Part 1: SWAPI

Instructions

For each of the following use the SWAPI docs, to figure out the complete URL(s) (including params or queries) that you need to go to in order to reach the following data:

1. the height of Darth Vader
    a. https://swapi.dev/api/people/4/
    b. res.data.height
2. the population of the planet Alderaan
    a. https://swapi.dev/api/planets/2/
    b. res.data.population
3. the name of the manufacturer of the Millennium Falcon
    a. https://swapi.dev/api/starships/10/
    b. res.data.manufacturer
4. the name of the species that C-3PO belongs to (multiple URLs)
    a. https://swapi.dev/api/people/2/
        i. res.data.species
    b. https://swapi.dev/api/species/2/
5. the title of each film that Obi-Wan Kenobi is in (multiple URLs)
    a. https://swapi.dev/api/people/10/
        i. res.data.films
    b. https://swapi.dev/api/films/1/
    c. https://swapi.dev/api/films/2/
    d. https://swapi.dev/api/films/3/
    e. https://swapi.dev/api/films/4/
    f. https://swapi.dev/api/films/5/
    g. https://swapi.dev/api/films/6/
        i. res.data.title
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.
    a. https://swapi.dev/api/starships/?search=millennium

Part 2: Social Mountain

Summary

In this section, you'll be looking through the documentation for the Social Mountain API and answering questions. You'll also be making requests and recording the URLs and some information about the responses. Run the requests in Postman. **Note: this API is live and viewable by your classmates and staff. Keep things appropriate for class.**

1. Check if the POST request accepts params, queries, and/or a body. Which one(s) and what information is it expecting to be sent?
    a. Requires body that is a json object with key-value of "text":"string".
2. What data type does the GET request return?
    a. Returns array of all posts in json, each containing id, text, date.
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,)
    a. Delete: https://practiceapi.devmountain.com/api/posts?id=555
4. List the possible response codes from the GET request at '/posts/filter'
    a. 200 and 409
5. Create a post whose text is your name, record the URL and body here:
    a. Giselle's URL: https://practiceapi.devmountain.com/api/posts and body: {"text": "Giselle"}
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?
    a. https://practiceapi.devmountain.com/api/posts?id=6719 and body: {"text": "hot pink"}
7. What is the URL to get posts that contain the text "blue"?
    a. https://practiceapi.devmountain.com/api/posts/filter?text=blue
8. Make a request to GET all the posts. What are the content type and charset of the response? (Hint: look on the Headers)
    a. Content Type: application/json. Charset: UTF-8
9. What would cause a PUT request to return a 409 status?
    a. Not passing in the required body that the PUT needs.
10. What happens if you try to send a query in the GET request URL? Why do you get that response?
    a. The front end does not have the functionality to handle a query on the get request.