SWAPI - Star Wars API Guide

Title Page

SWAPI - Star Wars API Documentation
Programmatic access to the Star Wars universe

Author: Vinay SP Kulkarni
Date: December 2025

Table of Contents

API Reference Document

Overview

SWAPI - the Star Wars API is a programmatic service to access data about the Star Wars universe including people, planets, starships, and more. It is a RESTful API that uses standard HTTP methods and returns responses in JSON format.

Getting Started

Base URL

- Every endpoint starts with the same base URL
  `https://swapi.py4e.com/api/`

Authentication / Authorization

- No authentication or API key is required.
- All endpoints are publicly accessible.

Headers

- Accept: application/json

Example End Point

1. Resource: /people/
2. HTTP Method: Use GET requests to retrieve data.
3. Request URL: `https://swapi.py4e.com/api/people/1/`

Prerequisites

- Systems consuming SWAPI should support HTTP/HTTPS networking and JSON parsing.
- SDKs are not required. But, HTTP clients or browsers with REST capability are required.

Sample Response


```
{
  "name": "Luke Skywalker",
  "height": "172",
  "mass": "77",
  "hair_color": "blond",
  "skin_color": "fair",
  "eye_color": "blue",
  "birth_year": "19BBY",
  "gender": "male",
  "homeworld": "https://swapi.py4e.com/api/planets/1/",
  "films": [
    "https://swapi.py4e.com/api/films/1/",
    "https://swapi.py4e.com/api/films/2/"
  ],
  "species": [],
  "vehicles": [
    "https://swapi.py4e.com/api/vehicles/14/"
  ],
  "starships": [
    "https://swapi.py4e.com/api/starships/12/"
  ],
  "created": "2014-12-09T13:50:51.644000Z",
  "edited": "2014-12-20T21:17:56.891000Z",
  "url": "https://swapi.py4e.com/api/people/1/"
}
```


Code Samples

```
import requests
url = "https://swapi.py4e.com/api/people/1/"
response = requests.get(url)
if response.status_code == 200:
    data = response.json()
    print("Name:", data["name"])
    print("Height:", data["height"])
    print("Films:", data["films"])
else:
    print("Error:", response.status_code)
```

```
const fetch = require("node-fetch");
const url = "https://swapi.py4e.com/api/people/1/";

fetch(url, { headers: { "Accept": "application/json" } })
  .then(response => {
    if (!response.ok) throw new Error("HTTP error " + response.status);
    return response.json();
  })
  .then(data => {
    console.log("Name:", data.name);
    console.log("Height:", data.height);
    console.log("Films:", data.films);
  })
  .catch(err => console.error("Error:", err.message));
```

Status Codes

These status codes help clients handle responses effectively. HTTP Status Codes

- 200 - OK
- 400 - Bad Request – Invalid parameters
- 404 - Not Found – Resource does not exist
- 500 - Internal Server Error

Rate limits

Swapi sets a limit on how many times you can hit the API each day to keep the service running smoothly and stop anyone from going overboard. Right now, you get 10,000 requests per day per IP address. If exceed, request will be throttled.

Pagination and Filtering

Pagination

The SWAPI returns large collections (e.g., /people/) in pages. Each response includes:

- count → total number of items available
- next → URL of the next page of results
- previous → URL of the previous page of results
- results → array of items for the current page

Request URL

GET `https://swapi.py4e.com/api/people/?page=2`

Response (truncated)


```
{
  "count": 87,
  "next": "https://swapi.py4e.com/api/people/?page=3",
  "previous": "https://swapi.py4e.com/api/people/?page=1",
  "results": [
    { "name": "C-3PO", "height": "167" },
    { "name": "R2-D2", "height": "96" }
  ]
}
```


Filtering

You can filter results using the search query parameter.

Request URL

GET `https://swapi.py4e.com/api/people/?search=Luke`

Response


```
{
  "count": 1,
  "results": [
    { "name": "Luke Skywalker", "height": "172", "mass": "77" }
  ]
}
```


Code Samples

```
# Pagination
data = requests.get("https://swapi.py4e.com/api/people/?page=2").json()
print([p["name"] for p in data["results"]])
# Filtering
search = requests.get("https://swapi.py4e.com/api/people/?search=Luke").json()
print([p["name"] for p in search["results"]])
```

```
// Pagination
fetch("https://swapi.py4e.com/api/people/?page=2")
  .then(res => res.json())
  .then(data => console.log(data.results.map(p => p.name)));
// Filtering
fetch("https://swapi.py4e.com/api/people/?search=Luke")
  .then(res => res.json())
  .then(data => console.log(data.results.map(p => p.name)));
```

Status Codes

- 200 - OK
- 400 - Bad Request – Invalid parameters
- 404 - Not Found – Resource does not exist
- 500 - Internal Server Error

Best Practices

Respect Rate Limits

- SWAPI allows 10,000 requests per day for one IP address.
- Cache responses to avoid hitting limits.

Use Caching

Most resources are static. Store them locally to avoid repeat calls.

Handle Errors Gracefully

Implement retries with exponential backoff for temporary errors.

Paginate Through Collections

Navigate datasets using next and previous fields.

Filter Results

Use search to reduce the payload size.

Secure Your Integration

Always use HTTPS.