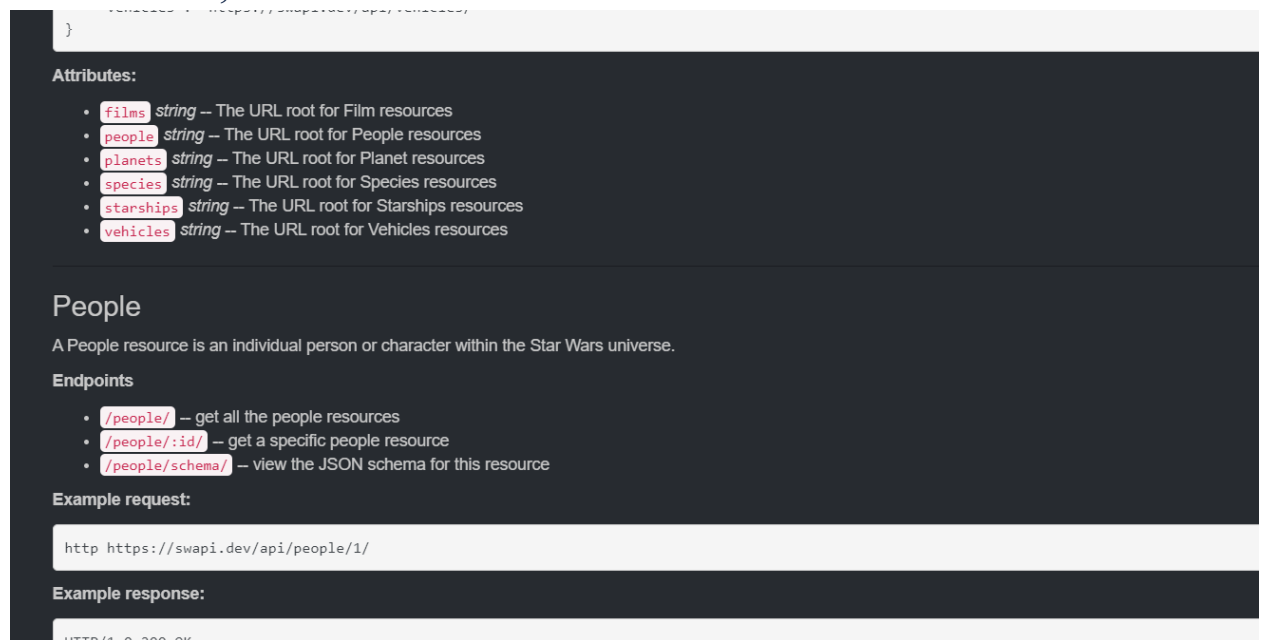


Tech Task #3

1. How can I see all endpoints in SWAPI

➔ We can find the End points by Exploring the SWAPI Documentation: <https://swapi.dev/> documentation. On the end of each Resources, there is End point Mentioned,



The screenshot shows the SWAPI documentation for the 'People' resource. It includes a list of attributes with their corresponding URL roots, a description of the 'People' resource, a list of endpoints, an example request, and an example response.

Attributes:

- `films` *string* – The URL root for Film resources
- `people` *string* – The URL root for People resources
- `planets` *string* – The URL root for Planet resources
- `species` *string* – The URL root for Species resources
- `starships` *string* – The URL root for Starships resources
- `vehicles` *string* – The URL root for Vehicles resources

People

A People resource is an individual person or character within the Star Wars universe.

Endpoints

- `/people/` – get all the people resources
- `/people/{id}/` – get a specific people resource
- `/people/schema/` – view the JSON schema for this resource

Example request:

```
http https://swapi.dev/api/people/1/
```

Example response:

```
HTTP/1.1 200 OK
```

2. How many people is saved in SWAPI database?

➔ Total 82 people is saved in SWAPI database.

3. How can I see JSON schema for people endpoint

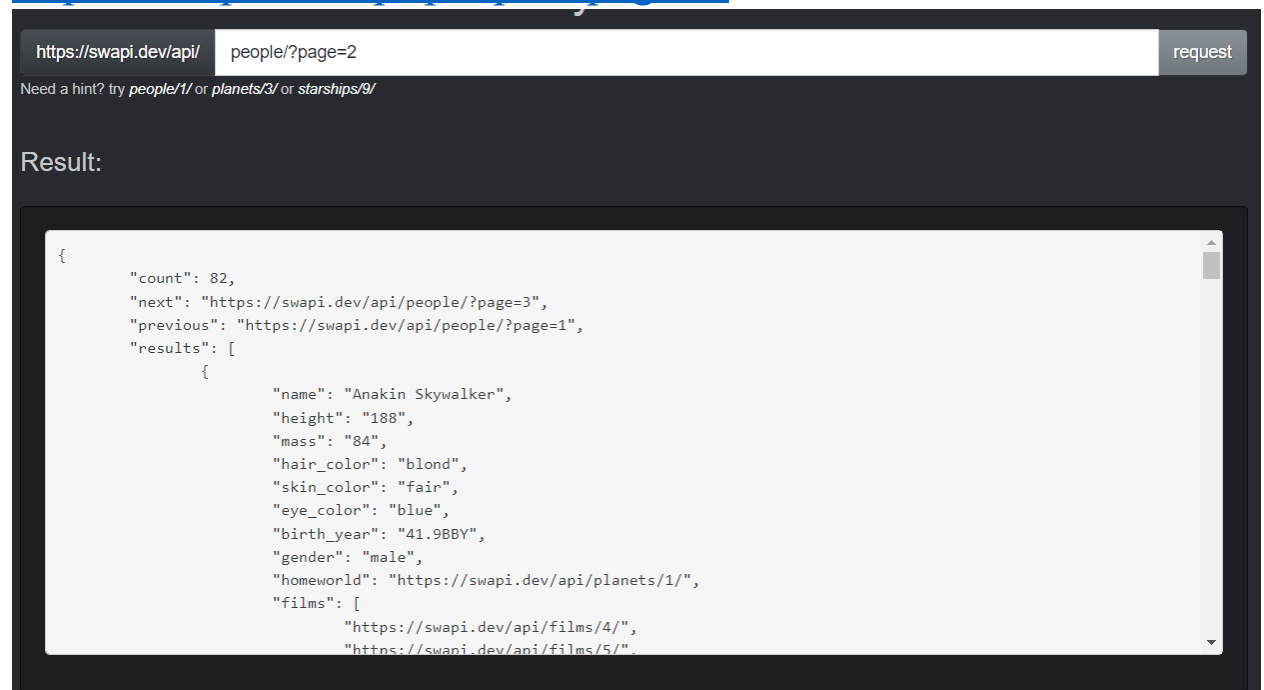
➔ To see Json Schema for people endpoint we need to use,

Request: GET - <https://swapi.dev/api/people/schema>

4. What query parameter do use for moving to the next page?

→ - Request: GET - <https://swapi.dev/api/people/>

- Add a query parameter: `page` with a value
<https://swapi.dev/api/people/?page=2>



5. What description have starships value?

→ Request: GET - <https://swapi.dev/api/starships/>

- Response:

Try it now!

[https://swapi.dev/api/](https://swapi.dev/api/starships/)

Need a hint? try [people/1/](#) or [planets/3/](#) or [starships/9/](#)

Result:

```
{
  "count": 36,
  "next": "https://swapi.dev/api/starships/?page=2",
  "previous": null,
  "results": [
    {
      "name": "CR90 corvette",
      "model": "CR90 corvette",
      "manufacturer": "Corellian Engineering Corporation",
      "cost_in_credits": "3500000",
      "length": "150",
      "max_atmosphering_speed": "950",
      "crew": "30-165",
      "passengers": "600",
      "cargo_capacity": "3000000",
      "consumables": "1 year",
      "hyperdrive_rating": "2.0",
      "MGLT": "60"
```

6. What ID have character named Han Solo

→ Han Solos ID is 14,

[https://swapi.dev/api/](https://swapi.dev/api/people/14)

Need a hint? try [people/1/](#) or [planets/3/](#) or [starships/9/](#)

Result:

```
{
  "name": "Han Solo",
  "height": "180",
  "mass": "80",
  "hair_color": "brown",
  "skin_color": "fair",
  "eye_color": "brown",
  "birth_year": "29BBY",
  "gender": "male",
  "homeworld": "https://swapi.dev/api/planets/22/",
  "films": [
    "https://swapi.dev/api/films/1/",
    "https://swapi.dev/api/films/2/",
    "https://swapi.dev/api/films/3/"
  ],
  "species": [],
  "vehicles": [],
  "starships": [
```

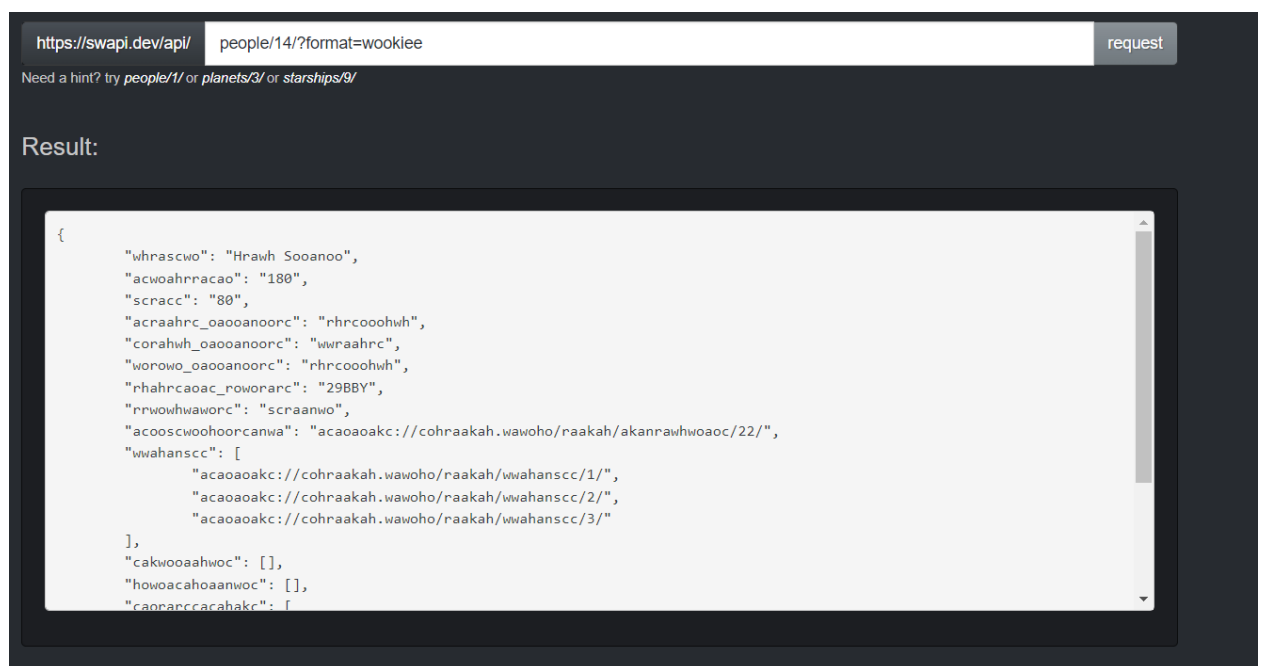
7. How to bring back Han Solo, but so that the answer was in the Wookiee language?

➔ Method: GET

URL: <https://swapi.dev/api/people/>

Wookiee language:

<https://swapi.dev/api/people/14/?format=wookiee>



8. How would the URL be in Wookiee?

➔ The URL in Wookiee language would be like:

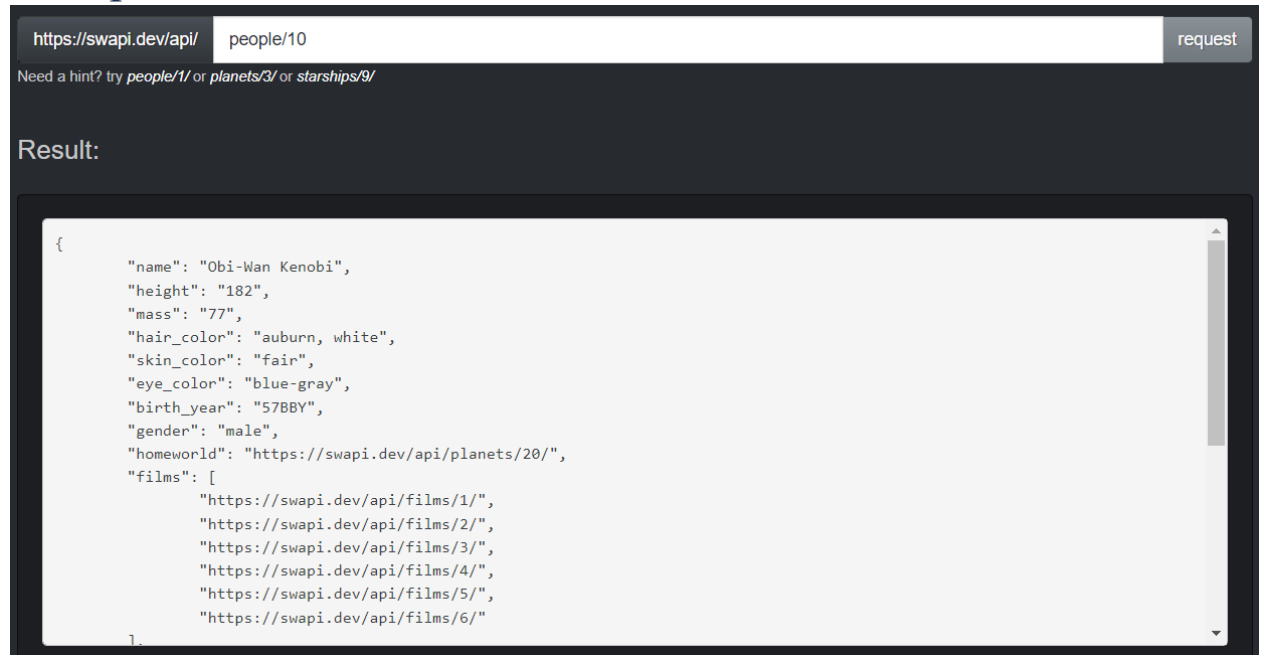
<https://swapi.dev/api/people/id/?format=wookiee>

9. How to find Obi-Wan Kenobi among people?

➔ We can find Obi- Wan Kenobi by his id which is 10,

Request: GET - <https://swapi.dev/api/people/10/>

- Response:



For Postman

1. ****Get Information about All Employees:****

- This query retrieves a list of all employees.

2. ****Get Information about Employee with ID = 2 + Check Existence:****

- This query retrieves information about the employee with ID 2 and includes a test to ensure the response indicates success and contains the correct employee ID.

3. ****Create a New Record about Employee in the Database:****

- This query creates a new employee record with the name "John Doe," a salary of "50000," and an age of "30."

4. ****Update Values of the Created Employee from Step 3:****

- This query updates the information of the employee created in Step 3, changing the name to "Updated John Doe," the salary to "55000," and the age to "32."

5. ****Delete the Created Employee:****

- This query deletes the employee created in Step 3.