One Possible Approach to Solution: Big Data Testing + Data Driven Test Automation

Problem: What is Big Data Testing

This is a kind of big data problem. We can apply big data testing principles:
These steps should be followed for successful result:

1. Manually check the data for quick grasp data
2. Create test cases in high level
3. Create a testable and controllable subset data from the original data set
4. Create low level test cases
5. Create data test scripts depending on data-driven test automation approach
6. Run test script with the subset data
7. Eliminate the bugs, test until to make sure it works for every low level test cases
8. Run test script with original big data

Methodology: What is Data-Driven Test Automation

Test script is written to test the data provided outsource of the script so new data can be tested with the same script. This approach is usable when the test data changes periodicall but the form of the data and requirement is not changing.

Application of Big Data Prenciples:

1. Data is json file and composed of two keys: 'books' and 'authors'
    - **Bug**: keys in a book should be : 'id', 'name', 'author_id' but they are 'id', 'name', '**author**'
2. These are the high level test cases:
    1. ['Blocker', 'Source should be in JSON format'],
    2. ['Major', 'Books should have 3 keys'],
    3. ['Major', 'Books should have following keys: author, id, name'],
    4. ['Major', 'Books should have following keys: author, id, name not empty'],
    5. ['Minor', 'There should be no duplicated book records with same id but different content'],
    6. ['Minor', 'There should be no duplicated book records with same id and same content'],
    7. ['Major', 'Author should have 2 keys'],
    8. ['Major', 'Author should have following keys: id, name'],
    9. ['Major', 'Author should have following keys: id, name not empty'],
    10. ['Minor', 'There should be no duplicated author records with same id but different content'],
    11. ['Minor', 'There should be no duplicated author records with same id and same content'],
    12. ['Major', 'Books should have a valid author'],
3. Check the data from subset_book_data.json which is composed of 100 books and 100 authors
    1. Low-level test cases are applied to the data
    2. Check the next item for low level test cases
4. Indented item are low level cases
    1. ['Major', 'Books should have 3 keys'],
        1. It can be 2 or less
        2. It can be 4 or more

    2. ['Major', 'Books should have following keys: author, id, name'],
        1. no author
        2. no id
        3. no name

3. ['Major', 'Books should have following keys: author, id, name not empty'],
    1. empty author
    2. empty id
    3. empty name

4. ['Minor', 'There should be no duplicated book records with same id but different content'],
    1. same id, different author
    2. same id, different name

5. ['Minor', 'There should be no duplicated book records with same id and same content'],
    1. same id, same name, same author

6. ['Major', 'Author should have 2 keys'],
    1. It can be 1
    2. It can be 3 or more

7. ['Major', 'Author should have following keys: id, name'],
    1. no id
    2. no name

8. ['Major', 'Author should have following keys: id, name not empty'],
    1. empty id
    2. empt name

9. ['Minor', 'There should be no duplicated author records with same id but different content'],
    1. same id, different name

10. ['Minor', 'There should be no duplicated author records with same id and same content'],
    1. same id, same name

11. ['Major', 'Books should have a valid author'],
    1. no author

5. Check 'src' folder
6. Check 'subset_book_data_validated.json' in 'result' folder
7. Check 'subset_book_data_result.txt'in 'result' folder
8. run this command: 'python run.py book_data.json' and check result folder