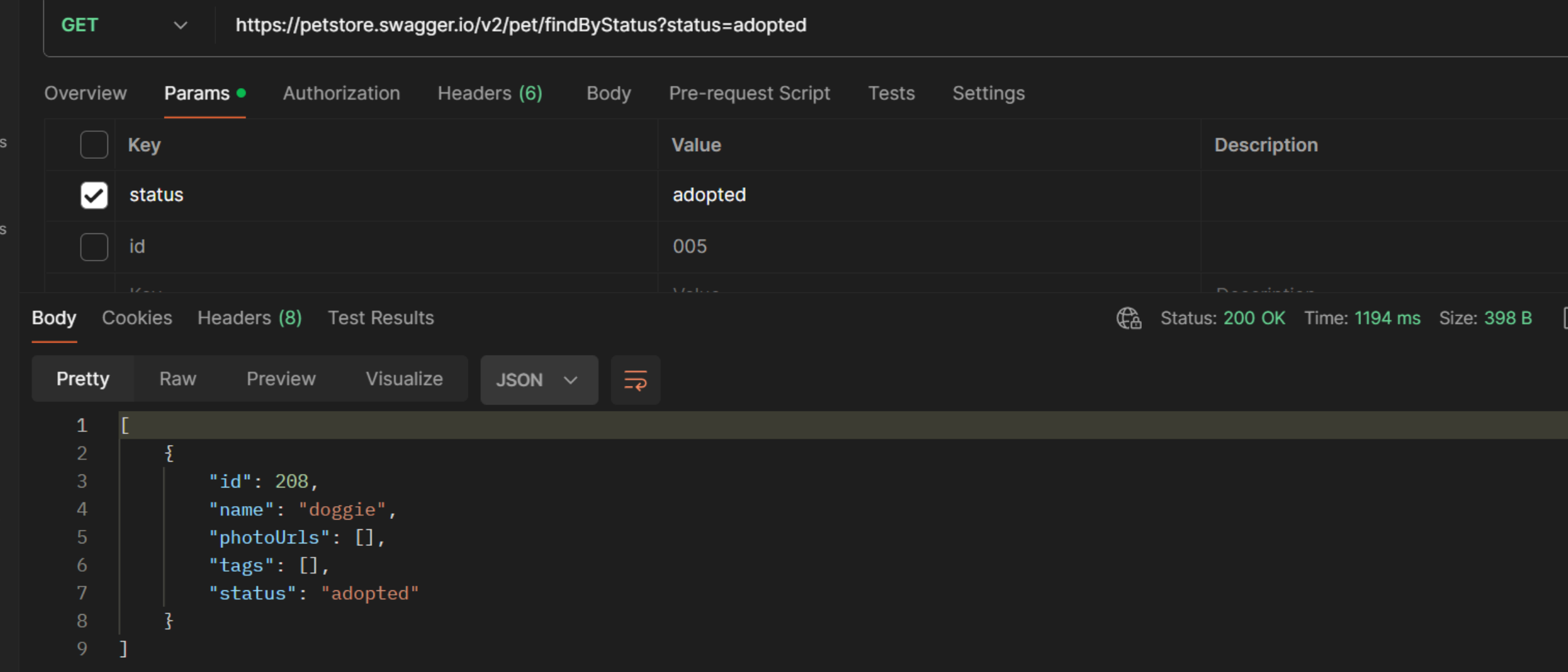
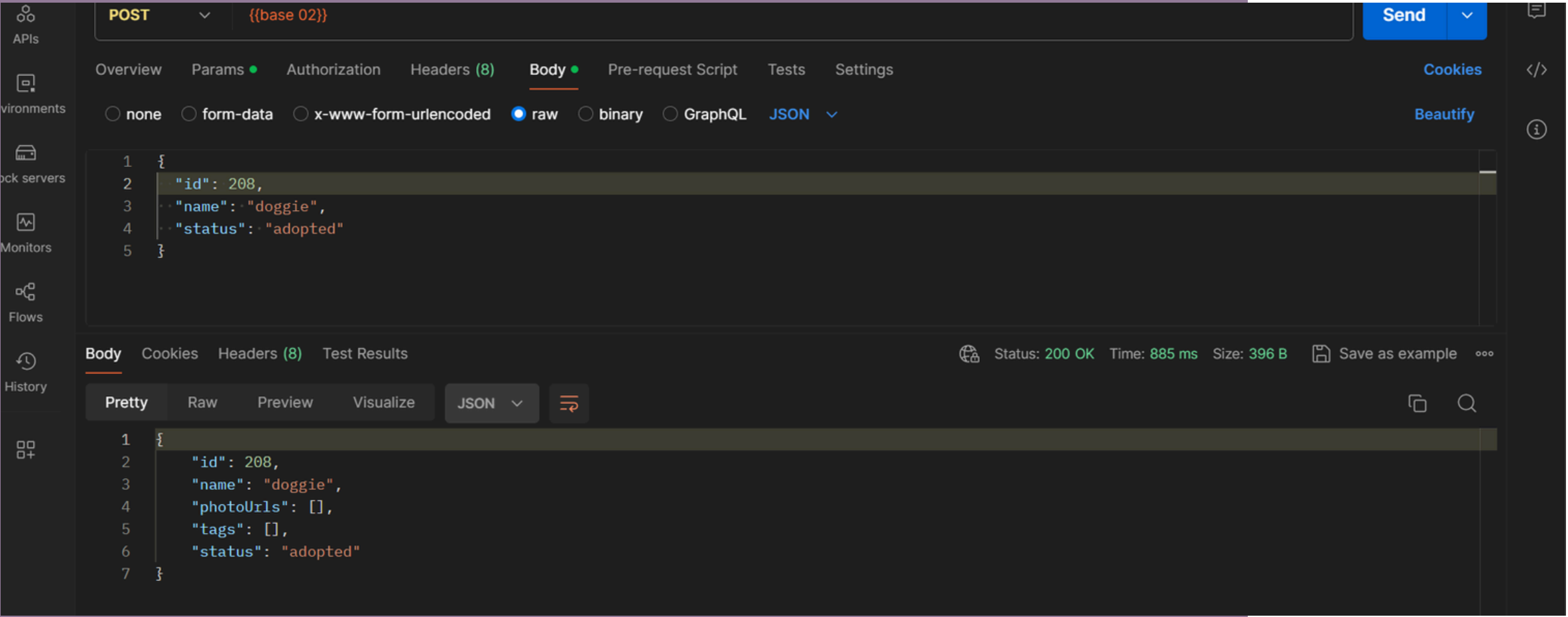
BasePath: `/v2`

ACTIVITY ONE

Adding Pet with id,name,status and Verifying the same (edited)

- Set Request Type to POST and Enter Request URL
- Prepare the JSON Payload
- Send the Request and Verify the Response

ACTIVITY ONE



ACTIVITY TWO

Add a pet as above
but randomize the
pet ID

The screenshot shows a REST client interface for a 'Pets' API. The request is a POST to '{{baseUrl}}'. The body is a JSON object with the following structure:

```
1 {
2   "id": "{{randomInt}}",
3   "name": "Tommy",
4   "status": "Available"
5 }
```

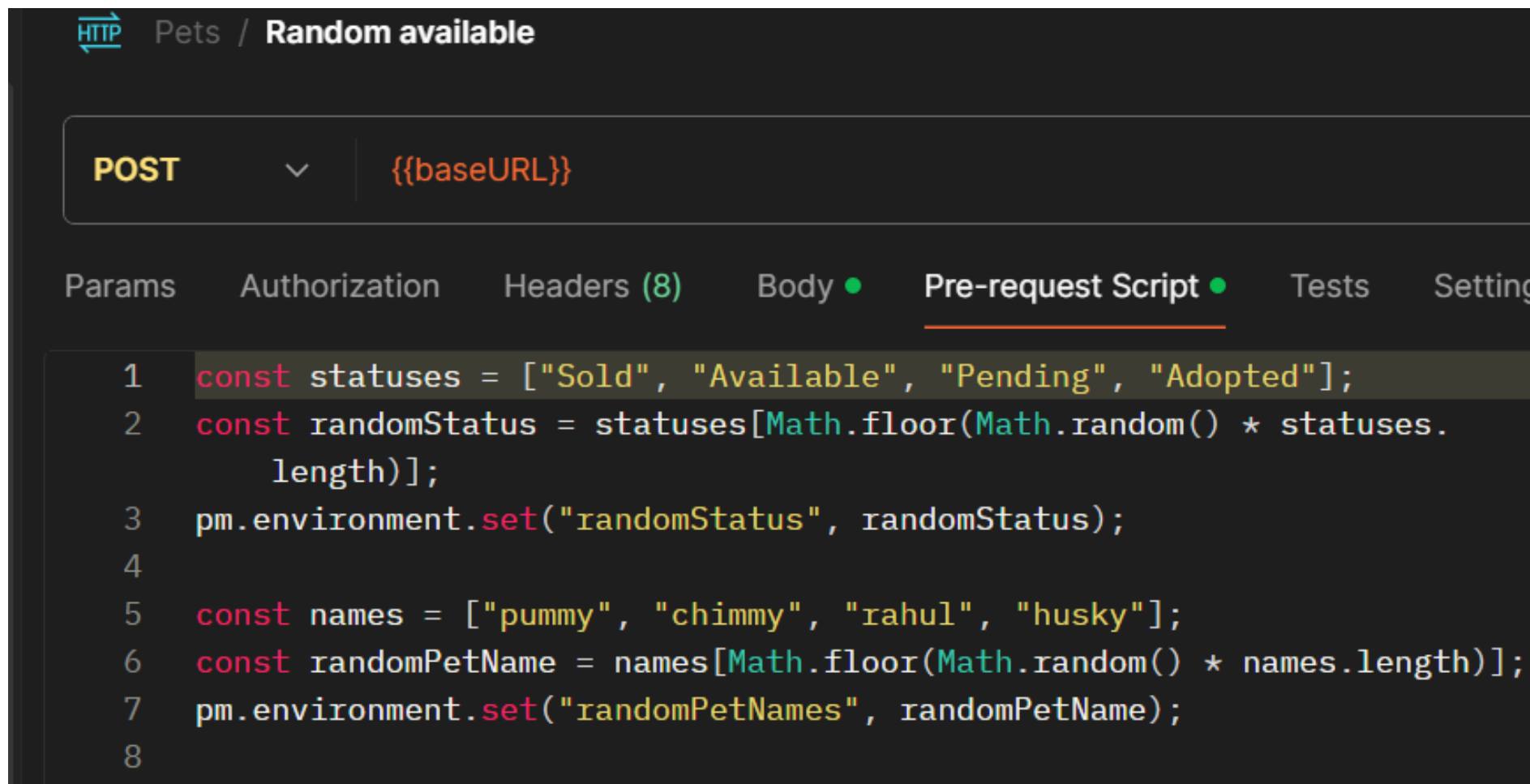
The response is shown in the bottom panel, which is formatted as JSON. The response body is:

```
1 {
2   "id": 621,
3   "name": "Tommy",
4   "photoUrls": [],
5   "tags": [],
6   "status": "Available"
7 }
```

The status bar indicates a successful response with status 200 OK, a response time of 308 ms, and a body size of 397 B. There are also options to 'Save as example' and a search icon.

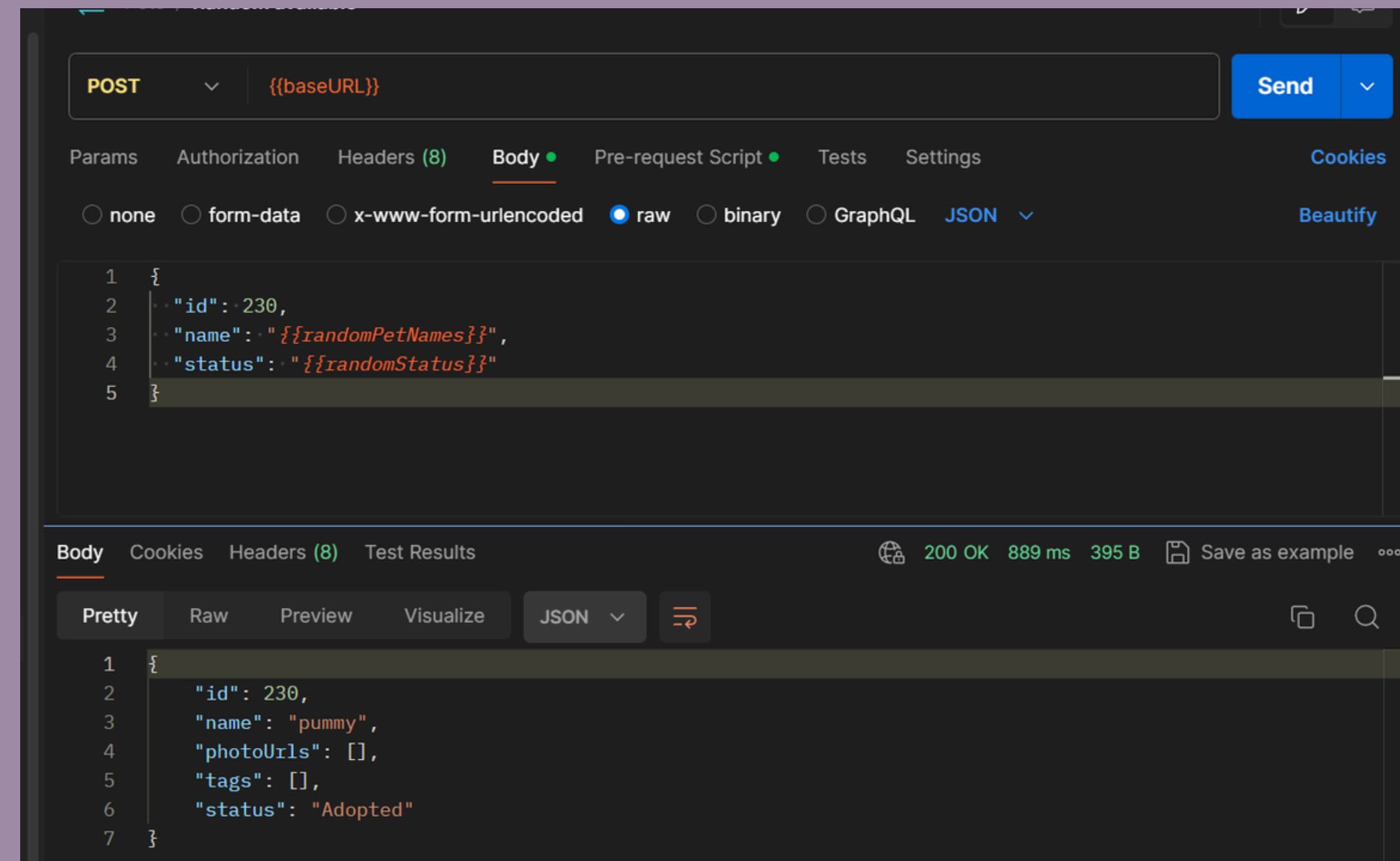
ACTIVITY THREE

Randomize pet Name as
well as the pet Status



The screenshot shows a REST client interface with a POST request to `{{baseUrl}}`. The 'Pre-request Script' tab is active, displaying the following JavaScript code:

```
1 const statuses = ["Sold", "Available", "Pending", "Adopted"];
2 const randomStatus = statuses[Math.floor(Math.random() * statuses.length)];
3 pm.environment.set("randomStatus", randomStatus);
4
5 const names = ["pummy", "chimmy", "rahul", "husky"];
6 const randomPetName = names[Math.floor(Math.random() * names.length)];
7 pm.environment.set("randomPetNames", randomPetName);
8
```



The screenshot shows the same REST client interface, but now the 'Body' tab is active. The request body is a JSON object:

```
1 {
2   "id": 230,
3   "name": "{{randomPetNames}}",
4   "status": "{{randomStatus}}"
5 }
```

Below the request body, the 'Body' tab of the response is shown, displaying the following JSON object:

```
1 {
2   "id": 230,
3   "name": "pummy",
4   "photoUrls": [],
5   "tags": [],
6   "status": "Adopted"
7 }
```


ACTIVITY FOUR

POST request above
where the pet id was
randomized

```
HTTP Pets / Random name of pet

POST {{baseUrl}}

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

1  const statuses = ["Sold", "Available", "Pending", "Adopted"];
2  const randomStatus = statuses[Math.floor(Math.random() * statuses.
    length)];
3  pm.environment.set("randomStatus", randomStatus);
4
5  const names = ["pummy", "chimmy", "rahul", "husky"];
6  const randomPetName = names[Math.floor(Math.random() * names.length)];
7  pm.environment.set("randomPetNames", randomPetName);
8
```

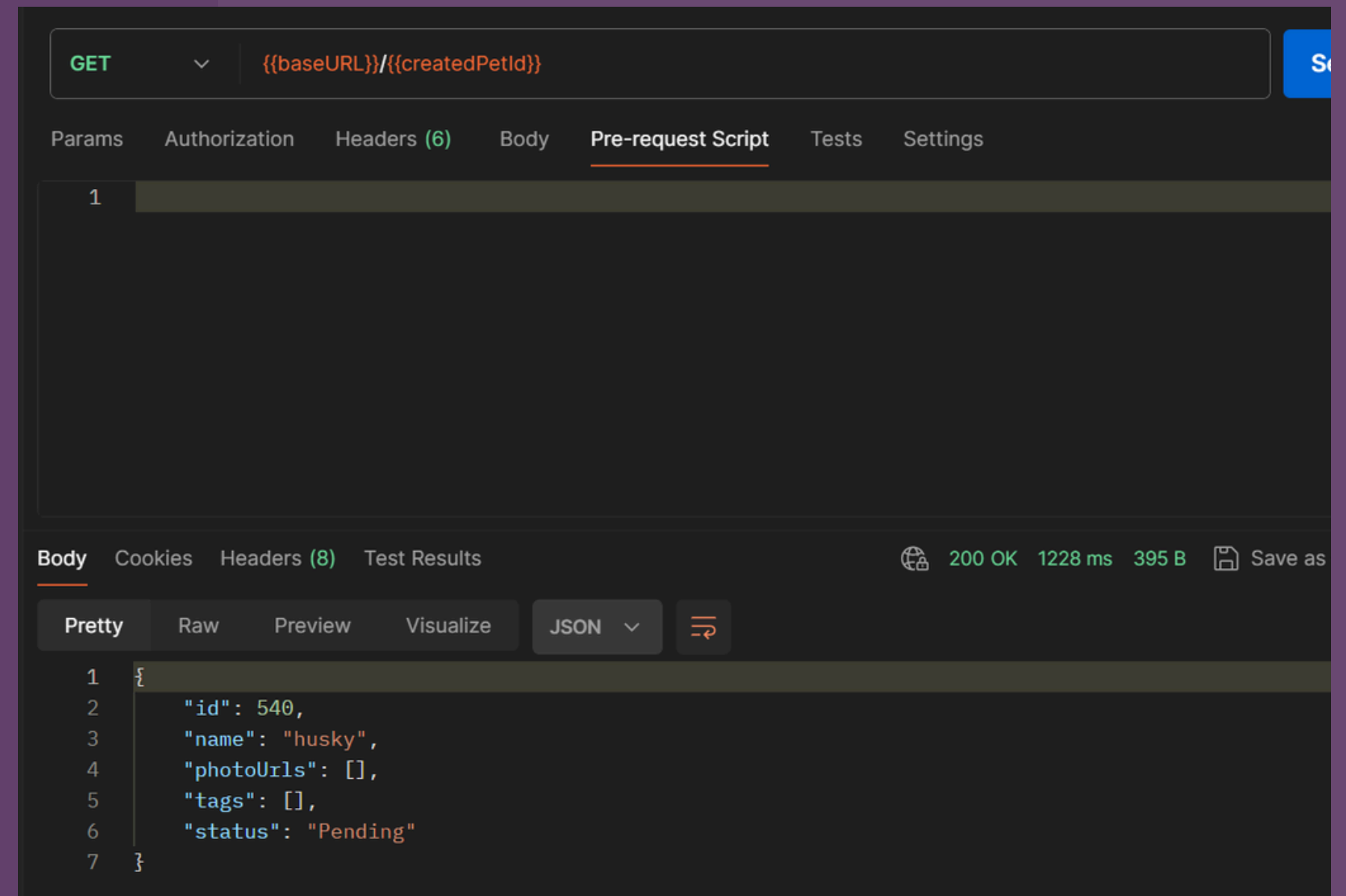
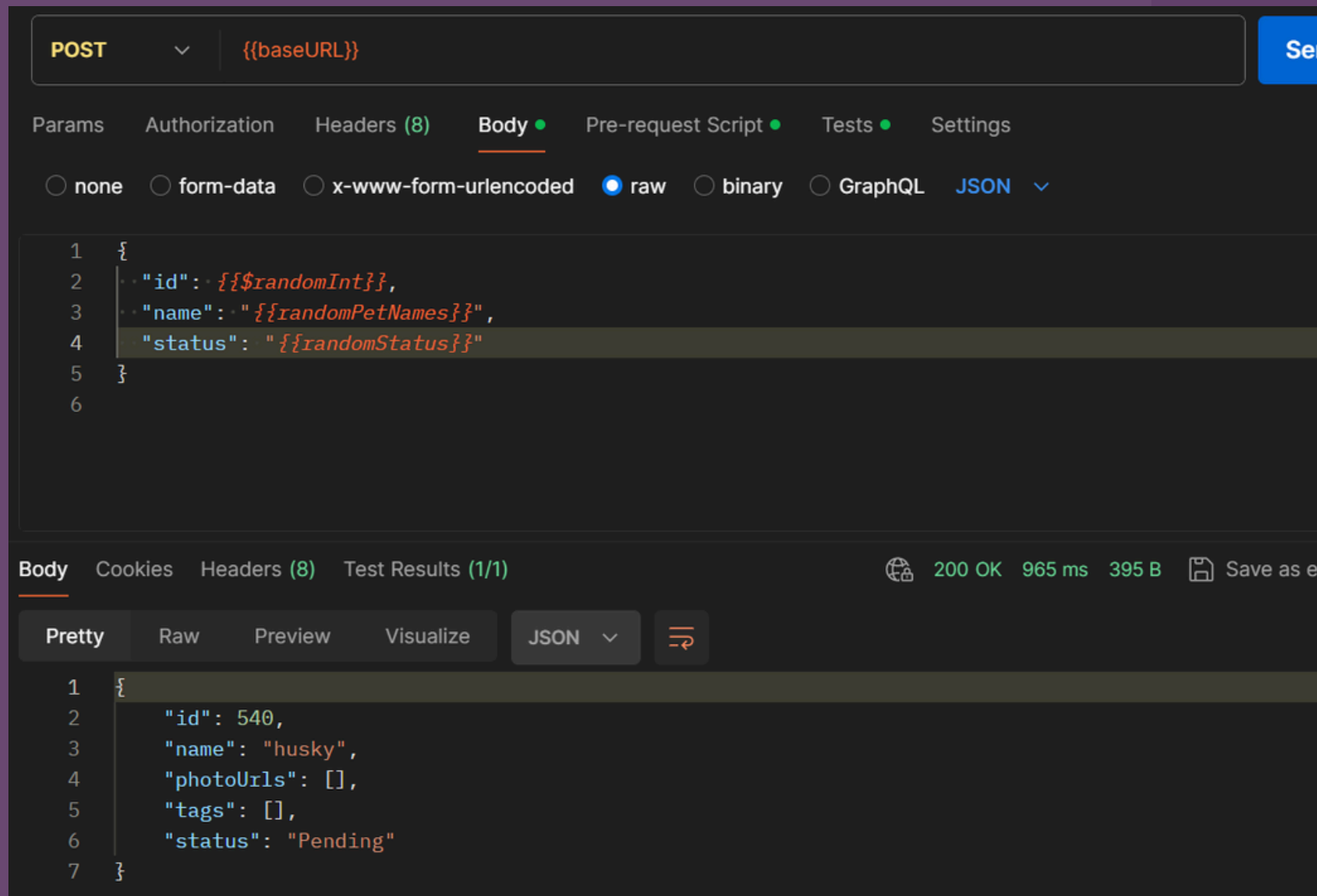
```
HTTP Pets / Random name of pet

POST {{baseUrl}}

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

1  pm.test("Successful post", ()=>{
2      pm.expect(pm.response.code).to.be.oneOf([200,201]);
3  });
4
5  const response=pm.response.json();
6  pm.globals.set("createdPetId",response.id);
7
```

GET request to get the details of the pet created



ACTIVITY FOUR

ACTIVITY FIVE

POST REQUEST

The screenshot displays a REST client interface for a POST request to `https://petstore.swagger.io/v2/pet`. The request body is a JSON object with three fields: `id`, `name`, and `status`, each using a random data placeholder. The response is a 200 OK status with a JSON body containing the same fields with actual values.

Request Details:

- Method: POST
- URL: `https://petstore.swagger.io/v2/pet`
- Body Type: raw (JSON)
- Body Content:

```
1 {  
2   "id": "{{randomInt}}",  
3   "name": "{{randomPetNames}}",  
4   "status": "{{randomStatus}}"  
5 }
```

Response Details:

- Status: 200 OK
- Time: 259 ms
- Size: 395 B
- Body Type: Pretty (JSON)
- Body Content:

```
1 {  
2   "id": 886,  
3   "name": "pummy",  
4   "photoUrls": [],  
5   "tags": [],  
6   "status": "Adopted"  
7 }
```

POST

https://petstore.swagger.io/v2/pet

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

```
1 pm.test("Successful post", ()=>{
2   pm.expect(pm.response.code).to.be.oneOf([200,201]);
3 });
4
5 const response=pm.response.json();
6 pm.globals.set("createdPetId",response.id);
7
```

ACTIVITY FIVE

POST REQUEST

POST

https://petstore.swagger.io/v2/pet

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

```
1 const statuses = ["Sold", "Available", "Pending", "Adopted"];
2
3 const randomStatus = statuses[Math.floor(Math.random() * statuses.length)];
4 pm.environment.set("randomStatus", randomStatus);
5
6 const names = ["pummy", "chimmy", "rahul", "husky"];
7 const randomPetName = names[Math.floor(Math.random() * names.length)];
8 pm.environment.set("randomPetNames", randomPetName);
9
10 console.log(randomPetName,randomStatus);
11
```


TP Pet 02 / Get New data

GET https://petstore.swagger.io/v2/pet/{{createdPetId}}

Params Authorization Headers (6) Body Pre-request Script Tests Settings

```
1 console.log(pm.response.json());
```

dy Cookies Headers (8) Test Results 200 OK 264 ms 395 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 886,
3   "name": "pummy",
4   "photoUrls": [],
5   "tags": [],
6   "status": "Adopted"
7 }
```

GET & PUT REQUEST

PUT https://petstore.swagger.io/v2/pet

Params Authorization Headers (8) Body Pre-request Script

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

```
1 {
2   "id": {{createdPetId}},
3   "name": "veer",
4   "status": "Sold"
5 }
```

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 886,
3   "name": "veer",
4   "photoUrls": [],
5   "tags": [],
6   "status": "Sold"
7 }
```

ACTIVITY FIVE

ACTIVITY FIVE

 Pet 02 / **Updated Data**

GET



https://petstore.swagger.io/v2/pet/{{createdPetId}}

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests ●

Settings

```
1 console.log(pm.response.json());
```

Body

Cookies

Headers (8)

Test Results



200 OK

341 ms

Pretty

Raw


Preview

Visualize

JSON



```
1 {
2   "id": 886,
3   "name": "veer",
4   "photoUrls": [],
5   "tags": [],
6   "status": "Sold"
7 }
```

 Pet 02 / **Delete**

DELETE



https://petstore.swagger.io/v2/pet/{{createdPetId}}

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests ●

Settings

☒ none

☐ form-data

☐ x-www-form-urlencoded

☐ raw

☐ binary

☐ GraphQL

This request does not have a body

Body

Cookies

Headers (8)

Test Results



200 OK

1358 ms

371

Pretty

Raw

Preview

Visualize

JSON



```
1 {
2   "code": 200,
3   "type": "unknown",
4   "message": "886"
5 }
```

HTTP Pet 02 / Updated Data Copy

GET https://petstore.swagger.io/v2/pet/886

Params Authorization Headers (6) Body Pre-request Script Tests ● Settings

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

This request does not have a body

Body Cookies Headers (8) Test Results 404 Not Found 255

Pretty Raw Preview Visualize JSON

```
1 {
2   "code": 1,
3   "type": "error",
4   "message": "Pet not found"
5 }
```

"husky" "Available"

► POST https://petstore.swagger.io/v2/pet

► GET https://petstore.swagger.io/v2/pet/94

► {id: 94, name: "husky", photoUrls: [0]...}

► PUT https://petstore.swagger.io/v2/pet

► {id: 94, name: "veer", photoUrls: [0]...}

► GET https://petstore.swagger.io/v2/pet/94

► {id: 94, name: "veer", photoUrls: [0]...}

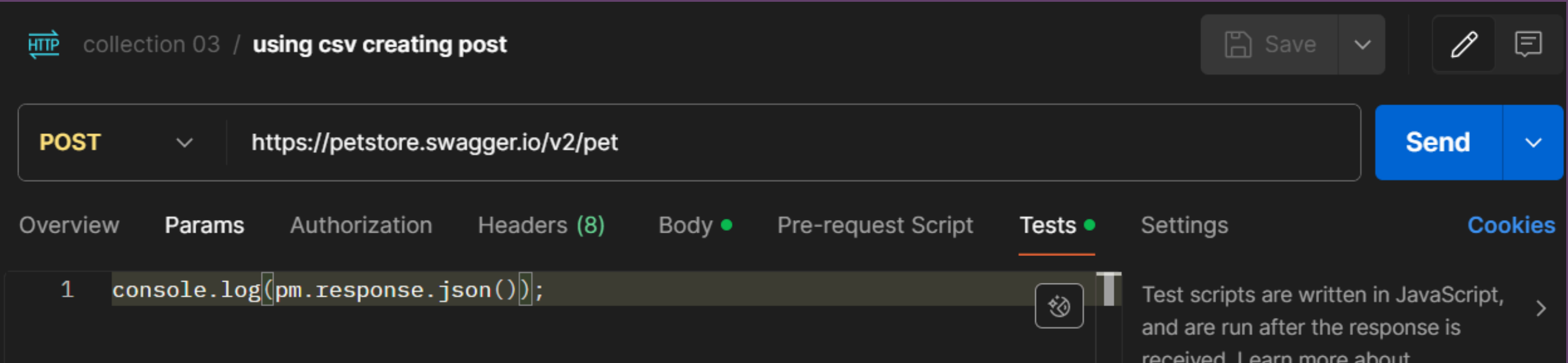
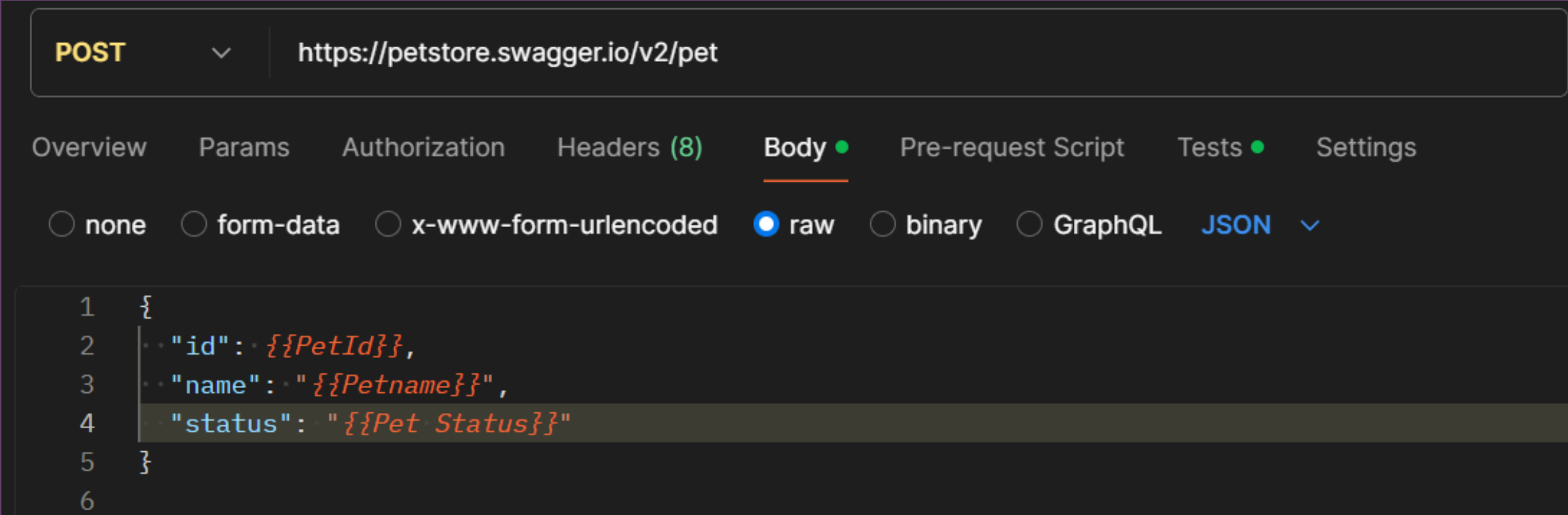
► DELETE https://petstore.swagger.io/v2/pet/94

"Pet is deleted"

ACTIVITY FIVE

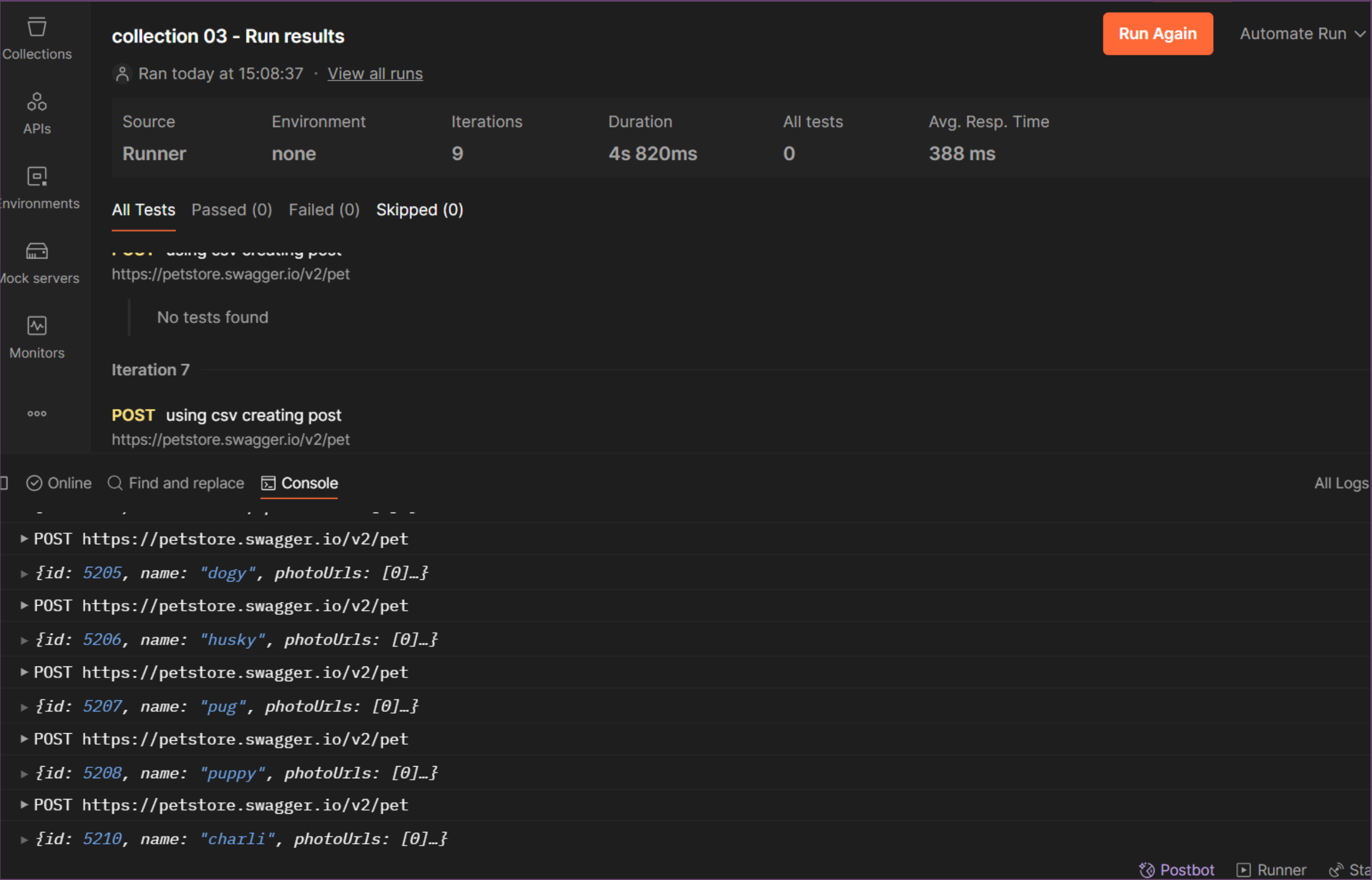
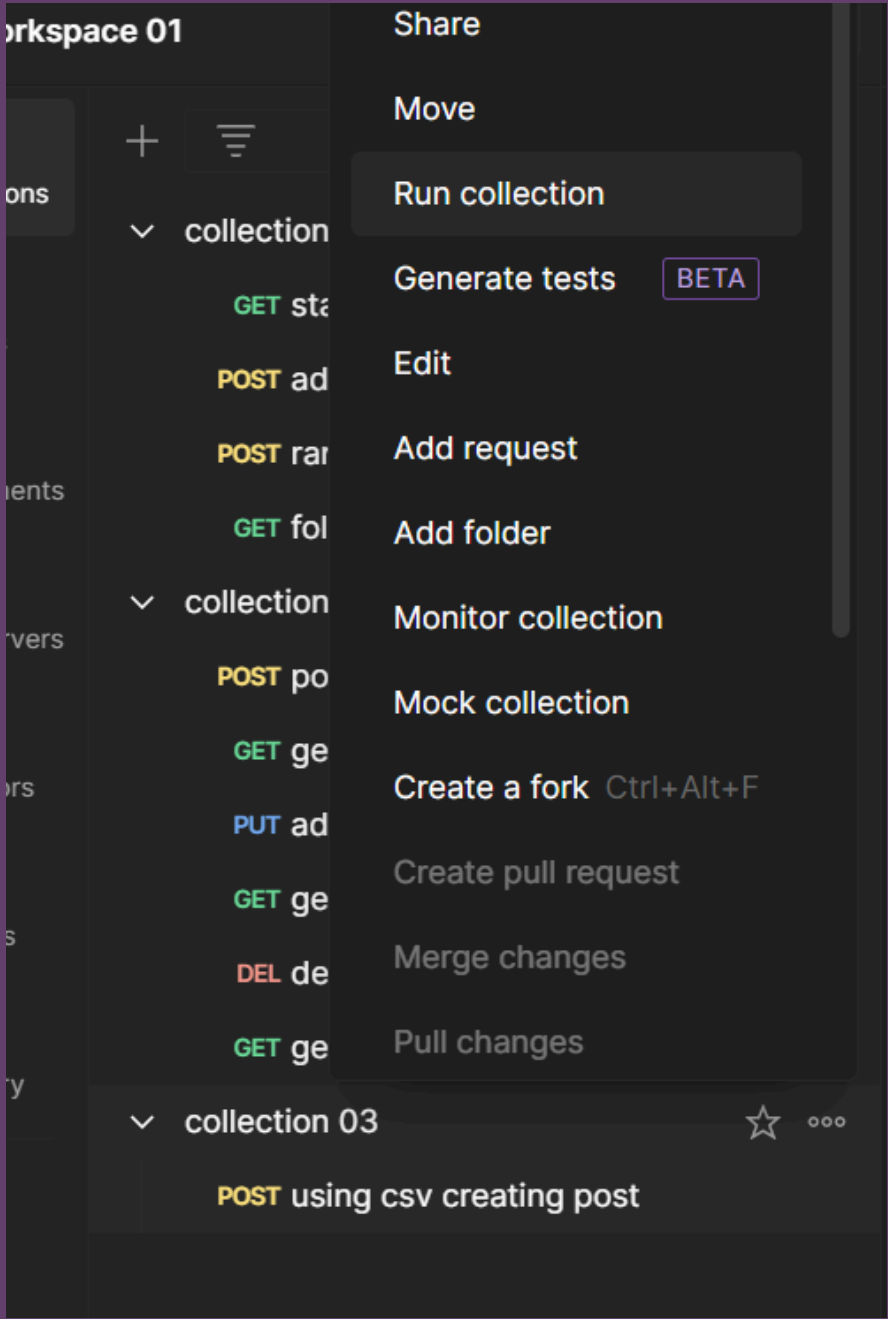
ACTIVITY SIX

```
PetId,Petname,Pet Status
5201,Timmy,alive
5202,devil,hungry
5203,rocky,happy
5204,sam,sad
5205,dogy,adopted
5206,husky,sleeping
5207,pug,playing
5208,puppy,shouting
5210,charli,alive
```



CSV FILE WITH TEST DATA

ACTIVITY SIX



CSV FILE WITH TEST DATA

THANK YOU

