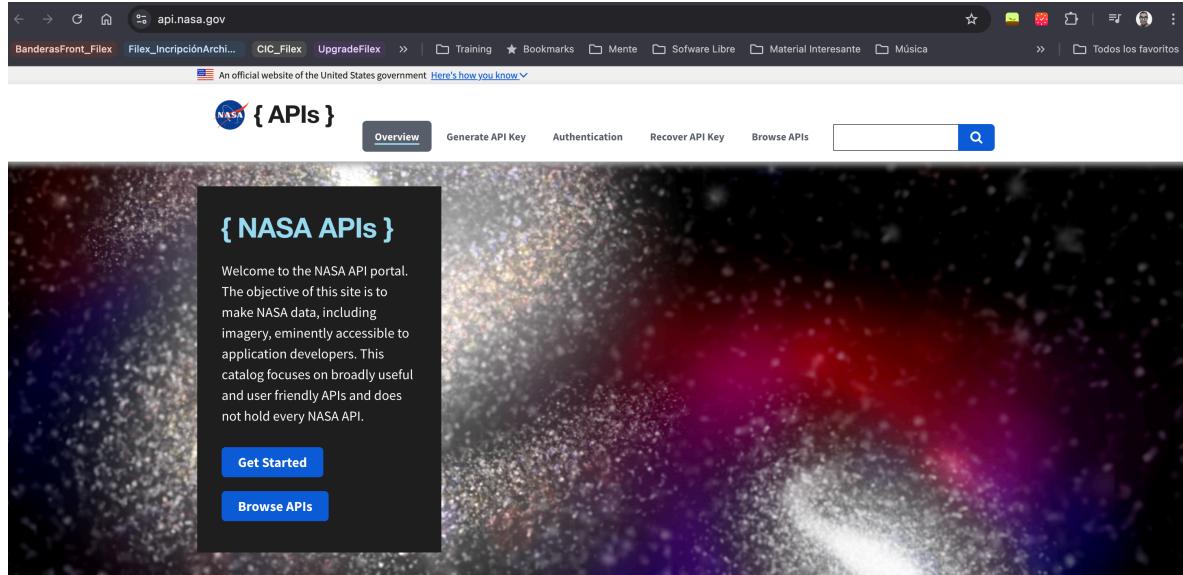


Evidencias Taller 3 – Reto 2

Generación API Key en el sitio web de la NASA <https://api.nasa.gov/>

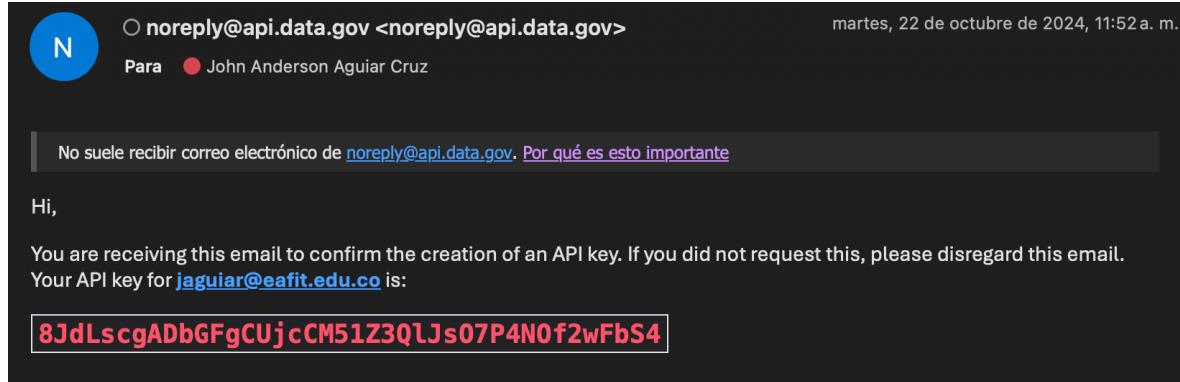


Generate API Key

A screenshot of the "Generate API Key" form on the NASA APIs website. The title "Generate API Key" is at the top. Below it, a note says "Required fields are marked with an asterisk (*)." There are four input fields: "First Name" with value "Anderson", "Last Name" with value "Aguiar", "Email" with value "jaguiar@eafit.edu.co", and a "How will you use the APIs? (optional)" field containing "Educational topics". A "Sign up" button is at the bottom. The form is protected by reCAPTCHA.

This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply.

Una vez diligenciamos el formulario, llega al correo el API Key proporcionado por la NASA, el cual se usará a nivel del header Authorization y en algunos request como queryParam en el GET.



Luego de la generación del API Key, comenzamos la generación de los request en Postman que pide el reto 2. Realizaremos 2 requests del API <>TechPort<> y del API <> Asteroids - NeoWs <> para un total de 4 request.

Cabe anotar que las API de la NASA solo permite el verbo HTTP GET. Adicionalmente, solo se generarán variables de ambiente, no se generarán ambientes (Environments) porque solo existe un solo ambiente para ejecutar las pruebas. Conforme a esto, se generan las variables globales a utilizar en cada request:

Variable	Type	Initial value	Current value
domainTechport	default	https://techport.nasa.gov	https://techport.nasa.gov
api_key	default		8JdLscgADbGFgCUjcCM51Z3QlJs07P4N0f2wFbS4
start_date	default		2024-10-17
end_date	default		2024-10-24
update_since_date	default		2024-07-26
domainNeo	default	https://api.nasa.gov	https://api.nasa.gov

A continuación, las evidencias de cada prueba:

TechPort

TechPort es el inventario de tecnología de la NASA, que muestra la cartera de proyectos tecnológicos activos y finalizados de la NASA. Una vez seleccionamos el

proyecto, se abre la documentación en Swagger del API en el sitio web <https://techport.nasa.gov/help/articles/api>.

The screenshot shows a browser window with the URL <https://techport.nasa.gov/help/articles/api>. The page title is "Using the TechPort API". The content area starts with a bold heading: "The NASA TechPort system provides a RESTful web services API to make technology project data available to other systems and services. This API can be used to export TechPort data into JSON format, which can be further processed and analyzed." Below this, another bold heading states: "Complete documentation of the available objects, properties, and RESTful URLs is available in the documentation at the bottom of this page." To the right, there is a "Glossary" section with definitions for API, JSON, and REST. The "API" definition is: "Application Programming Interface – a set of rules governing how different applications can communicate with each other. TechPort's APIs give outside systems access to the same functions and data structures used to create techport.nasa.gov." The "JSON" definition is: "JavaScript Object Notation – an open standard file and data interchange format that uses attribute-value pairs, and array data types, to represent structured data." The "REST" definition is: "Representational State Transfer – an architectural style that describes how distributed systems expose an interface. When people use the term 'REST API,' they are usually referring to an API accessed via HTTP, at a predefined set of URLs." At the bottom of the main content area, there is a section titled "Start with a token" containing instructions about API tokens.

Get projects: request que me entrega los proyectos creados y finalizados de la NASA. Se debe parametrizar una fecha de inicio, la cual se tomará desde una variable global.

The screenshot shows the Postman API client interface. The left sidebar shows collections like "Topics_Eafit" and "Globals". The main area shows a "GET Get Projects" request for the URL `(domainTechport) /api/projects?updatedSince={{update_since_date}}`. The "Params" tab shows two parameters: "Key" and "updatedSince" with the value `{{update_since_date}}`. The "Body" tab shows the response in JSON format, which is a list of projects. The response body is as follows:

```
1 {
2     "projects": [
3         {
4             "acronym": "",
5             "projectId": 116315,
6             "title": "",
7             "website": "",
8             "lastUpdated": "2024-10-11"
9         },
10        {
11            "acronym": "",
12            "projectId": 157205,
13            "title": "",
14            "website": "",
15            "lastUpdated": "2024-10-10"
16        }
17    ]
}
```

The screenshot shows the Postman interface with a collection named 'Topicos_Eafit'. A 'GET' request is made to 'Taller3Topics_Reto2 / Techport API | Get Projects'. The 'Pre-request' script contains JavaScript code to calculate a date 7 days ago and set it as a global variable 'api_key'. The 'Post-response' script sets an environment variable 'update_since_date' to the calculated date. The response body is a JSON array of project objects, each with fields like 'acronym', 'projectId', 'title', 'website', and 'lastUpdated'. The status bar at the bottom indicates a 200 OK response with 376 ms latency and 14.78 KB size.

Project information: request que me entrega la información detallada del proyecto proporcionando un ID de proyecto.

The screenshot shows the Postman interface with a collection named 'Topicos_Eafit'. A 'GET' request is made to 'Taller3Topics_Reto2 / Techport API | Project information'. The 'Params' tab is selected, showing a query parameter 'Key' with the value '116315'. The response body is a detailed JSON object for a specific project, including fields like 'project', 'acronym', 'projectId', 'title', 'parentNodeId', 'level', 'code', 'definition', and 'exampleTechnologies'. The status bar at the bottom indicates a 200 OK response with 176 ms latency and 9.33 KB size.

Se evalúa que el resultado sea un HTTP 200:

The screenshot shows the Postman interface with the following details:

- Collection:** Topics_Eafit
- Environment:** Taller3Topics_Reto2 / Techport API
- Request Type:** GET
- URL:** {{domainTechport}}/api/projects/116315
- Pre-request Script:**

```
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
});
```
- Post-response Script:**

```
pm.setEnvironmentVariable("ProjectID", pm.response.json().project.id);
```
- Body:** JSON response (Pretty)

```
{
  "project": {
    "acronym": "SQRL",
    "projectId": 116315,
    "title": "High IMU Rover Lidar",
    "primaryTaxonomyNodes": [
      {
        "taxonomicNodeId": 11266,
        "taxonomicRootId": 8817,
        "parentNodeId": 11265,
        "level": 3,
        "code": "TX18.1.1",
        "title": "Sensing and Perception for Autonomous Systems",
        "definition": "Sensing and perception technologies for autonomous systems collect and process information internal and external to the system from sensors and instruments. Perception can be done at the sensor or at a centralized computer.",
        "exampleTechnologies": "Three dimensional (3D) sensing and perception from stereo vision or light detection and ranging (LIDAR), force and tactile sensing, science-instrument sensing (e.g. spectrometers) that is eventually used in decision-making, tools that assess data"
      }
    ]
  }
}
```
- Response Status:** 200 OK
- Time:** 176 ms
- Size:** 9.33 KB

Asteroids – NeoWs

API que permite obtener información sobre asteroides cercanos a la Tierra. El API permite buscar asteroides en función de su fecha de aproximación más cercana a la Tierra, buscar un asteroide específico con su identificador, así como explorar el conjunto de datos general.

Neo feed: request que recupera una lista de asteroides en función de su fecha de aproximación más cercana a la Tierra.

The screenshot shows the Postman interface with the following details:

- Collection:** Topics_Eafit
- Environment:** Taller3Topics_Reto2 / Asteroids - NeoWs
- Request Type:** GET
- URL:** {{domainNeo}}/neo/rest/v1/feed?start_date={{start_date}}&end_date={{end_date}}&api_key={{api_key}}
- Params:**

Key	Value	Description
start_date	({{start_date}})	
end_date	({{end_date}})	
api_key	({{api_key}})	
- Body:** JSON response (Pretty)

```
{
  "element_count": 158,
  "near_earth_objects": {
    "2024-10-20": [
      {
        "links": [
          {
            "self": "http://api.nasa.gov/neo/rest/v1/neo/2412977?api_key=83d1sg40gGf8UjicM51Z3Q1ls07P4N0f2wFh54"
          }
        ],
        "id": "2412977",
        "neo_reference_id": "2412977",
        "name": "2024 SDP 1099",
        "nasa_url": "https://ssd.jpl.nasa.gov/tools/sbdb_lookup.html?&sst=2412977",
        "absolute_magnitude_h": 19.57,
        "estimated_diameter": {
          "kilometers": {
            "estimated_diameter_min": 0.3240074354,
            "estimated_diameter_max": 0.7245026508
          }
        }
      }
    ]
  }
}
```
- Response Status:** 200 OK
- Time:** 1477 ms
- Size:** 36.22 KB

The screenshot shows the Postman interface with the following details:

- Collection:** Topics_Eafit
- Environment:** Taller3Topics_Reto2 / Asteroids - NeoWs
- Request Method:** GET
- URL:** `((domainNeo))/neo/rest/v1/feed?start_date=((start_date))&end_date=((end_date))&api_key=((api_key))`
- Pre-request Script:**

```

1 // Obtener la fecha actual
2 const today = new Date();
3 const todayFormatted = today.toISOString().split('T')[0];
4
5 // Obtener la fecha de hace 7 días
6 const sevenDaysAgo = new Date();
7 sevenDaysAgo.setDate(today.getDate() - 7);
8 const sevenDaysAgoFormatted = sevenDaysAgo.toISOString().split('T')[0];
9
10
11 //pm.globals.set("api_key", "8JdLscgADbGfgCUjcCM51Z3Q1J3s07P4N0f2wFbS4");
12 pm.globals.set("start_date", sevenDaysAgoFormatted);
13 pm.globals.set("end_date", todayFormatted);

```
- Body:** Cookies (2) Headers (24) Test Results
- Response Status:** 200 OK
- Response Body (Pretty JSON):**

```

6     },
7     "element_count": 198,
8     "near_earth_objects": {
9       "2024-10-20": [
10         {
11           "links": [
12             {
13               "self": "http://api.nasa.gov/neo/rest/v1/neo/2412977?api_key=8JdLscgADbGfgCUjcCM51Z3Q1J3s07P4N0f2wFbS4"
14             },
15             {
16               "id": "2412977",
17               "neo_reference_id": "2412977",
18               "name": "A12977 (1996 UO)",
19               "nasa_jpl_url": "https://ssd.jpl.nasa.gov/tools/sbdb_lookup.html#?sstr=2412977",
20               "absolute_magnitude_h": 19.57,
21               "estimated_diameter": {
22                 "kilometers": [
23                   {
24                     "estimated_diameter_min": 0.3240974354,
25                     "estimated_diameter_max": 0.7245026508
26                   }
27                 ]
28               }
29             }
30           ]
31         }
32       }
33     }
34   }
35 }
```

Neo Lookup: request que permite conocer la información específica de un asteroide según el ID de la NASA.

The screenshot shows the Postman interface with the following details:

- Collection:** Topics_Eafit
- Environment:** Taller3Topics_Reto2 / Asteroids - NeoWs
- Request Method:** GET
- URL:** `((domainNeo))/neo/rest/v1/neo/2412977?api_key=((api_key))`
- Pre-request Script:**

```

1 Use JavaScript to write tests, visualize response, and more.

```
- Body:** Cookies (2) Headers (24) Test Results
- Response Status:** 200 OK
- Response Body (Pretty JSON):**

```

1 {
2   "links": [
3     {
4       "self": "http://api.nasa.gov/neo/rest/v1/neo/2412977?api_key=8JdLscgADbGfgCUjcCM51Z3Q1J3s07P4N0f2wFbS4"
5     }
6   ],
7   "id": "2412977",
8   "neo_reference_id": "2412977",
9   "name": "A12977 (1996 UO)",
10  "designation": "2412977",
11  "nasa_jpl_url": "https://ssd.jpl.nasa.gov/tools/sbdb_lookup.html#?sstr=2412977",
12  "absolute_magnitude_h": 19.57,
13  "estimated_diameter": {
14    "kilometers": [
15      {
16        "estimated_diameter_min": 0.3240974354,
17        "estimated_diameter_max": 0.7245026508
18      }
19    ]
20  },
21  "meters": [
22    {
23      "estimated_diameter_min": 324.097435942,
24      "estimated_diameter_max": 724.5026508
25    }
26  ]
27 }
```