Part 1: SWAPI

1.the height of Darth Vader - https://swapi.dev/api/people/4

2.The population of the planet Alderaan - https://swapi.dev/api/planets?name=Alderaan

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/?name=C-3PO

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

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

https://swapi.dev/api/people?name=Obi-Wan Kenobi

https://swapi.dev/api/films?characters=10

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/?name=Millennium Falcon

Part 2: Social Mountain

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?

   Specifies to post parameters

   Model is listed as a text object :body

2. What data type does the GET request return?

   Returns an array of filtered posts

   ```
   string

   (query)
   ```

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

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

   Request query is missing required text property.

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

   https://practiceapi.devmountain.com/api/posts


   ```
   {
       "id": 6745,
       "text": "Moira Leon",
   ```


```
    "date": "21 Jul 2021"
  },
  {
    "id": 6747,
    "text": "Sarah",
    "date": "21 Jul 2021"
  },
```


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": 6748,
    "text": "aquamarine",
    "date": "21 Jul 2021"
  },
```


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 are the content type and charset of the response? (Hint: look on the Headers)

   application/json; charset=utf-8

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

   If we are missing the expected params in our put request

   409 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 we enter a data that is actually in the data, it will display all of the responses; however if we enter a param that is not in the api, it will throw a 409 code.

Part 3: Front End (Advanced)

In this section, you'll be making a front end that uses data from SWAPI. The goal is to be able to click a button and get all of the residents of the planet Alderaan listed out on the page.

Setup

1. Create a folder called "swapi" and three files inside - index.html, styles.css, and main.js
2. Open the folder up in VS Code
3. Run **npm init -y** which will create a package.json file
4. Install axios using npm

index.html

1. Create a basic HTML layout (doctype, html, head, body)
2. Connect the CSS file (using a link) and the JS file (script tag)
3. Add another script tag, above the main.js script, to import axios (since axios is in our node modules folder, the src of the script can just use the file path to get to axios, which is **"./node_modules/axios/dist/axios.min.js"**)
4. In the body tag create a button that says "get residents" on it

main.js

1. Select the button using **querySelector** and save it to a variable
2. Write a function that just console logs a string like 'button clicked'
3. Use **addEventListener** to attach the function you just wrote to a click event on the button
4. Open **index.html** in the browser (right click and copy path)
5. Click the button and check the console, if it's working, move on to the next section

making a request

- As you complete this section, be sure to test along the way using Postman and console.logs

1. Now you'll modify the function to make an axios call to SWAPI
2. Make an axios request that gets the information about the planet Alderaan (use the search query to request it, the how to on the search query is at the bottom of the Getting Started section of the documentation)https://swapi.dev/api/planets?name=Alderaan
3. Inside the callback passed to the .then, loop over the residents array returned on the results. It's full of URLs.

```
// axios.post('/api/users', user).then(...)
```

```
//axios.post('/api/planets/ ', residents).then(...)
        "residents": [
            "https://swapi.dev/api/people/1/",
            "https://swapi.dev/api/people/2/",
            "https://swapi.dev/api/people/4/",
            "https://swapi.dev/api/people/6/",
            "https://swapi.dev/api/people/7/",
            "https://swapi.dev/api/people/8/",
            "https://swapi.dev/api/people/9/",
            "https://swapi.dev/api/people/11/",
            "https://swapi.dev/api/people/43/",
            "https://swapi.dev/api/people/62/"
```

```
let user = {
  username: 'Dragon',
  firstName: 'Joey',
  lastName: 'Tribiani'
}

axios.post('/api/users', user).then(...)
```

```
app.post('/api/users', (req, res) => {
  console.log(req.body) // { username: 'Dragon", firstName: 'Joey', lastName: 'Tribiani' }
  let username = req.body.username
```

```
  let firstName = req.body.firstName
  let lastName = req.body.lastName
  res.status(200).send(`Welcome, ${username}! It is nice to meet you ${firstName} ${lastName}.`)
}
```

4. In the loop, make another get request for each URL in the array.
5. You'll have another .then that has its own callback, inside which you should create an h2 element whose content is the name of the resident that you just requested. Append the h2 to your HTML document.

styles.css

- add any styles you'd like to your page