Red Hat Technical Assessment - Data Engineer

This is meant to assess a number of things for potential candidates:
- Comfort with the programming language - Python
- Comfort with potentially troubleshooting problem when implementing a technology - Setting up MySQL database locally
- Proficiency with documentation

There are some requirements for the tasks:
- You cannot use the Pandas library to pull, push, or transform data.
  - We recommend using the Requests library to perform API requests.
- Use Python3.x not Python2.x
- You will not be allowed to use the Python wrapper for the Star Wars API.
- You should use MySQL for your database.
- SQL statements will be limited to *select* or *insert* or *create*.
- Data transformations should take place in Python during runtime.
- You should document all work so that a user can easily setup your project and verify your results.
- Your project needs to be hosted on Github

For the tasks below you will be using the Star Wars API - https://swapi.co/

Task 1

The Star Wars API lists 87 main characters in the Star Wars saga. For the first task we would like you to use a random number generator that picks a number between 1-87. Using these random numbers you will be pulling 15 characters from the API using Python. (*reference below*)
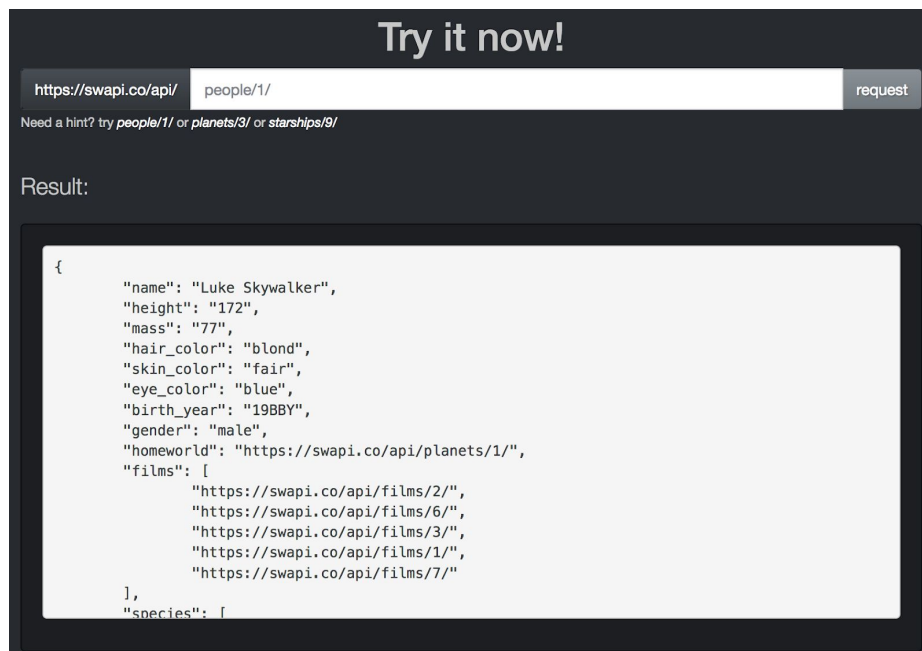
We would like you to:

1. GET 15 random characters and the names of the films they have been in using Python.
2. Insert into MySQL - Please include SQL database schema(s) for any table(s) created in your Github repo
3. Write a script called *task_one.py* that when called will output something like this to the console. (*reference below*).

```
 1   [
 2       {
 3           "film": "The Empire Strikes Back",
 4           "character":
 5           [
 6               "Luke Skywalker",
 7               "C-3PO",
 8               "R2-D2",
 9               "Darth Vader"
10           ]
11       },
12       {
13           "film": "Return of the Jedi",
14           "character":
15           [
16               "Luke Skywalker",
17               "C-3PO",
18               "R2-D2",
19               "Darth Vader"
20           ]
21       }
22   ]
```

Task 2
Now we would like you to pull data from the *films* endpoint - *see reference below*

Try it now!

https://swapi.co/api/   films/2/   request

Need a hint? try *people/1/* or *planets/3/* or *starships/9/*

Result:

```
{
    "title": "The Empire Strikes Back",
    "episode_id": 5,
    "opening_crawl": "It is a dark time for the\r\nRebellion. Although the Death\r\nStar has be
    "director": "Irvin Kershner",
    "producer": "Gary Kurtz, Rick McCallum",
    "release_date": "1980-05-17",
    "characters": [
        "https://swapi.co/api/people/1/",
        "https://swapi.co/api/people/2/",
        "https://swapi.co/api/people/3/",
        "https://swapi.co/api/people/4/",
        "https://swapi.co/api/people/5/",
        "https://swapi.co/api/people/10/",
        "https://swapi.co/api/people/13/",
        "https://swapi.co/api/people/14/",
        "https://swapi.co/api/people/18/",
        "https://swapi.co/api/people/20/".
```

We would like you to do the following:
1. Pull data for the movie *A New Hope*
2. Replace the data for each of the endpoints listed in the JSON object you receive from the API request (e.g. - In the example above you would take all the character endpoints and pull the data from each of those endpoints then insert the data into the JSON object, etc.)
   a. *A New Hope* has character, planet, starship, vehicle, and species data you will need to retrieve and replace.
3. We also ask that you convert the metric heights and weights of each character to standard units.
4. You will also need to remove all cross referencing material from the data you replace (e.g. - When you pull Luke Skywalker you would want to remove cross-referencing urls from Luke's JSON object (like films, species, vehicles, and spaceships.)
5. Lastly, you will take the dictionary you created and write it out to a JSON file locally named *task_two.json*.

Bonus Task
This is not required but if you would like to attempt it, please do so:
● We would like you to write unit tests for your functions and end to end testing for the tasks.

Good luck, and may the force be with you!