**Part 1: SWAPI**

**Instructions**

For each of the following use the [SWAPI docs](https://swapi.dev/documentation), 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
   * **Answer = 202 cm**
   * [**https://swapi.dev/api/people/?search=Darth%20Vader**](https://swapi.dev/api/people/?search=Darth%20Vader)
   * [**https://swapi.dev/api/people/4**](https://swapi.dev/api/people/4)
2. the population of the planet Alderaan
   * **answer = 2000000000**
   * [**https://swapi.dev/api/planets/?search=Alderaan**](https://swapi.dev/api/planets/?search=Alderaan)
   * [**https://swapi.dev/api/planets/2**](https://swapi.dev/api/planets/2)
3. the name of the manufacturer of the Millennium Falcon
   * **answer = "Corellian Engineering Corporation"**
   * **https://swapi.dev/api/starships/10**
4. the name of the species that C-3PO belongs to (multiple URLs)
   * **answer = Droid**
   * [**https://swapi.dev/api/species/2**](https://swapi.dev/api/species/2)
5. the title of each film that Obi-Wan Kenobi is in (multiple URLs)
   * **answer = "A New Hope", "The Empire Strikes Back", "Return of the Jedi", "The Phantom Menace", “Attack of the Clones", "Revenge of the Sith"**
   * **"films":["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
   * [**https://swapi.dev/api/starships/?search=Millennium%20Falcon**](https://swapi.dev/api/starships/?search=Millennium%20Falcon)
   * **{"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**

**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.**

You can view the documentation for the Social Mountain API [here](https://app.swaggerhub.com/apis/DevMountain/social_mountain/1.0.0#/)

The base URL of your requests is: <https://practiceapi.devmountain.com/api> (make sure to have the “s” in “https”)

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 sends an array of all the questions in a JSON object.**
2. What data type does the GET request return?
   * **GET returns an array of all trivia questions also in JSON object.**
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, )
   * [**https://practiceapi.devmountain.com/api/posts?id=555**](https://practiceapi.devmountain.com/api/posts?id=555)
   * **In Postman, select DELETE, use URL:** [**https://practiceapi.devmountain.com/api/posts**](https://practiceapi.devmountain.com/api/posts)**?, and add ‘id’ to key field and ‘1782’ to value field.**
4. List the possible response codes from the GET request at ‘/posts/filter’
   * **200: description: Returns an array of filtered posts**
   * **409: description: Request query is missing required text property.**
5. Create a post whose text is your name, record the URL and body here:
   * **POST** [**https://practiceapi.devmountain.com/api/posts/**](https://practiceapi.devmountain.com/api/posts/)
   * **Body: {"text": "AnAn"} (has to select JSON file in Postman)**
   * **Response BODY:**

**{**

**"id": 1786,**

**"text": "AnAn",**

**"date": "12 Jul 2023"**

**},**

1. What would the URL and body object be to update the post you just made to contain your favorite color instead of your name?
   * **Request URL:** [**https://practiceapi.devmountain.com/api/posts?id=1786**](https://practiceapi.devmountain.com/api/posts?id=1786)

**Body: {**

**"text": "pinkpinkpinkpink"**

}

1. What is the URL to get posts that contain the text “blue”?
   * **https://practiceapi.devmountain.com/api/posts/filter?text=blue**
2. Make a request to GET all the posts. What are the content type and charset of the response? (Hint: look on the Headers)
   * **Content-Type: application/json; charset=utf-8**
3. What would cause a PUT request to return a 409 status?
   * **409: description: Request was missing req.query.id or req.body.text**
4. What happens if you try to send a query in the GET request URL? Why do you get that response?
   * **It returns an array of all the posts in JSON object. This is because the GET request does not need a parameter and it will always return all the posts.**