**Task 04\_HomeTask\_REST\_Automation**

**Task - 1**

**Find the event names which are in English (En) language from below. Verify the list of names returned with expected list of values.**

**Note:** Manually tried to fetch data from the given endpoint **Observation:** User not able to fetch the results from the endpoint (<https://events.epam.com/api/v2/events>). User is getting the error as “could not send request”.

Screenshot:

**A screenshot of a computer

Description automatically generated**

However, tried to proceed with approach by taking dummy API and written the code with only the skeleton part. Since the JSONSLURPER and GPATH are new to me I have taken reference from internet and learned the concept of these two libraries.

**Steps To Approach:**

Step 1: Fetch JSON Response from the API

Firstly, to make a GET request to the provided endpoint and fetch the JSON response.

Step 2: Parse the JSON Response

Use JsonSlurper to parse the JSON response into a Groovy object, which allows to navigate and manipulate the JSON data easily.

Step 3: Filter Event Names in English

Use GPath expressions to filter out the events that are in the English language.

Step 4: Verify the Filtered List Against the Expected Values

Finally, compare the list of English event names with the expected list of values.

**Code Snippet:**

class EventNameVerification {

static void main(String[] args) {

// Step 1: Fetch JSON Response

RestAssured.baseURI = "https://events.epam.com/api/v2";

String response = given()

.when()

.get("/events")

.then()

.statusCode(200) // Ensure the request was successful

.extract().asString(); // Extract the response as a string

// Step 2: Parse JSON Response

def jsonSlurper = new JsonSlurper();

def jsonResponse = jsonSlurper.parseText(response);

// Step 3: Filter Event Names in English

def englishEventNames = jsonResponse.events.findAll {

it.language == "En"

}.collect { it.name };

// Expected list of event names

def expectedEventNames = ["Event 1", "Event 2", "Event 3"]; // Replace with actual expected event names

// Step 4: Verify the filtered list against expected values

assert englishEventNames == expectedEventNames : "Event names do not match!";

println("All English event names match the expected values.");

}

}

**Task - 2**

**Scenario:** openweathermap.org exposes APIs for reading the weather data from various cities. The documentation is at https://openweathermap.org/guide#how.

1. Use the below weather endpoint to get the current weather details of Hyderabad

http://api.openweathermap.org/data/2.5/weather?q=hyderabad&appid={your\_key}

**Obtain an API Key**

To use the OpenWeatherMap API, need to sign up at <https://openweathermap.org/appid> and obtain an API key.

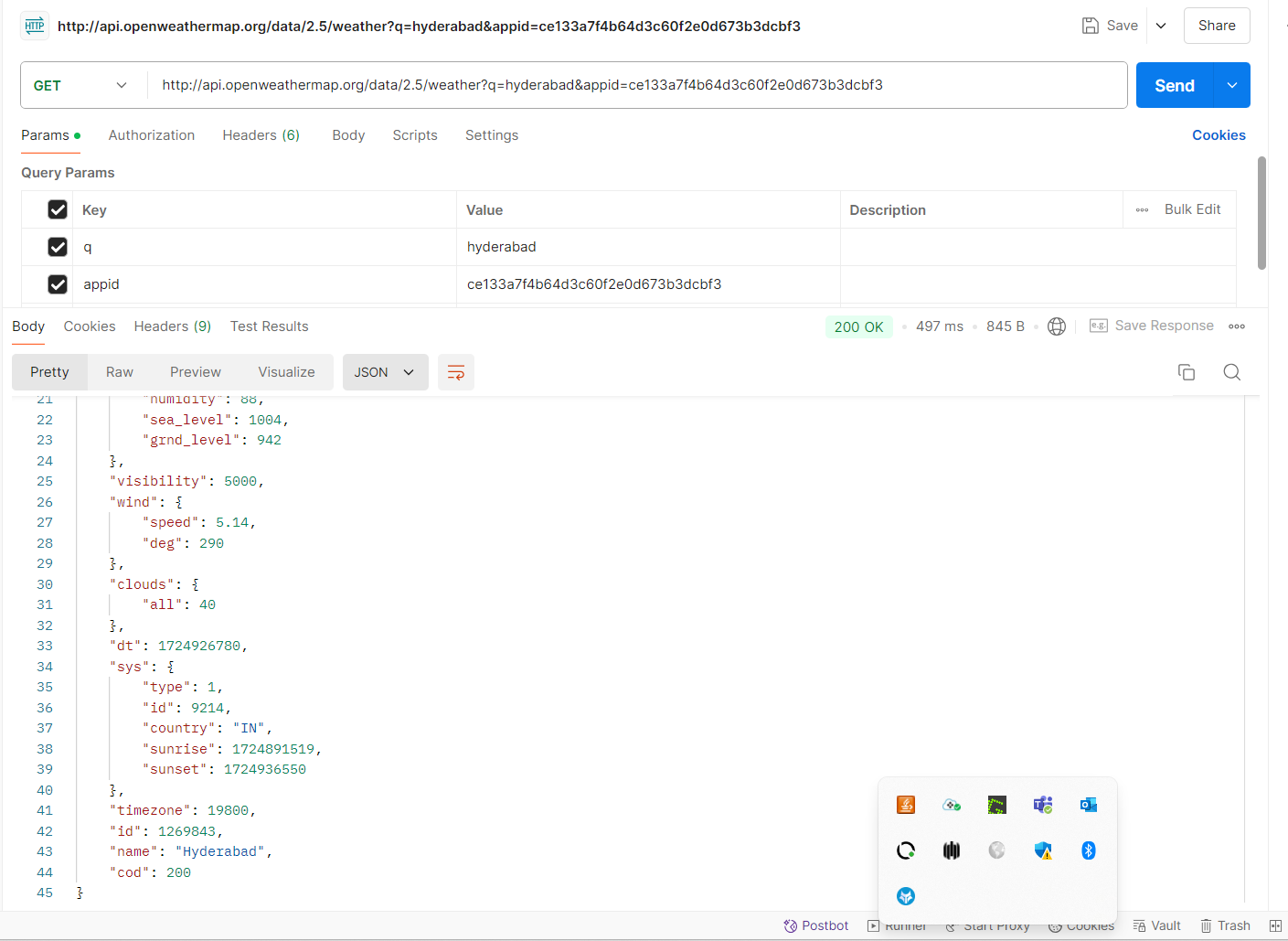
API Key = ce133a7f4b64d3c60f2e0d673b3dcbf3

**Construct the API Endpoint**

End Point: <http://api.openweathermap.org/data/2.5/weather?q=hyderabad&appid=ce133a7f4b64d3c60f2e0d673b3dcbf3>

A screenshot of a computer

Description automatically generated



**Response:**

{

    "coord": {

        "lon": 78.4744,

        "lat": 17.3753

    },

    "weather": [

        {

            "id": 300,

            "main": "Drizzle",

            "description": "light intensity drizzle",

            "icon": "09d"

        }

    ],

    "base": "stations",

    "main": {

        "temp": 298.38,

        "feels\_like": 299.26,

        "temp\_min": 298.38,

        "temp\_max": 304.88,

        "pressure": 1004,

        "humidity": 88,

        "sea\_level": 1004,

        "grnd\_level": 942

    },

    "visibility": 5000,

    "wind": {

        "speed": 5.14,

        "deg": 290

    },

    "clouds": {

        "all": 40

    },

    "dt": 1724926780,

    "sys": {

        "type": 1,

        "id": 9214,

        "country": "IN",

        "sunrise": 1724891519,

        "sunset": 1724936550

    },

    "timezone": 19800,

    "id": 1269843,

    "name": "Hyderabad",

    "cod": 200

}

**API Request Using REST Assured**

public class WeatherAPITest {

public static void main(String[] args) {

// Set the API key

String apiKey = "ce133a7f4b64d3c60f2e0d673b3dcbf3";

// Set the base URI

RestAssured.baseURI = "http://api.openweathermap.org/data/2.5";

// Send the GET request

Response response = RestAssured

.given()

.queryParam("q", "hyderabad")

.queryParam("appid", apiKey)

.when()

.get("/weather")

.then()

.statusCode(200) // Ensure the request was successful

.extract()

.response();

// Print the response

System.out.println("Weather Details of Hyderabad:");

System.out.println(response.prettyPrint());

}

}

**Task – 3**

**Scenario :** Use the coordinates (longitude and latitude) of of the above response to the end-point

http://api.openweathermap.org/data/2.5/weather?lat={latitude}&lon={longitude}&appid={your\_key} and verify the below in response

- name --> Hyderabad

- sys.country --> IN

- main.temp\_min --> greater than 0

- main.temp --> greater than 0

**Fetch the Coordinates from the First API Call**

First to extract the latitude and longitude from the response of the initial API call

**Use the Coordinates to Make the Second API Request**

Once user have the latitude and longitude, user can use them in the second API request to verify the details.

**Using REST Assured:**

import io.restassured.RestAssured;

import io.restassured.response.Response;

import static io.restassured.RestAssured.\*;

import static org.hamcrest.Matchers.\*;

public class WeatherVerificationTest {

public static void main(String[] args) {

// Set API key

String apiKey = "ce133a7f4b64d3c60f2e0d673b3dcbf3";

// Set the base URI

RestAssured.baseURI = "http://api.openweathermap.org/data/2.5";

// Step 1: Fetch the coordinates from the first API call

Response initialResponse = given()

.queryParam("q", "hyderabad")

.queryParam("appid", apiKey)

.when()

.get("/weather")

.then()

.statusCode(200) // Ensure the request was successful

.extract()

.response();

// Extract latitude and longitude from the response

float latitude = initialResponse.path("coord.lat");

float longitude = initialResponse.path("coord.lon");

// Step 2: Use the coordinates to make the second API call

given()

.queryParam("lat", latitude)

.queryParam("lon", longitude)

.queryParam("appid", apiKey)

.when()

.get("/weather")

.then()

.statusCode(200) // Ensure the request was successful

.body("name", equalTo("Hyderabad")) // Verify the city name is Hyderabad

.body("sys.country", equalTo("IN")) // Verify the country is India (IN)

.body("main.temp\_min", greaterThan(0.0f)) // Verify the minimum temperature is greater than 0

.body("main.temp", greaterThan(0.0f)); // Verify the current temperature is greater than 0

}

}

==================================================================