



Instructions to use CAPIF APIs with Postman

EVOLVED-5G

- Install POSTMAN

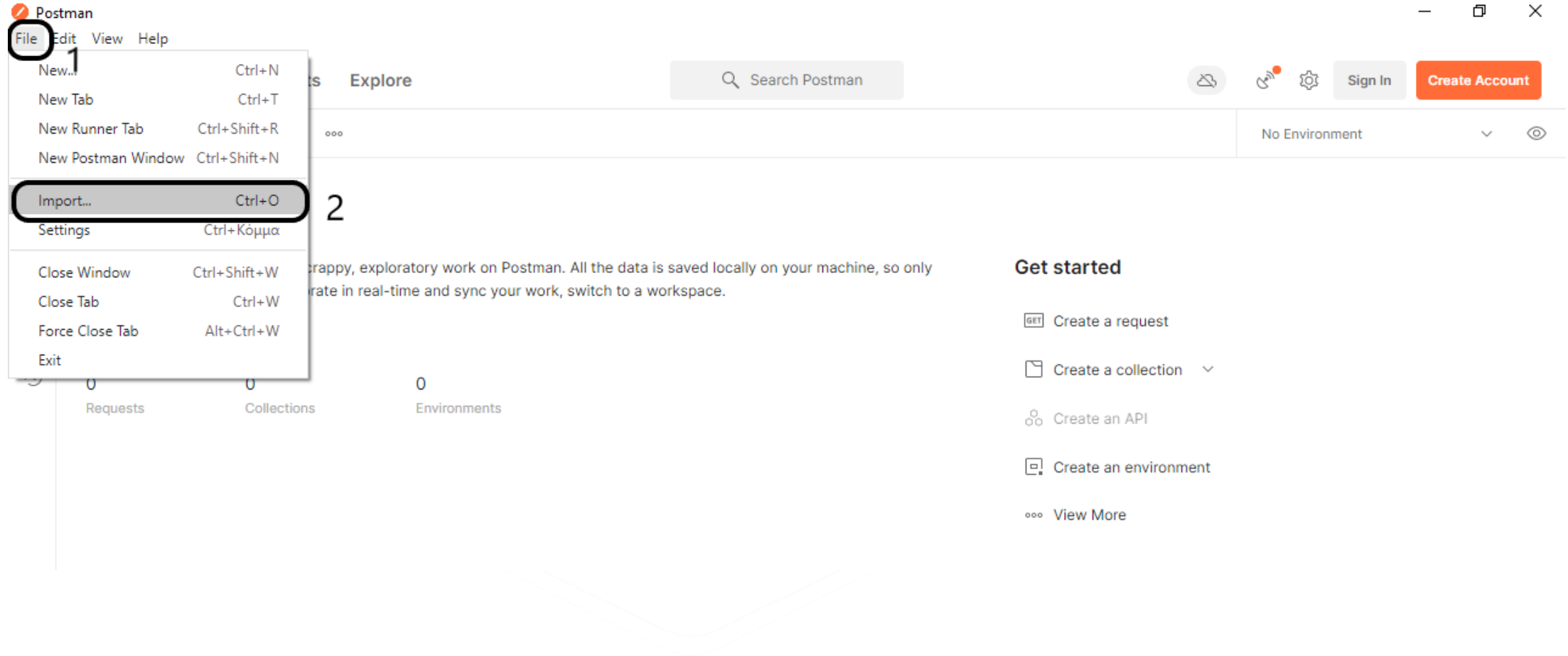
- Postman is an API platform for building and using APIs.
- Download from: <https://www.postman.com/downloads/>
(for Linux distros, POSTMAN is available also in the corresponding Software Center Application)



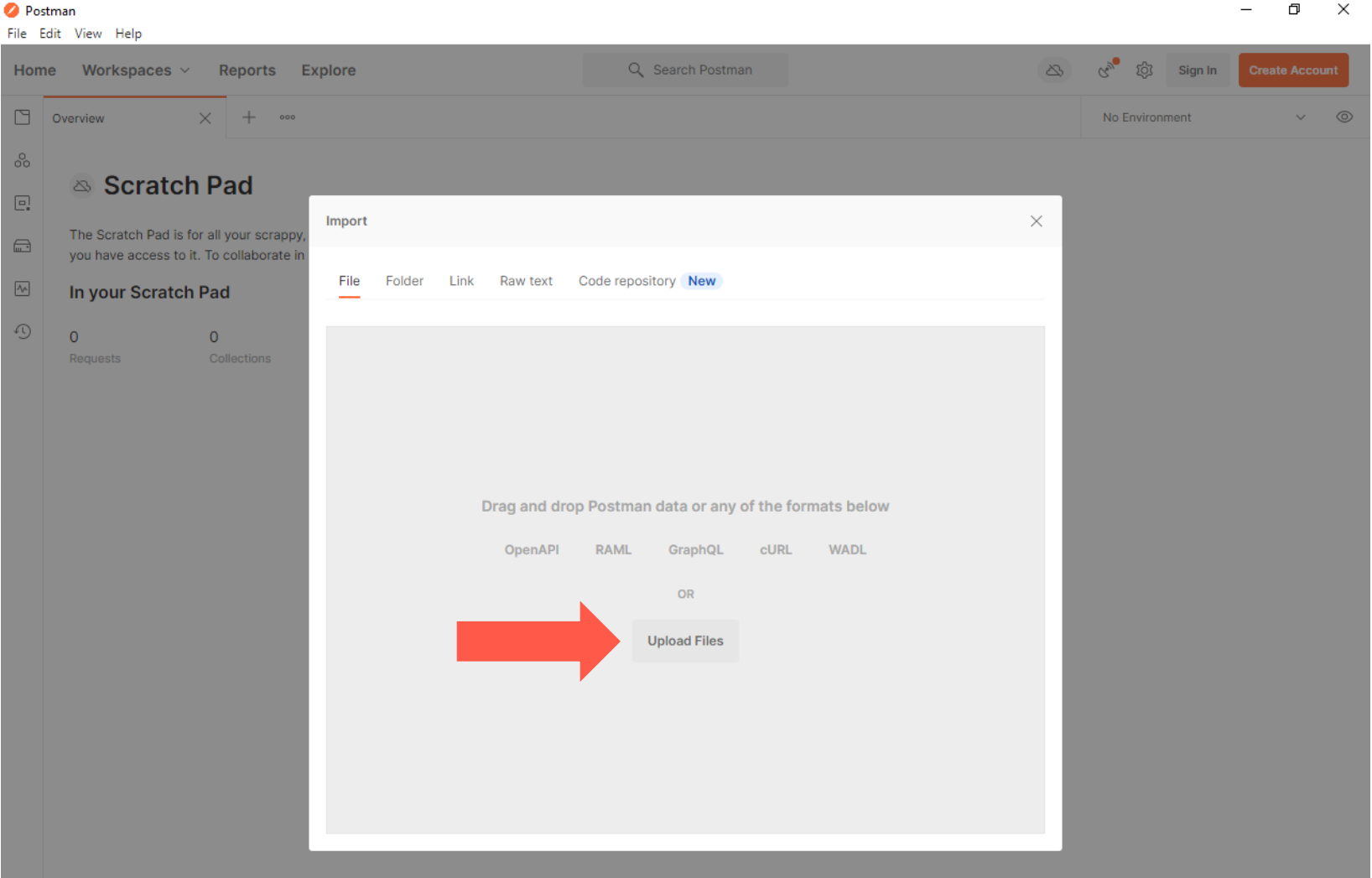
POSTMAN

- Clone CAPIF Tool from Github and deploy

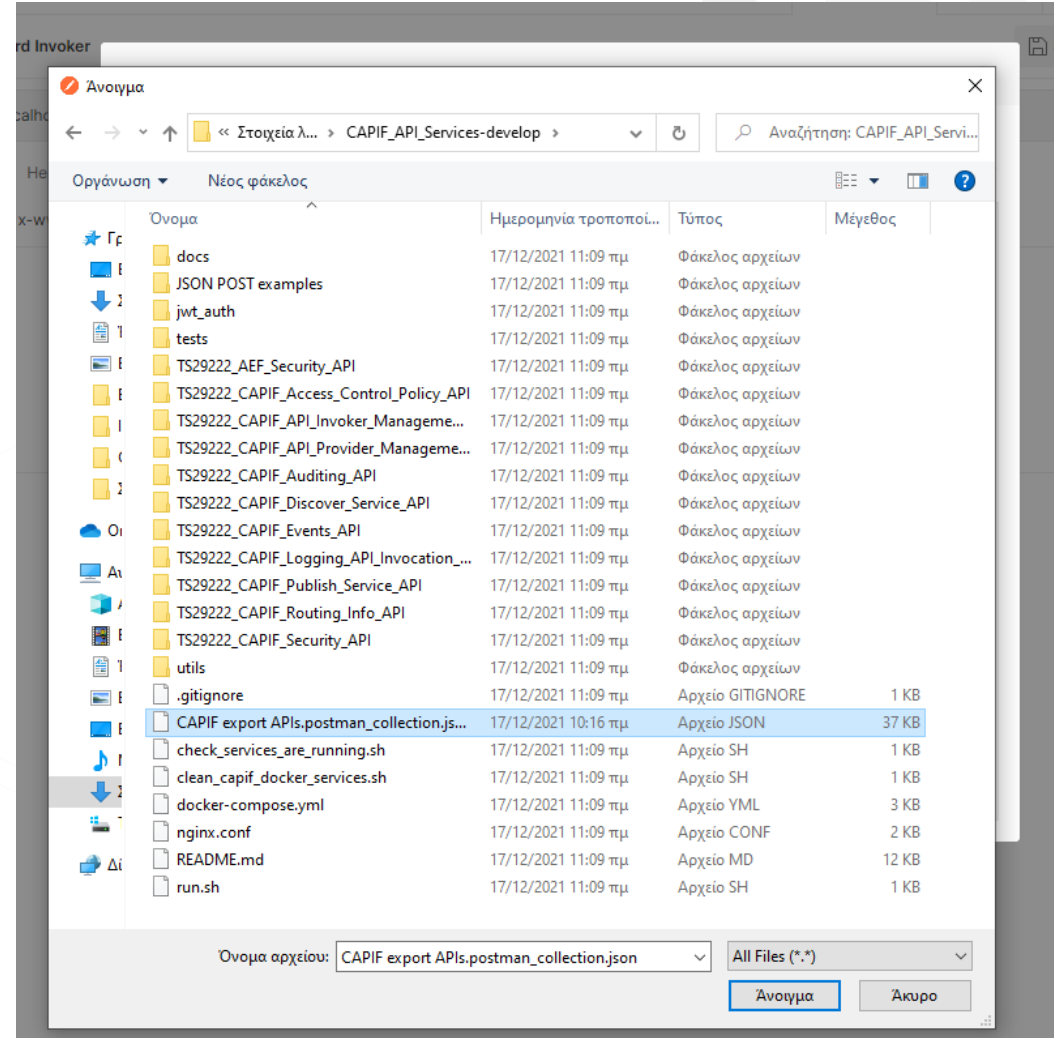
Open POSTMAN



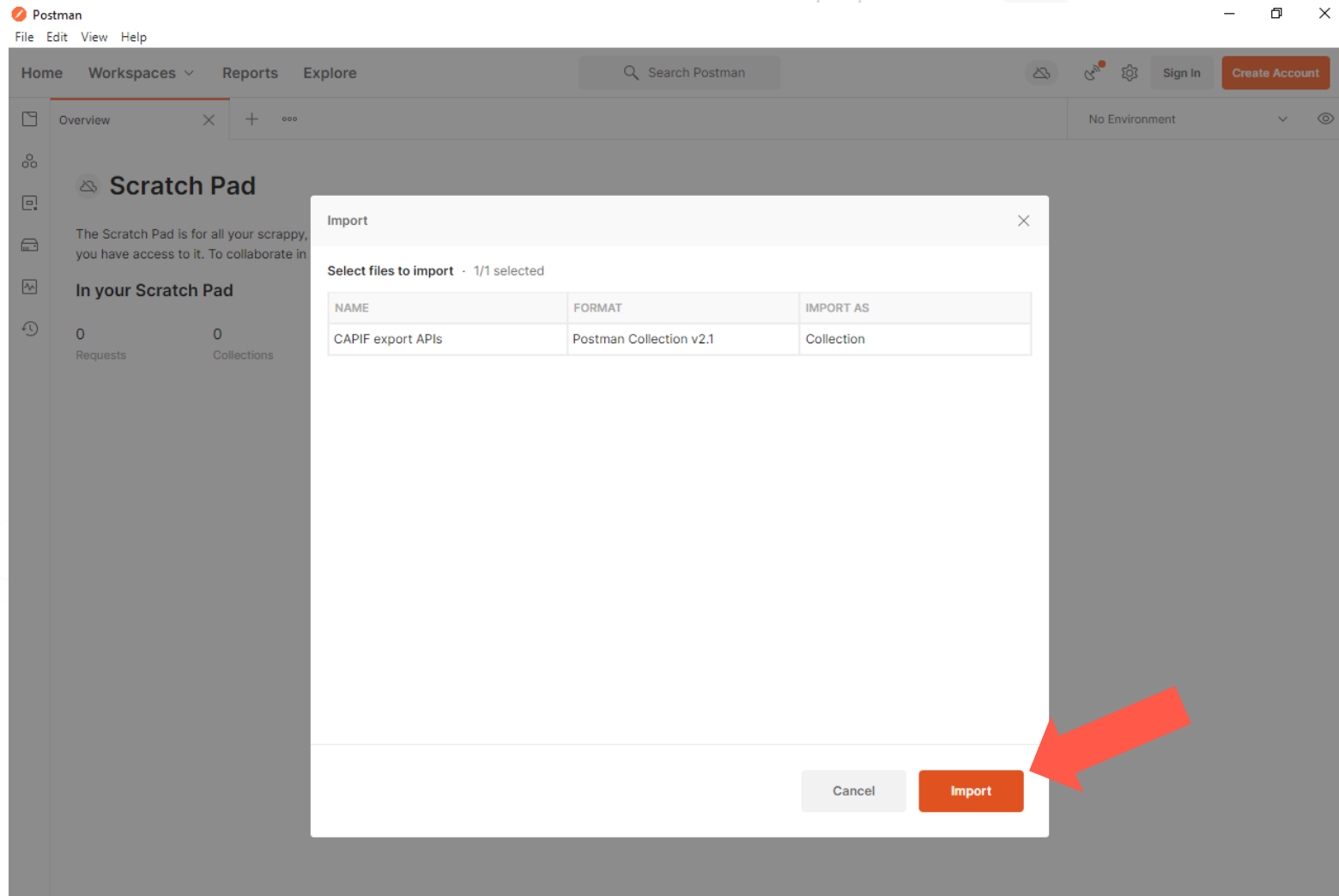
Import Menu



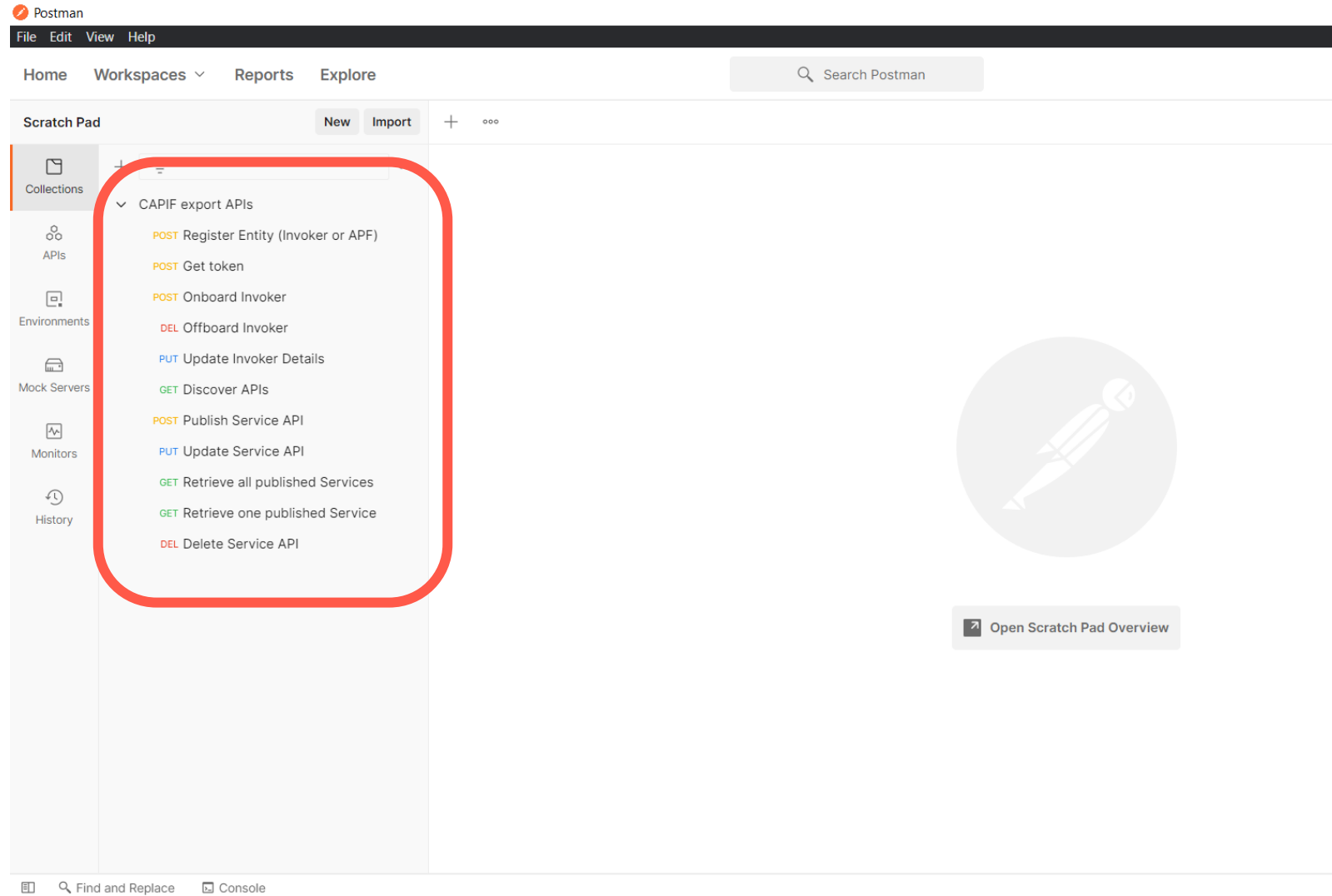
Select “CAPIF export APIs.postman_collection.json”



Final Import



List of CAPIF APIs imported



Process of Executing an API call

1. Double-click an API

2. Select method

3. Edit URL

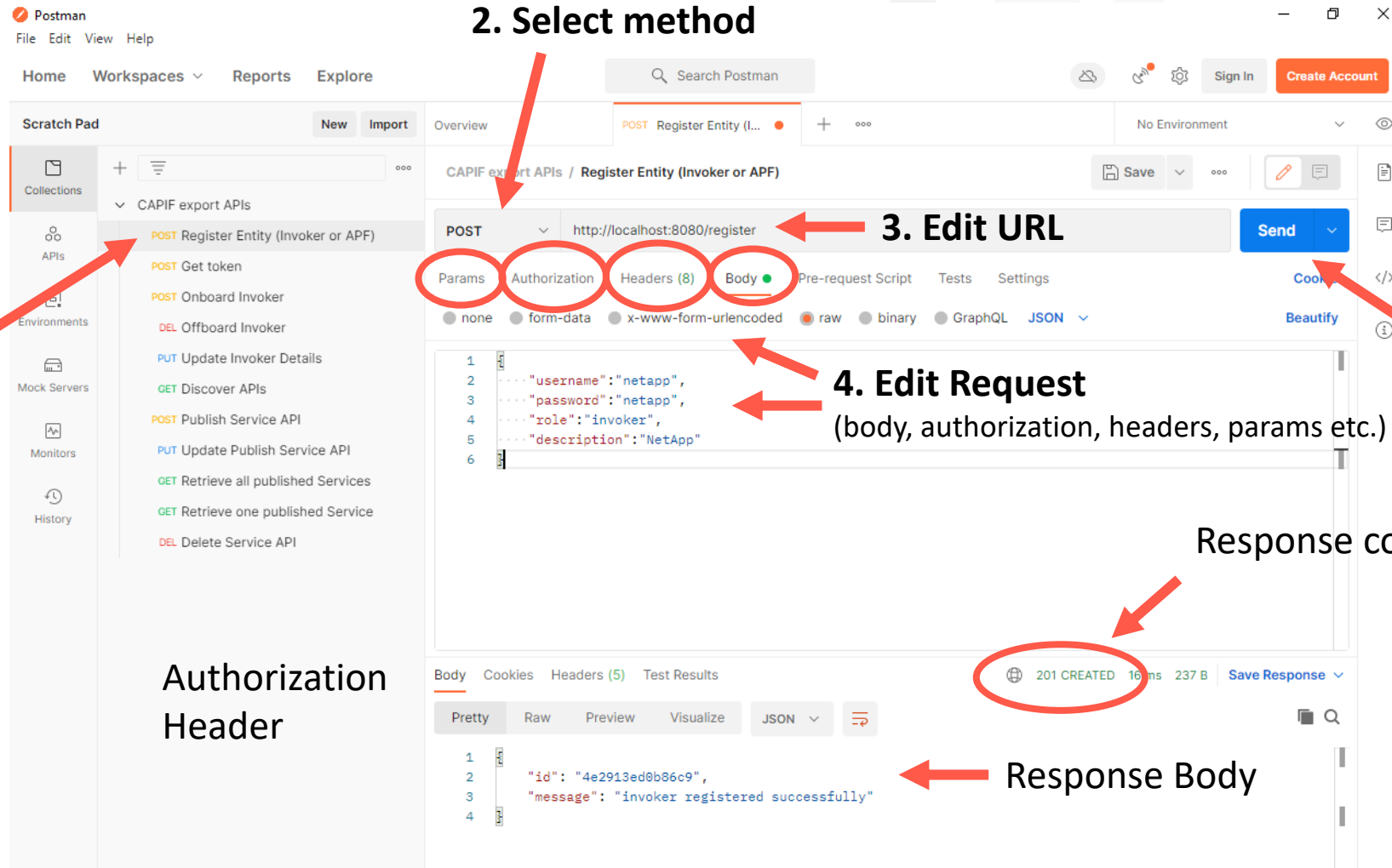
4. Edit Request
(body, authorization, headers, params etc.)

4. Press to execute

Authorization Header

Response code

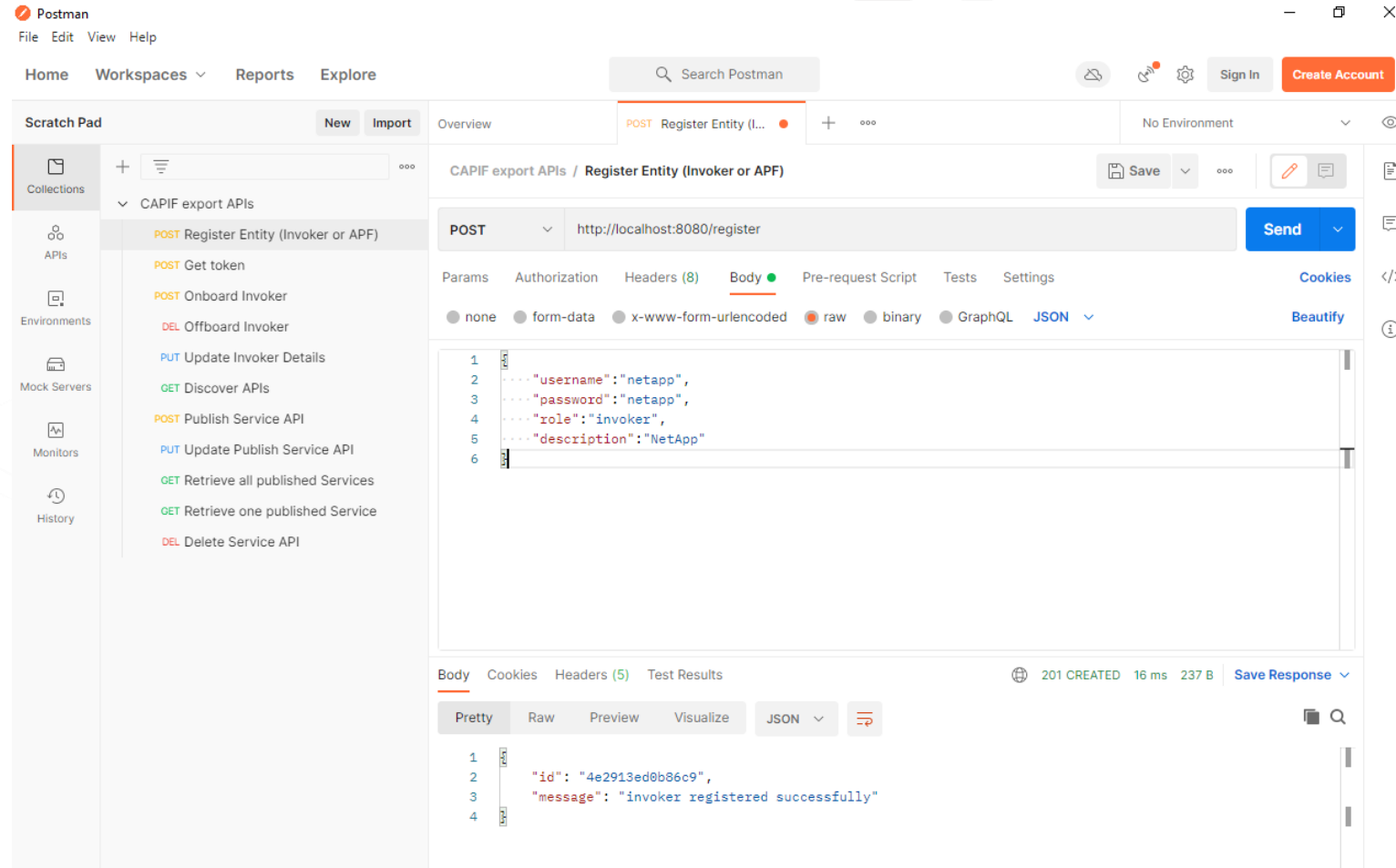
Response Body



Postman interface showing the process of executing an API call. The interface includes a sidebar with 'Collections' and 'Environments', a main workspace with a 'POST' method selected, and a 'Send' button. The URL is 'http://localhost:8080/register'. The request body is a JSON object: {\"username\": \"netapp\", \"password\": \"netapp\", \"role\": \"invoker\", \"description\": \"NetApp\"}. The response is shown at the bottom: 201 CREATED 16ms 237 B. The response body is: {\"id\": \"4e2913ed0b86c9\", \"message\": \"invoker registered successfully\"}.

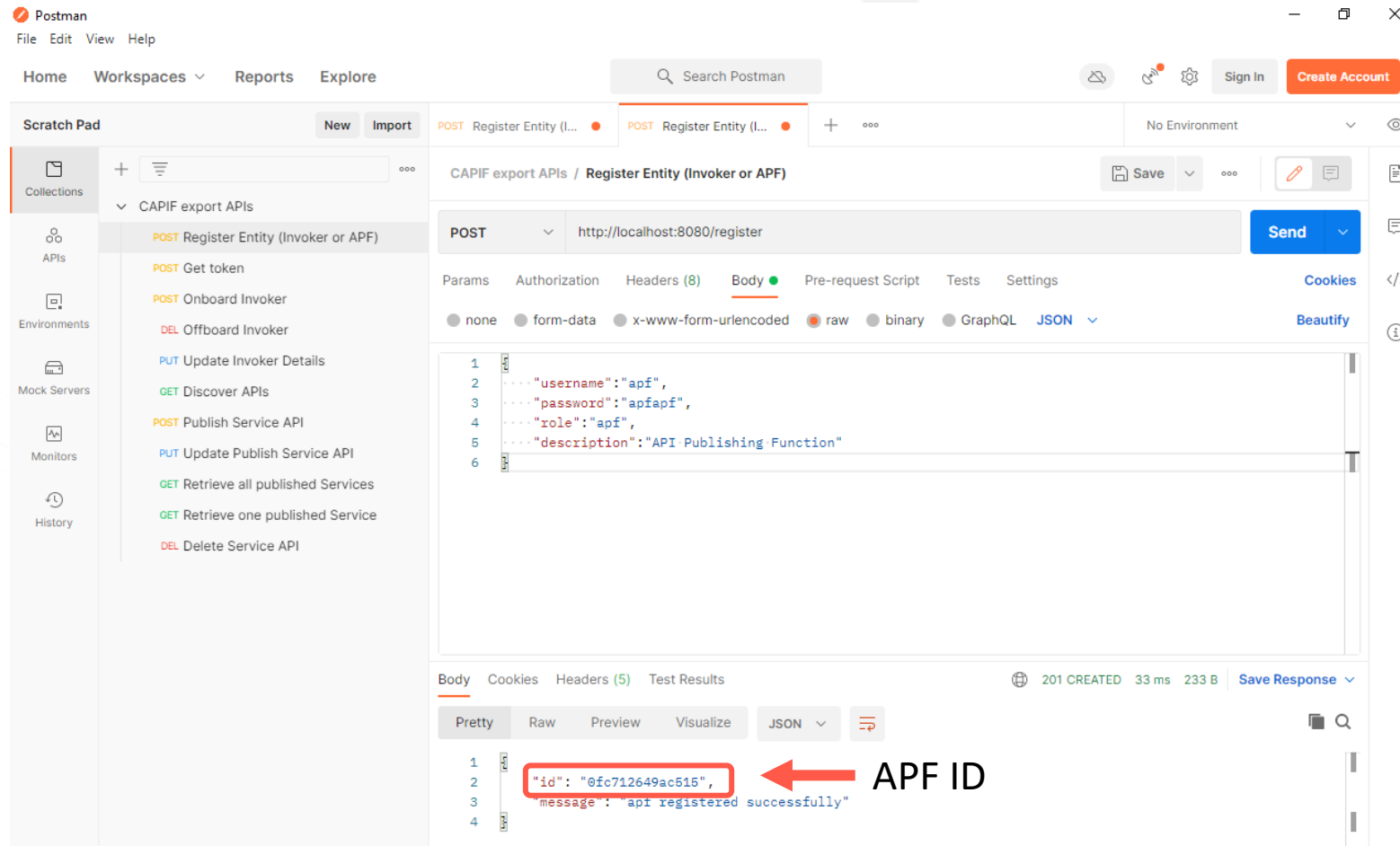
Register Entity (NetApp)

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



Register Entity (APF)

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



- broken

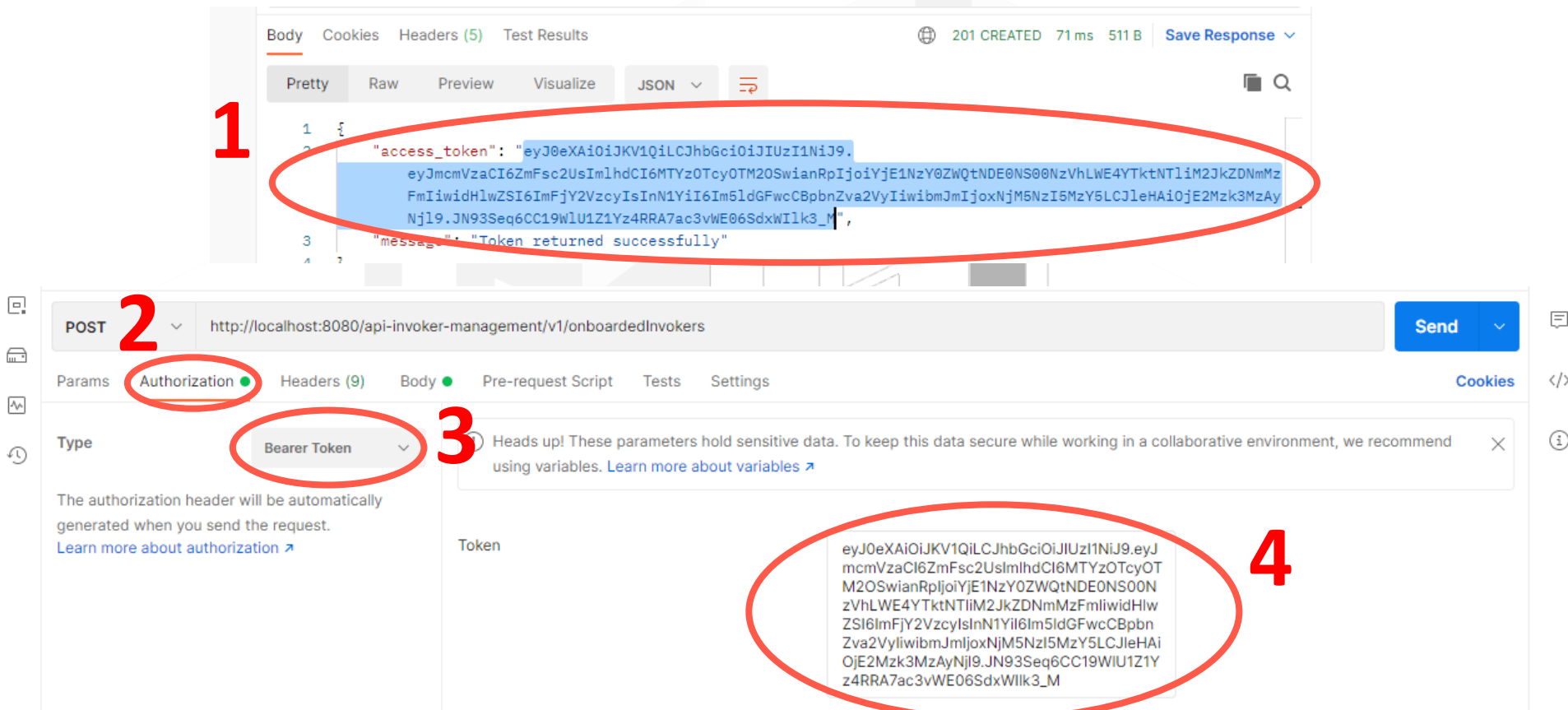


- token



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

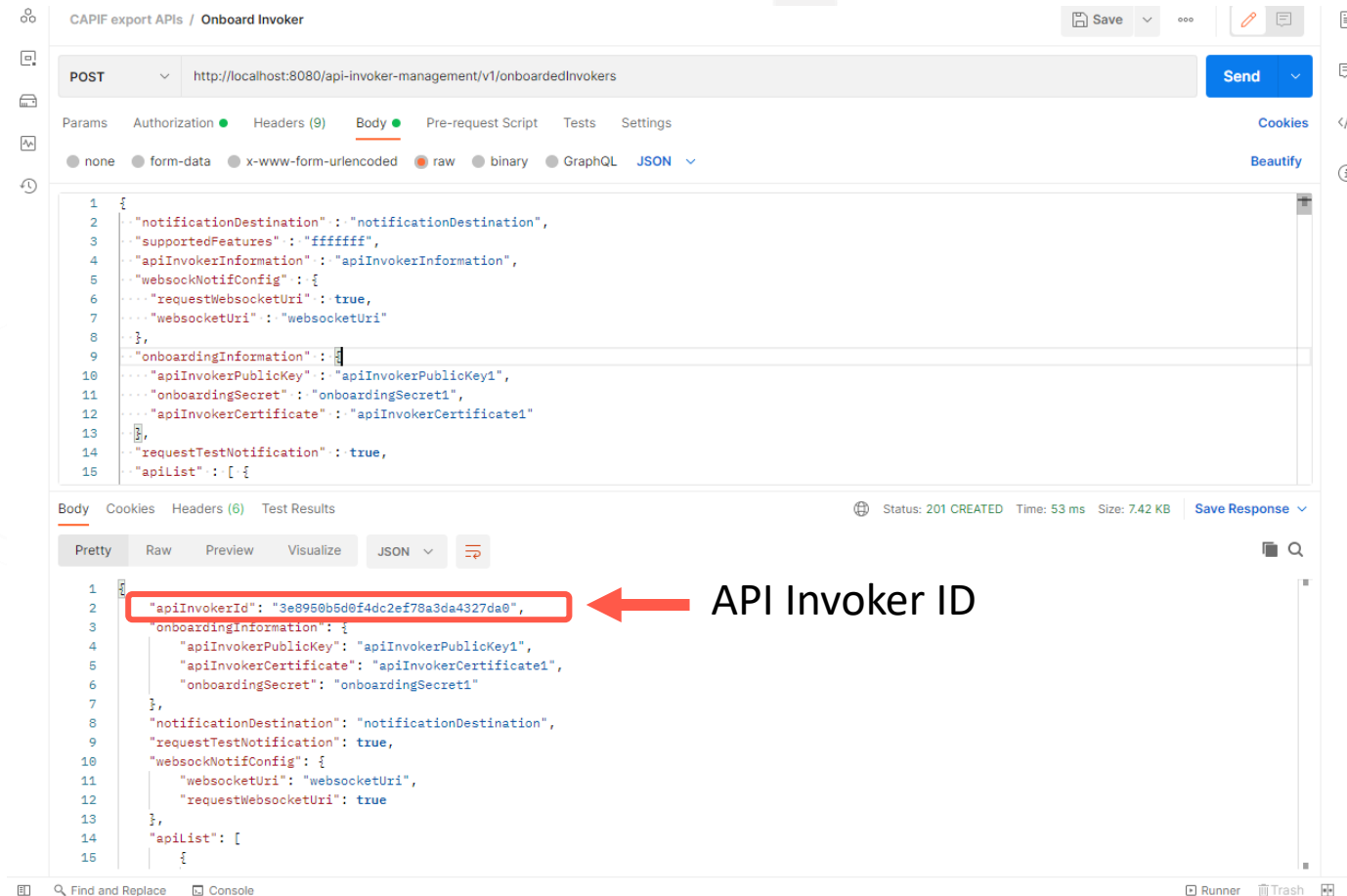


The screenshot illustrates the steps to use a JSON Web Token (JWT) for API authentication in Postman:

- 1**: The response body of a successful API call is shown in JSON format. The `access_token` field contains the JWT token, which is highlighted with a red oval.
- 2**: The API request is configured with the `Authorization` header tab selected.
- 3**: The `Bearer Token` type is selected for the authorization header.
- 4**: The JWT token is pasted into the `Token` field of the authorization header.

Onboard Invoker (NetApp)

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



CAPIF export APIs / Onboard Invoker

POST `http://localhost:8080/api-invoker-management/v1/onboardedInvokers` **Send**

Params Authorization Headers (9) Body Pre-request Script Tests Settings

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

```
1 {
2   "notificationDestination": "notificationDestination",
3   "supportedFeatures": "ffffff",
4   "apiInvokerInformation": "apiInvokerInformation",
5   "websocketNotifConfig": {
6     "requestWebSocketUri": true,
7     "websocketUri": "websocketUri"
8   },
9   "onboardingInformation": {
10    "apiInvokerPublicKey": "apiInvokerPublicKey1",
11    "onboardingSecret": "onboardingSecret1",
12    "apiInvokerCertificate": "apiInvokerCertificate1"
13  },
14   "requestTestNotification": true,
15   "apilist": [ {
```

Body Cookies Headers (6) Test Results Status: 201 CREATED Time: 53 ms Size: 7.42 KB Save Response

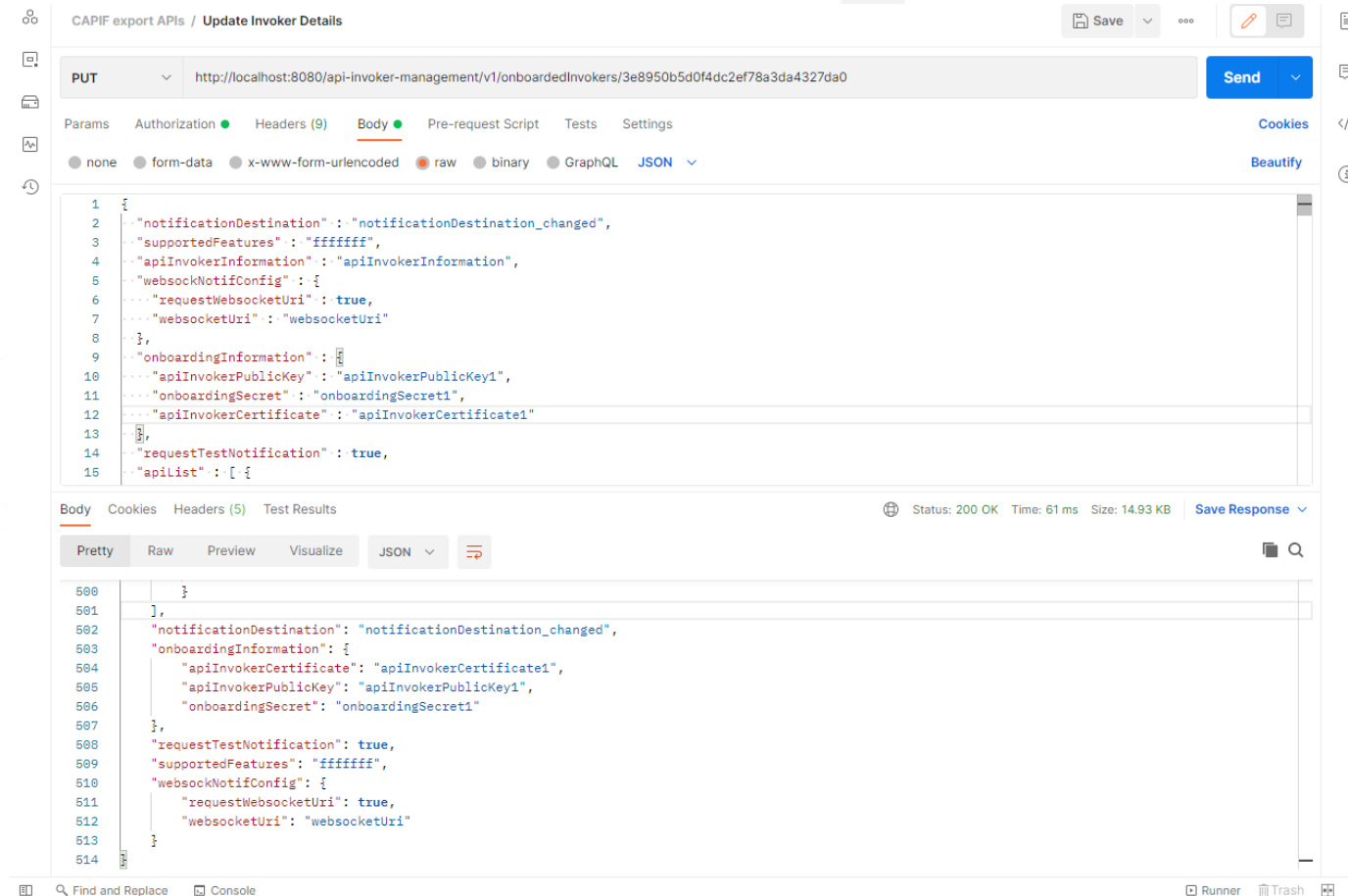
Pretty Raw Preview Visualize JSON

```
1 {
2   "apiInvokerId": "3e8950b5d8f4dc2ef78a3da4327da0",
3   "onboardingInformation": {
4     "apiInvokerPublicKey": "apiInvokerPublicKey1",
5     "apiInvokerCertificate": "apiInvokerCertificate1",
6     "onboardingSecret": "onboardingSecret1"
7   },
8   "notificationDestination": "notificationDestination",
9   "requestTestNotification": true,
10  "websocketNotifConfig": {
11    "websocketUri": "websocketUri",
12    "requestWebSocketUri": true
13  },
14  "apilist": [
15    {
```

API Invoker ID

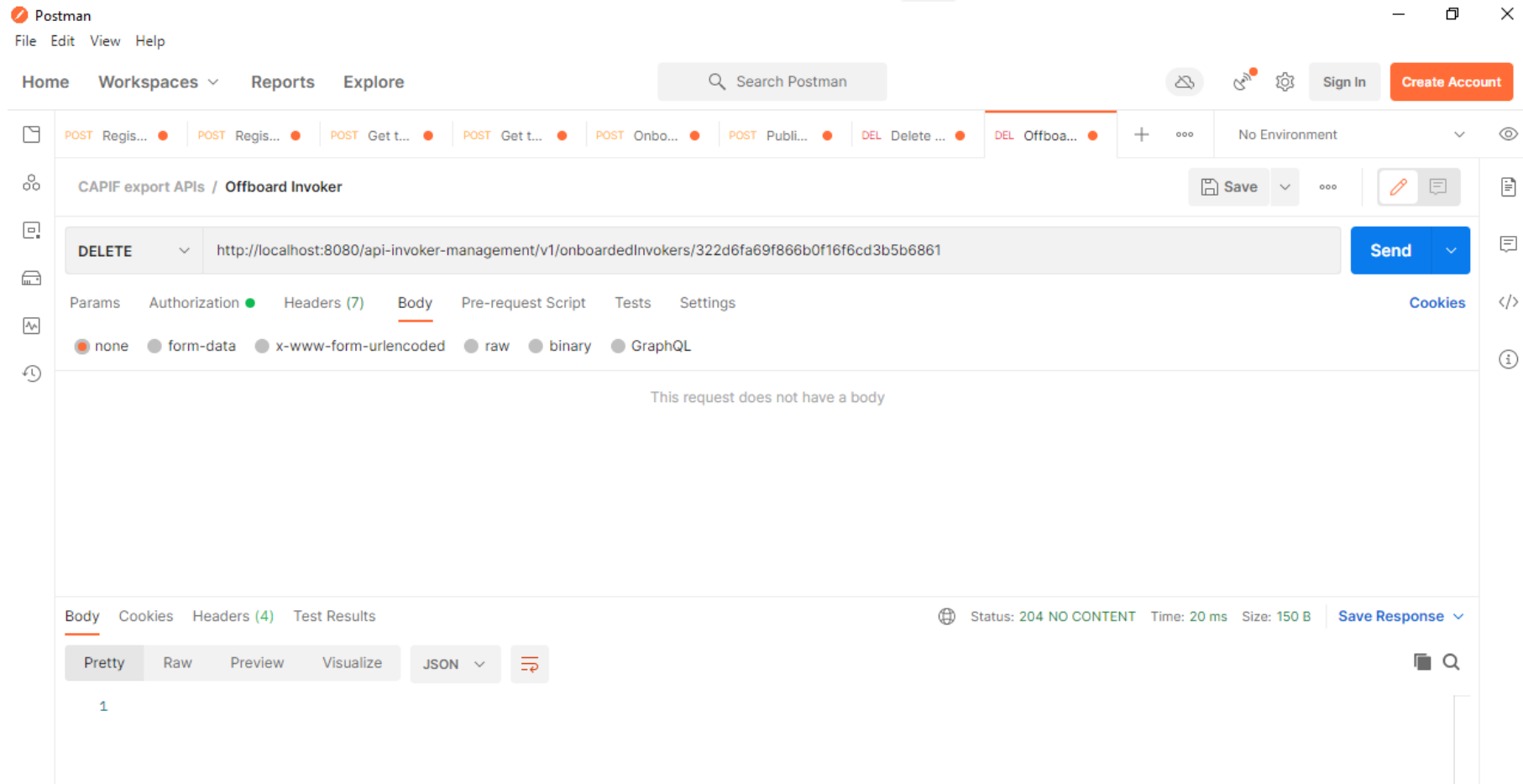
Update Invoker Details (NetApp)

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



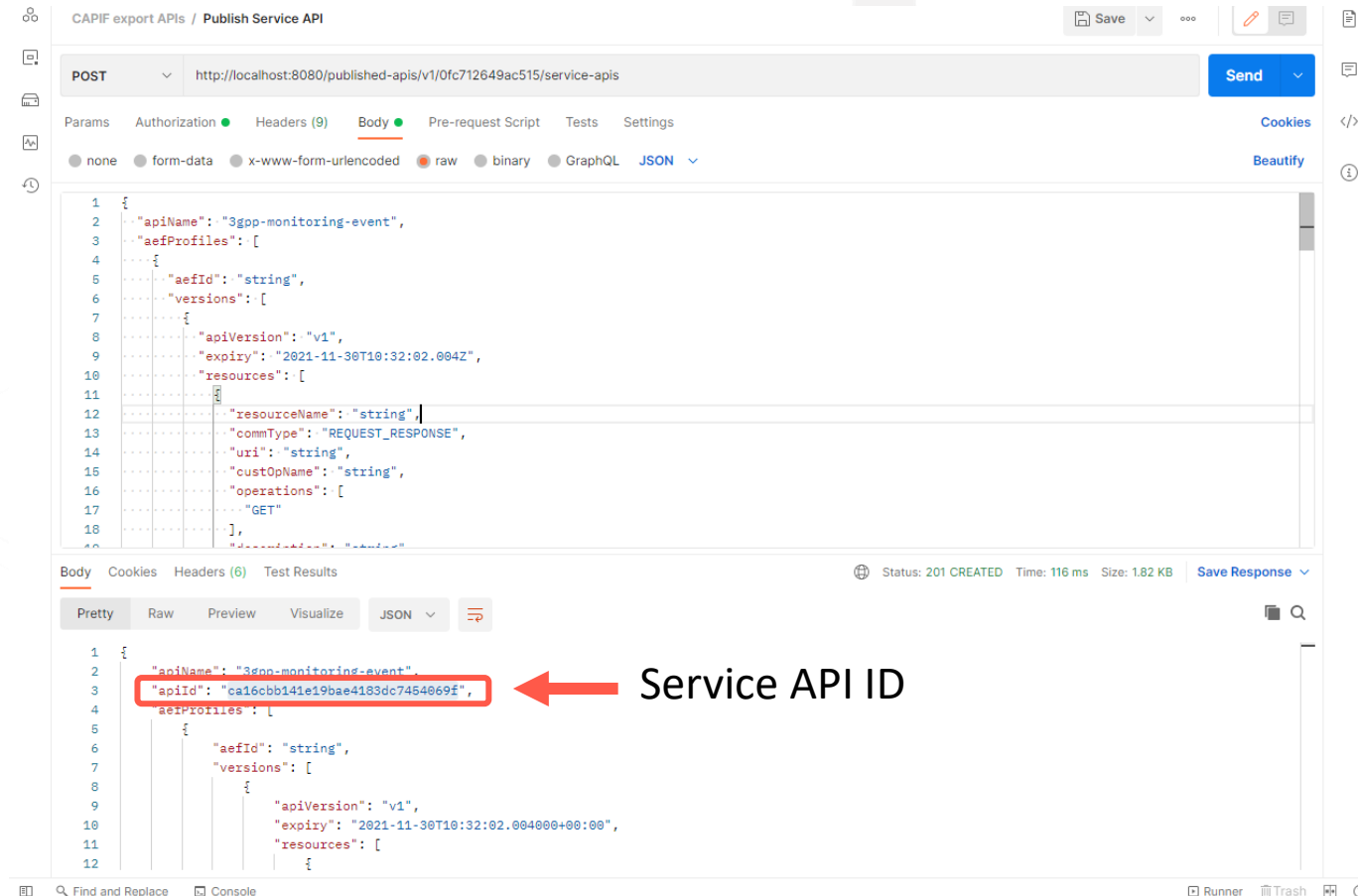
Offboard Invoker (NetApp)

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



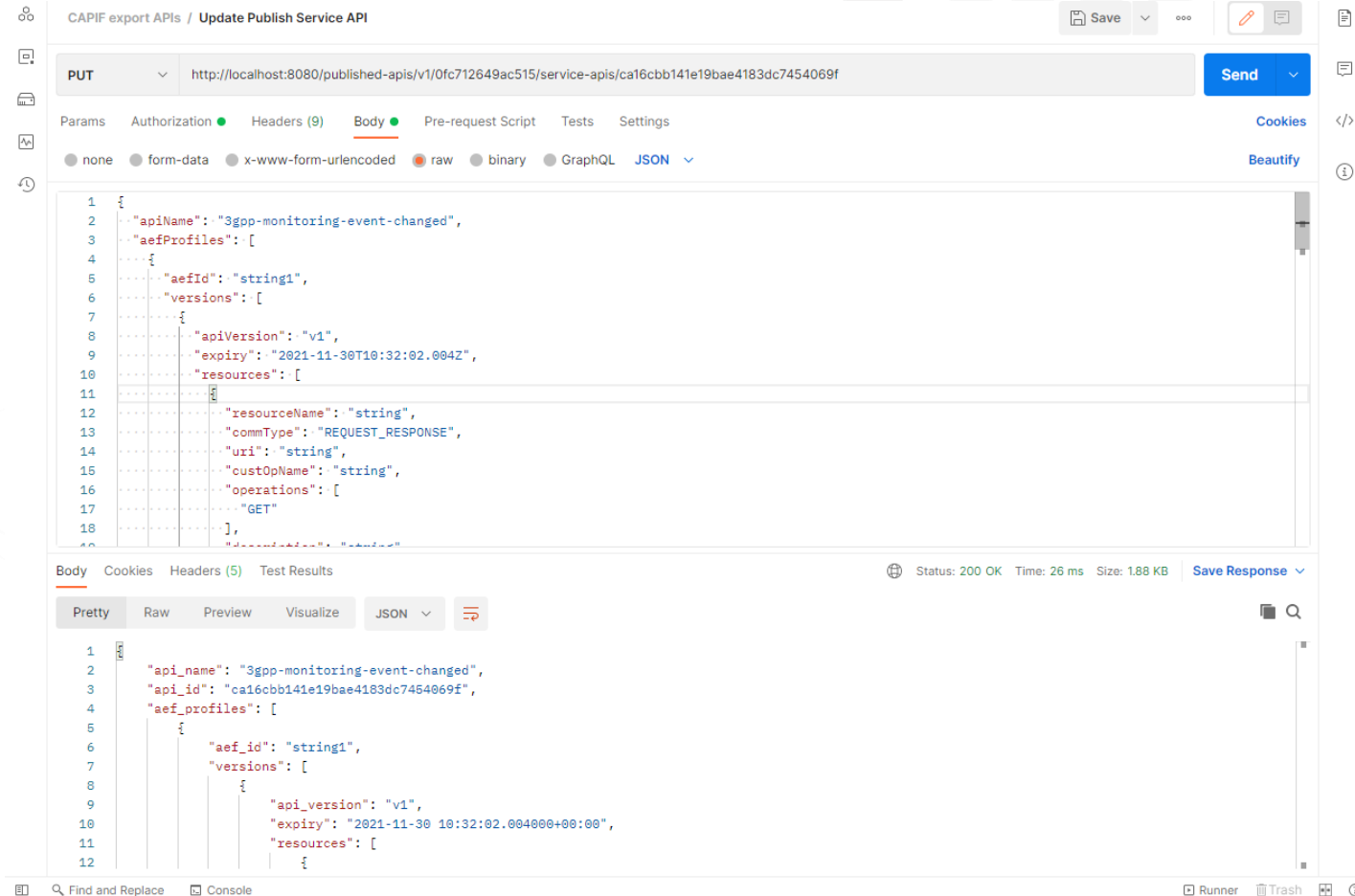
Publish Service API (APF)

- 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 with the following details:

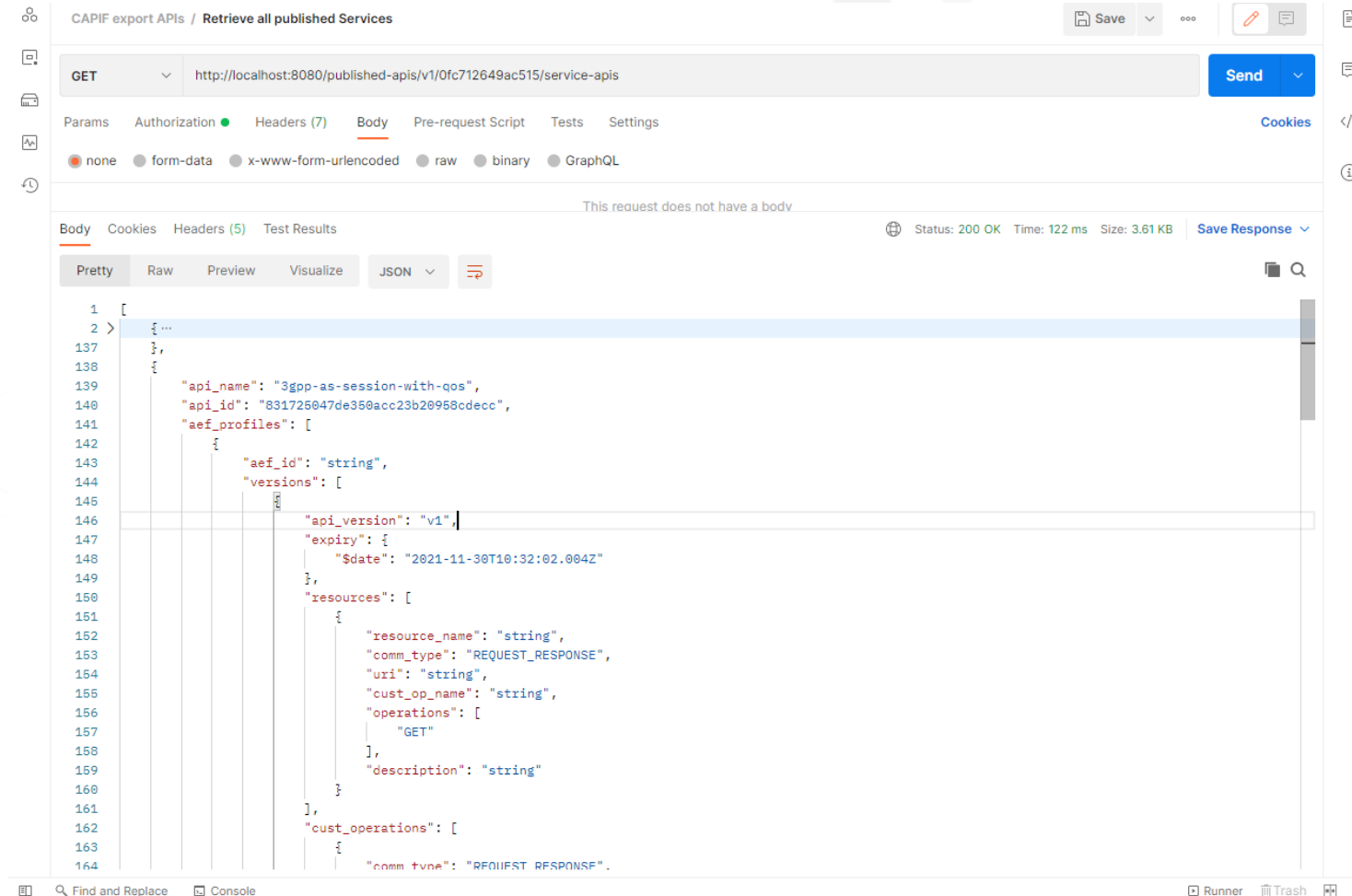
- Method:** PUT
- URL:** `http://localhost:8080/published-apis/v1/0fc712649ac515/service-apis/ca16cbb141e19bae4183dc7454069f`
- Body (Request):**

```
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 }
```
- Status:** 200 OK, Time: 26 ms, Size: 1.88 KB
- Body (Response):**

```
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 }
```

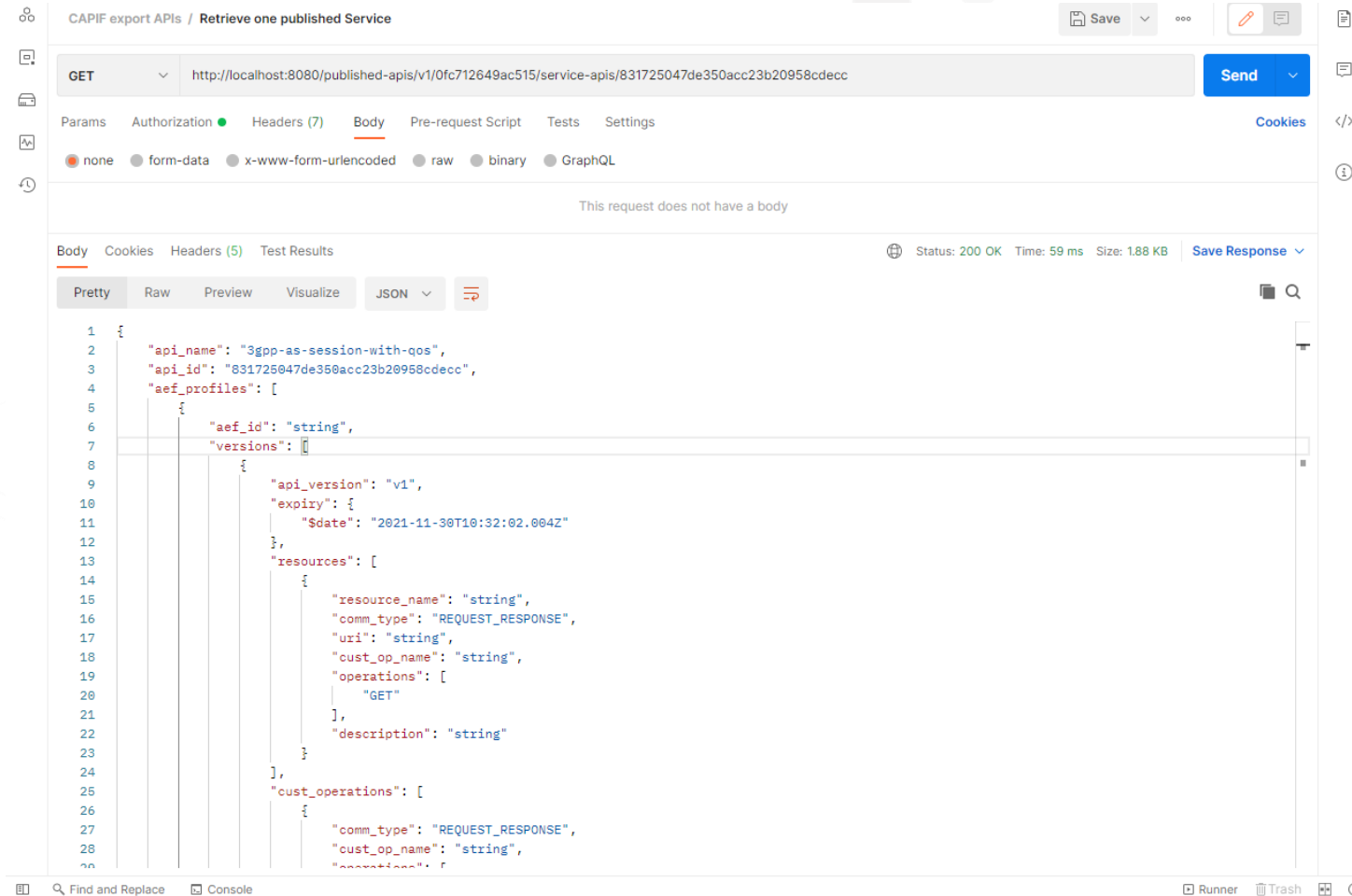
Retrieve all published services (APF)

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



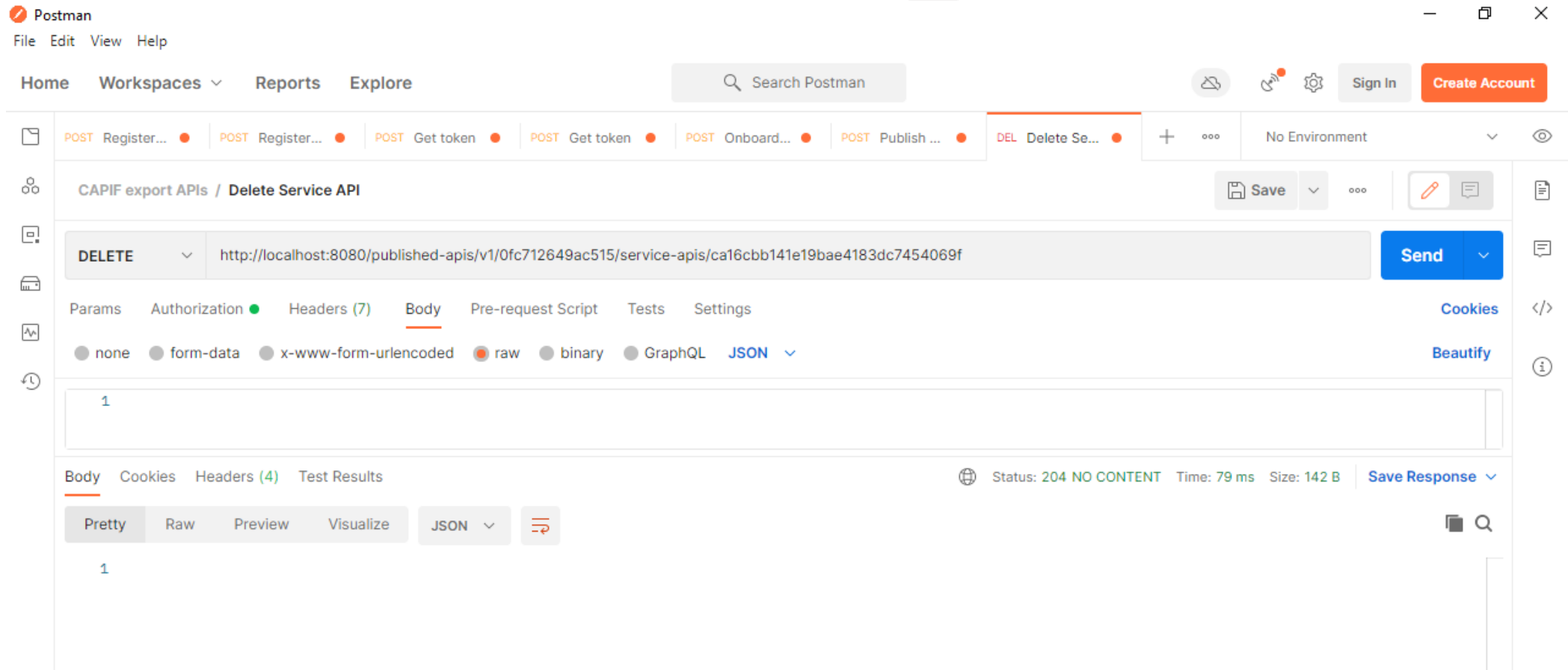
Retrieve one published services (APF)

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



Delete Service API (APF)

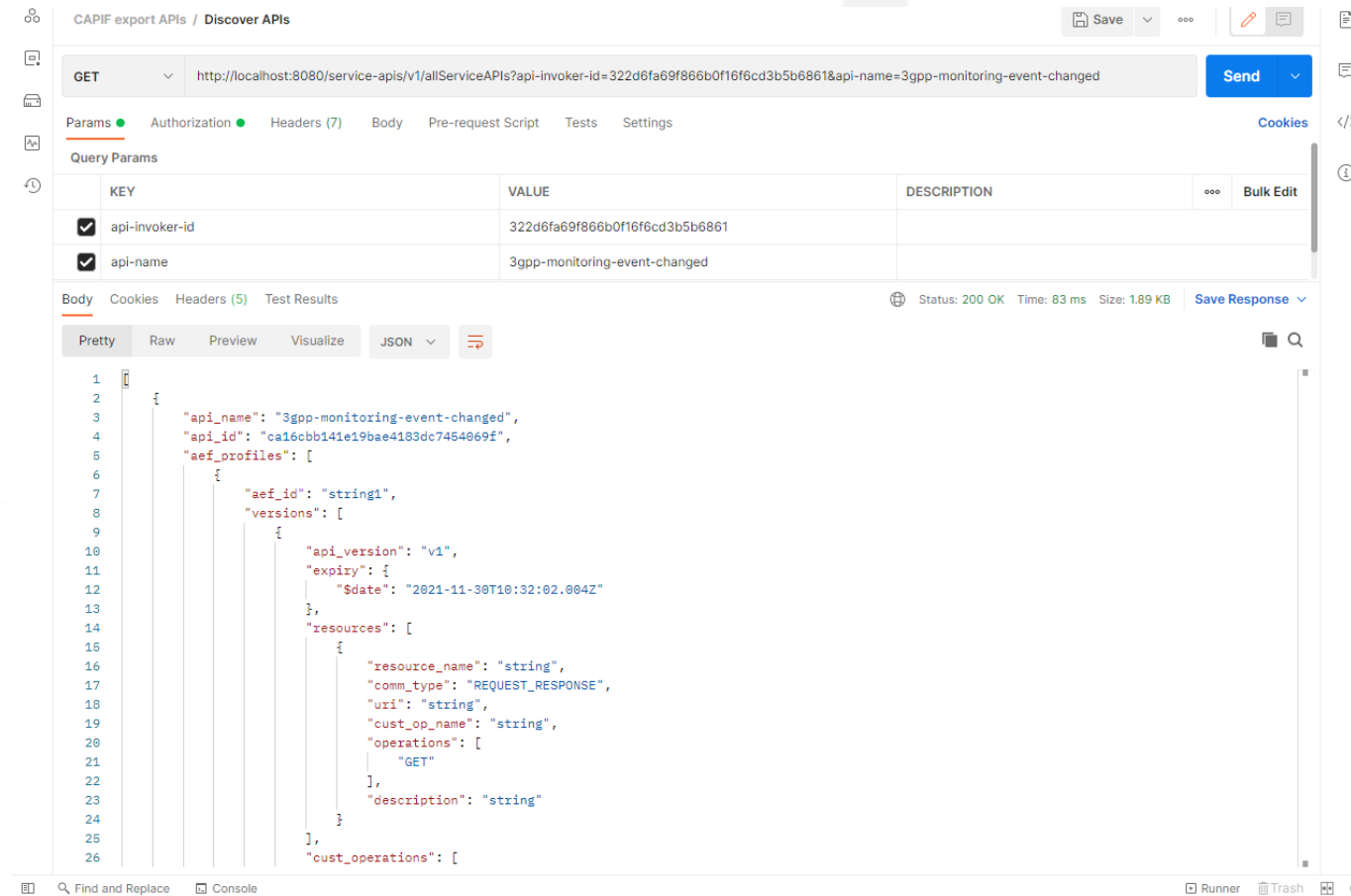
- DELETE `http://<CAPIF Host IP>:8080/published-apis/v1/<APF ID>/service-apis/<Service API ID>`



The image shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'Reports', and 'Explore'. Below this, a search bar and 'Sign In'/'Create Account' buttons are visible. The main workspace shows a collection of APIs under 'CAPIF export APIs'. The selected API is 'Delete Service API', which is a DELETE request. The URL is `http://localhost:8080/published-apis/v1/0fc712649ac515/service-apis/ca16cbb141e19bae4183dc7454069f`. The request body is empty. The response status is '204 NO CONTENT', with a time of 79 ms and a size of 142 B. The response body is also empty.

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