Harry Potter API testing

In this assignment, you will test Happy Potter API created based on Harry Potter movie series. Full documentation for Happy Potter API can be found here: https://www.potterapi.com/. Watch all the movies or read all the books to better understanding of this API (this is optional). To test this API, you have to register and get access key. Sign up for access key. Access key is required for most of the requests.

Import the Postman collection using this link: https://www.getpostman.com/collections/5ceaa3e2705188383075
In postman create an environment for this API. Add the following variables:

1. Variable name: baseURL
   Variable value: `https://www.potterapi.com/v1/`

2. Variable name: apiKey
   Variable value: your api key from the https://www.potterapi.com/

Automate the given test cases. You can use any existing project. You can automate all test cases in same class or different classes.
For verifying all of the use *pojos*. Create pojo classes for **Character** and **House** in *pojos* package based on the provided json files.

Verify sorting hat

1. Send a get request to `/sortingHat`. Request includes :
2. Verify status code 200, content type application/json; charset=utf-8
3. Verify that response body contains one of the following houses:
   "Gryffindor", "Ravenclaw", "Slytherin", "Hufflepuff"

Verify bad key

1. Send a get request to `/characters`. Request includes :
   - Header Accept with value application/json
   - Query param key with value invalid
2. Verify status code 401, content type application/json; charset=utf-8
3. Verify response status line include message Unauthorized
4. Verify that response body says *"error": "API Key Not Found"*

Verify no key

1. Send a get request to `/characters`. Request includes :
   - Header Accept with value application/json
2. Verify status code 409, content type application/json; charset=utf-8
3. Verify response status line include message Conflict
4. Verify that response body says *"error": "Must pass API key for request"*

Verify number of characters

1. Send a get request to `/characters`. Request includes :
   - Header Accept with value application/json
   - Query param key with value {{apiKey}}
2. Verify status code 200, content type application/json; charset=utf-8
3. Verify response contains 194 characters

Verify number of character id and house

1. Send a get request to `/characters`. Request includes :
   - Header Accept with value application/json
   - Query param key with value {{apiKey}}
2. Verify status code 200, content type application/json; charset=utf-8
3. Verify all characters in the response have id field which is not empty
4. Verify that value type of the field dumbledoresArmy is a boolean in all characters in the response
5. Verify value of the house in all characters in the response is one of the following:
   "Gryffindor", "Ravenclaw", "Slytherin", "Hufflepuff"

Verify all character information

1. Send a get request to `/characters`. Request includes :
   - Header Accept with value application/json
   - Query param key with value {{apiKey}}
2. Verify status code 200, content type application/json; charset=utf-8
3. Select name of any ***random*** character
4. Send a get request to `/characters`. Request includes :
   - Header Accept with value application/json
   - Query param key with value {{apiKey}}
   - Query param name with value from step 3
5. Verify that response contains the same character information from step 3. Compare all fields.

Verify name search

1. Send a get request to `/characters`. Request includes :
   - Header Accept with value application/json
   - Query param key with value {{apiKey}}
   - Query param name with value Harry Potter
2. Verify status code 200, content type application/json; charset=utf-8
3. Verify name Harry Potter
4. Send a get request to `/characters`. Request includes :
   - Header Accept with value application/json
   - Query param key with value {{apiKey}}
   - Query param name with value Marry Potter
5. Verify status code 200, content type application/json; charset=utf-8
6. Verify response body is empty

Verify house members

1. Send a get request to `/houses`. Request includes :
   - Header Accept with value application/json
   - Query param key with value {{apiKey}}
2. Verify status code 200, content type application/json; charset=utf-8
3. Capture the id of the Gryffindor house
4. Capture the ids of the all members of the Gryffindor house
5. Send a get request to `/houses/:id`. Request includes :
   - Header Accept with value application/json
   - Query param key with value {{apiKey}}
   - Path param id with value from step 3
6. Verify that response contains the same member ids as the step 4

Verify house members again

1. Send a get request to `/houses/:id`. Request includes :
   - Header Accept with value application/json
   - Query param key with value {{apiKey}}
   - Path param id with value 5a05e2b252f721a3cf2ea33f
2. Capture the ids of all members
3. Send a get request to `/characters`. Request includes :
   - Header Accept with value application/json
   - Query param key with value {{apiKey}}
   - Query param house with value Gryffindor
4. Verify that response contains the same member ids from step 2

Verify house with most members

1. Send a get request to `/houses`. Request includes :
   - Header Accept with value application/json
   - Query param key with value {{apiKey}}
2. Verify status code 200, content type application/json; charset=utf-8
3. Verify that Gryffindor house has the most members