Table Of Contents

Objective

The test plan is designed to validate the REST endpoint GET https://images-api.nasa.gov/search with query parameters “q”, “media_type”, “start_year” and “end_year” based on the requirements stated in the api document https://images.nasa.gov/docs/images.nasa.gov_api_docs.pdf

Scope

Testing the GET https://images-api.nasa.gov/search with query parameters “q”, “media_type”, “start_year” and “end_year”.

Test Strategy

REST EndPoints

1. GET https://images-api.nasa.gov/search
2. GET https://images-api.nasa.gov/search?q=<no text | null>
3. GET https://images-api.nasa.gov/search?q=<valid|invalid text>
4. GET https://images-api.nasa.gov/search?media_type=<audio|image|video>

5. GET https://images-api.nasa.gov/search?q=<text>&year_start=<year between 1921 to 2021>
6. GET https://images-api.nasa.gov/search?q=<text>&year_end=<year between 1921 to 2021>
7. GET https://images-api.nasa.gov/search?q=<text>&media_type=<media>&start_year=<date>end_year=<date>

Functional Testing

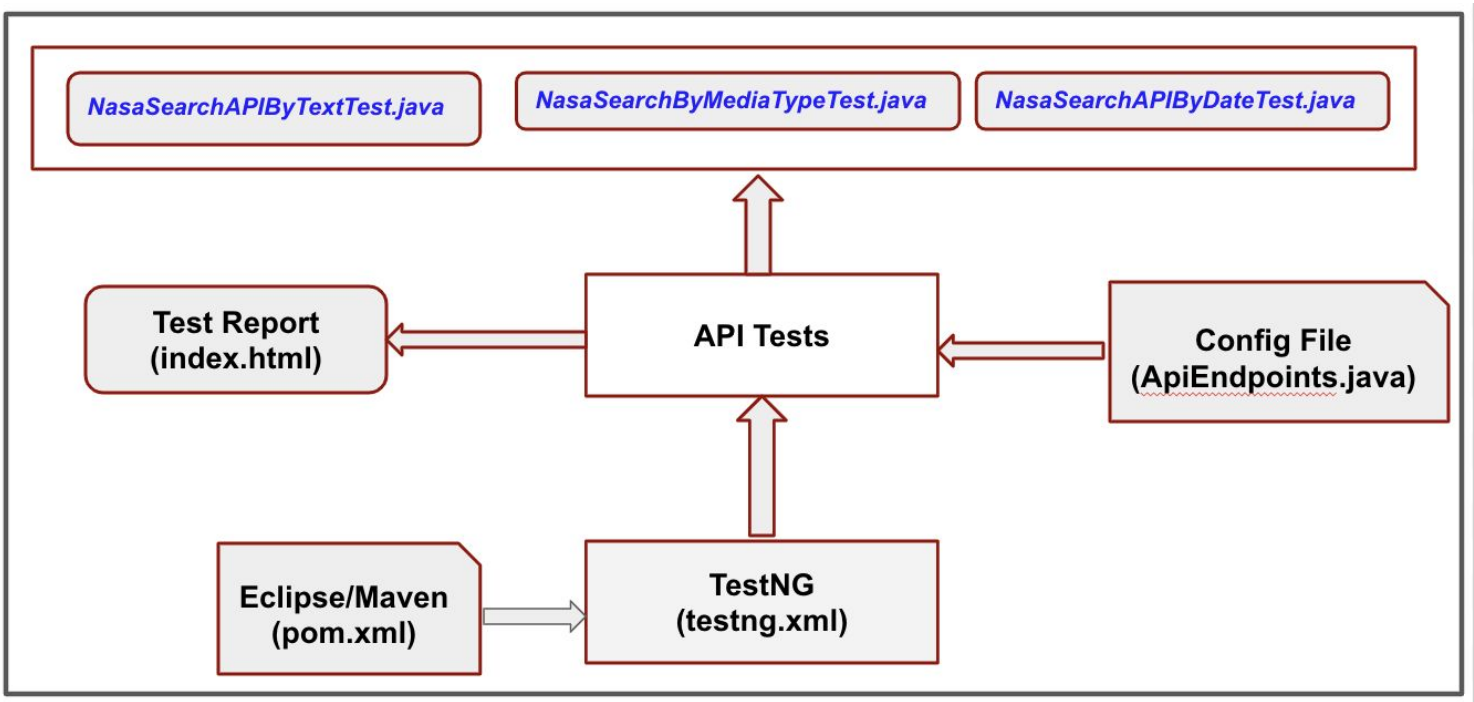
Test Cases

Test case	Verification	Expected
Search with optional q parameter		
/search?q=<text in lowercase>	1. Verify the elements href, items, version, metadata, links returned under collection 2. Verify the sub elements of the elements 3. Verify the links in href are active 4. Verify the pagination: links should show the link for next page and should not display any link for previous page 5. Verify the arrays for items return 100 items/page 6. Verify response headers	200, OK. Collection should return the records requested
/search?q=<text in uppercase>		200, OK. Collection should return the records requested
/search?q=<text in mixedcase>		200, OK. Collection should return the records requested
/search?q=<invalid text>		200, OK. Zero records
/search?q=<text with special characters>	test with each special character to check the backend logic handles the special characters correctly	Expected: 200, OK when supported special characters are sent. Collection should return the records when match found, else should return empty items array. **Actual**: **200 OK, returned records for the special characters ` ! @ $ % ^ * ( ) { [ ] “ ‘ \ : < >** **200 OK, empty items array for the special characters ~ - _ + } | . ? /**

		400 Bad Request for the special characters # & ; ,
/search?q=<text with max characters>	Boundary condition check for max limit on the search text	200, OK. Collection should return the records requested
/search?q=<text with greater than max characters>	Boundary condition check for greater than max limit on the search text	400, Bad Request.
/search	Check for optional parameter missing	400, Bad Request.
/search?q =<null>	Check for null	400, Bad Request.
/search?q=<empty string>	Check for empty string	400, Bad Request.
Search with optional media_type parameter		
/search?media_type<supported media type in lower case> eg: image	Verify the records under data array has media_type = image and does not include audio/video	200, OK
/search?media_type<supported media type in upper case> eg: IMAGE	Verify the records under data array has media_type = image and does not include audio/video	Expected: 200 OK with records displayed **Actual: 200 OK, with zero records**
/search?media_type<supported media type in mixed case> eg: IMAGE	Verify the records under data array has media_type = image and does not include audio/video	Expected: 200 OK with records displayed **Actual: 200 OK, with zero records**
/search?media_type<multiple supported media types> eg: image, audio	Verify the records under data array has media_type = image and audio and does not include video	200, OK. Returns records for image and audio
/search?media_type =<null>	Check for null	400, Bad Request.
/search?media_type =<empty string>	Check for empty string	400, Bad Request.
Test the parameters (q, media_type, year_start, year_end) combinations		
/search?q=apollo&media_type=video&year_start=2010	Verify the listing of records after 2010 Verify the date creation for the media is listed correctly	200, OK. should display records dated after the year 2010
/search?q=apollo&media_type=video&year_start=2010&year_end=2015	Verify the listing of records within the year range 2010 - 2015 Verify the date creation for the media is listed correctly	200, OK. should display records dated between the year 2010-2015

/search?q=apollo&media_type=video&year_end=2015	Verify the listing of records till the year 2015 Verify the date creation for the media is listed correctly	200, OK. should display records dated till the year 2015
/search?q=apollo&media_type=video&year_start=2025	Verify future year is accepted	200, OK. Zero records displayed
/search?q=apollo&media_type=video&year_end=1900	Verify past year is accepted	200, OK. Zero records displayed
/search?q=apollo&media_type=video&year_end=0000	Verify invalid year is not accepted	400, Bad Request
Test for pagination		
https://images-api.nasa.gov/search?q=mars&page=3	Check the links array contain link for previous page(page 2) and link for next page(page 4)	200, OK. Should display links for next and previous pages
https://images-api.nasa.gov/search?q=mars&page=101	Check for max page size limit	**Expected: 200 Ok, with no link to next page** **Actual: 400, Bad Request**
https://images-api.nasa.gov/search?q=mars&page=1	Check the links array contain link for next page(page 2) and link for previous page should not be displayed	200, OK with link for next page should be displayed
Test for API limit		
Hit the api endpoint <max api limit times>		200, OK
Hit the api endpoint <greater than max api limit times>		429, rate limit exceeded
Hit the api endpoint <lesser than max api limit times>		200, OK
Test for 5xx errors		
Hit /search endpoint	Test the endpoint when db/system is down	500/503, Internal Server Error
Test for 404		
Hit the endpoint as /search ?q=apollo	Verify space between the base url and the root path is not accepted	404, The resource could not be found.
Hit the endpoint as /search1?q=apollo		404, The resource could not be foun

Automation RestAssured Test Framework



Automation Testing

Automation test cases were added for the following functional scenarios.

NasaSearchAPIByTextTest

1. *happy path - search by valid text in lowercase*
2. *happy path - search by valid text in uppercase*
3. *happy path - search by valid text in mixedcase*
4. *search by invalid text*
5. *search text is null*
6. *search text is empty string*

NasaSearchByMediaTypeTest

1. *happy path - search by valid media type in lowercase*
2. *happy path - search by valid media type in uppercase*
3. *happy path - search by valid media type in mixedcase*
4. *media type invalid text*
5. *media type is null*
6. *media type is empty string*

NasaSearchAPIByDateTest

1. happy path - test for optional parameters combinations
2. test for optional parameters combinations and year_start is invalid
3. test for optional parameters combinations and year_end is invalid
4. test for optional parameters combinations and year_start is in future

Performance Testing

1. Verify response time when accessing different pages with max records.

2. Verify response time when accessing same page by concurrent users

Security Testing

1. Verify the endpoint can be accessed with no headers and authorization sent in request
2. Verify the response headers for the below set (verification for cross origin resource sharing)
   `Access-Control-Allow-Origin: *`
   `Access-Control-Allow-Headers: Origin,Content-Type,Accept,Authorization,X-Requested-With`
   `Access-Control-Allow-Methods:GET`
3. SQL injection and verify the response

Issues found / ambiguities

1. When sending null or empty string in the optional parameters q or media_type, the response codes returned are inconsistent
   https://images-api.nasa.gov/search?media_type=<null | empty string> returns 200. Expected: 400.
   https://images-api.nasa.gov/search?q=<null | empty string> returns 400, Expected: 400

2. https://images-api.nasa.gov/search?q=<special characters> When sending special characters, only few characters are blocked and others fetch the records successfully.
   200 OK, returned records for the special characters ` ! @ $ % ^ * ( ) { [ ] “ ‘ \ : < >
   200 OK, empty items array for the special characters ~ - _ + } | . ? /
   400 Bad Request for the special characters # & ; ,

3. Information on supported vs unsupported special characters is not stated in the api document
4. Max limit of characters for search pattern string for q parameter is not stated in the api document
5. API Document states the release version as 1.5.3 but api response returns version as 1.0.


```
{
   "collection": {
       "metadata": {
           "total_hits": 1
       },
       "version": "1.0",
       "items": [
           {
```


6. media_type in upper case is returning zero records.
7. Information on api limit is not stated in the api documentation
8. Limitation on date range year_start and year_end is not stated in the api document.
   Hitting endpoint https://images-api.nasa.gov/search?q=mars&year_start=<2022-2027> is returning 200.
   Hitting endpoint https://images-api.nasa.gov/search?q=mars&media_type=video&year_start=2026 returns 200 and the records display date created as future date


```
"items": [
           {
               "data": [
```

```
{
    "date_created": "2027-12-16T00:00:00Z",
    "center": "KSC",
    "title": "Greenhouse in Antarctica Helping Astronauts on Long-Duration
Missions",
    "media_type": "video",
    "nasa_id": "Antarctica Eden ISS",
    "photographer": "NASA/Francisco Martin",
    "keywords": [
        "EDEN ISS",
        "Meumayer II Station",
        "Antarctica"
    ],
```


Test Environment

Automation Testing: Automation framework built with Java, TestNG, Maven, RestAssured
Manual Testing: Postman
Performance Testing: jmeter

Future enhancements to automation framework:

1. Add utility to verify the broken links
2. Add utility to generate custom report
3. Add utility to parse json objects
4. Add utility to generate and extract date, year
5. Add utility to generate random valid and invalid texts for search patterns