Report

1. Introduction

- Purpose of the study.

This task study is designed to test the CoinGecko API to ensure its functionality, reliability, and performance through both automated and manual testing methods. The aim is to validate the API endpoints, response times, data accuracy, and error handling capabilities.

- Overview of CoinGecko API.

CoinGecko API offers the most comprehensive and reliable crypto market data through RESTful JSON endpoints

2. Test Methodology

- Manual, Functional, Automation testing is done using Postman. Assertion are applied in postman script and Jmeter.
- Performance Testing is done using Jmeter

3. Test Execution - Summary and Observations from manual tests.

*Test cases are in excel sheet.
*Newman html report is attached.

Command to run in newman and generate html report : `newman run ApiCollection.json -e coinGecko.postman_environment.json -r html CoinGecko`

```
CoinGecko API
→ Ping: GET /ping
  GET https://api.coingecko.com/api/v3/ping [200 OK, 1.37kB, 1153ms]
  ✓  Status code is 200
  ✓  Body is correct
→ Simple Price: GET /simple/price
  GET https://api.coingecko.com/api/v3/simple/price?ids=bitcoin&vs_currencies=usd [200 OK, 1.1kB, 1056ms]
  ✓  Status code is 200
  ✓  Body matches string
  ✓  Body matches string
→ Coins List: `GET /coins/list
  GET https://api.coingecko.com/api/v3/coins/list?include_platform=false [200 OK, 869.43kB, 1709ms]
  ✓  Status code is 200
  ✓  Body matches string
  ✓  Body matches string
  ✓  Body matches string
→ Coin Markets: GET /coins/markets
  GET https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd [200 OK, 81.68kB, 1084ms]
  ✓  Status code is 200
  ✓  Body matches string
  ✓  Body matches string
  ✓  Body matches string
→ Coin Data: GET /coins/{id}
  GET https://api.coingecko.com/api/v3/coins/bitcoin [200 OK, 99.37kB, 1114ms]
  ✓  Status code is 200
  ✓  Check if id is 'bitcoin'
  ✓  Check if symbol is 'btc'
  ✓  Check if name is 'btc'
┌─────────────────────────┬───────────────────┬──────────────────┐
│                         │          executed │           failed │
├─────────────────────────┼───────────────────┼──────────────────┤
│              iterations │                 1 │                0 │
├─────────────────────────┼───────────────────┼──────────────────┤
│                requests │                 5 │                0 │
├─────────────────────────┼───────────────────┼──────────────────┤
│            test-scripts │                 5 │                0 │
├─────────────────────────┼───────────────────┼──────────────────┤
│      prerequest-scripts │                 0 │                0 │
├─────────────────────────┼───────────────────┼──────────────────┤
│              assertions │                17 │                0 │
├─────────────────────────┴───────────────────┴──────────────────┤
│ total run duration: 6.3s                                       │
├────────────────────────────────────────────────────────────────┤
│ total data received: 1.05MB (approx)                           │
├────────────────────────────────────────────────────────────────┤
│ average response time: 1223ms [min: 1056ms, max: 1709ms, s.d.: 245ms] │
└────────────────────────────────────────────────────────────────┘
```

Test case 1. Ping Endpoint

- Test ID: TC001
- Description: Verify that the Ping endpoint returns a successful response. - Method: Automated/Manual
- Expected Result: `{"gecko_says": "(V3) To the Moon!"}`

Observation : Ping API typically requires an API key or token for authentication and authorization. This is to ensure secure access and to track usage.

Test case 2. Simple Price Endpoint

- Test ID: TC002
- Description: Retrieve the current price of Bitcoin in USD. - Method: Automated/Manual
- Expected Result: `{"bitcoin": {"usd": value}}`

Observation :

1. There should be JWT Token / Session Token in Header for authorization and to make it secure.

2. If invalid value is passed in ids and vs_currencies then it should give a proper error message with error code 400. Currently it’s giving 200 which is a success response.
3. If i pass two ids and two precision like in below URL, the it’s giving the result for first id and applying second precision and vice versa.

https://api.coingecko.com/api/v3/simple/price?ids=bitcoin&ids=ethercoin&vs_currencies=usd,inr&precision=full&precision=5

Test Case 3. Coins List Endpoint

- Test ID: TC003
- Description: Fetch the list of all available coins.
- Method: Automated/Manual
- Expected Result: List of coin objects with `id`, `symbol`, and `name`.

Observation :

1. If i give value of include_platform = abcd or blank, then also it’s giving me success response 200 ok. Here it should give error because it’s a boolean field and should accept only true or false.
2. There should be JWT Token / Session Token in Header for authorization and to make it secure.

Test case 4. Coin Markets Endpoint

- Test ID: TC004
- Description: Get market data for a specific coin.
- Method: Automated/Manual
- Expected Result: Market data including `id`, `symbol`, `current_price`, etc.

Observation :

1. There should be JWT Token / Session Token in Header for authorization and to make it secure.

Test case 5.Coin Data Endpoint

- Test ID: TC005
- Description: Retrieve detailed data for Bitcoin.

- Method: Automated/Manual
- Expected Result: Detailed data object for Bitcoin.

Observation :

1. There should be some limit and offset in query params because the response is very big and it can have performance issues for GET api.
2. There should be JWT Token / Session Token in Header for authorization and to make it secure.

4. Performance Analysis , Load testing results, Stress testing results - Jmeter script attached.

Not able to do much Performance and Stress testing due to Rate limit on APIs. I asked for API key in email but did not get any reply regarding that. Was getting below error message : `"error_message": "You've exceeded the Rate Limit. Please visit https://www.coingecko.com/en/api/pricing to subscribe to our API plans for higher rate limits."`

5. Security Analysis (Not able to do must testing because i don't have exposure in this as there were separate Security testing teams in my every previous organization)

1. While testing through postman, i saw that in Headers of the API, the Cookies was having SameSite=None. This can expose application to Cross-Site Request Forgery (CSRF) attacks, where malicious websites can perform actions on behalf of the user without their consent.
2. There are some cases in which Error 500 was throwing, in such cases custom error should show.

6. Conclusion

- Recommendations for improvements - Some APIs have undocumented behavior and should have token in Headers

7. Appendices
- Detailed test case documentation. - Attached in Excel sheet

Run your tests and validations on CI/CD using Postman CLI configuration

```
# Jenkins runs pipelines on the host system that it is setup on. To run it on mac
# install the server on a mac machine by following https://www.jenkins.io/doc/book/installing/macos/
# once it has been setup add the below step to your pipeline file to run automated tests using Postman
CLI.
pipeline {
  agent any
  tools {nodejs "{your_nodejs_configured_tool_name}"}
  stages {
    stage('Install Postman CLI') {
      steps {
        sh 'curl -o- "https://dl-cli.pstmn.io/install/osx_64.sh" | sh'
      }
    }
    stage('Postman CLI Login') {
      steps {
        sh 'postman login --with-api-key $POSTMAN_API_KEY'
      }
    }
    stage('Running collection') {
      steps {
        sh 'postman collection run "31298752-9155bbaa-fd0d-4492-ad4a-a64a016fa69a" -e
"31298752-eb893bb6-64d0-4e53-8bff-c09d87ca93d7"'
      }
    }
  }
}
```