CoinGecko API Testing

Project Name: CoinGecko API Testing
Prepared By: Ashwini Parashuram Saravannavar
Tested By: Ashwini Parashuram Saravannavar
Date: 3/3/2025

---

1. Introduction

This document provides an overview of the API testing methodology, test implementation, and automation approach for testing the CoinGecko API endpoints.

---

2. Objectives

- Validate API responses against expected outputs.

- Ensure API error handling works correctly.

- Verify performance, security, and data consistency.

- Automate test execution using RestAssured + TestNG.

- Generate structured test execution reports.

---

3. Tools & Technologies

- **Programming Language:** Java

- **Frameworks:** TestNG, RestAssured

- **Reporting:** Extent Reports, TestNG Reports, HTML Report

- **Dependency Management:** Maven

- **Version Control:** GitHub

- **Execution Platform:** Postman, Newman, Eclipse

---

4. API Endpoints Tested

Endpoint	Description
/api/v3/coins/markets	Fetch cryptocurrency market data

---

5. Test Cases

Added in **Test_Plan_CoinGecko** excel file with the Sheet name as "**TestCase**"

---

6. Test Implementation

6.1 Setup & Configuration

- **Base URL:** Defined in config.properties.
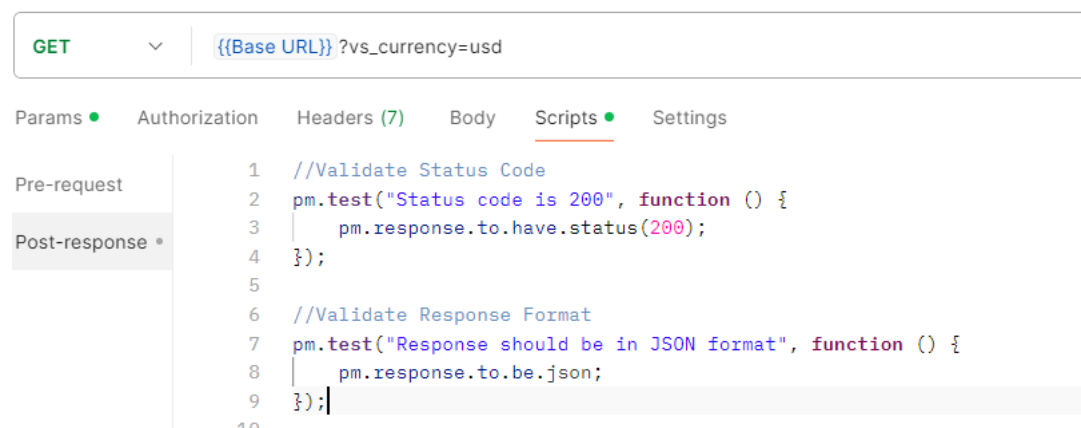
- **Endpoints:** Retrieved from properties file.

6.2 Sample Test Script

RestAssured + TestNG (Java-Based Automation)

```java
@Test(description = "Test Case 1: Validate Status Code", priority = 1)
Run | Debug
public void verifyAPIStatusCode()
{
    Response response = given()
            .queryParam("vs_currency", "usd")
            .when()
            .get(javaUtils.propertyData("endpoint"))
            .then()
            .contentType(ContentType.JSON)
            .extract()
            .response();

        Assert.assertEquals(response.getStatusCode(), 200, "Status code is not 200");
        test.pass("Status code is 200");
        test.info("API response time: " + response.getTime() + "ms");
}
```

Postman Test Script (JavaScript in Postman)

```
GET    ∨    {{Base URL}} ?vs_currency=usd

Params ●   Authorization   Headers (7)   Body   Scripts ●   Settings

Pre-request
                1   //Validate Status Code
                2   pm.test("Status code is 200", function () {
Post-response ●  3       pm.response.to.have.status(200);
                4   });
                5
                6   //Validate Response Format
                7   pm.test("Response should be in JSON format", function () {
                8       pm.response.to.be.json;
                9   });
                10
```

6.3 Test Scripts Execution

- **TestNG Execution (Java-based test scripts):**

Run the test scripts using **Run ALL** option in TestNG class

```
    Run All
13  public class StatusCodeAndFormatTest extends BaseClass
14  {
15⊖      @Test(description = "TC_001: Validate Status Code", priority = 1)
        Run | Debug
16      public void verifyAPIStatusCode()
```

- **Postman Collection Execution using Newman (Command Line Tool):**

```
C:\Users\hp>newman run "C:\Users\hp\Desktop\QSpiders\Software Testing Projects\APITesting.postman_collection.json" -e "C
:\Users\hp\Desktop\QSpiders\Software Testing Projects\CoinGeckoEnv.postman_environment.json"
```

7. Reporting & Logs

- **Extent Reports:** Generates structured reports with test results and logs.

- **TestNG Reports:** Provides summary and detailed execution results.

- **Newman Reports (Postman Tests):** Newman generates CLI logs and an HTML report of Postman API test execution.

8. Issues & Improvements

Identified Issues:

Postman Test Scripts:-

1. Data doesn't get updated within the last 60 seconds.
2. 429 Too Many Requests occurs sometimes for some test cases and sometimes doesn't.

RestAssured + TestNG Test Scripts:-

1. API Response time exceeds 500ms in some cases.
2. 429 Too Many Requests occurs sometimes for some test cases and sometimes doesn't.

Proposed Improvements:

- Reduce response time using caching mechanisms.

9. Conclusion

The API testing framework validates CoinGecko's API endpoints with automated scripts, structured reporting, and issue tracking. Identified issues will be reported for further improvements.