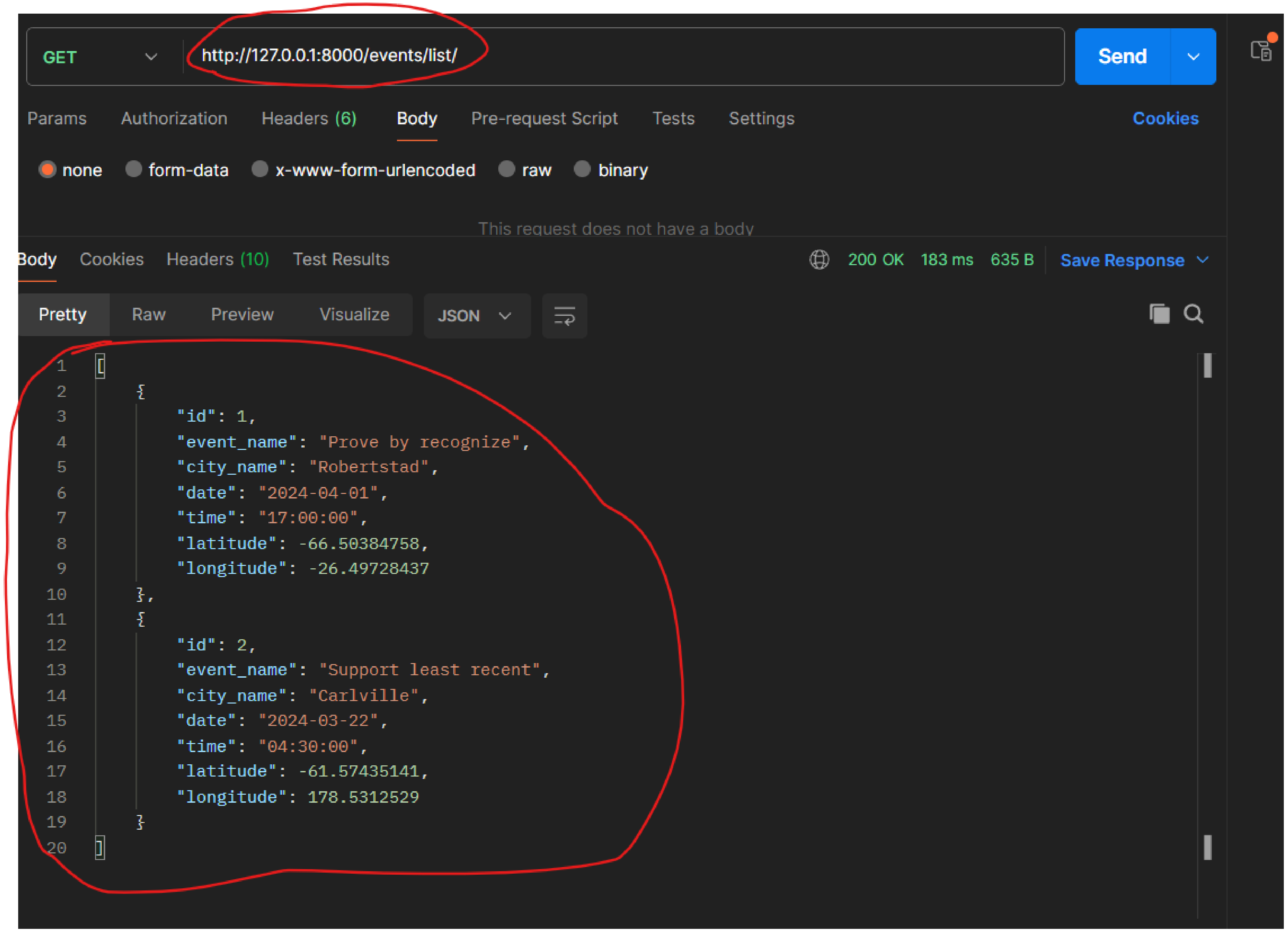
Upon examination of the screenshot, it's evident that the event list was successfully retrieved using a GET request.



GET http://127.0.0.1:8000/events/list/

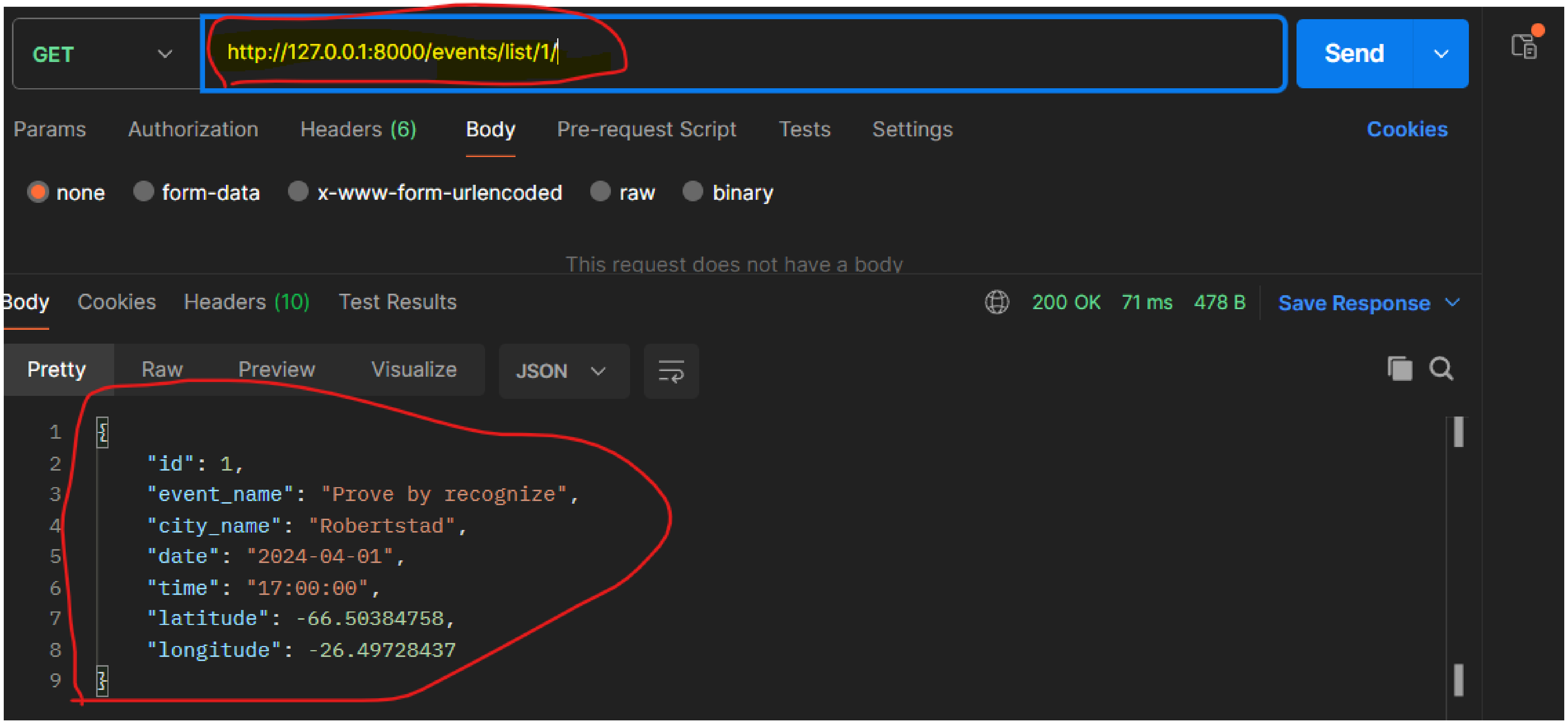
Body: none

This request does not have a body

200 OK 183 ms 635 B

```
[
    {
        "id": 1,
        "event_name": "Prove by recognize",
        "city_name": "Robertstad",
        "date": "2024-04-01",
        "time": "17:00:00",
        "latitude": -66.50384758,
        "longitude": -26.49728437
    },
    {
        "id": 2,
        "event_name": "Support least recent",
        "city_name": "Carlville",
        "date": "2024-03-22",
        "time": "04:30:00",
        "latitude": -61.57435141,
        "longitude": 178.5312529
    }
]
```

An examination of the screenshot reveals the successful retrieval of a specific event using its ID through a GET request.



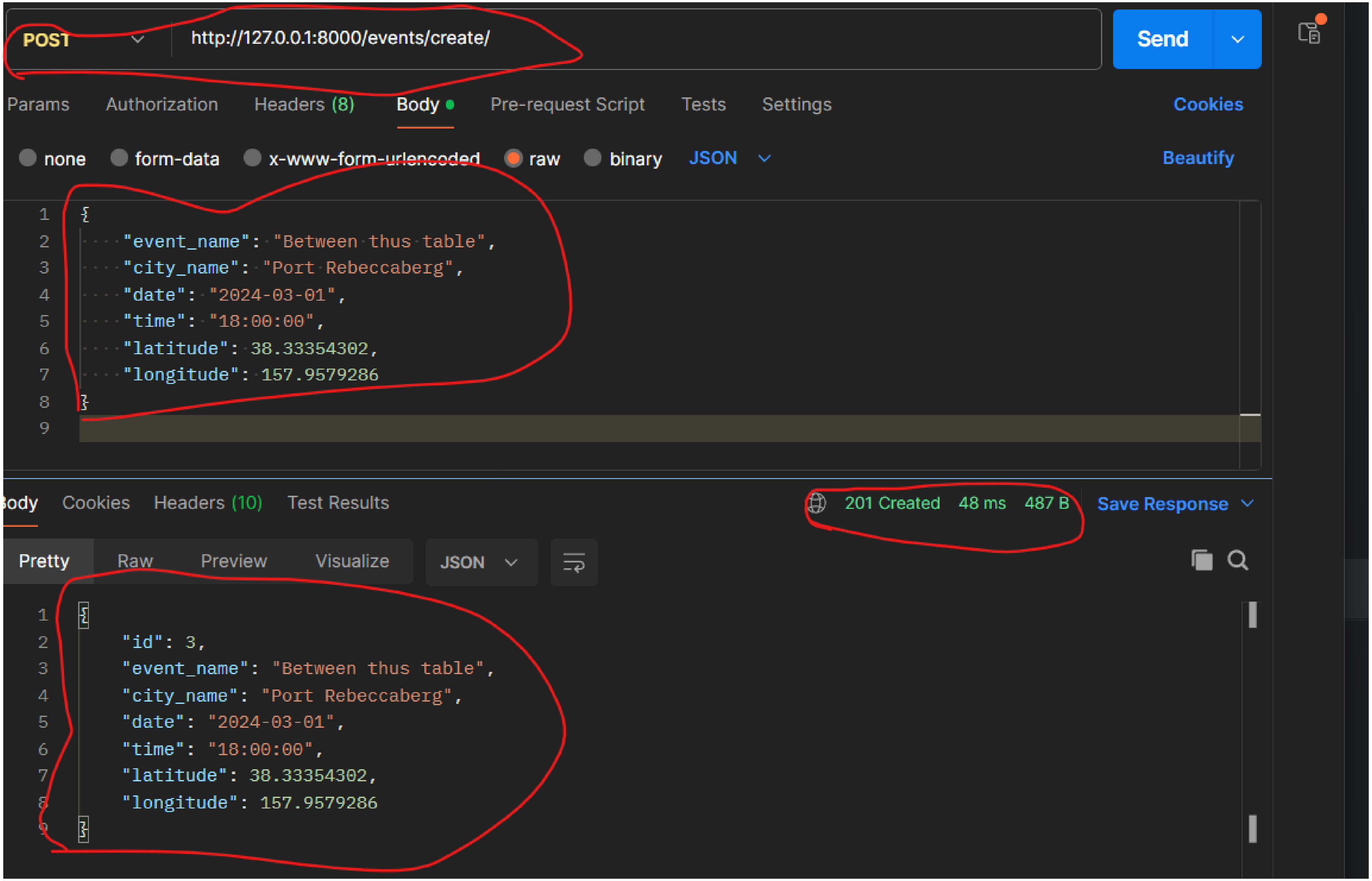
```
GET http://127.0.0.1:8000/events/list/1/
```

This request does not have a body

200 OK 71 ms 478 B

```
{
    "id": 1,
    "event_name": "Prove by recognize",
    "city_name": "Robertstad",
    "date": "2024-04-01",
    "time": "17:00:00",
    "latitude": -66.50384758,
    "longitude": -26.49728437
}
```

The creation of a new event is clearly demonstrated in the provided screenshot using POST request



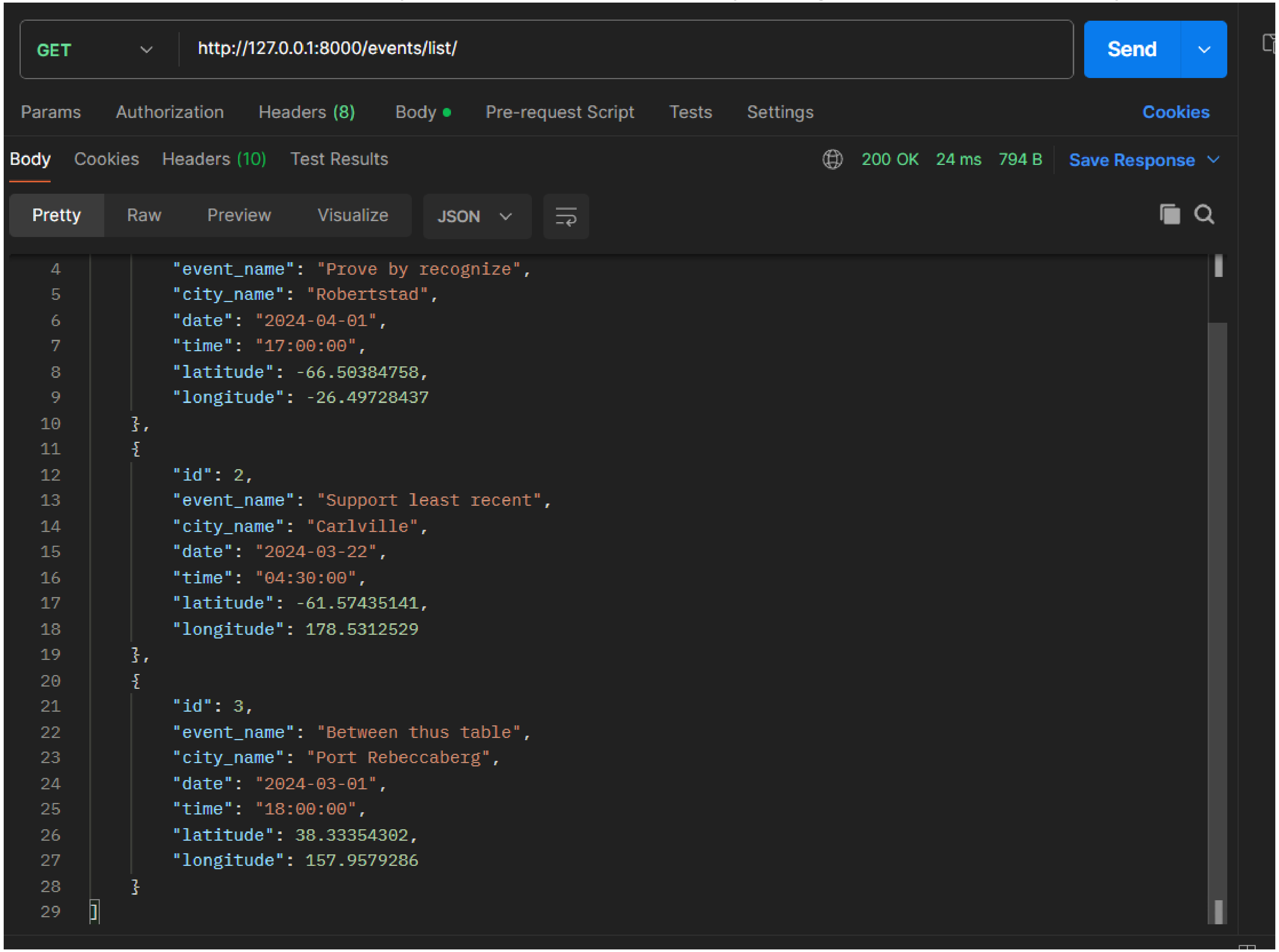
```
http://127.0.0.1:8000/events/create/
```

```
{
    "event_name": "Between thus table",
    "city_name": "Port Rebeccaberg",
    "date": "2024-03-01",
    "time": "18:00:00",
    "latitude": 38.33354302,
    "longitude": 157.9579286
}
```

201 Created 48 ms 487 B

```
{
    "id": 3,
    "event_name": "Between thus table",
    "city_name": "Port Rebeccaberg",
    "date": "2024-03-01",
    "time": "18:00:00",
    "latitude": 38.33354302,
    "longitude": 157.9579286
}
```

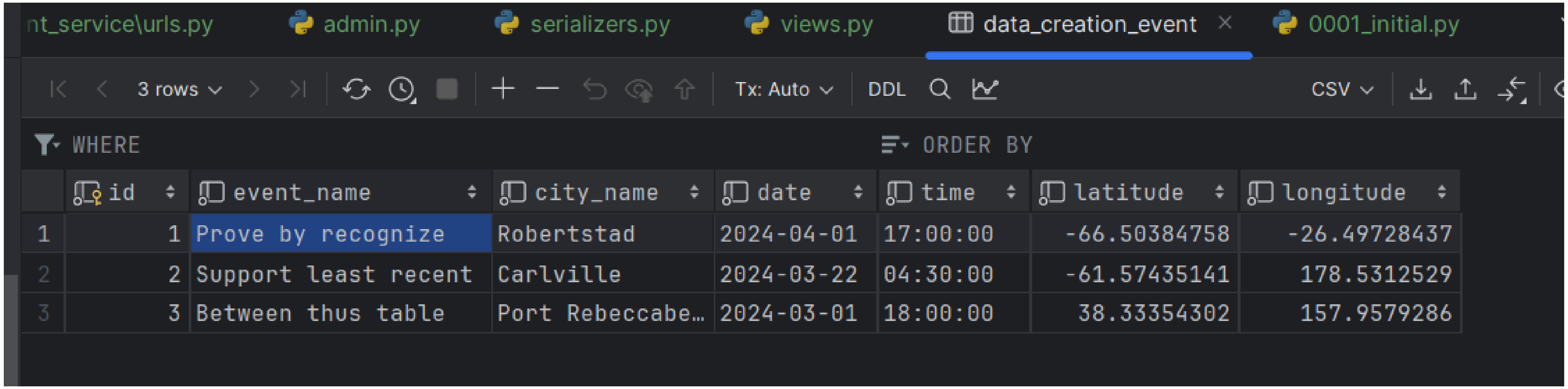
The screenshot illustrates the update of the events list, incorporating a new event identified by the id =3.



```
http://127.0.0.1:8000/events/list/
```

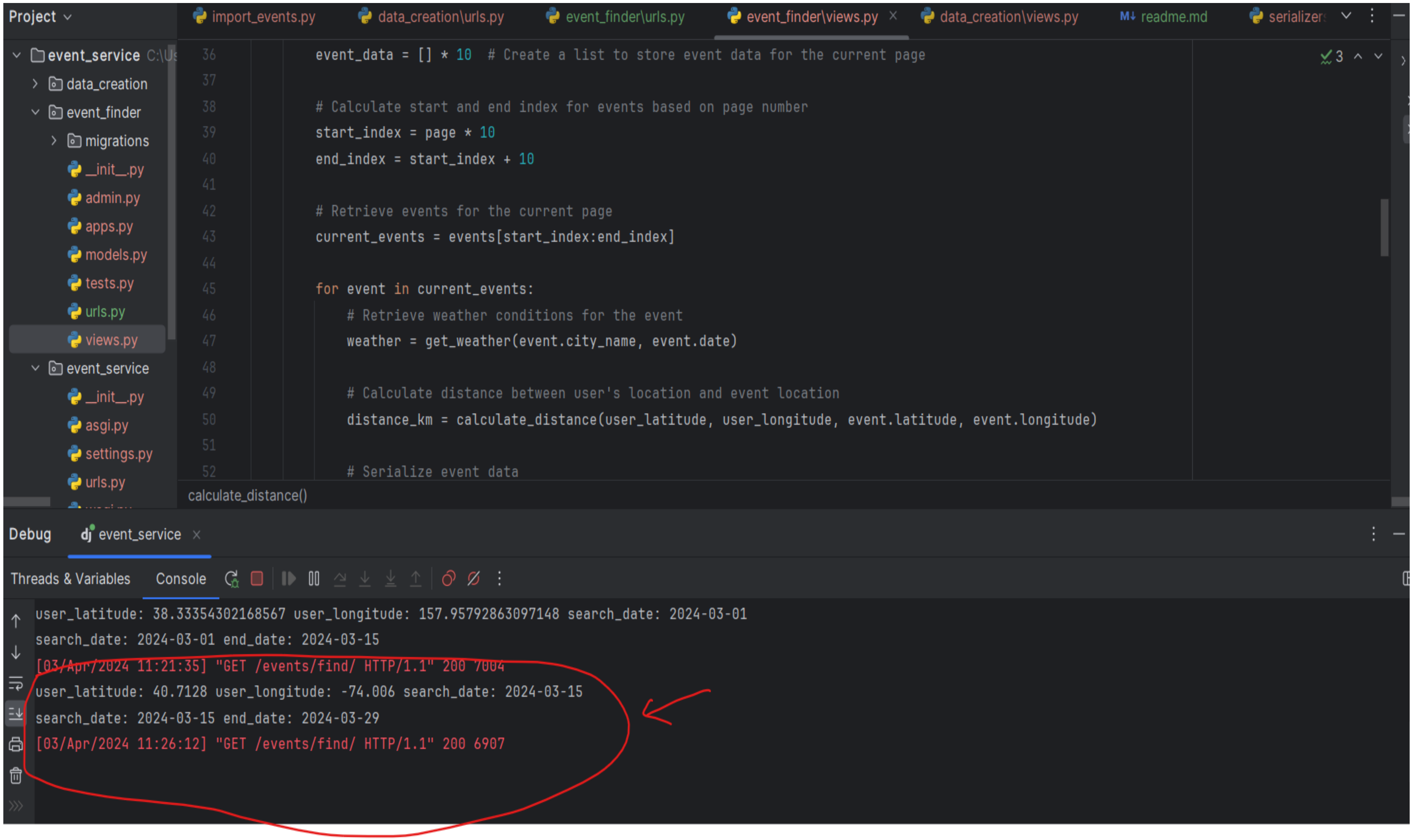
```
        "event_name": "Prove by recognize",
        "city_name": "Robertstad",
        "date": "2024-04-01",
        "time": "17:00:00",
        "latitude": -66.50384758,
        "longitude": -26.49728437
    },
    {
        "id": 2,
        "event_name": "Support least recent",
        "city_name": "Carlville",
        "date": "2024-03-22",
        "time": "04:30:00",
        "latitude": -61.57435141,
        "longitude": 178.5312529
    },
    {
        "id": 3,
        "event_name": "Between thus table",
        "city_name": "Port Rebeccaberg",
        "date": "2024-03-01",
        "time": "18:00:00",
        "latitude": 38.33354302,
        "longitude": 157.9579286
    }
]
```

Furthermore, the modifications are appropriately reflected within the internal database, as confirmed by the screenshot.



	id	event_name	city_name	date	time	latitude	longitude
1	1	Prove by recognize	Robertstad	2024-04-01	17:00:00	-66.50384758	-26.49728437
2	2	Support least recent	Carlville	2024-03-22	04:30:00	-61.57435141	178.5312529
3	3	Between thus table	Port Rebeccabe…	2024-03-01	18:00:00	38.33354302	157.9579286

Test Case for the finder API as per the assignment :
Input: User's Source Latitude: 40.7128, User's Source Longitude: -74.0060, Search Date: 2024-03-15

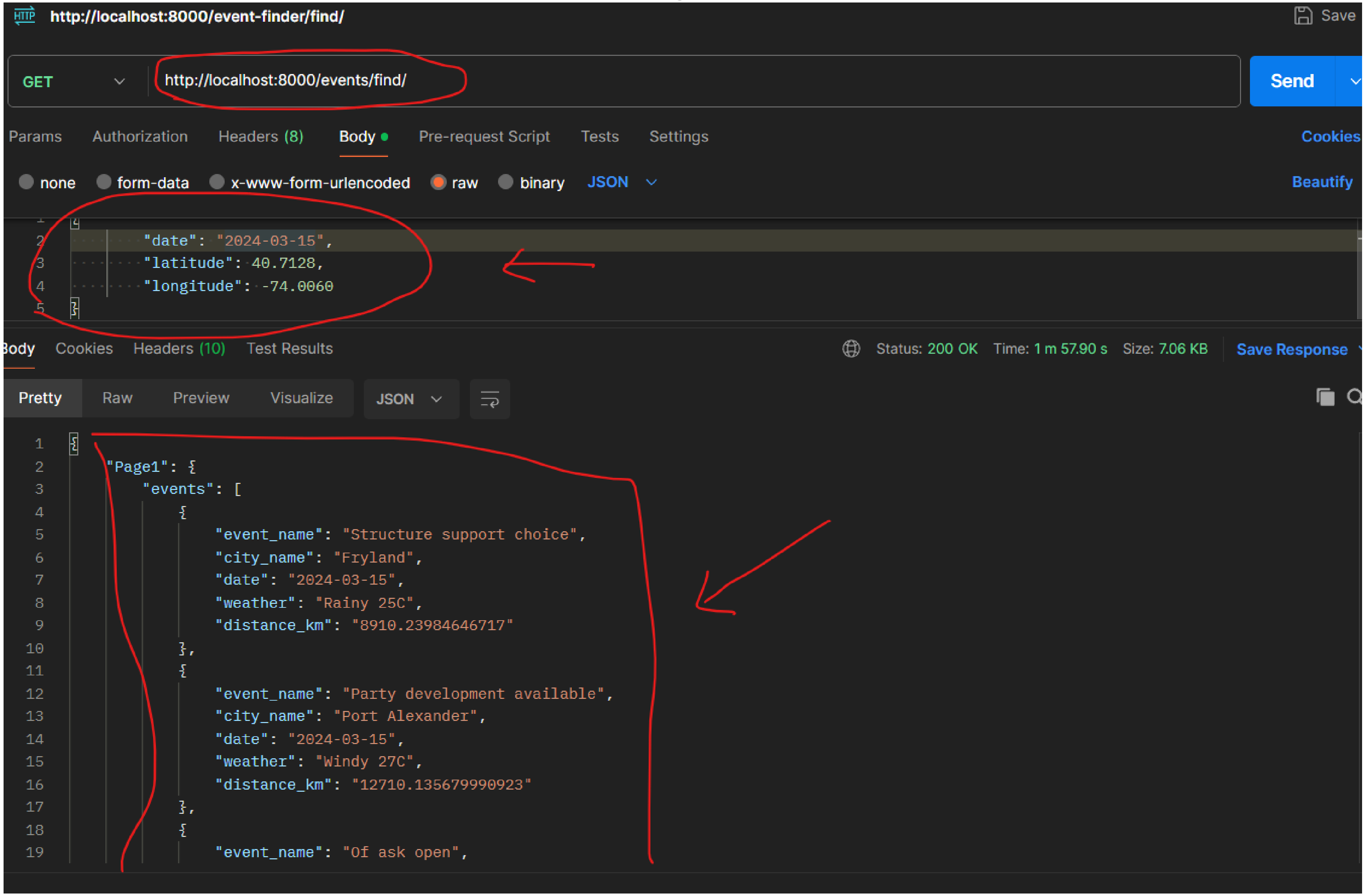


```python
        event_data = [] * 10  # Create a list to store event data for the current page

        # Calculate start and end index for events based on page number
        start_index = page * 10
        end_index = start_index + 10

        # Retrieve events for the current page
        current_events = events[start_index:end_index]

        for event in current_events:
            # Retrieve weather conditions for the event
            weather = get_weather(event.city_name, event.date)

            # Calculate distance between user's location and event location
            distance_km = calculate_distance(user_latitude, user_longitude, event.latitude, event.longitude)

            # Serialize event data
```

```
user_latitude: 38.33354302168567 user_longitude: 157.95792863097148 search_date: 2024-03-01
search_date: 2024-03-01 end_date: 2024-03-15
[03/Apr/2024 11:21:35] "GET /events/find/ HTTP/1.1" 200 7004
user_latitude: 40.7128 user_longitude: -74.006 search_date: 2024-03-15
search_date: 2024-03-15 end_date: 2024-03-29
[03/Apr/2024 11:26:12] "GET /events/find/ HTTP/1.1" 200 6907
```

Output format as per the assignment in (`/events/find` REST API endpoint) :

Page 1



http://localhost:8000/event-finder/find/

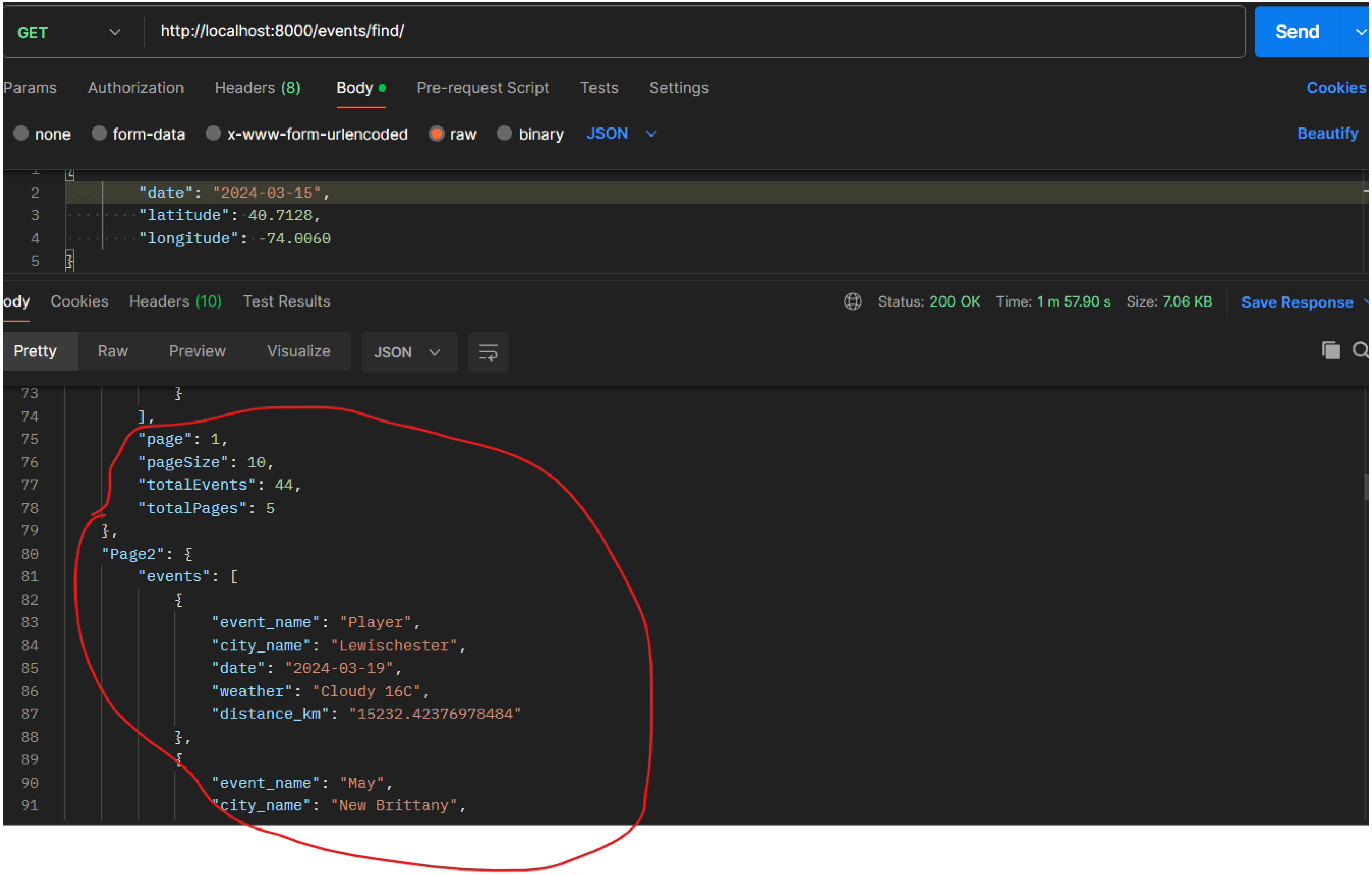
GET http://localhost:8000/events/find/

```
{
    "date": "2024-03-15",
    "latitude": 40.7128,
    "longitude": -74.0060
}
```

Status: 200 OK Time: 1 m 57.90 s Size: 7.06 KB

```
{
"Page1": {
    "events": [
        {
            "event_name": "Structure support choice",
            "city_name": "Fryland",
            "date": "2024-03-15",
            "weather": "Rainy 25C",
            "distance_km": "8910.23984646717"
        },
        {
            "event_name": "Party development available",
            "city_name": "Port Alexander",
            "date": "2024-03-15",
            "weather": "Windy 27C",
            "distance_km": "12710.135679990923"
        },
        {
            "event_name": "Of ask open",
```

PAGE 2



```
http://localhost:8000/events/find/
```

```
{
    "date": "2024-03-15",
    "latitude": 40.7128,
    "longitude": -74.0060
}
```

Status: 200 OK   Time: 1 m 57.90 s   Size: 7.06 KB

```
            }
        ],
        "page": 1,
        "pageSize": 10,
        "totalEvents": 44,
        "totalPages": 5
    },
    "Page2": {
        "events": [
            {
                "event_name": "Player",
                "city_name": "Lewischester",
                "date": "2024-03-19",
                "weather": "Cloudy 16C",
                "distance_km": "15232.42376978484"
            },
            {
                "event_name": "May",
                "city_name": "New Brittany",
```