

1. Create topic

The screenshot shows the Postman interface with a POST request to `http://localhost:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics`. The request body is a JSON object:

```
{  "topic_name": "topicPostman2"}
```

The response is a 201 Created status with the following JSON body:

```
{  "kind": "KafkaTopic",  "metadata": {    "self": "http://rest-proxy:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics/topicPostman2",    "resource_name": "crn:///kafka=MyatF13LP6er_9SBlywsA/topic=topicPostman2"  },  "cluster_id": "MyatF13LP6er_9SBlywsA",  "topic_name": "topicPostman2",  "is_internal": false,  "replication_factor": 0,  "partitions_count": 0,  "partitions": {    "related": "http://rest-proxy:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics/topicPostman2/partitions"  },  "configs": {    "related": "http://rest-proxy:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics/topicPostman2/configs"  },  "partition_reassignments": {}}
```

The status bar shows: Status: 201 Created, Time: 106 ms, Size: 970 B.

2. List all topics

The screenshot shows the Postman interface with a GET request to `http://localhost:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics`. The response is a 200 OK status with the following JSON body:

```
{  "self": "http://rest-proxy:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics/topicPostman2",  "resource_name": "crn:///kafka=MyatF13LP6er_9SBlywsA/topic=topicPostman2",  "cluster_id": "MyatF13LP6er_9SBlywsA",  "topic_name": "topicPostman2",  "is_internal": false,  "replication_factor": 1,  "partitions_count": 1,  "partitions": {    "related": "http://rest-proxy:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics/topicPostman2/partitions"  },  "configs": {    "related": "http://rest-proxy:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics/topicPostman2/configs"  },  "partition_reassignments": {    "related": "http://rest-proxy:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics/topicPostman2/partitions/-/reassignment"  }}
```

The status bar shows: Status: 200 OK, Time: 86 ms, Size: 5416 KB.

3. Describe topic

The screenshot shows the Postman interface with a GET request to `http://localhost:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics/topicPostman2`. The request is sent, and the response is displayed in the Body tab, showing a JSON object with metadata and configuration details.

Request:

```
GET http://localhost:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics/topicPostman2
```

Response (JSON):

```
{
  "kind": "KafkaTopic",
  "metadata": {
    "self": "http://rest-proxy:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics/topicPostman2",
    "resource_name": "crn:///kafka-MyatF13LP6er_9SBlywsA/topic=topicPostman2"
  },
  "cluster_id": "MyatF13LP6er_9SBlywsA",
  "topic_name": "topicPostman2",
  "is_internal": false,
  "replication_factor": 1,
  "partitions_count": 1,
  "partitions": {
    "related": "http://rest-proxy:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics/topicPostman2/partitions"
  },
  "configs": {
    "related": "http://rest-proxy:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics/topicPostman2/configs"
  },
  "partition_reassignments": {}
}
```

Status: 200 OK Time: 139 ms Size: 875 B

4. Send data

The screenshot shows the Postman interface with a POST request to `http://localhost:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics/topicPostman2/records`. The request is sent, and the response is displayed in the Body tab, showing a JSON object with metadata and configuration details.

Request:

```
POST http://localhost:8082/v3/clusters/MyatF13LP6er_9SBlywsA/topics/topicPostman2/records
```

Response (JSON):

```
{
  "cluster_id": "MyatF13LP6er_9SBlywsA",
  "topic_name": "topicPostman2",
  "partition_id": 0,
  "offset": 0,
  "timestamp": "2023-11-06T19:45:14.948Z",
  "key": {
    "type": "JSON",
    "size": 6
  },
  "value": {
    "type": "JSON",
    "size": 16
  }
}
```

Status: 200 OK Time: 103 ms Size: 377 B

5. Create consumer group

The screenshot shows the Postman application interface. The top bar includes navigation links (Home, Workspaces, API Network, Explore), a search bar, and user settings. The left sidebar shows the 'Collections' panel with a collection named 'http://localhost:8082/consumers/HappyGroup'. The main workspace displays a POST request to 'http://localhost:8082/consumers/HappyGroup'. The 'Body' tab is selected, showing a JSON payload:

```
{  "name": "FirstConsumerPostman",  "format": "json",  "auto.offset.reset": "earliest"}
```

The 'Body' tab is also selected in the bottom panel, showing the same JSON payload in 'Pretty' format. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 79 ms', and 'Size: 314 B'. The Windows taskbar at the bottom shows the system clock as 9:46 PM on 11/6/2023.

6. Create subscription

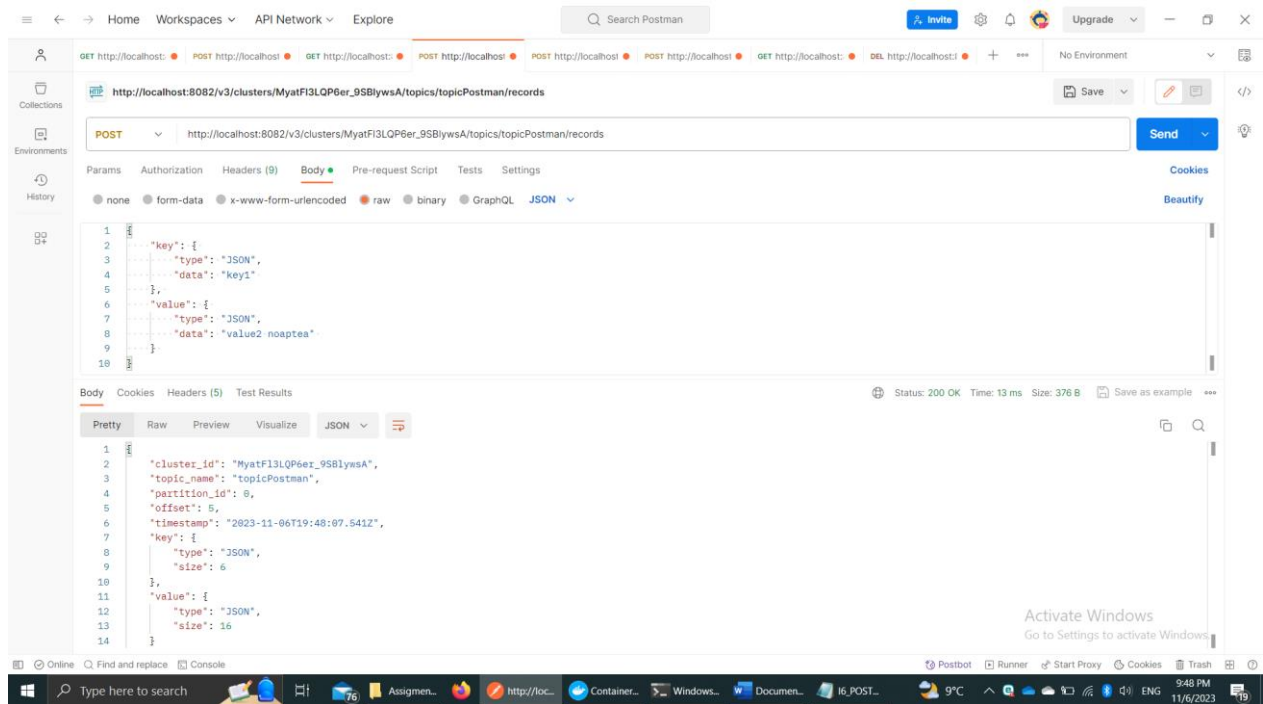
The screenshot shows the Postman application interface. The top bar is identical to the previous screenshot. The left sidebar shows the 'Collections' panel with a collection named 'http://localhost:8082/consumers/HappyGroup/instances/FirstConsumerPostman/subscription'. The main workspace displays a POST request to 'http://localhost:8082/consumers/HappyGroup/instances/FirstConsumerPostman/subscription'. The 'Body' tab is selected, showing a JSON payload:

```
{  "name": "FirstConsumerPostman",  "format": "json",  "auto.offset.reset": "earliest"}
```

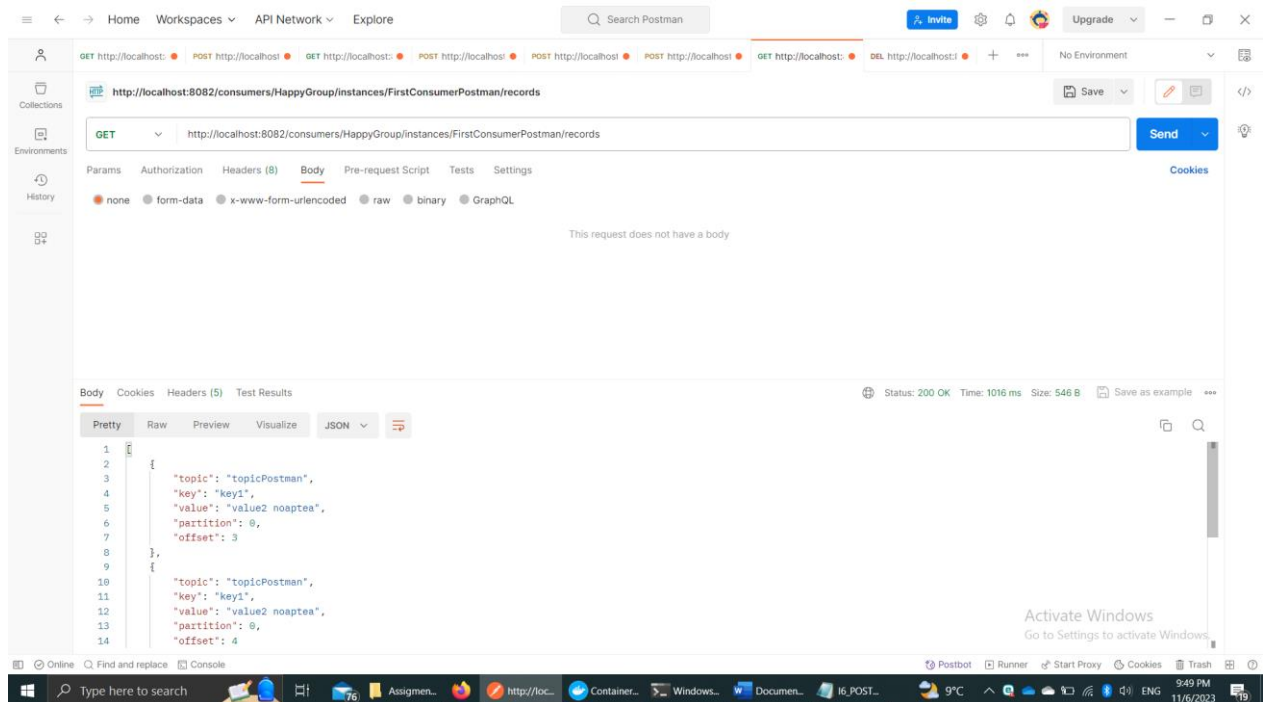
The 'Body' tab is also selected in the bottom panel, showing the same JSON payload in 'Pretty' format. The status bar at the bottom indicates 'Status: 204 No Content', 'Time: 92 ms', and 'Size: 64 B'. The Windows taskbar at the bottom shows the system clock as 9:47 PM on 11/6/2023.

7. Get messages

2 messages sent on topicPostman



Messages received



8. Delete consumer group

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', 'API Network', and 'Explore'. A search bar labeled 'Search Postman' is present. Below this, a collection of requests is visible, with the selected request being a DELETE request to 'http://localhost:8082/consumers/HappyGroup/instances/FirstConsumerPostman'. The request is configured with the method 'DELETE' and the URL 'http://localhost:8082/consumers/HappyGroup/instances/FirstConsumerPostman'. The 'Params' tab is active, showing a table for 'Query Params' with columns 'Key', 'Value', and 'Description'. The 'Body' tab is also visible, showing a status of '204 No Content', a time of '40 ms', and a size of '64 B'. The bottom of the screen shows the Windows taskbar with various open applications and the system clock indicating 9:50 PM on 11/6/2023.

DELETE http://localhost:8082/consumers/HappyGroup/instances/FirstConsumerPostman

Query Params

Key	Value	Description
Key	Value	Description

Body

Status: 204 No Content Time: 40 ms Size: 64 B

Activate Windows
Go to Settings to activate Windows.