*Nicholas Kingston, Justin Owen, Angela L.*

# **Part 1: SWAPI**

1. the height of Darth Vader

**202**

https://swapi.dev/api/people/4

2. the population of the planet Alderaan

https://swapi.dev/api/planets/2

3. the name of the manufacturer of the Millennium Falcon

https://swapi.dev/api/starships/10

4. the name of the species that C-3PO belongs to (multiple URLs)

https://swapi.dev/api/people/2

https://swapi.dev/api/species/2

5. the title of each film that Obi-Wan Kenobi is in (multiple URLs)

https://swapi.dev/api/films/1/

https://swapi.dev/api/films/2/

https://swapi.dev/api/films/3/

https://swapi.dev/api/films/4/

https://swapi.dev/api/films/5/

https://swapi.dev/api/films/6/

6. use the search query (the how to on the search query is at the bottom of the Getting Started section of the documentation) to get the information about the Millennium Falcon, it’s a starship

{

"count": 1,

"next": **null**,

"previous": **null**,

"results": [

{

"name": "Millennium Falcon",

"model": "YT-1300 light freighter",

"manufacturer": "Corellian Engineering Corporation",

"cost\_in\_credits": "100000",

"length": "34.37",

"max\_atmosphering\_speed": "1050",

"crew": "4",

"passengers": "6",

"cargo\_capacity": "100000",

"consumables": "2 months",

"hyperdrive\_rating": "0.5",

"MGLT": "75",

"starship\_class": "Light freighter",

"pilots": [

"https://swapi.dev/api/people/13/",

"https://swapi.dev/api/people/14/",

"https://swapi.dev/api/people/25/",

"https://swapi.dev/api/people/31/"

],

"films": [

"https://swapi.dev/api/films/1/",

"https://swapi.dev/api/films/2/",

"https://swapi.dev/api/films/3/"

],

"created": "2014-12-10T16:59:45.094000Z",

"edited": "2014-12-20T21:23:49.880000Z",

"url": "https://swapi.dev/api/starships/10/"

}

]

}

# 

# **Part 2: Social Mountain**

1. Check if the POST request accept params, queries, and/or a body. Which one(s) and what information is it expecting to be sent?

**The POST request accepts a body parameter named “text”. It expects the text of the new post to be sent in the request body.**

2. What data type does the GET request return?

The GET request returns an **array** of all posts in **JSON** format.

3. What would the URL look like for deleting the post with the id 555? (This post does not exist anymore, but the syntax is the same for existing posts, )

curl -X 'DELETE' \ **'https://practiceapi.devmountain.com/api/posts?id=555**' \ -H 'accept: application/json'

4. List the possible response codes from the GET request at ‘/posts/filter’

**200: Returns an Array of Filtered Posts**

**409: Request query is missing the required “text” property.**

5. Create a post whose text is your name, record the URL and body here:

https://practiceapi.devmountain.com/api/posts/filter?text=Justin

{

"id": 1759,

"text": "Justin",

"date": "12 Jul 2023"

}

6. What would the URL and body object be to update the post you just made to contain your favorite color instead of your name?

https://practiceapi.devmountain.com/api/posts?id=1759

{“text”: “Blue”}

7. What is the URL to get posts that contain the text “blue”?

https://practiceapi.devmountain.com/api/posts/filter?text=blue

8. Make a request to GET all the posts. What is the content type and charset of the response? (Hint: look on the Headers)

**The content type of the response is application/json.**

**The charset of the response is `UTF-8`.**

9. What would cause a PUT request to return a 409 status?

**Request was missing req.query.id or req.body.text**

10. What happens if you try to send a query in the GET request URL? Why do you get that response?

**If you put a query in the /posts endpoint you receive all the posts no matter the query because it is not expecting one but if you put a query on the /posts/filter endpoint and you specify either the ID or the text you will receive the filtered response of what you specified.**