GoodReads.com Scraper Manual Test Plan

Prerequisites

OS:

- Windows
- MacOS
- Linux

OS should not matter much as long as it supports the necessary python version

<u>Python 3.7+</u>

Not necessary but useful to install ‘pip’ for python for downloading the necessary python packages for the library.

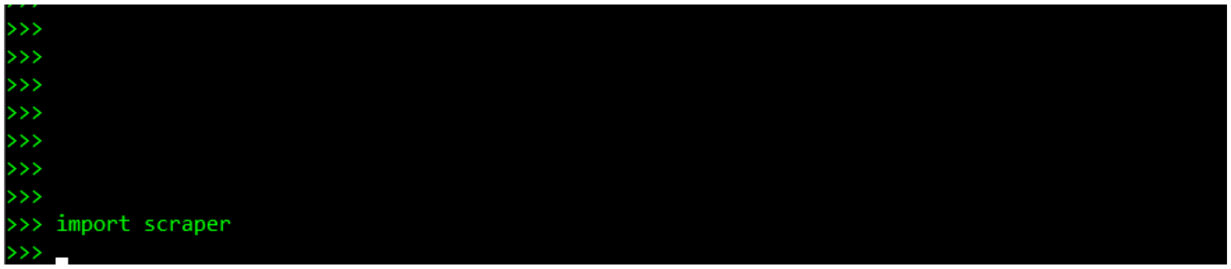
Python packages: json, requests, bs4, tinydb

Manual Test 1:

<u>Scraping Good Reads for One or More Books</u>

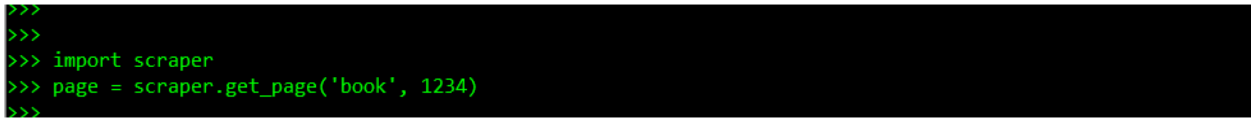
Open a python terminal from the directory in which you have ‘scraper.py’

Import scraper into python



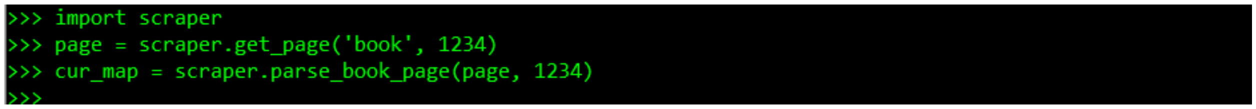
```
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> import scraper
>>>
```

Use get_page to extract the page from good reads, suppling ‘book’ and the book id.



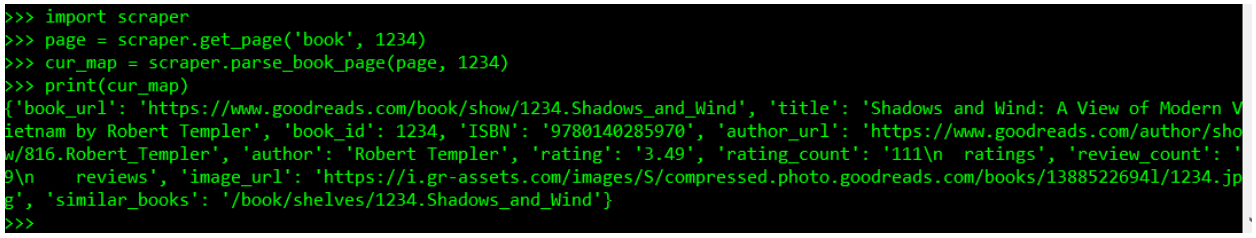
```
>>>
>>>
>>> import scraper
>>> page = scraper.get_page('book', 1234)
>>>
```

Now, that you have the book page loaded, extract the useful contents into a map using the parse_book_page function, supplying the current page you downloaded and the page id again



```
>>> import scraper
>>> page = scraper.get_page('book', 1234)
>>> cur_map = scraper.parse_book_page(page, 1234)
>>>
```

Now that you have a map of the data you can print it out to ensure it parsed the data correctly.



```
>>> import scraper
>>> page = scraper.get_page('book', 1234)
>>> cur_map = scraper.parse_book_page(page, 1234)
>>> print(cur_map)
{'book_url': 'https://www.goodreads.com/book/show/1234.Shadows_and_Wind', 'title': 'Shadows and Wind: A View of Modern V
ietnam by Robert Templer', 'book_id': 1234, 'ISBN': '9780140285970', 'author_url': 'https://www.goodreads.com/author/sho
w/816.Robert_Templer', 'author': 'Robert Templer', 'rating': '3.49', 'rating_count': '111\n  ratings', 'review_count': '
9\n    reviews', 'image_url': 'https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1388522694l/1234.jp
g', 'similar_books': '/book/shelves/1234.Shadows_and_Wind'}
>>>
```

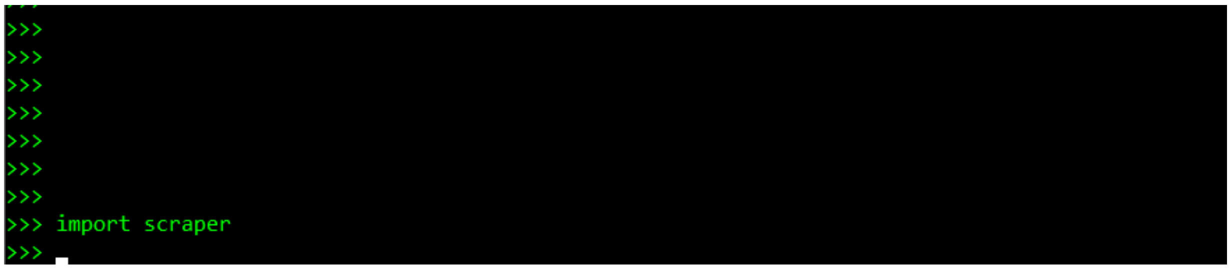
At this point you should check for missing fields in the map. The program will try to get each necessary element and if it failed on any of them it should print to the console which element it failed to retrieve. Use this to ensure it is working properly, beware, site may be missing the field so double check that the site does that the missing element to diagnose any issues.

Manual Test 2:

Scraping an Authors Page

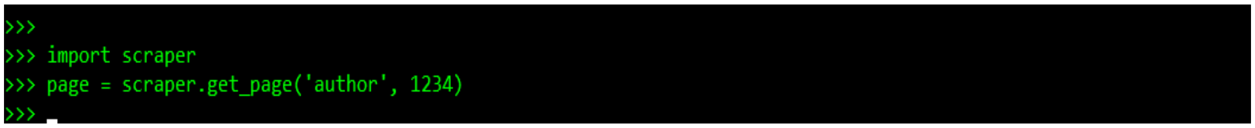
Similarly to the previous test you want to set up the same python environment.

Again, import in the scraper file.



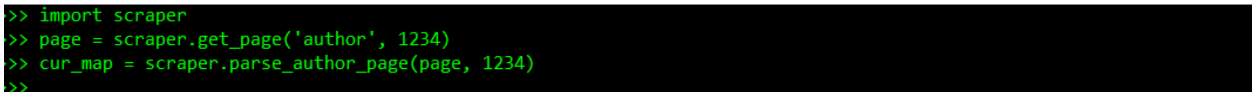
```
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> import scraper
>>> 
```

After this, you again want to get the page using the same function (scraper.get_page), however for an author page one must supply ‘author’ followed by the authors ID as the parameters



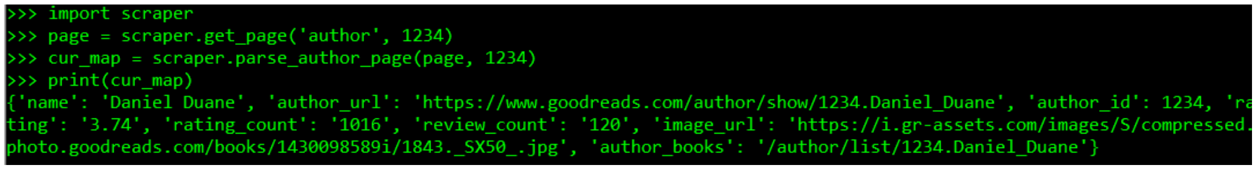
```
>>>
>>> import scraper
>>> page = scraper.get_page('author', 1234)
>>> 
```

Following this, you again want to parse the outputted page however you must now use the parse_author_page function to extract the useful author information



```
>> import scraper
>> page = scraper.get_page('author', 1234)
>> cur_map = scraper.parse_author_page(page, 1234)
>>
```

Now that the map has been created from scraping the page you can view it again by simply printing it out.



```
>>> import scraper
>>> page = scraper.get_page('author', 1234)
>>> cur_map = scraper.parse_author_page(page, 1234)
>>> print(cur_map)
{'name': 'Daniel Duane', 'author_url': 'https://www.goodreads.com/author/show/1234.Daniel_Duane', 'author_id': 1234, 'ra
ting': '3.74', 'rating_count': '1016', 'review_count': '120', 'image_url': 'https://i.gr-assets.com/images/S/compressed.
photo.goodreads.com/books/1430098589i/1843._SX50_.jpg', 'author_books': '/author/list/1234.Daniel_Duane'}
```

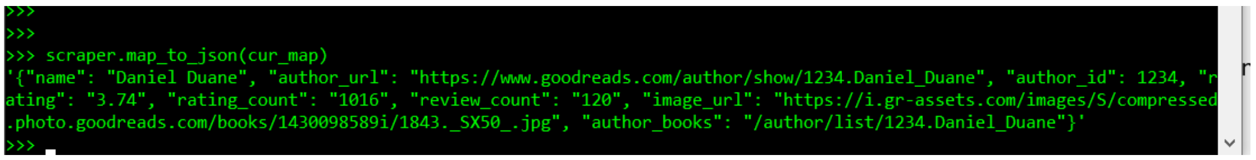
Here you would again want to check for missing values and ensure that they are actually missing from the page and not a failure of the function to extract it.

Manual Test 3:

Testing conversion of map to json and from json to map

First set up the python terminal as done before. You will want to follow one of the previous two tests in order to create map to be converted into a json and back.

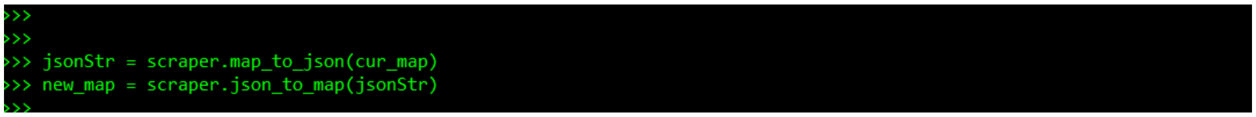
Use the map_to_json function to convert the map into a json string



```
>>>
>>>
>>> scraper.map_to_json(cur_map)
'{"name": "Daniel Duane", "author_url": "https://www.goodreads.com/author/show/1234.Daniel_Duane", "author_id": 1234, "r
ating": "3.74", "rating_count": "1016", "review_count": "120", "image_url": "https://i.gr-assets.com/images/S/compressed
.photo.goodreads.com/books/1430098589i/1843._SX50_.jpg", "author_books": "/author/list/1234.Daniel_Duane"}'
>>> 
```

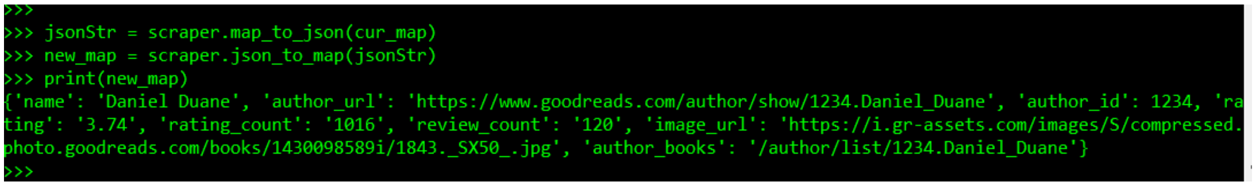
Here it is seen that the output is similar to the map supplied but it is in the json string format.

Then you can check that the opposite function works currently by calling the json_to_map function to convert it back into a map.



```
>>>
>>>
>>> jsonStr = scraper.map_to_json(cur_map)
>>> new_map = scraper.json_to_map(jsonStr)
>>>
```

Then this can be printed out to view the map.



```
>>>
>>> jsonStr = scraper.map_to_json(cur_map)
>>> new_map = scraper.json_to_map(jsonStr)
>>> print(new_map)
{'name': 'Daniel Duane', 'author_url': 'https://www.goodreads.com/author/show/1234.Daniel_Duane', 'author_id': 1234, 'ra
ting': '3.74', 'rating_count': '1016', 'review_count': '120', 'image_url': 'https://i.gr-assets.com/images/S/compressed.
photo.goodreads.com/books/1430098589i/1843._SX50_.jpg', 'author_books': '/author/list/1234.Daniel_Duane'}
>>>
```

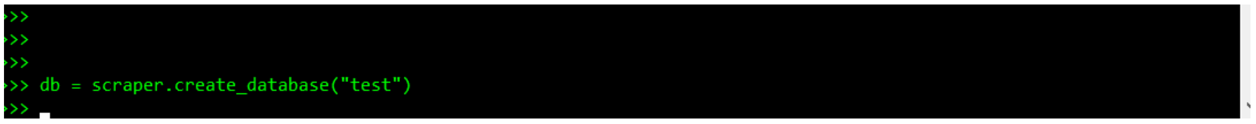
At this point, you should checked to ensure that the json string and the map from before which was supplied to both functions came back unchanged. This should show that the scraper can convert to a json string and correctly back into a map.

Manual Test 4:

Creating a database using the map

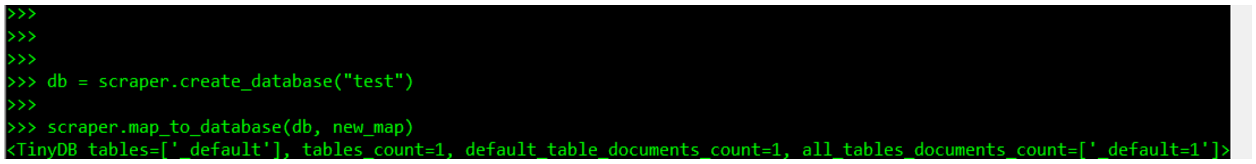
This manual test instruction will focus on two functions used for creating a low weight database from a map.

Similarly to the previous tests set up the test environment. Now you should use the create_database function supplying a name for the according json created.



```
>>
>>
>>
>> db = scraper.create_database("test")
>> 
```

Once this data base is created you can add items to it using the map_to_database function to insert the maps created by parsing authors and books into the database.

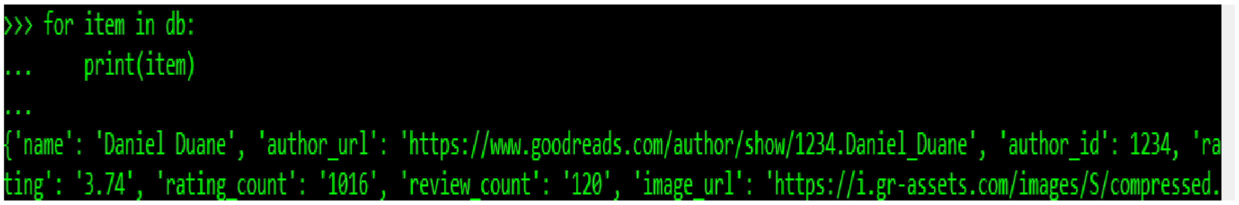


```
>>>
>>>
>>>
>>> db = scraper.create_database("test")
>>>
>>> scraper.map_to_database(db, new_map)
<TinyDB tables=['_default'], tables_count=1, default_table_documents_count=1, all_tables_documents_count=['_default=1']>
```

As seen here, it adds the item into the database. It returns the tb object with some information.

This will show whether it successfully added the item into the database.

Further, one can print out the table to check it using the following call.



```
>>> for item in db:
...     print(item)
...
{'name': 'Daniel Duane', 'author_url': 'https://www.goodreads.com/author/show/1234.Daniel_Duane', 'author_id': 1234, 'ra
ting': '3.74', 'rating_count': '1016', 'review_count': '120', 'image_url': 'https://i.gr-assets.com/images/S/compressed.
```

As seen here, you must loop through the items in the database and print them out and view the contents of it.

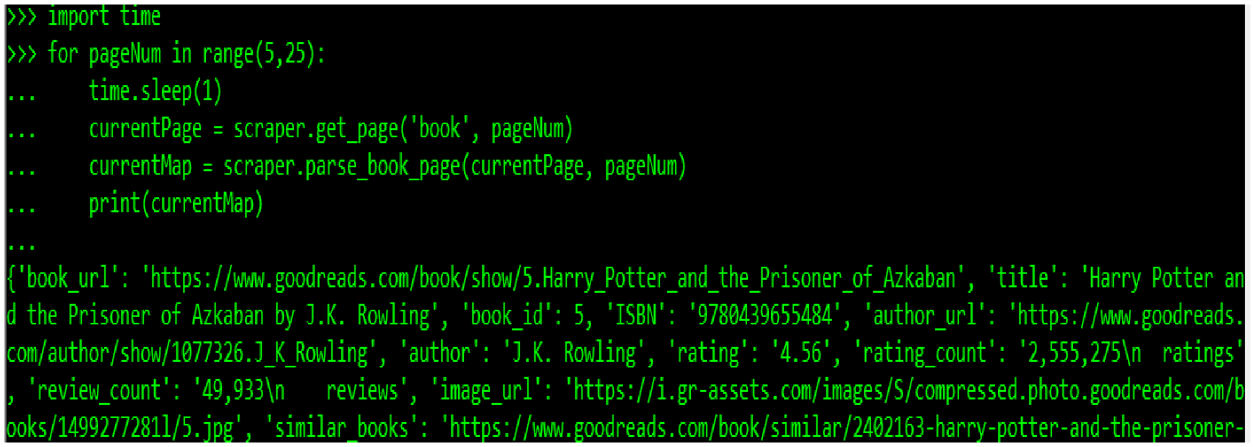
It is a good idea to test adding more items and ensure that they are successfully added to the database.

Manual Test 5:

Testing a loop of books

This test is similar to the first one but focuses on automatically pulling books by an ID rather than hardcoding the ID in.

Similarly, set up the same python environment and imports.



```
>>> import time
>>> for pageNum in range(5,25):
...     time.sleep(1)
...     currentPage = scraper.get_page('book', pageNum)
...     currentMap = scraper.parse_book_page(currentPage, pageNum)
...     print(currentMap)
...
{'book_url': 'https://www.goodreads.com/book/show/5.Harry_Potter_and_the_Prisoner_of_Azkaban', 'title': 'Harry Potter an
d the Prisoner of Azkaban by J.K. Rowling', 'book_id': 5, 'ISBN': '9780439655484', 'author_url': 'https://www.goodreads.
com/author/show/1077326.J_K_Rowling', 'author': 'J.K. Rowling', 'rating': '4.56', 'rating_count': '2,555,275\n  ratings'
, 'review_count': '49,933\n    reviews', 'image_url': 'https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/b
ooks/1499277281l/5.jpg', 'similar_books': 'https://www.goodreads.com/book/similar/2402163-harry-potter-and-the-prisoner-
```

As seen in the above image, one must set up a look of integers to be used for the book id in the query.

It is also recommended that you have the 1 second sleep in order to avoid an IP ban from excessive web scraping.

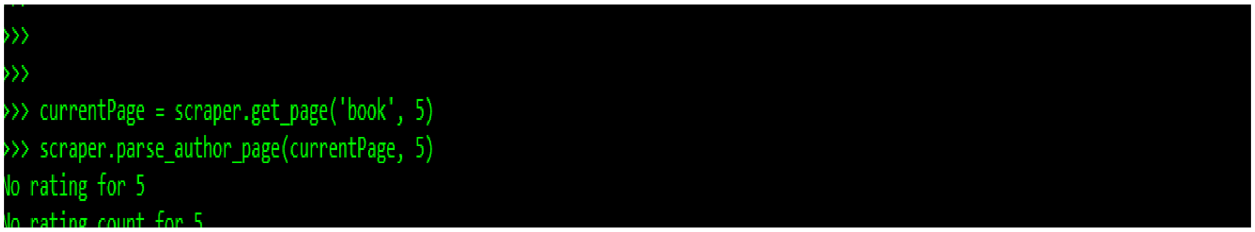
This loop will get the page by the book number and parse the page into the map, and finally it will print that.

In the image it is visible that the first book in the iterations is printed out in its parsed map form.

It is recommended that one ensures that more than one of these works. Observe that the latter outputs work and have the expected output, with deviations being compared to the relevant website.

Error Messages:

There are various error messages one my come by when using the functions in this package. Most of them have try and except’s with a failure in the attempt printing out an informative string to the console from with the user can work through to debug. One example of this is shown below for parsing a book page using the author parser.



```
>>>
>>>
>>> currentPage = scraper.get_page('book', 5)
>>> scraper.parse_author_page(currentPage, 5)
No rating for 5
No rating count for 5
```

Where the first print comes from the failure to parse the book page using the author parser. Printing that there is no rating for book ‘ID’ 5.