



# Instructions to use CAPIF APIs with Postman

---

EVOLVED-5G

- Install POSTMAN

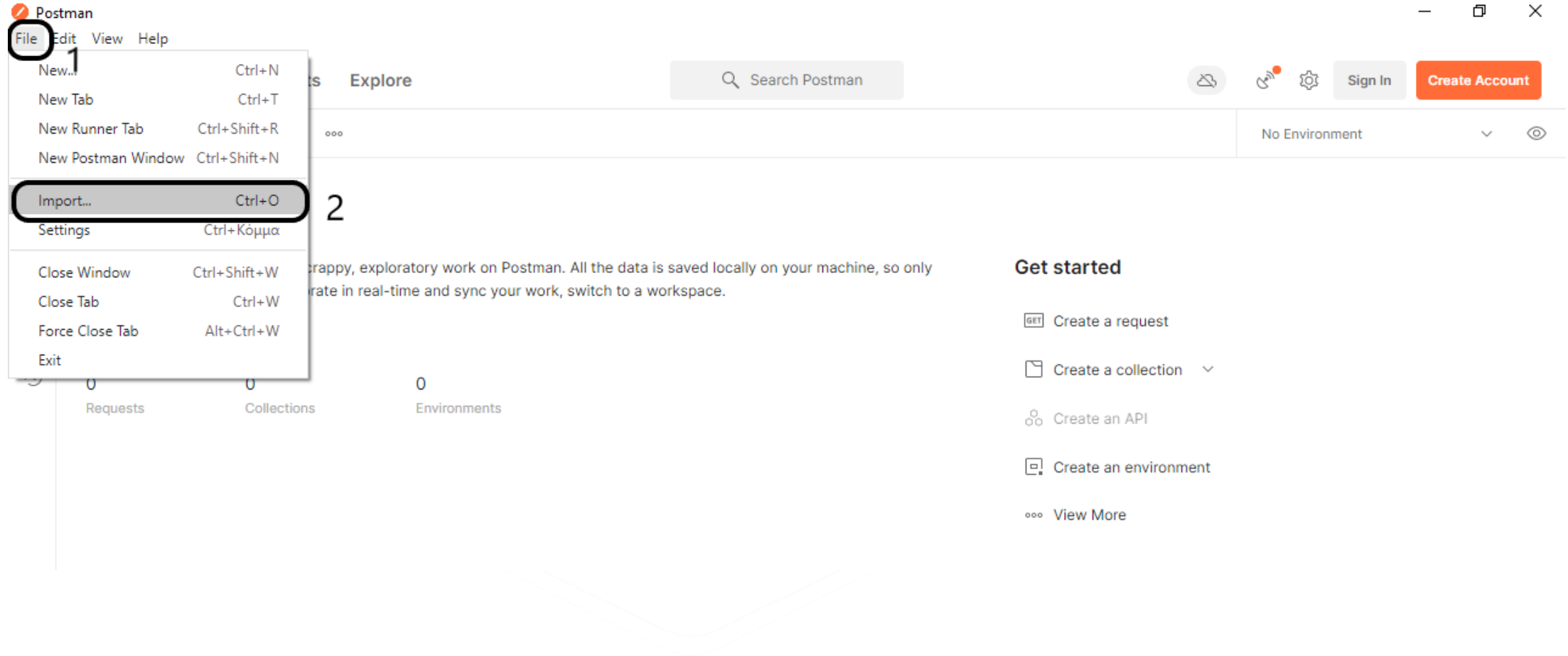
- Postman is an API platform for building and using APIs.
- Download from: <https://www.postman.com/downloads/>  
(for Linux distros, POSTMAN is available also in the corresponding Software Center Application)



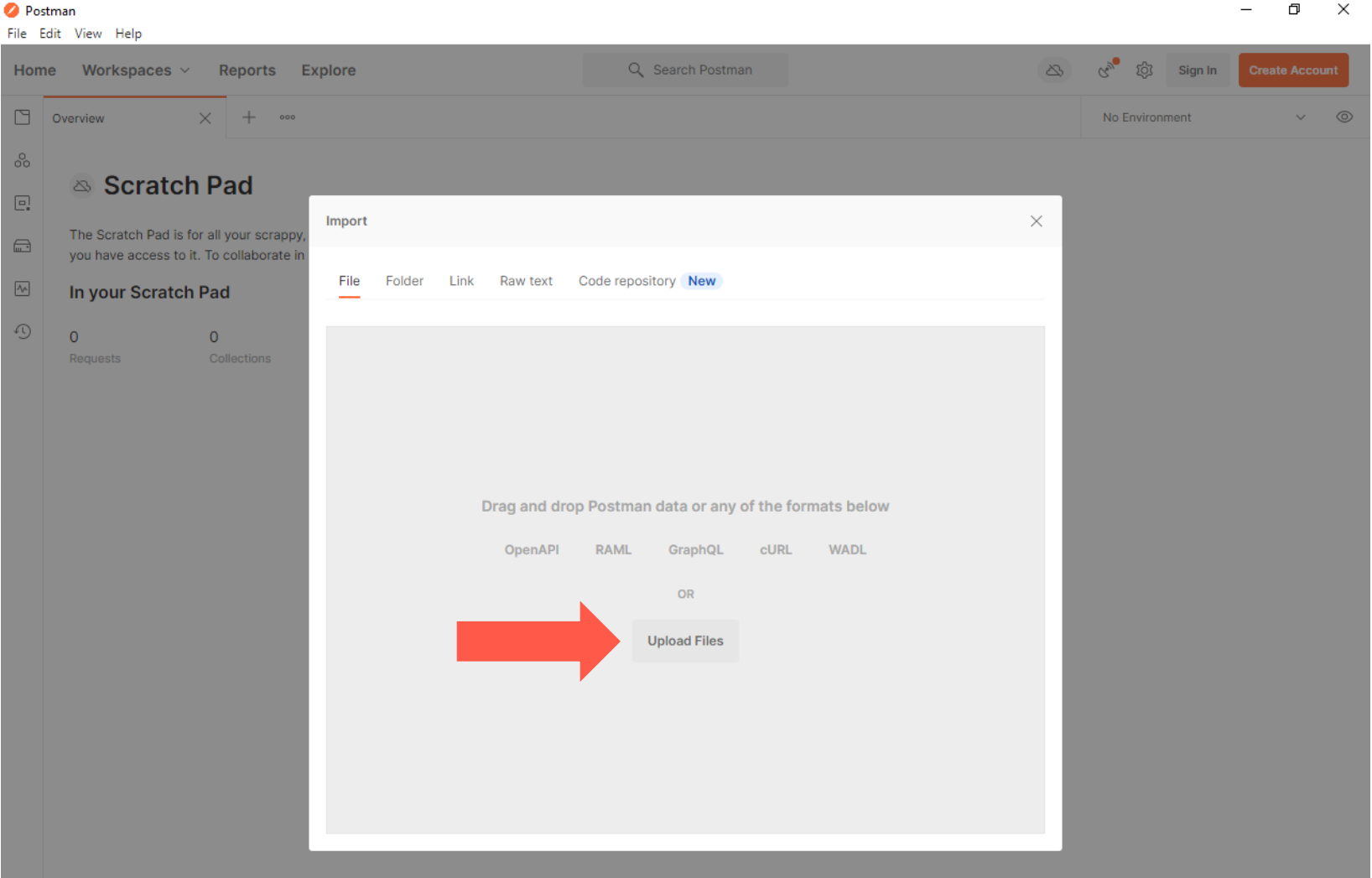
POSTMAN

- Clone CAPIF Tool from Github and deploy

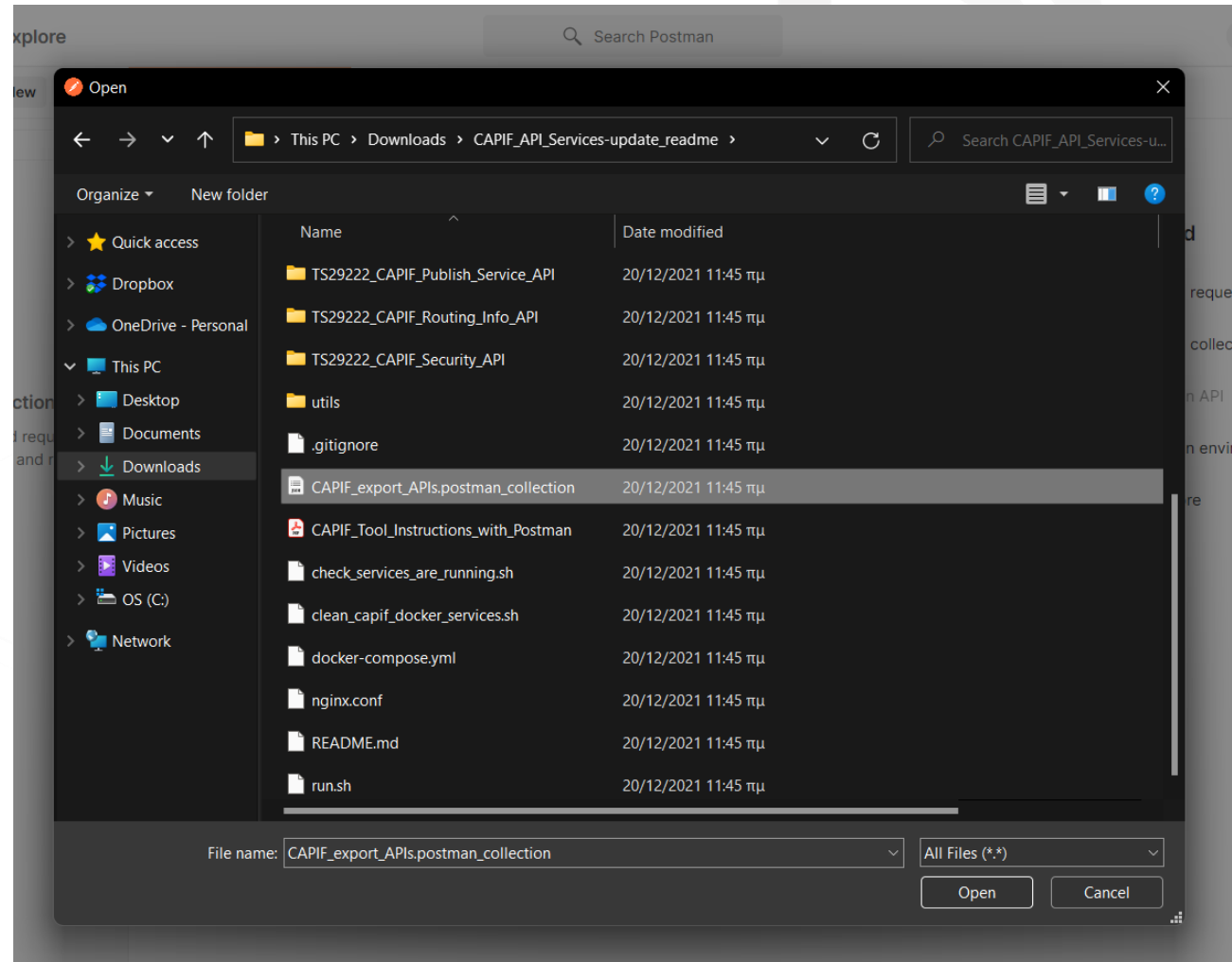
# Open POSTMAN



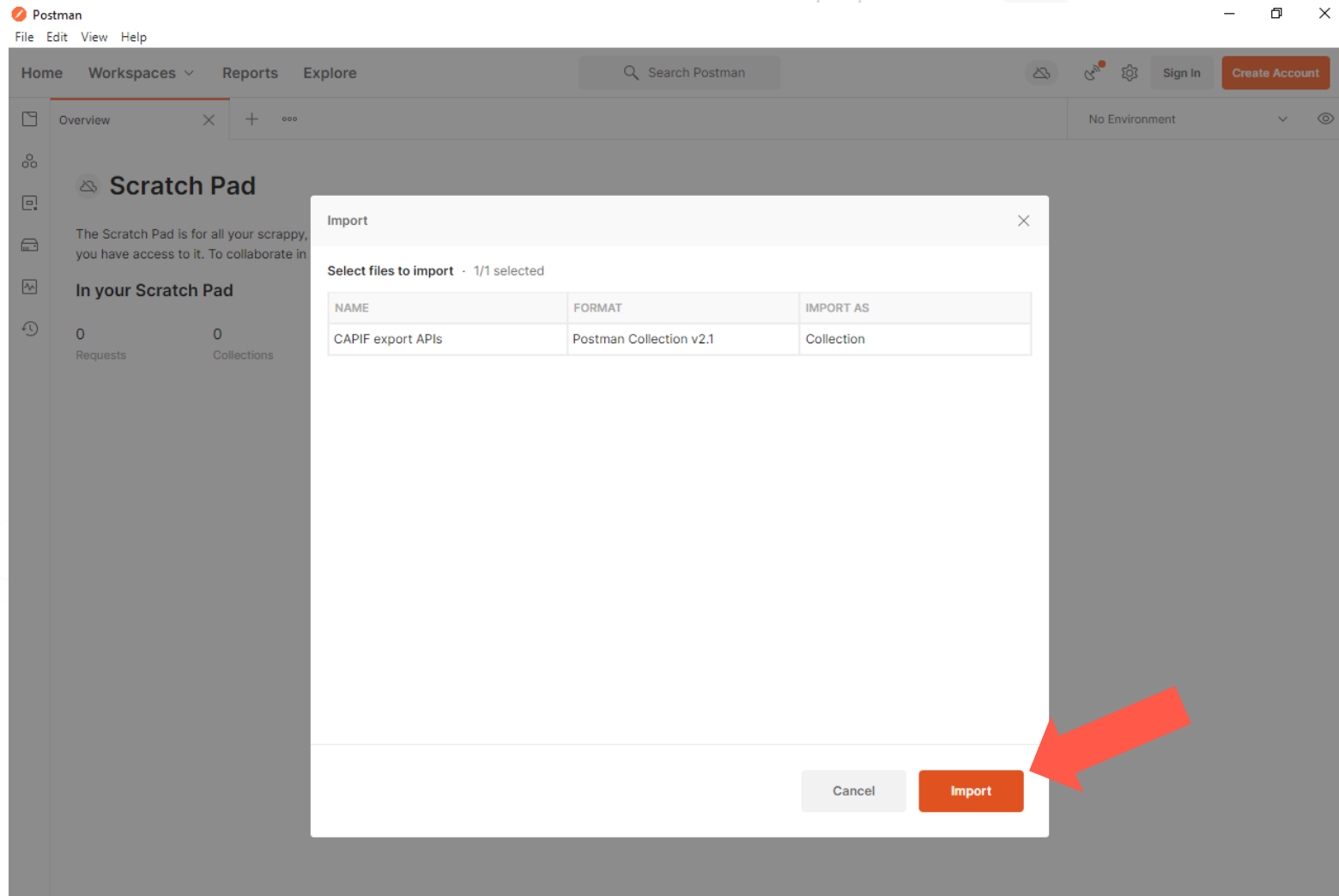
# Import Menu



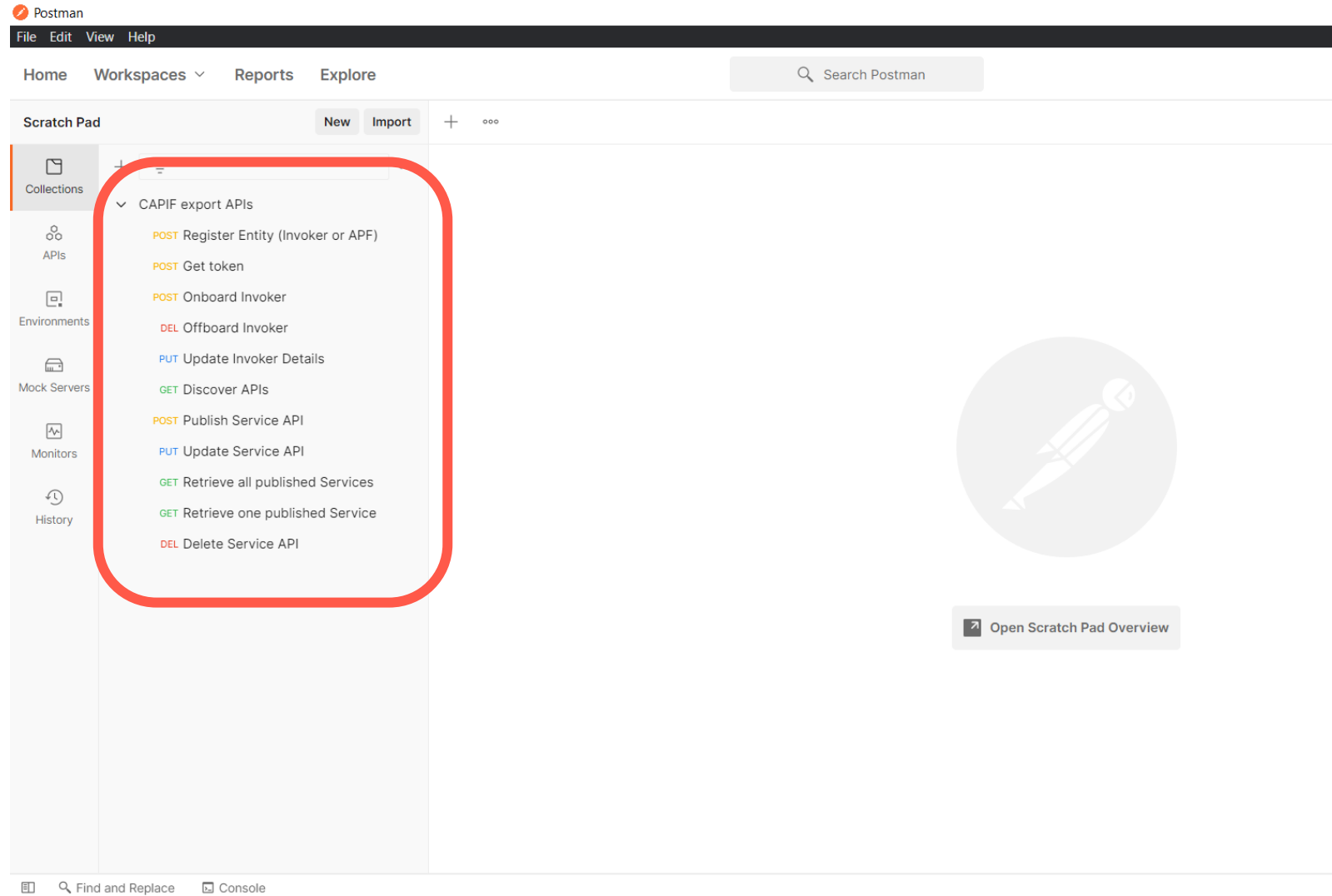
# Select “CAPIF\_export\_APIs.postman\_collection.json”



# Final Import



# List of CAPIF APIs imported



# Process of Executing an API call

1. Double-click an API

2. Select method

3. Edit URL

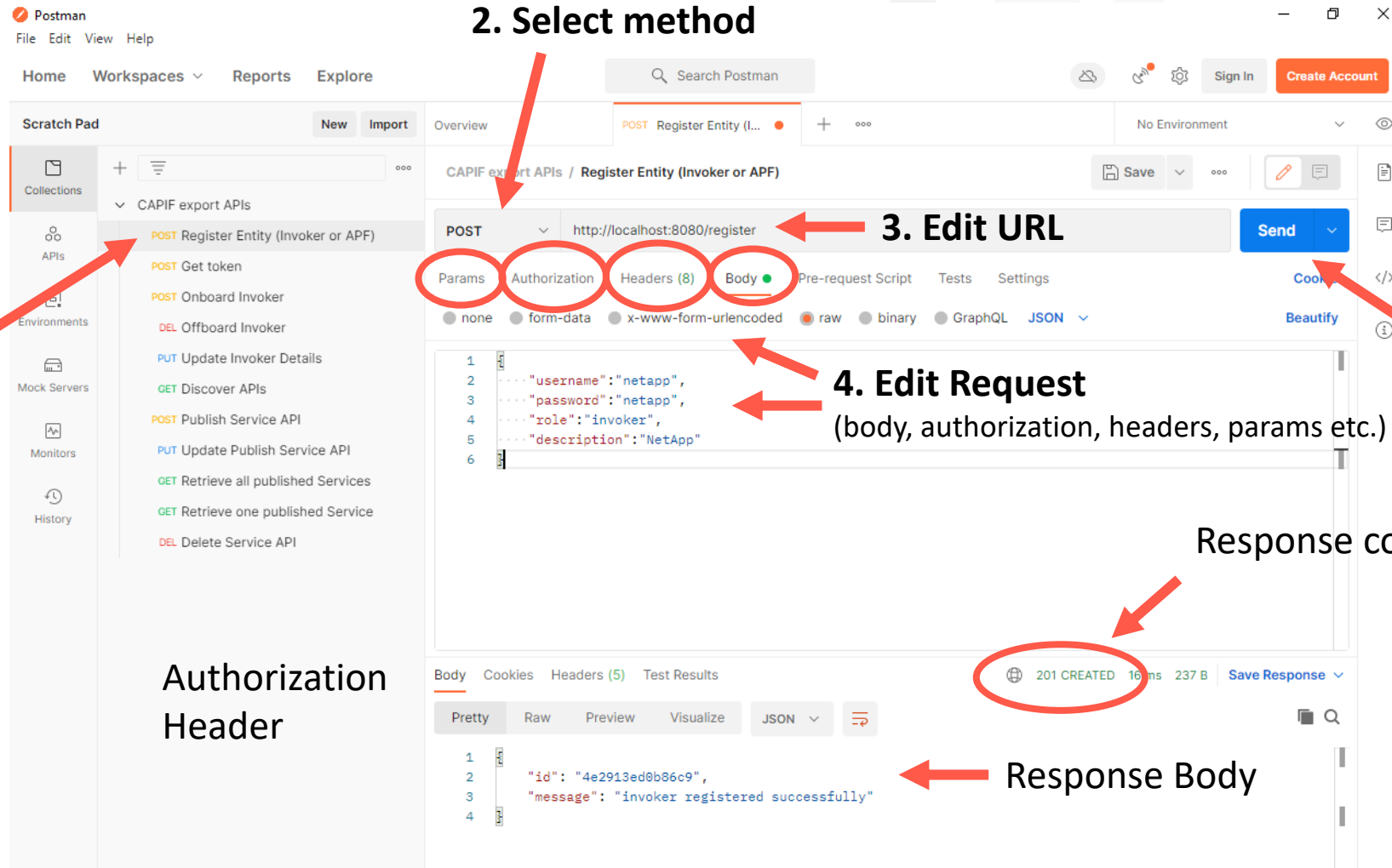
4. Edit Request  
(body, authorization, headers, params etc.)

4. Press to execute

Authorization Header

Response code

Response Body



Postman interface showing the process of executing an API call. The interface includes a left sidebar with 'Collections' and 'Environments', a main workspace with a 'POST' method selected, and a bottom section showing the 'Response code' and 'Response Body'.

API: POST Register Entity (Invoker or APF)

URL: http://localhost:8080/register

Method: POST

Request Body (JSON):

```
1 {
2   "username": "netapp",
3   "password": "netapp",
4   "role": "invoker",
5   "description": "NetApp"
6 }
```

Response Code: 201 CREATED

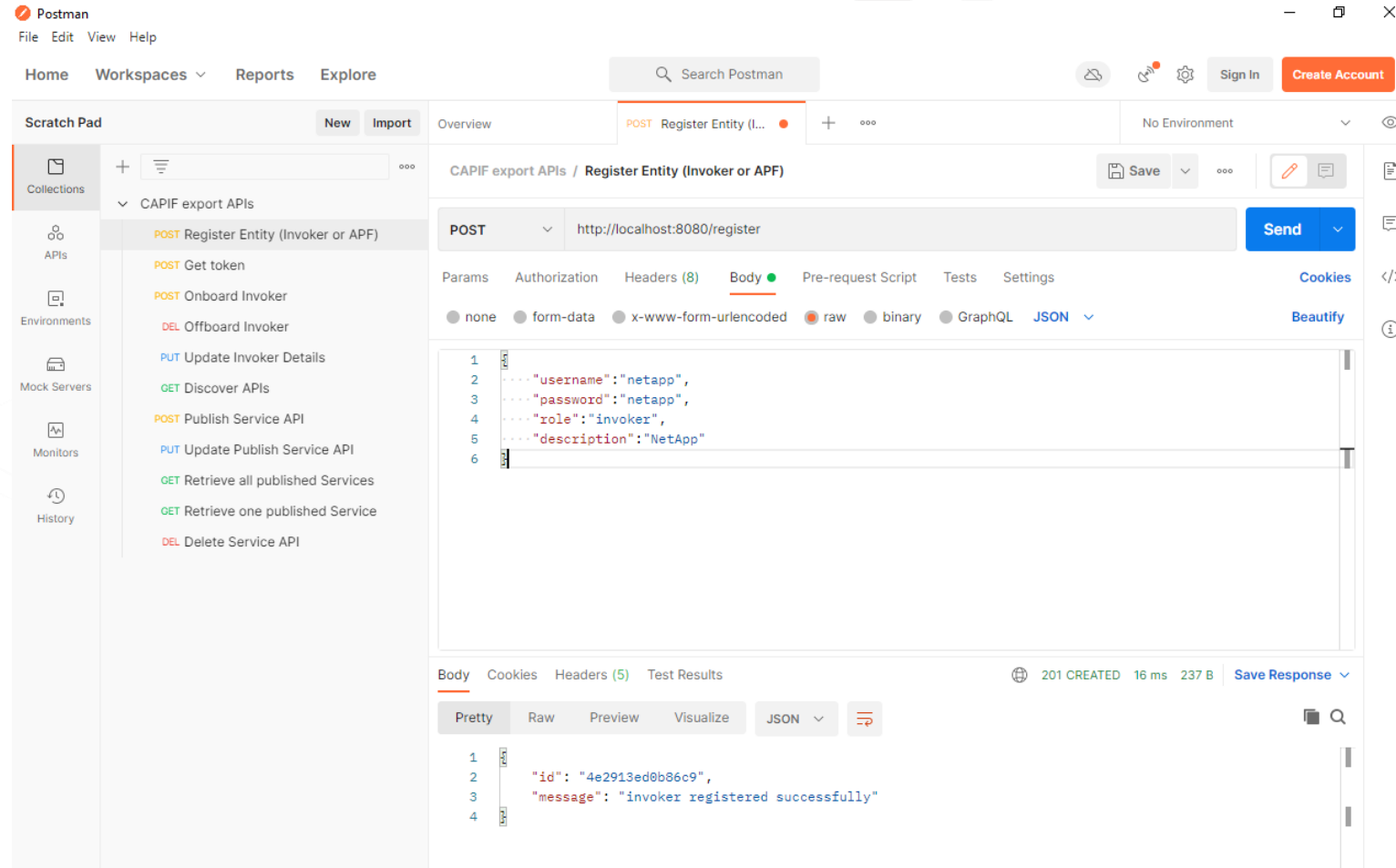
Response Body (JSON):

```
1 {
2   "id": "4e2913ed0b86c9",
3   "message": "invoker registered successfully"
4 }
```



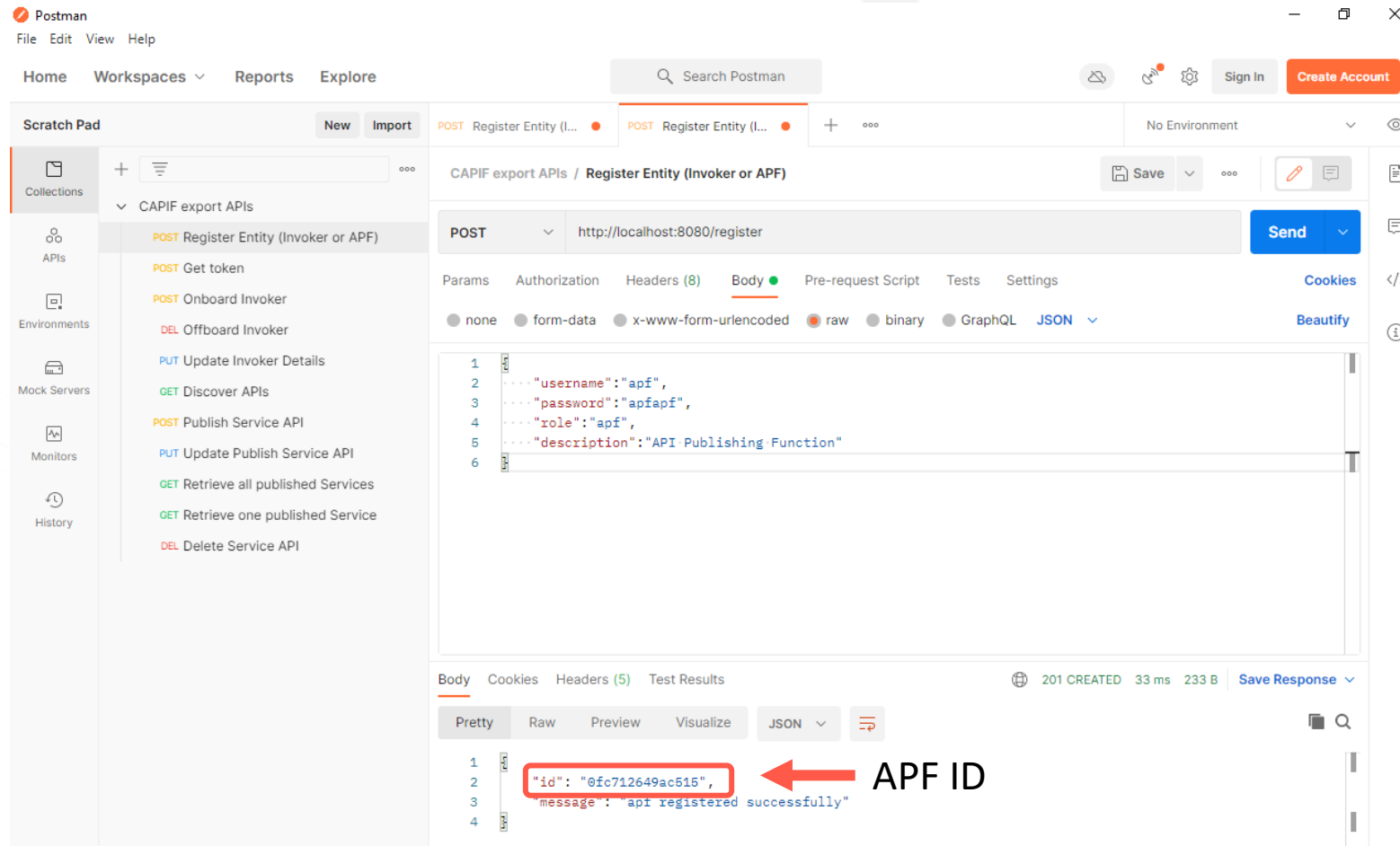
# Register Entity (NetApp)

- POST `http://<CAPIF Host IP>:8080/register`



# Register Entity (APF)

- POST `http://<CAPIF Host IP>:8080/register`



- broken

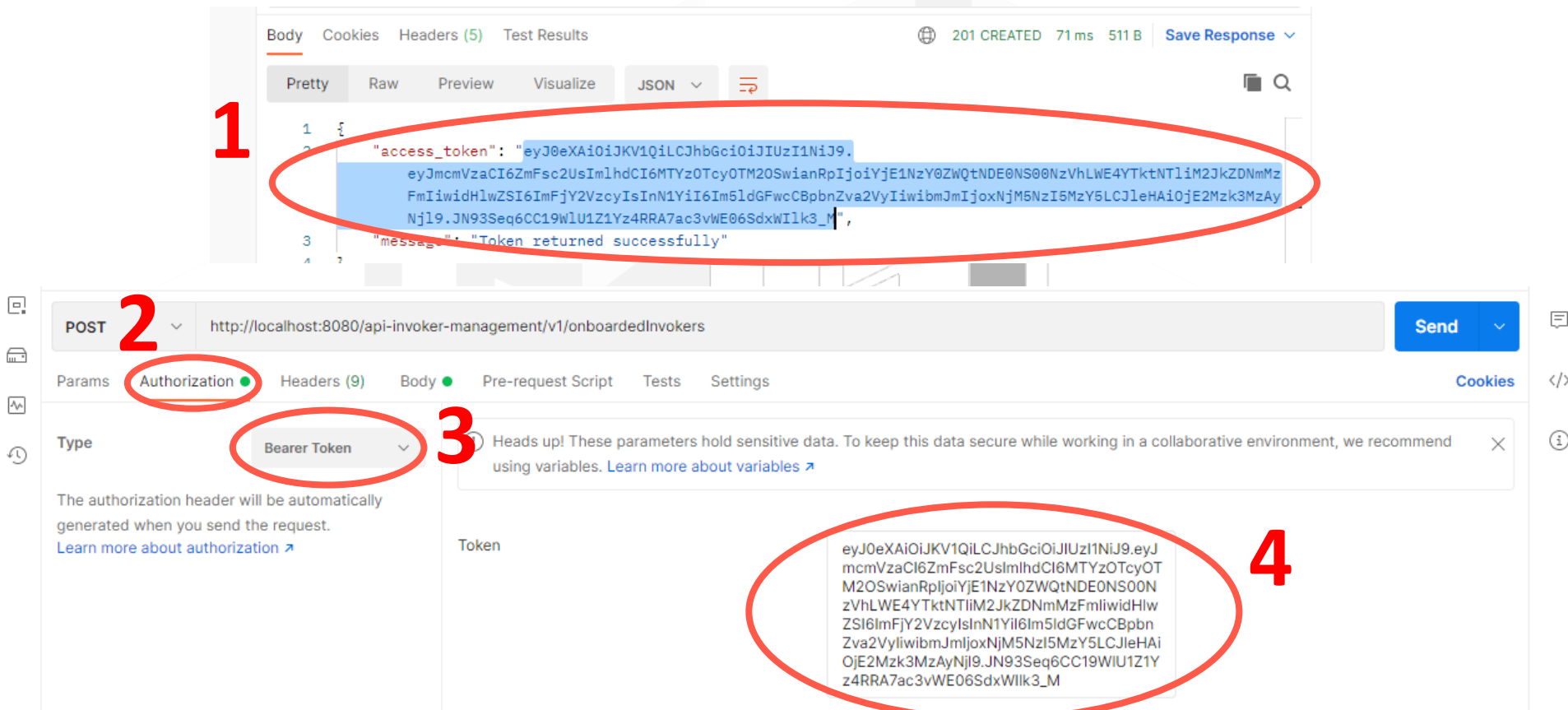


- token



# Use JSON Web Token

- For each API call of CAPIF, we have to copy the corresponding token (of the NetApp or the APF) to Authorization header

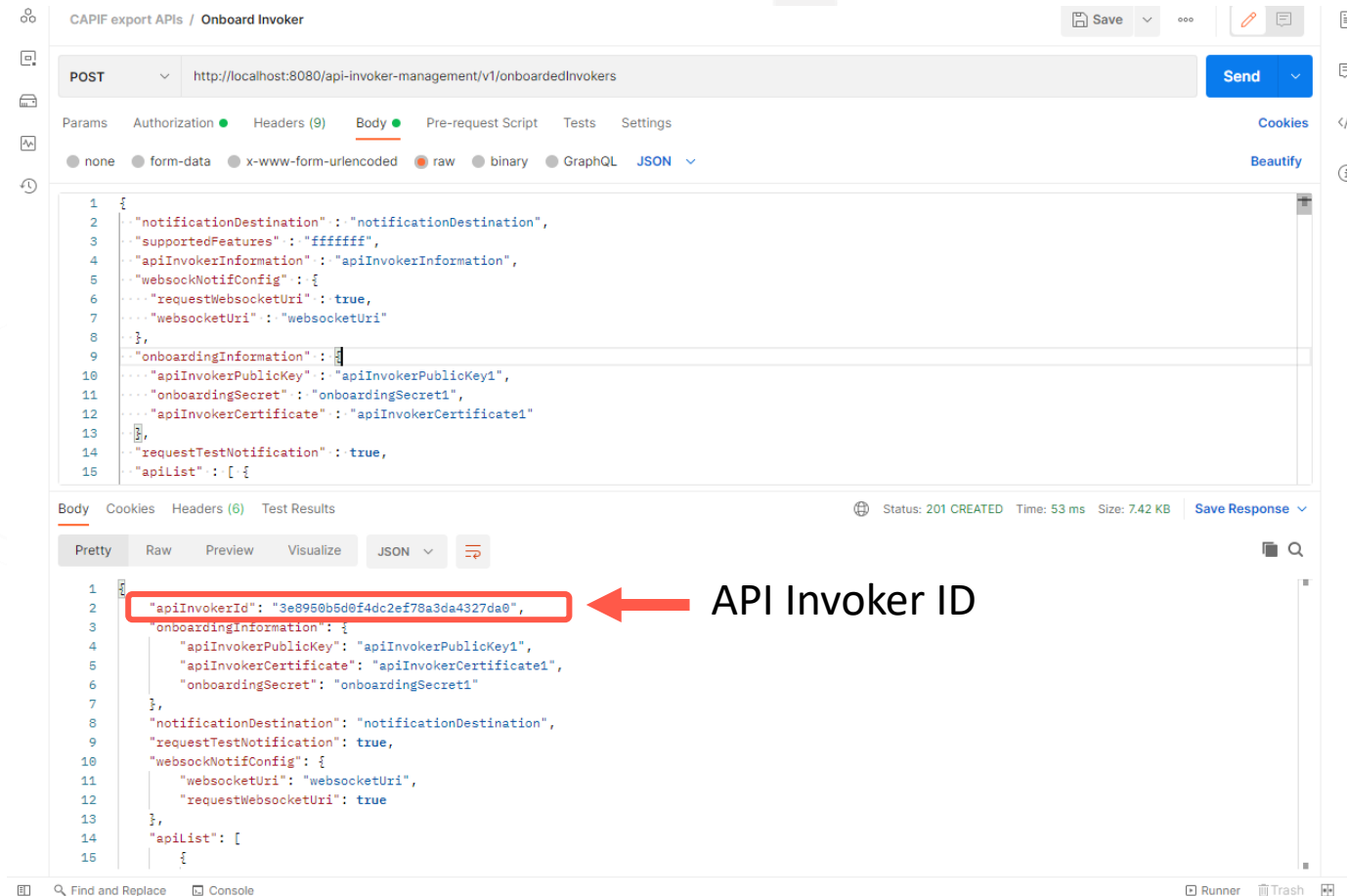


The screenshot illustrates the steps to use a JSON Web Token (JWT) for API authentication in Postman:

- 1**: The response body of the API call is displayed in JSON format. The `access_token` value is highlighted in blue.
- 2**: The **Authorization** tab is selected in the Postman interface.
- 3**: The **Bearer Token** type is selected for the authorization header.
- 4**: The JWT token is copied from the response body and pasted into the **Token** field of the authorization header.

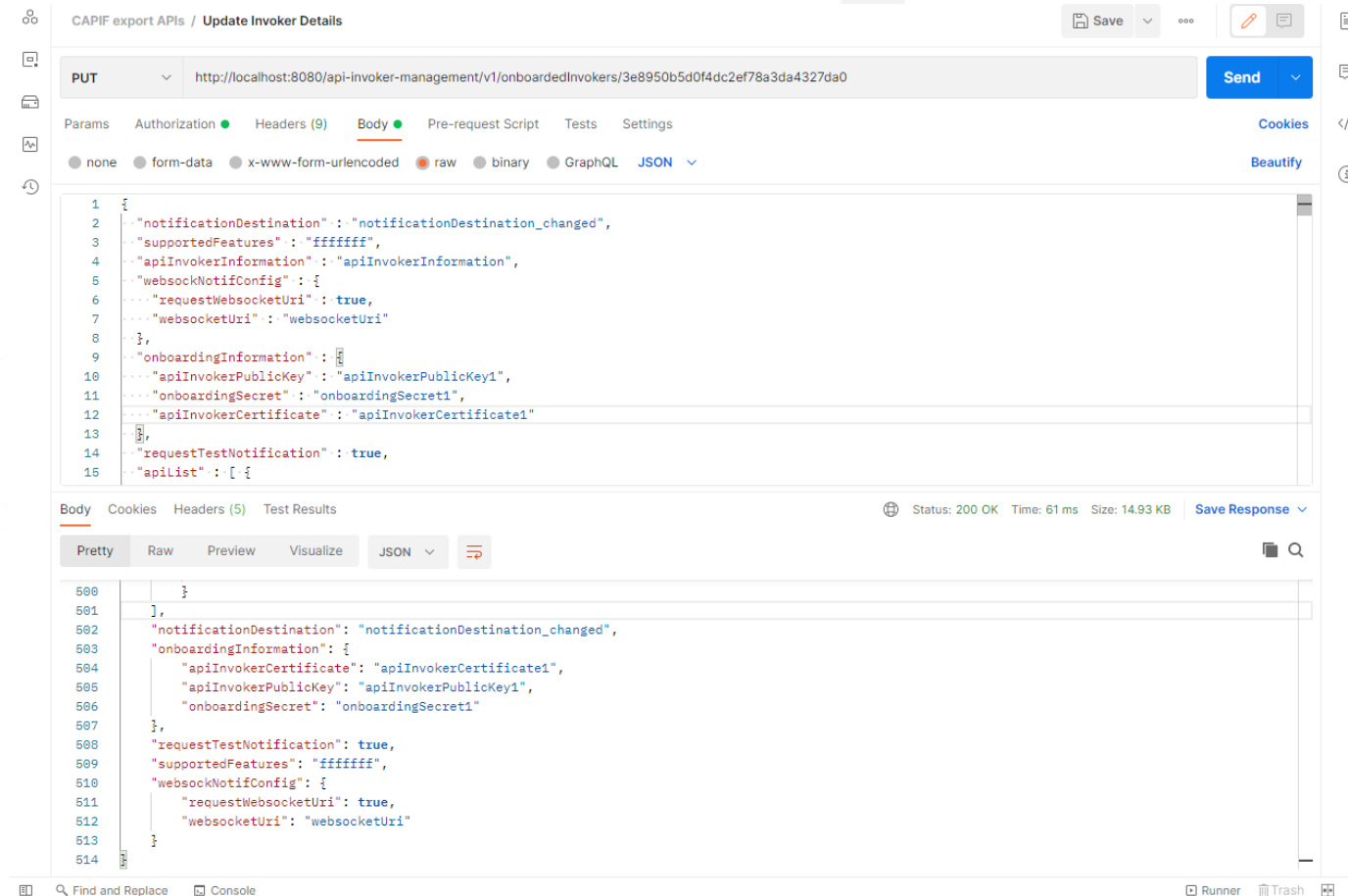
# Onboard Invoker (NetApp)

- POST `http://<CAPIF Host IP>:8080/api-invoker-management/v1/onboardedInvokers`



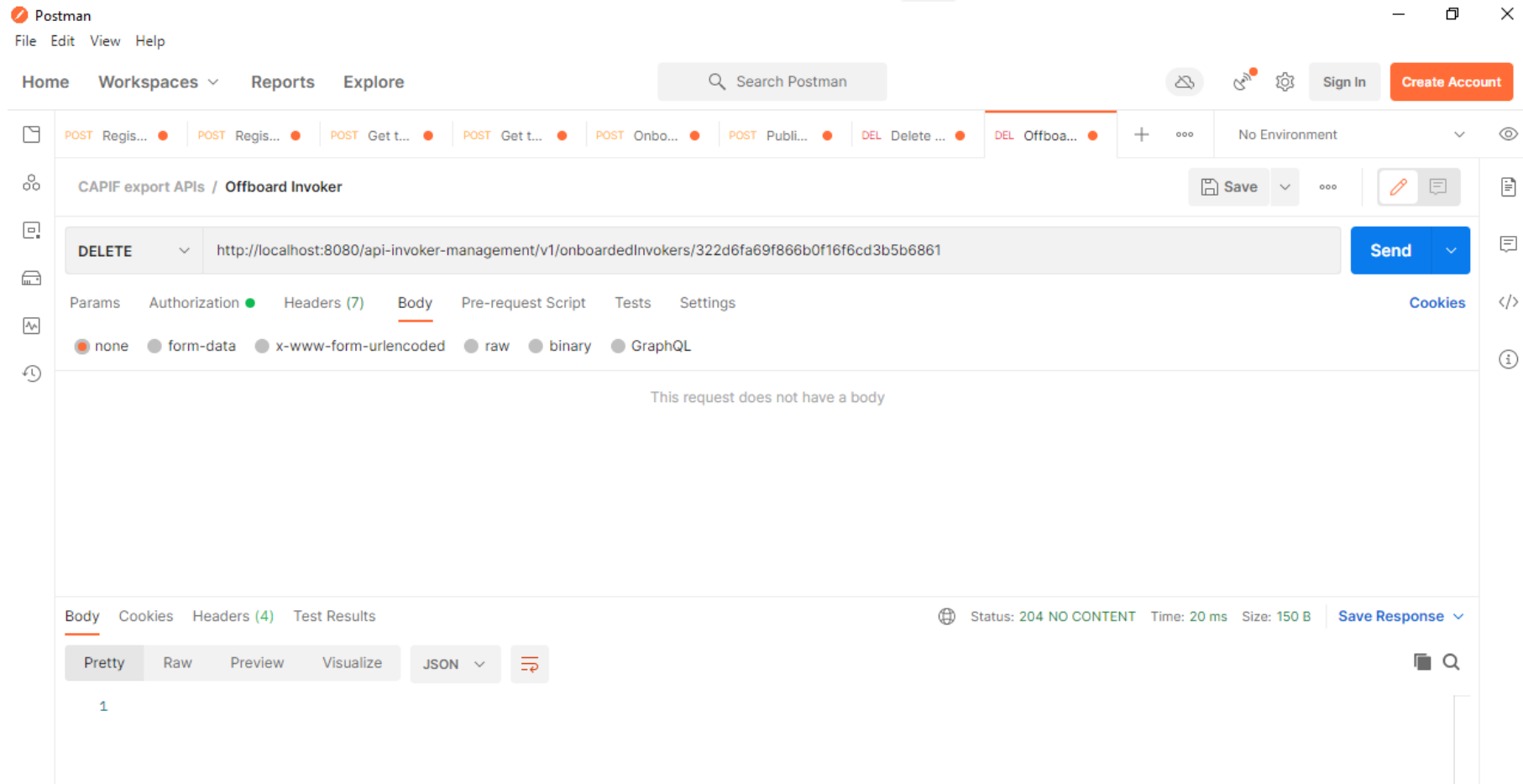
# Update Invoker Details (NetApp)

- PUT `http://<CAPIF Host IP>:8080/api-invoker-management/v1/onboardedInvokers/<API Invoker ID>`



# Offboard Invoker (NetApp)

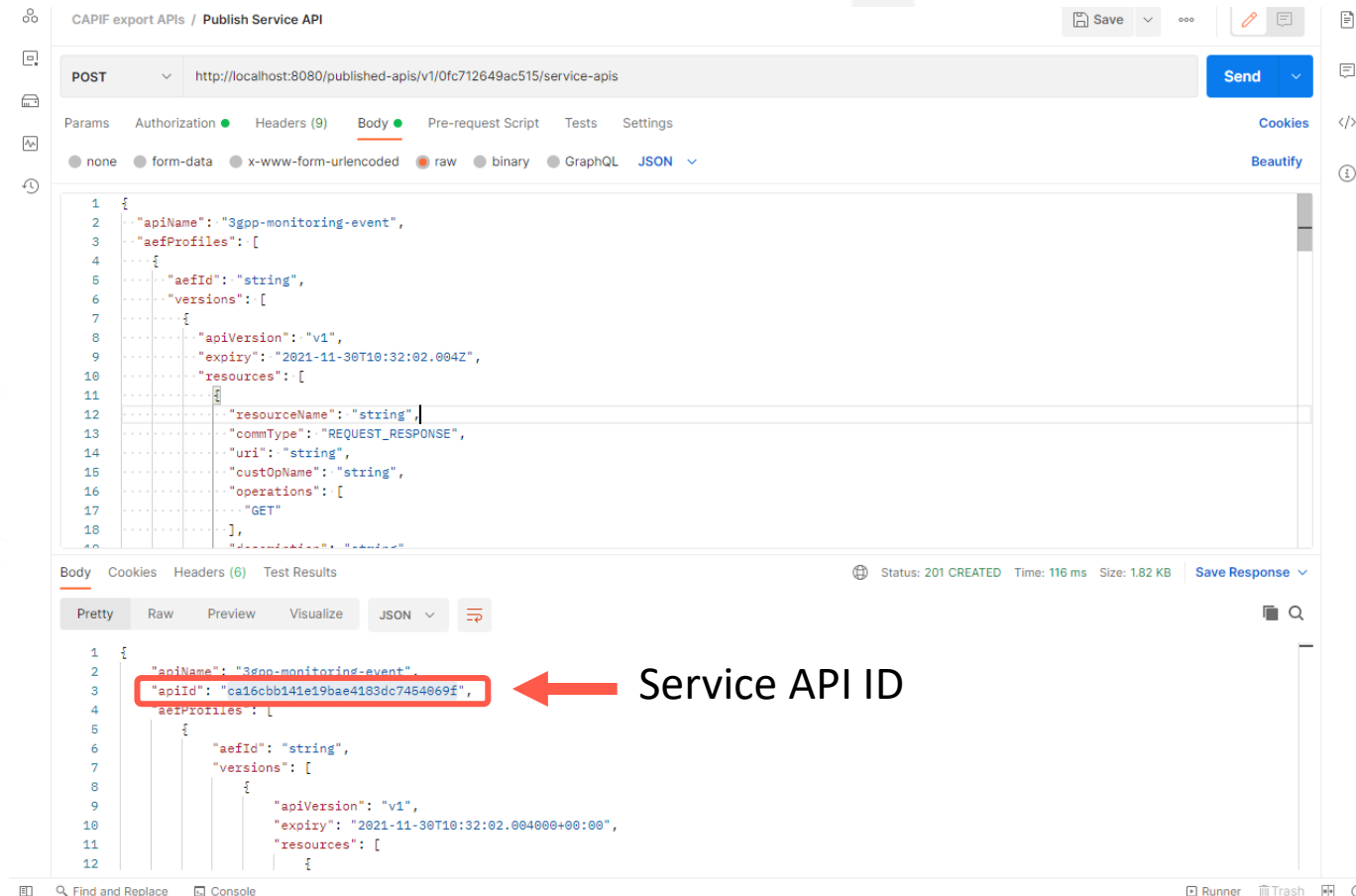
- DELETE `http://<CAPIF Host IP>:8080/api-invoker-management/v1/onboardedInvokers/<API Invoker ID>`





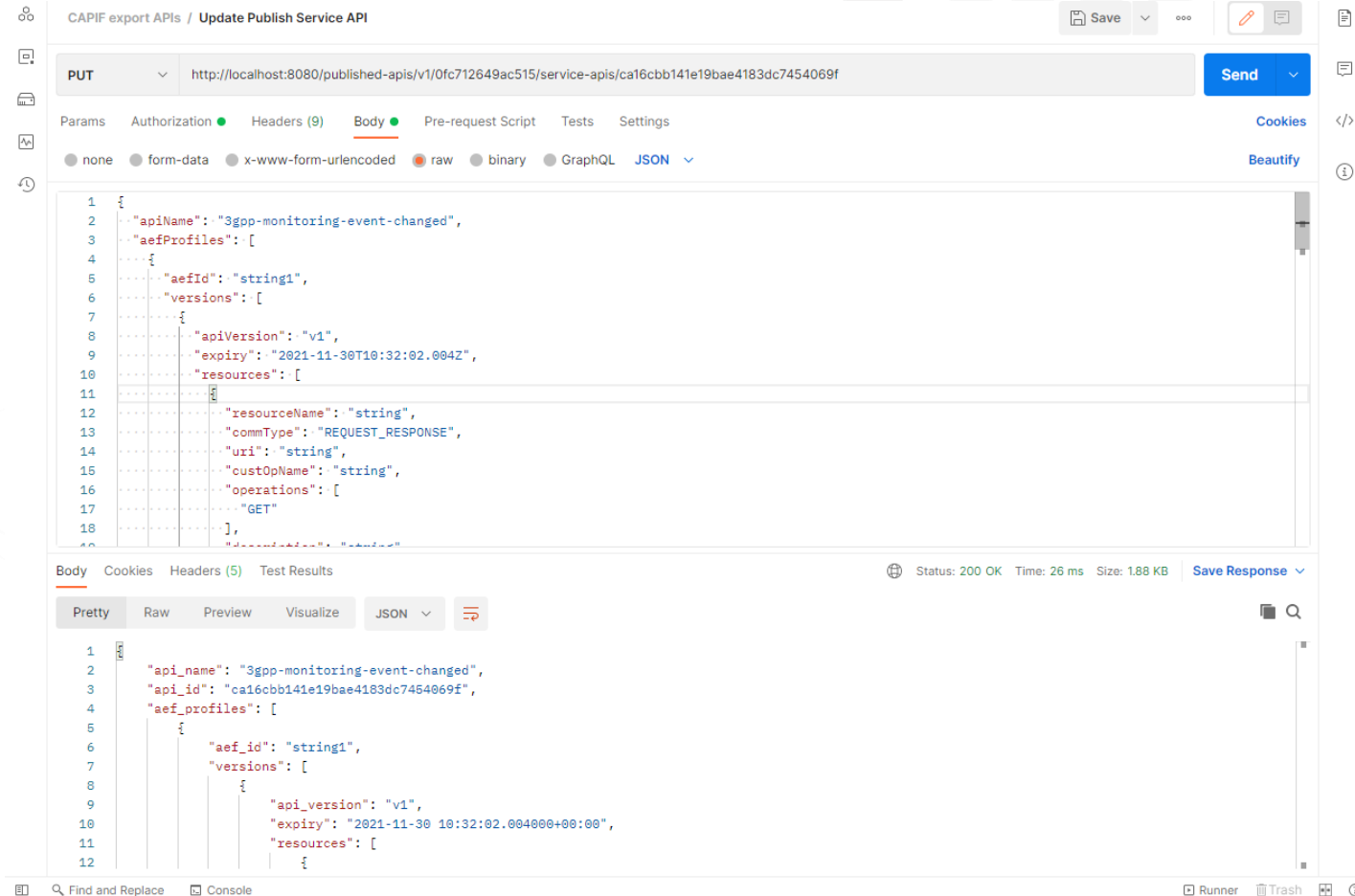
# Publish Service API (APF)

- POST `http://<CAPIF Host IP>:8080/published-apis/v1/<APF ID>/service-apis`



# Update Service API (APF)

- PUT `http://<CAPIF Host IP>:8080/published-apis/v1/<APF ID>/service-apis/<Service API ID>`



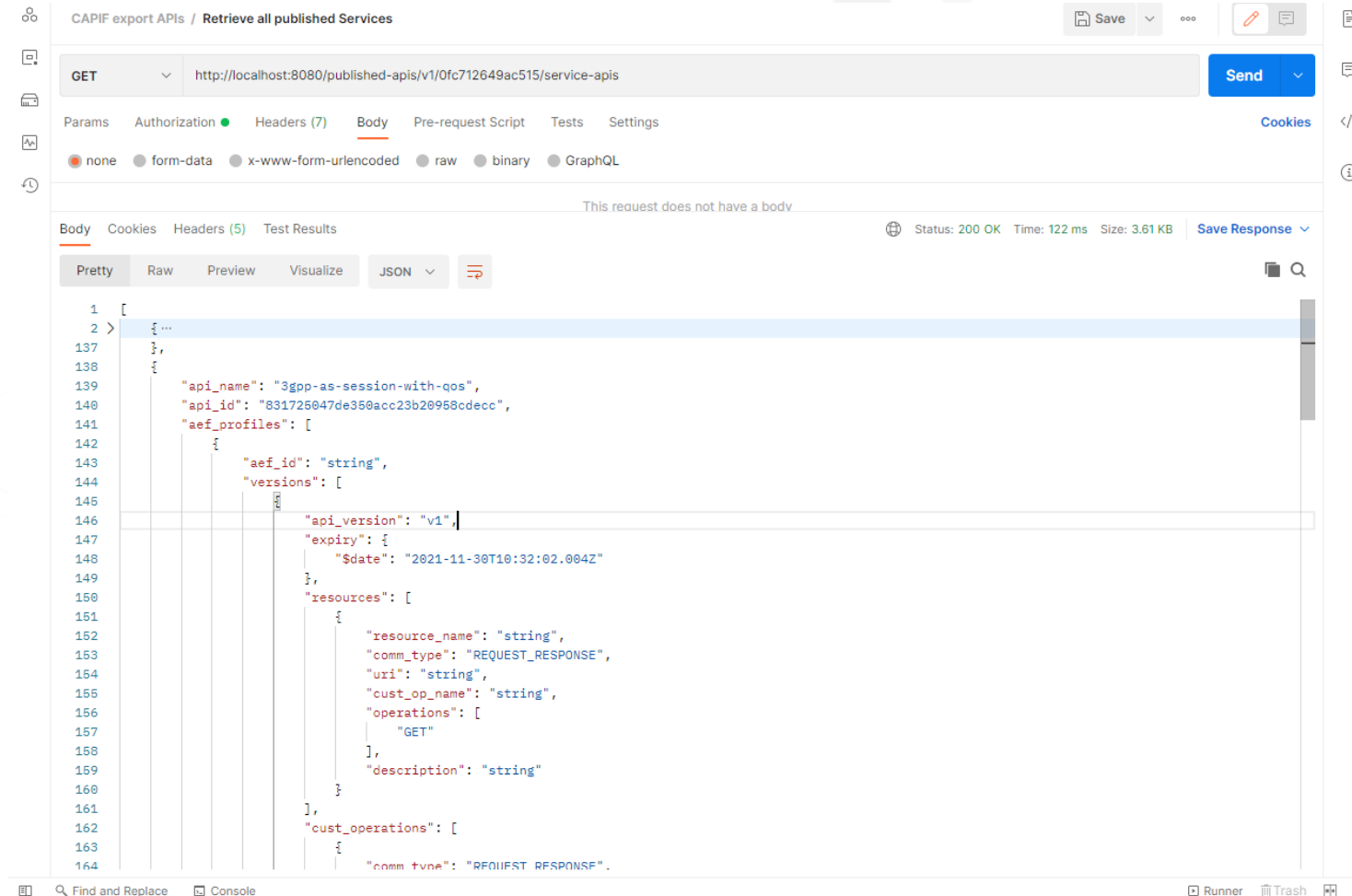
The screenshot displays a REST client interface for a PUT request. The URL bar shows the endpoint: `http://localhost:8080/published-apis/v1/0fc712649ac515/service-apis/ca16cbb141e19bae4183dc7454069f`. The request body is a JSON object defining an API profile. The response status is 200 OK, with a time of 26 ms and a size of 1.88 KB. The response body is a JSON object containing the same data as the request body.

```
1 {
2   "apiName": "3gpp-monitoring-event-changed",
3   "aefProfiles": [
4     {
5       "aefId": "string1",
6       "versions": [
7         {
8           "apiVersion": "v1",
9           "expiry": "2021-11-30T10:32:02.004Z",
10          "resources": [
11            {
12              "resourceName": "string",
13              "commType": "REQUEST_RESPONSE",
14              "uri": "string",
15              "custOpName": "string",
16              "operations": [
17                "GET"
18              ]
19            }
20          ]
21        }
22      ]
23    }
24  ]
25 }
```

```
1 {
2   "api_name": "3gpp-monitoring-event-changed",
3   "api_id": "ca16cbb141e19bae4183dc7454069f",
4   "aef_profiles": [
5     {
6       "aef_id": "string1",
7       "versions": [
8         {
9           "api_version": "v1",
10          "expiry": "2021-11-30 10:32:02.004000+00:00",
11          "resources": [
12            {
13              "resource_name": "string",
14              "comm_type": "REQUEST_RESPONSE",
15              "uri": "string",
16              "cust_op_name": "string",
17              "operations": [
18                "GET"
19              ]
20            }
21          ]
22        }
23      ]
24    }
25  ]
26 }
```

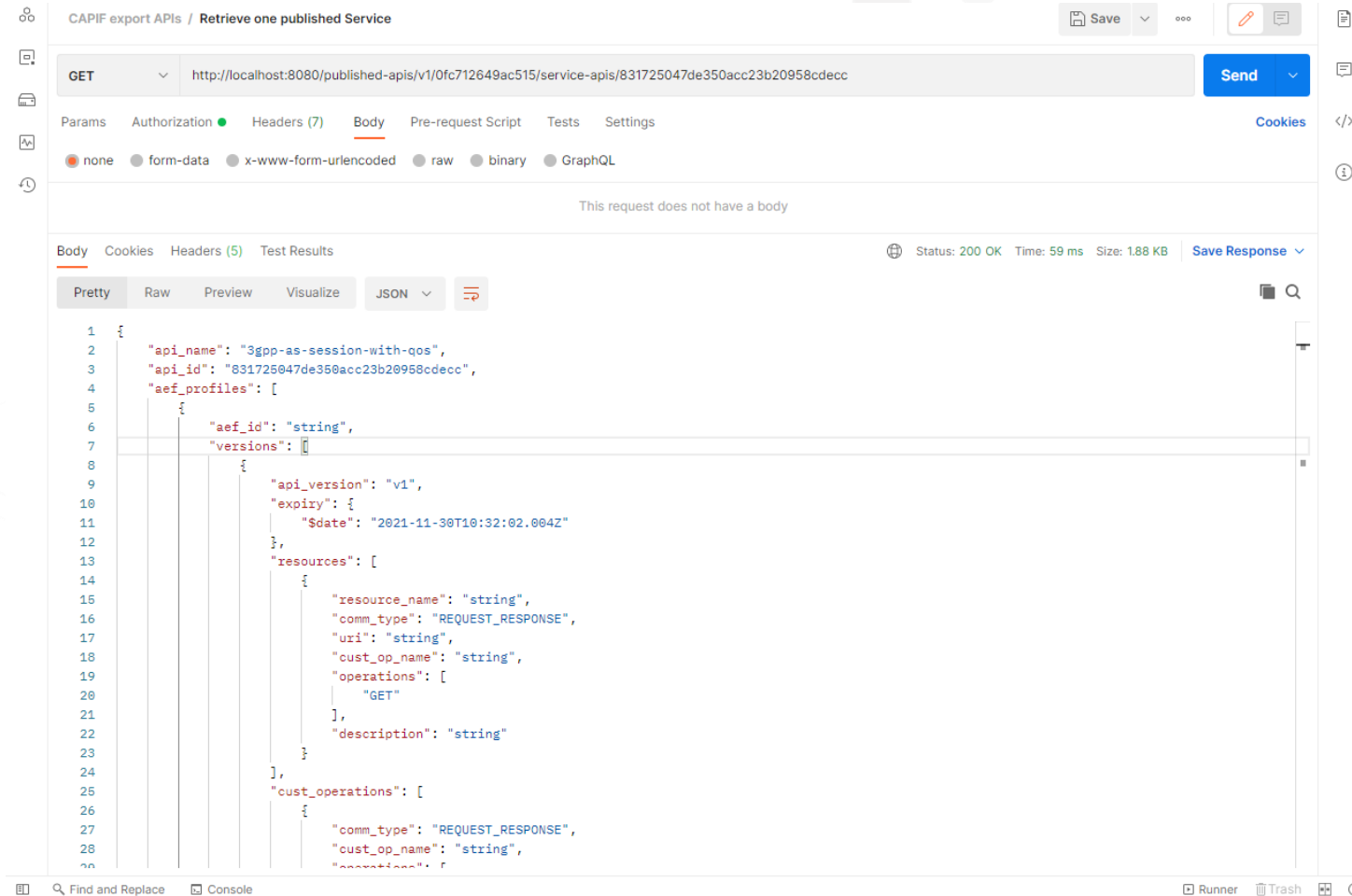
# Retrieve all published services (APF)

- GET `http://<CAPIF Host IP>:8080/published-apis/v1/<APF ID>/service-apis`



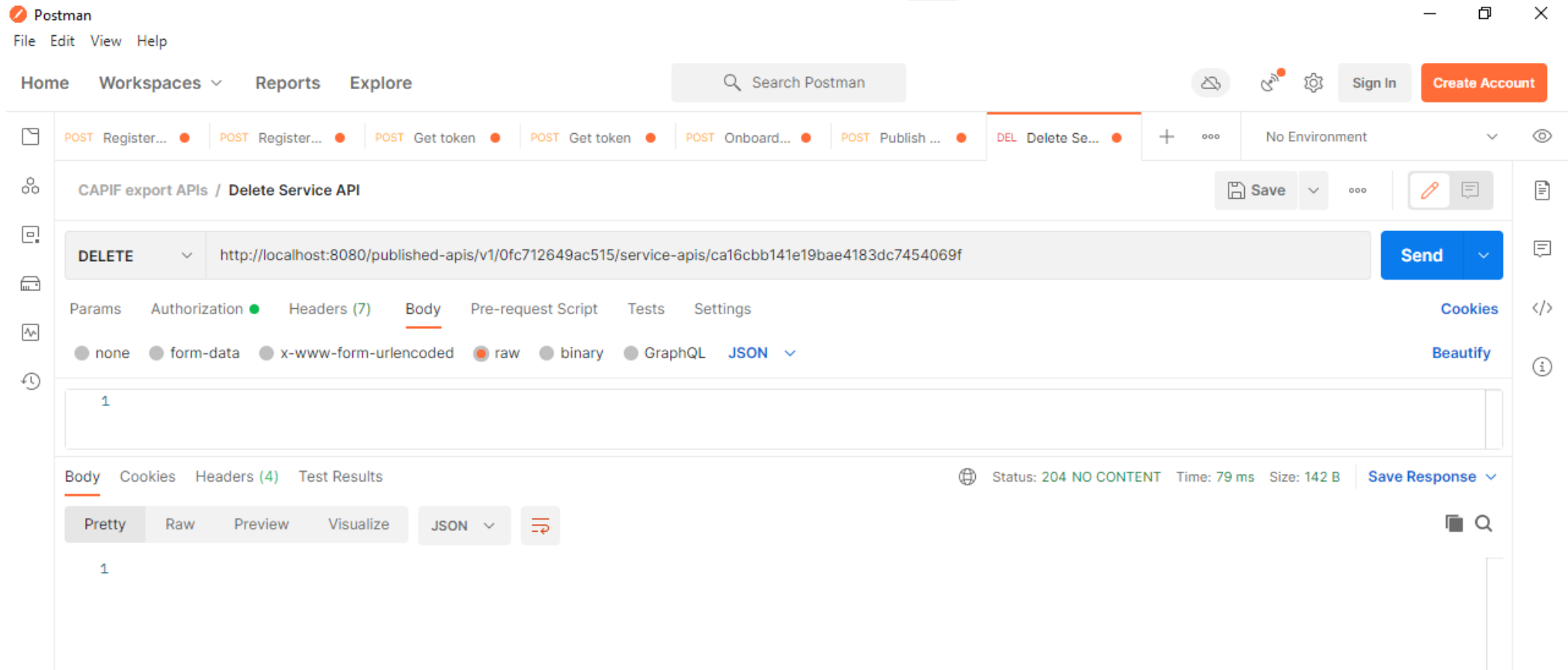
# Retrieve one published services (APF)

- GET `http://<CAPIF Host IP>:8080/published-apis/v1/<APF ID>/service-apis/<Service API ID>`



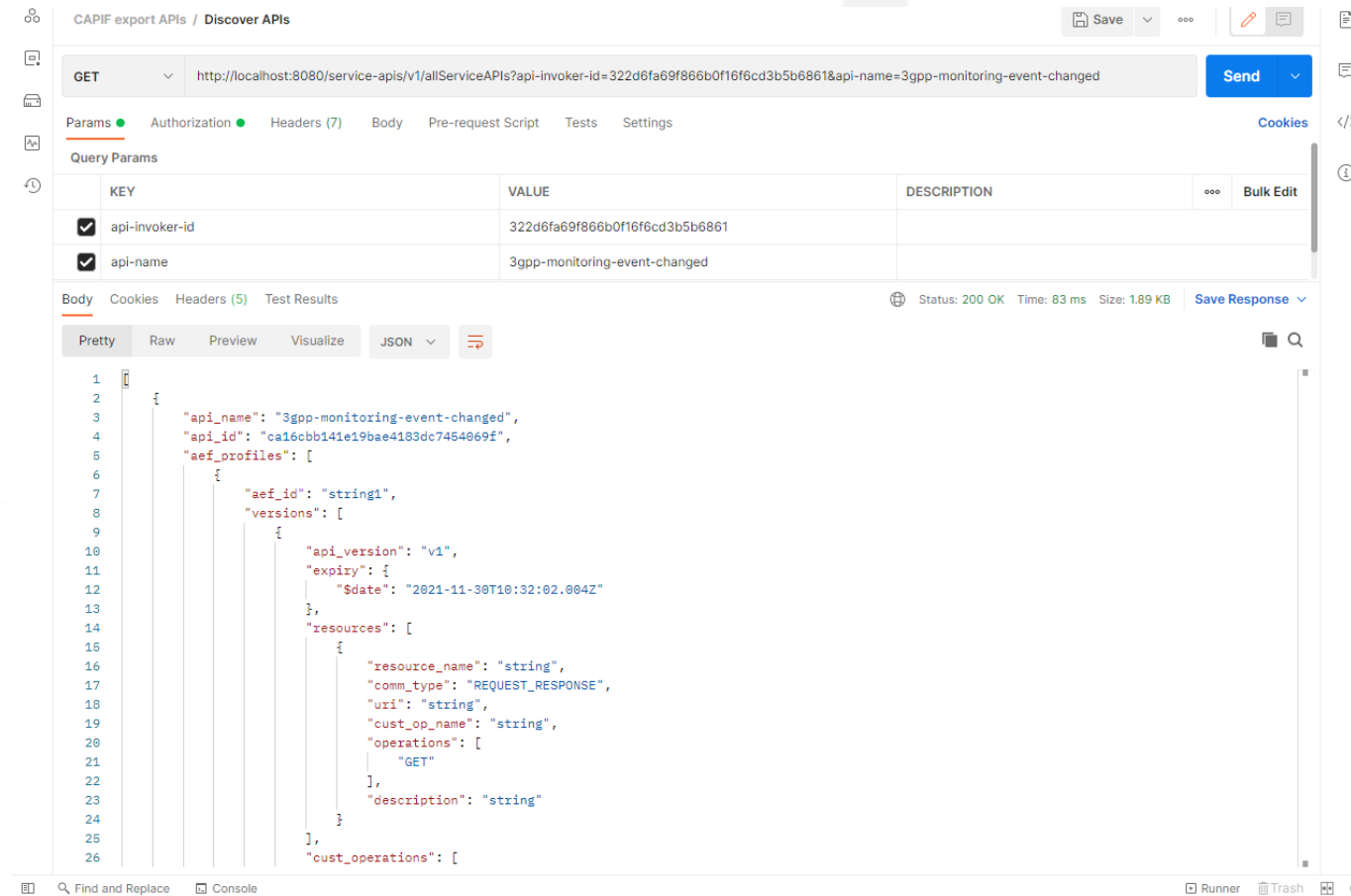
# Delete Service API (APF)

- DELETE `http://<CAPIF Host IP>:8080/published-apis/v1/<APF ID>/service-apis/<Service API ID>`



# Discover APIs (NetApp)

- GET `http://<CAPIF Host IP>:8080/service-apis/v1/allServiceAPIs?api-invoker-id=<API Invoker ID>&api-name=<API Name>`



CAPIF export APIs / Discover APIs

GET `http://localhost:8080/service-apis/v1/allServiceAPIs?api-invoker-id=322d6fa69f866b0f16f6cd3b5b6861&api-name=3gpp-monitoring-event-changed` **Send**

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

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> api-invoker-id	322d6fa69f866b0f16f6cd3b5b6861	
<input checked="" type="checkbox"/> api-name	3gpp-monitoring-event-changed	

Body Cookies Headers (5) Test Results Status: 200 OK Time: 83 ms Size: 1.89 KB **Save Response**

Pretty Raw Preview Visualize JSON

```
1 {
2   "api_name": "3gpp-monitoring-event-changed",
3   "api_id": "ca16cbb141e19bae4183dc7454069f",
4   "aef_profiles": [
5     {
6       "aef_id": "string1",
7       "versions": [
8         {
9           "api_version": "v1",
10          "expiry": {
11            "$date": "2021-11-30T10:32:02.004Z"
12          },
13          "resources": [
14            {
15              "resource_name": "string",
16              "comm_type": "REQUEST_RESPONSE",
17              "uri": "string",
18              "cust_op_name": "string",
19              "operations": [
20                "GET"
21              ],
22              "description": "string"
23            }
24          ],
25          "cust_operations": [
```



Thank you