

**PROJECT 1 – CMSC 621**  
**Nipun, Rishabh, Arshita, Aniruddha**

**Project Group Members:**

1. Nipun Ramagiri ([nipunr1@umbc.edu](mailto:nipunr1@umbc.edu) )
2. Rishabh Sachdeva ([rishabs1@umbc.edu](mailto:rishabs1@umbc.edu))
3. Arshita Jain ([a253@umbc.edu](mailto:a253@umbc.edu) )
4. Aniruddha Saha ([anisaha1@umbc.edu](mailto:anisaha1@umbc.edu) )

**INSTRUCTIONS TO RUN WEB APIs/WEB SERVER:**

- Unzip the submitted Zipfile (**Project\_1\_Nipun\_Aniruddha\_Arshita\_Rishab.zip**).
- The rest-api.go file in Project 1 folder is to be run.
  - **go run rest-api.go**
- The server runs on **localhost** port **8001 (localhost:8001)**

**Persistent Storage:**

- JSON file (**sites.json**)
- Persistence is maintained using the “**sync**” package’s **Mutex lock**.

## **DOCUMENTATION**

**Web Toolkit Used:** [Gorilla web toolkit](#)

The following documentation contains all the APIs with their respective functions described. Different scenarios of the APIs being used are also demonstrated with appropriate action being taken, which can be seen implemented in the screenshots following the API description. The APIs were run and tested using a RESTful client tool/API Development Tool [Postman](#).

**CRUD OPERATIONS FOR SITE:**

**Create APIs**

**1. /sites/{name}:**

Type: **PUT**

Function: Updates the site Name = {name} if it exists, else creates a new site based on details passed in BODY. In this case, access points are not updated.

**2. /sites**

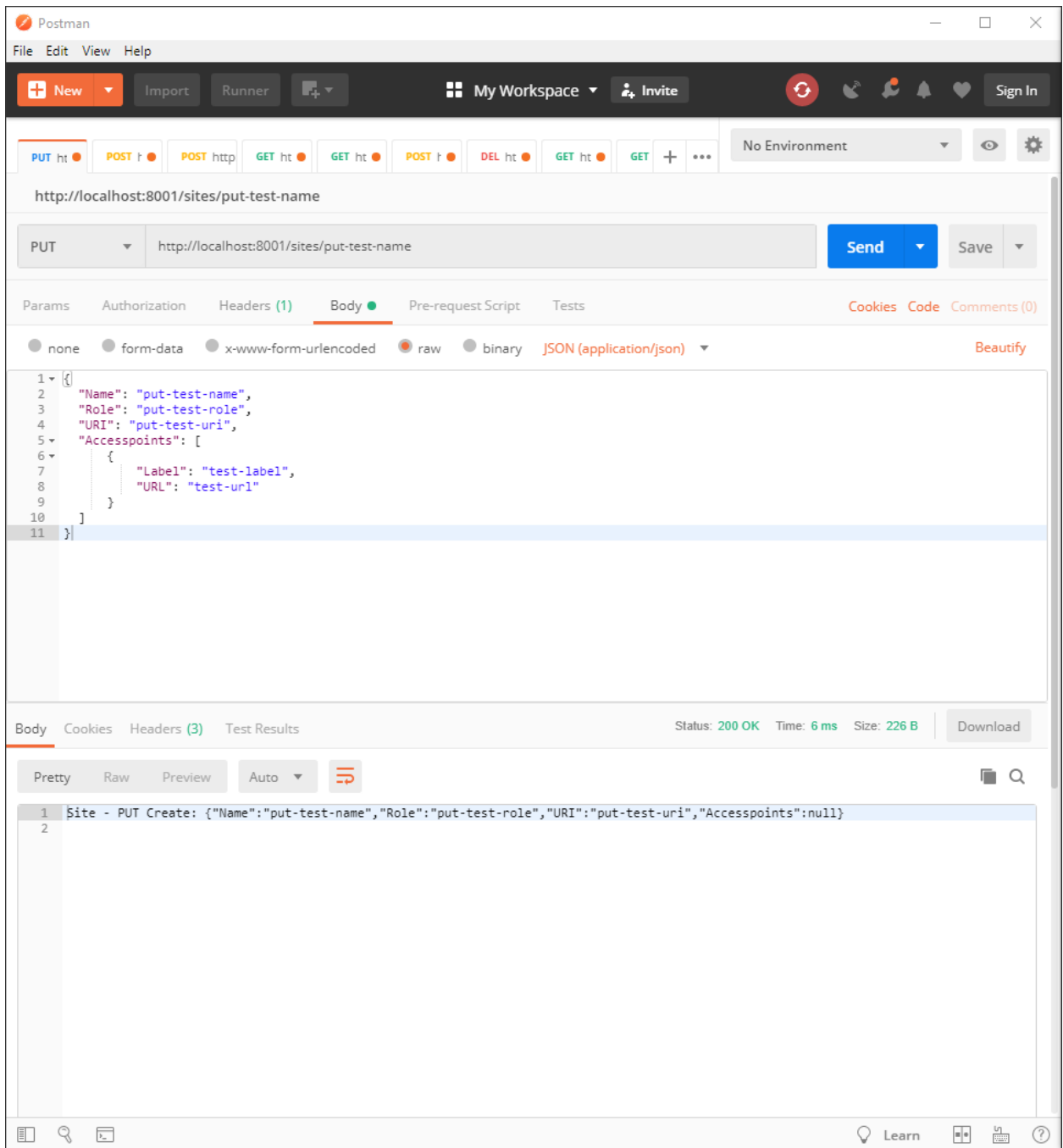
Type: **POST**

Function: Creates a new site based on details passed in Body if it doesn't exist.

Screenshots:

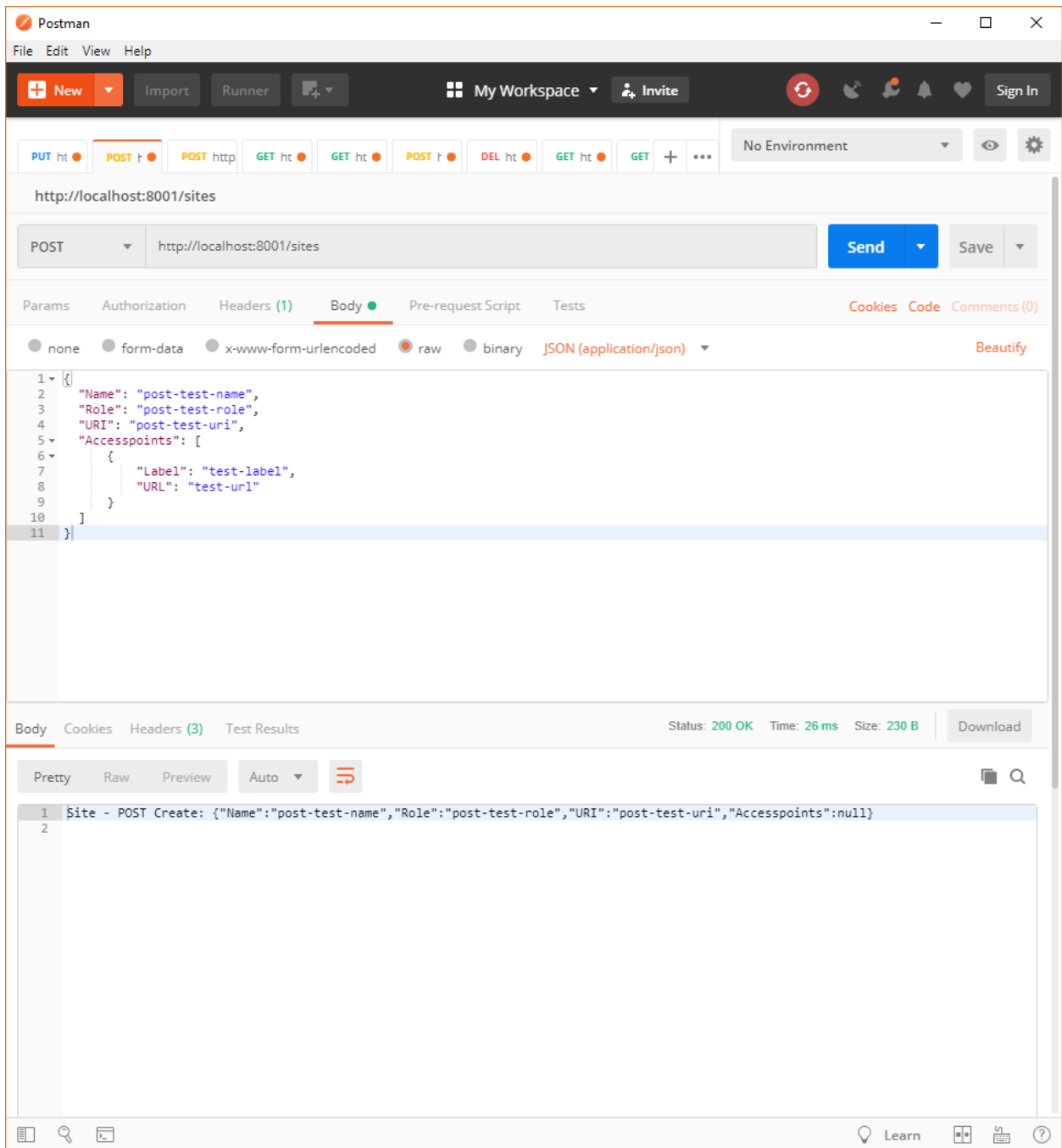
# PROJECT 1 – CMSC 621

Nipun, Rishabh, Arshita, Aniruddha



# PROJECT 1 – CMSC 621

Nipun, Rishabh, Arshita, Aniruddha



# PROJECT 1 – CMSC 621

Nipun, Rishabh, Arshita, Aniruddha

The screenshot shows the Postman application interface. At the top, there's a menu bar with 'File', 'Edit', 'View', and 'Help'. Below it is a toolbar with buttons for 'New', 'Import', 'Runner', and 'My Workspace'. The main area is divided into several sections. The top section shows the request URL 'http://localhost:8001/sites' and the method 'POST'. Below this, there are tabs for 'Params', 'Authorization', 'Headers (1)', 'Body', 'Pre-request Script', and 'Tests'. The 'Body' tab is selected, showing a JSON body with the following structure:

```
1 {
2   "Name": "",
3   "Role": "post-test-role",
4   "URI": "post-test-uri",
5   "Accesspoints": [
6     {
7       "Label": "test-label",
8       "URL": "test-uri"
9     }
10  ]
11 }
```

Below the body, there's a status bar showing 'Status: 200 OK', 'Time: 1 ms', and 'Size: 134 B'. At the bottom, there's a response section with tabs for 'Body', 'Cookies', 'Headers (3)', and 'Test Results'. The 'Body' tab is selected, showing the response body in 'Pretty' format:

```
1 No Name in Body.
2
```

**PROJECT 1 – CMSC 621**  
**Nipun, Rishabh, Arshita, Aniruddha**

**Read API:**

**1. /sites/{name}:**

Type: **GET**

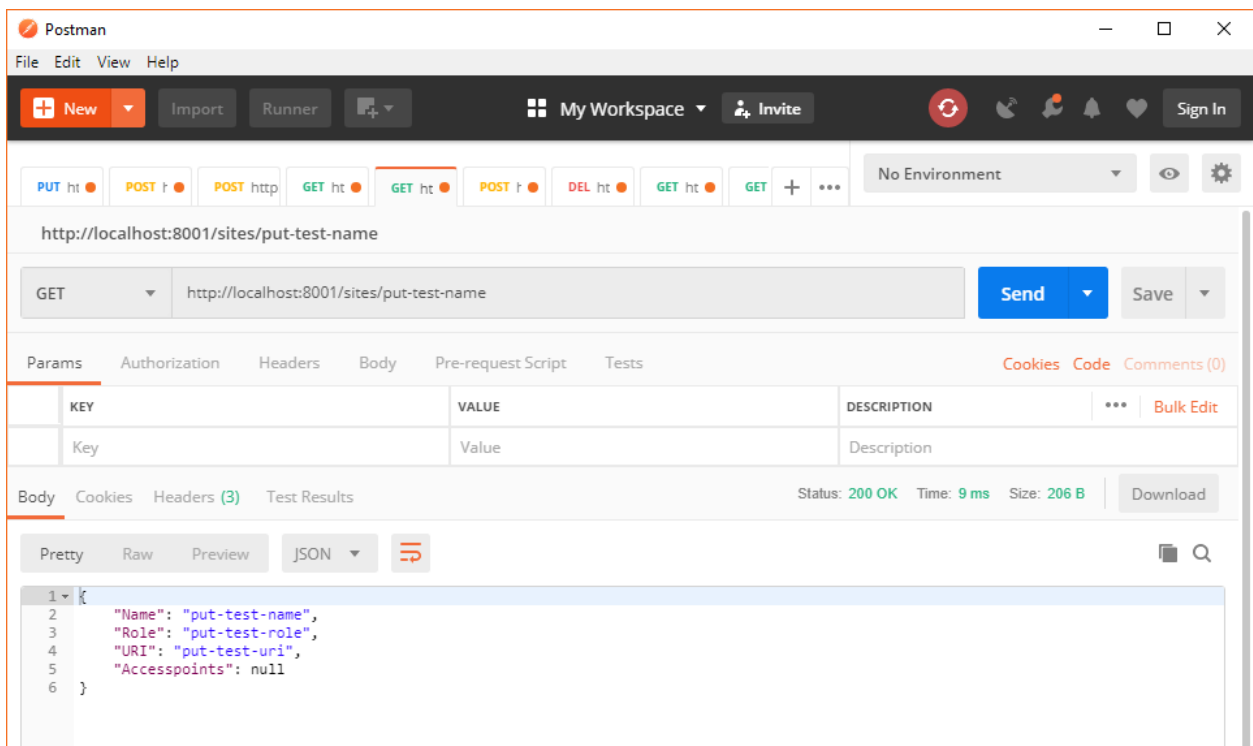
Function: Retrieves the site if present, i.e. Name = {name}

**2. /sites**

Type: **GET**

Function: Retrieves all the sites.

Screenshots:



# PROJECT 1 – CMSC 621

Nipun, Rishabh, Arshita, Aniruddha

Postman

File Edit View Help

+ New Import Runner My Workspace Invite No Environment Sign In

PUT ht POST ht POST http GET ht GET ht POST ht DEL ht GET ht GET + ...

http://localhost:8001/sites

GET http://localhost:8001/sites Send Save

Params Authorization Headers Body Pre-request Script Tests Cookies Code Comments (0)

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (3) Test Results Status: 200 OK Time: 21 ms Size: 567 B Download

Pretty Raw Preview JSON

```
1 [
2   {
3     "Name": "name-1",
4     "Role": "role-1",
5     "URI": "uri-1",
6     "Accesspoints": [
7       {
8         "Label": "label-1",
9         "URL": "url-1"
10      },
11      {
12        "Label": "label-2",
13        "URL": "url-2"
14      }
15    ]
16  },
17  {
18    "Name": "name-2",
19    "Role": "role-2",
20    "URI": "uri-2",
21    "Accesspoints": [
22      {
23        "Label": "label-3",
24        "URL": "url-3"
25      },
26      {
27        "Label": "label-2",
28        "URL": "url-2"
29      }
30    ]
31  },
32  {
33    "Name": "put-test-name",
34    "Role": "put-test-role",
35    "URI": "put-test-uri",
36    "Accesspoints": null
37  },
38  {
39    "Name": "post-test-name",
40    "Role": "post-test-role",
41  }
42 ]
```

Learn

## PROJECT 1 – CMSC 621

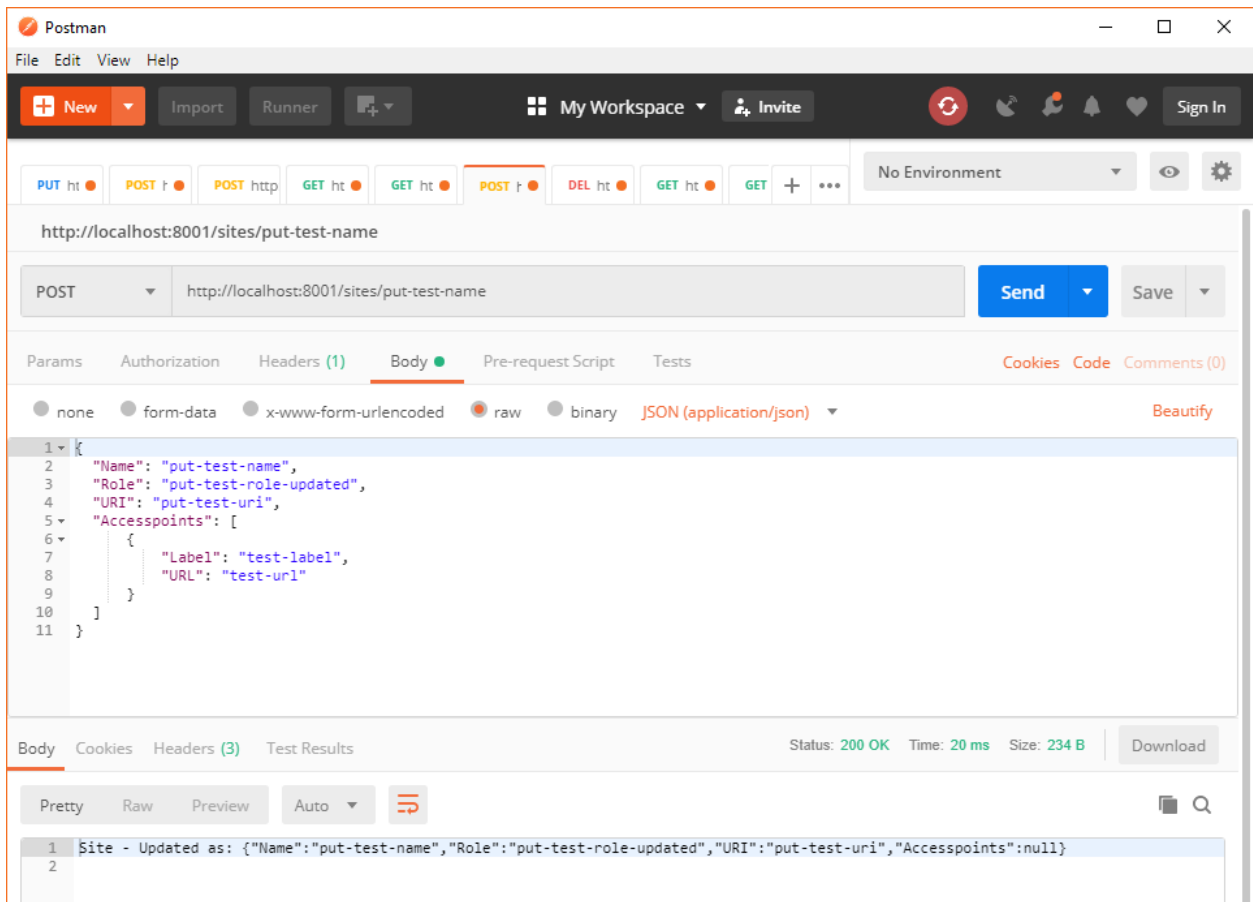
Nipun, Rishabh, Arshita, Aniruddha

### Update API:

#### 1. /sites/{name}:

Type: **POST**

Function: Updates the site if exists of Name = {name} based on details passed in BODY. It doesn't update its access points.



### Delete API:

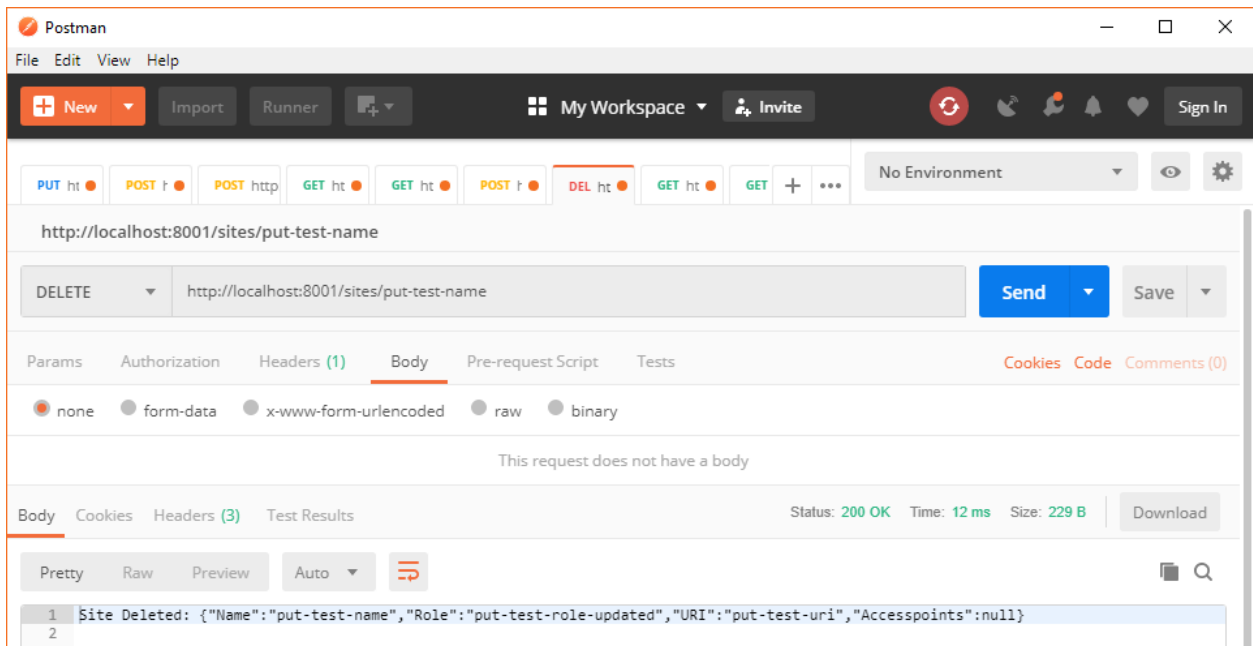
#### 1. /sites/{name}:

Type: **DELETE**

Function: Deletes the site if present of name = {name}, along with its access points.

# PROJECT 1 – CMSC 621

Nipun, Rishabh, Arshita, Aniruddha



## CRUD OPERATIONS FOR ACCESS POINTS:

### Create APIs

#### 1. /sites/{name}/accesspoints/{label}

Type: **PUT**

Function: Updates the Access point if exists of label = {label} associated with site name={name}, else creates a new site based on details passed in BODY. Else, creates a new one.

#### 2. /sites/{name}/accesspoints

Type: **POST**

Function: Creates a new access point under site of name={name} based on details passed in Body.

Screenshots:



**PROJECT 1 – CMSC 621**  
**Nipun, Rishabh, Arshita, Aniruddha**

http://localhost:8001/sites/post-test-name/accesspoints/test-label

PUT http://localhost:8001/sites/post-test-name/accesspoints/test-label

Params Authorization Headers **Body** Pre-request Script Tests

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

Body Cookies Headers (3) Test Results

Pretty Raw Preview Auto ↻

```
1 AP - PUT Create: {"Label":"test-label","URL":""}  
2
```

http://localhost:8001/sites/post-test-name/accesspoints/test-label

PUT http://localhost:8001/sites/post-test-name/accesspoints/test-label

Params Authorization Headers (1) **Body** Pre-request Script Tests

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {  
2   "Label": "no-change-test-label",  
3   "URL": "test-ur1"  
4 }
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview Auto ↻

```
1 AP - PUT Update: {"Name":"post-test-name","Role":"post-test-role","URI":"post-test-uri","Accesspoints":[{"Label":"test-label","URL":"test-ur1"}]}  
2
```

http://localhost:8001/sites/post-test-name/accesspoints/test-label

POST http://localhost:8001/sites/post-test-name/accesspoints/test-label

Params Authorization Headers (1) **Body** Pre-request Script Tests

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```
1 {  
2   "Label": "test-label-new",  
3   "URL": "test-ur1-new"  
4 }
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview Auto ↻

```
1 Site AP - Updated as: {"Name":"post-test-name","Role":"post-test-role","URI":"post-test-uri","Accesspoints":[{"Label":"test-label","URL":"test-ur1-new"}]}  
2
```

**PROJECT 1 – CMSC 621**  
**Nipun, Rishabh, Arshita, Aniruddha**

**Read API**

**1. /sites/{name}/accesspoints**

Type: **GET**

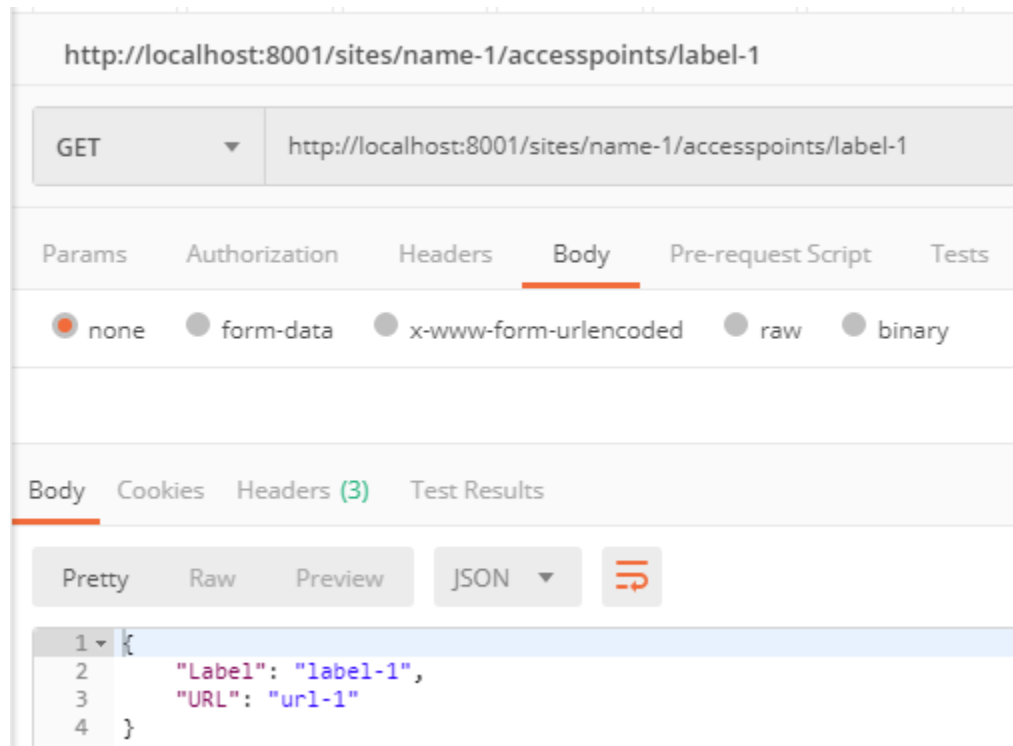
Function: Retrieve all the access points of site with name = {name}

**2. /sites/{name}/accesspoints/{label}**

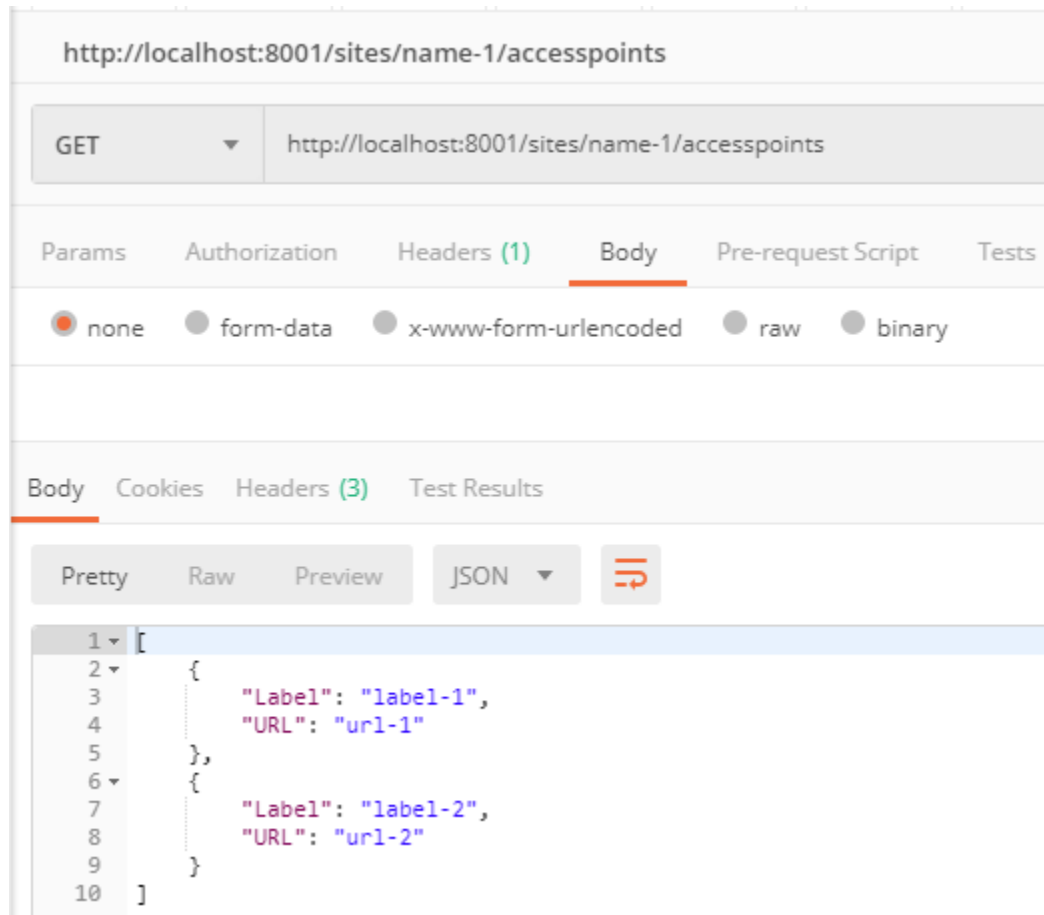
Type: **GET**

Function: Retrieves the specific Access point of label={label}, associated with site name = {name}.

Screenshots:



**PROJECT 1 – CMSC 621**  
**Nipun, Rishabh, Arshita, Aniruddha**



**Update API:**

**1. /sites/{name}/accesspoints/{label}**

Type: **POST**

Function: Updates the access point site if exists of label = {label} associated with site Name={name}, based on details passed in BODY.

# PROJECT 1 – CMSC 621

Nipun, Rishabh, Arshita, Aniruddha

http://localhost:8001/sites/post-test-name/accesspoints/test-label

POST http://localhost:8001/sites/post-test-name/accesspoints/test-label

Params Authorization Headers (1) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "Label": "test-label-new",
3   "URL": "test-uri-new"
4 }
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview Auto

```
1 Site AP - Updated as: {"Name":"post-test-name","Role":"post-test-role","URI":"post-test-uri","Accesspoints":[{"Label":"test-label","URL":"test-uri-new"}]}
2
```

http://localhost:8001/sites/name-1/accesspoints/label-1

POST http://localhost:8001/sites/name-1/accesspoints/label-1

Params Authorization Headers (1) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "URL": "uri-123"
3 }
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview Auto

```
1 Site AP - Updated as: {"Name":"name-1","Role":"role-1","URI":"uri-1","Accesspoints":[{"Label":"label-1","URL":"uri-123"},{"Label":"label-2","URL":"uri-2"}]}
2
```

http://localhost:8001/sites/name-1/accesspoints

POST http://localhost:8001/sites/name-1/accesspoints

Params Authorization Headers (1) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "URL": "uri-123"
3 }
```

Body Cookies Headers (3) Test Results

Pretty Raw Preview Auto

```
1 No Label in Body.
2
```

**PROJECT 1 – CMSC 621**  
**Nipun, Rishabh, Arshita, Aniruddha**

**Delete API:**

**1. /sites/{name}/accesspoints/{label}:**

Type: **DELETE**

Function: Deletes the Access Point if present of label = {label}, associated with the site of name={name}.

Screenshots:

The first screenshot shows a REST client interface for a DELETE request to `http://localhost:8001/sites/post-test-name/accesspoints/test-label`. The request body is empty. The response is shown in the 'Body' tab, indicating 'AP Deleted: {"Label": "test-label", "URL": "test-url-new"}'.

The second screenshot shows a REST client interface for a DELETE request to `http://localhost:8001/sites/post-test-name/accesspoints/test-label-new`. The request body is empty. The response is shown in the 'Body' tab, indicating 'Label not found. Nothing deleted.'