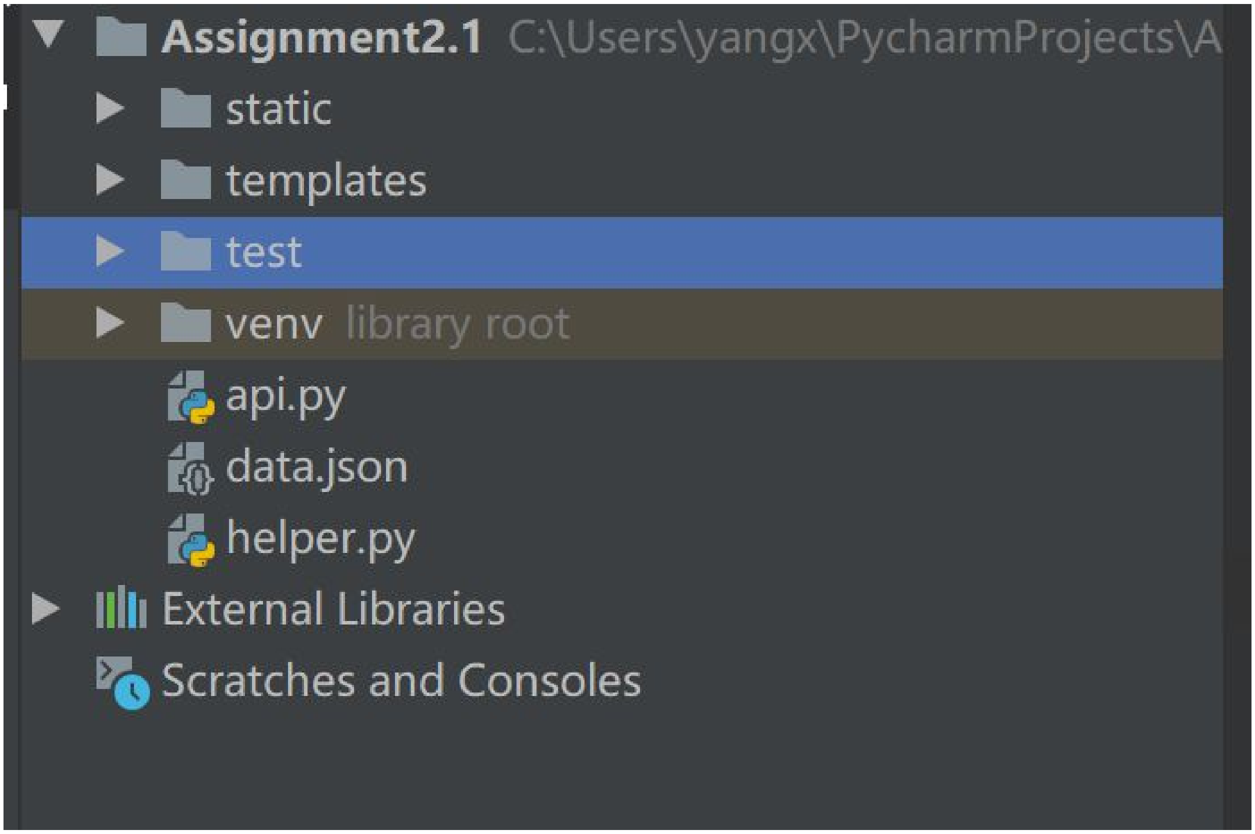
Data structure:



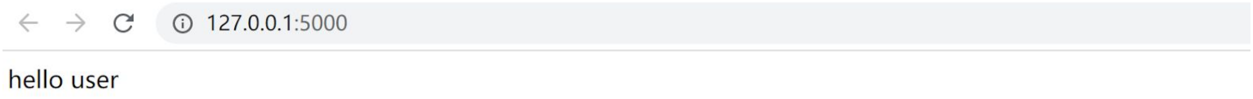
```
▼ Assignment2.1  C:\Users\yangx\PycharmProjects\A
    ▶ static
    ▶ templates
    ▶ test
    ▶ venv  library root
      api.py
      data.json
      helper.py
▶ External Libraries
  Scratches and Consoles
```

This program follows structure above.

- Static folder stores js file and css file.
- templates folder stores html template for rendering
- api.py is the api functions in flask structure
- data.json is the data file we crawled.
- helper.py includes functions that reads, parse, store and manipulate data file.

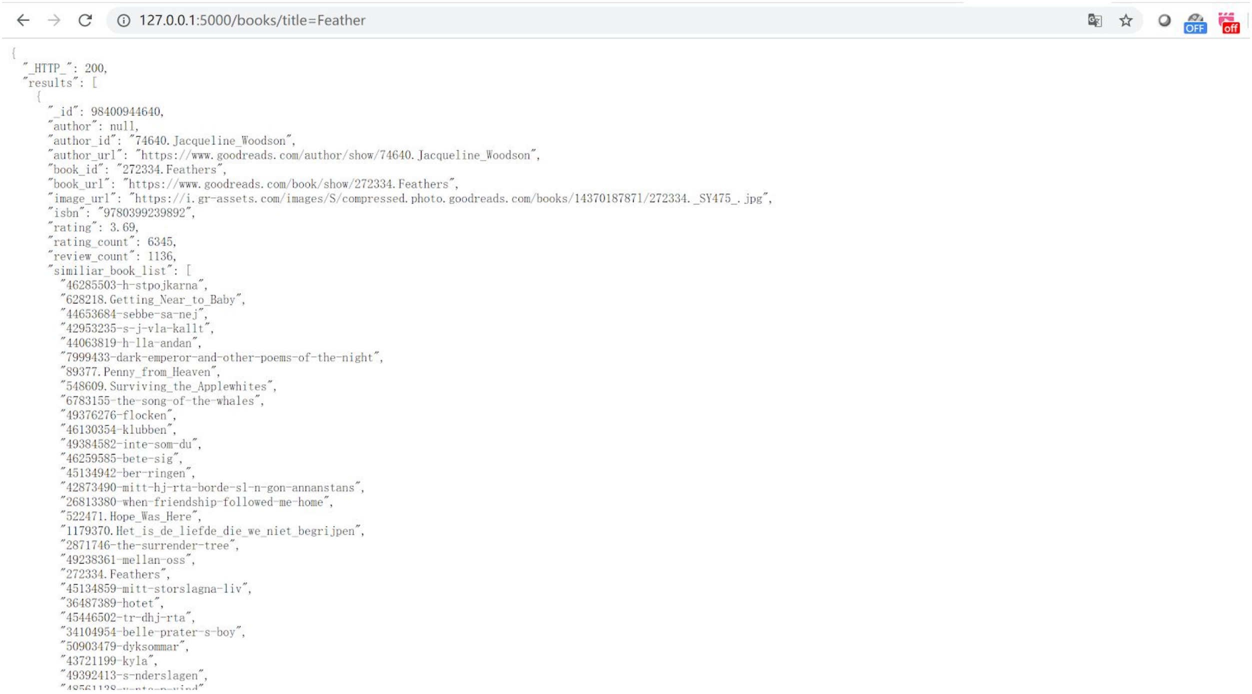
How to use:

Run api.ty or in python console run $ flask run api.ty will redirect you to the index page.



hello user

1) Find books with key word title, author,similarbook
   http://127.0.0.1:5000/books/title=Feather
   You can replace title with author or similar_book_list
   And Words after = sign is the key word you want to search. It will return all items meet your requirements. For example

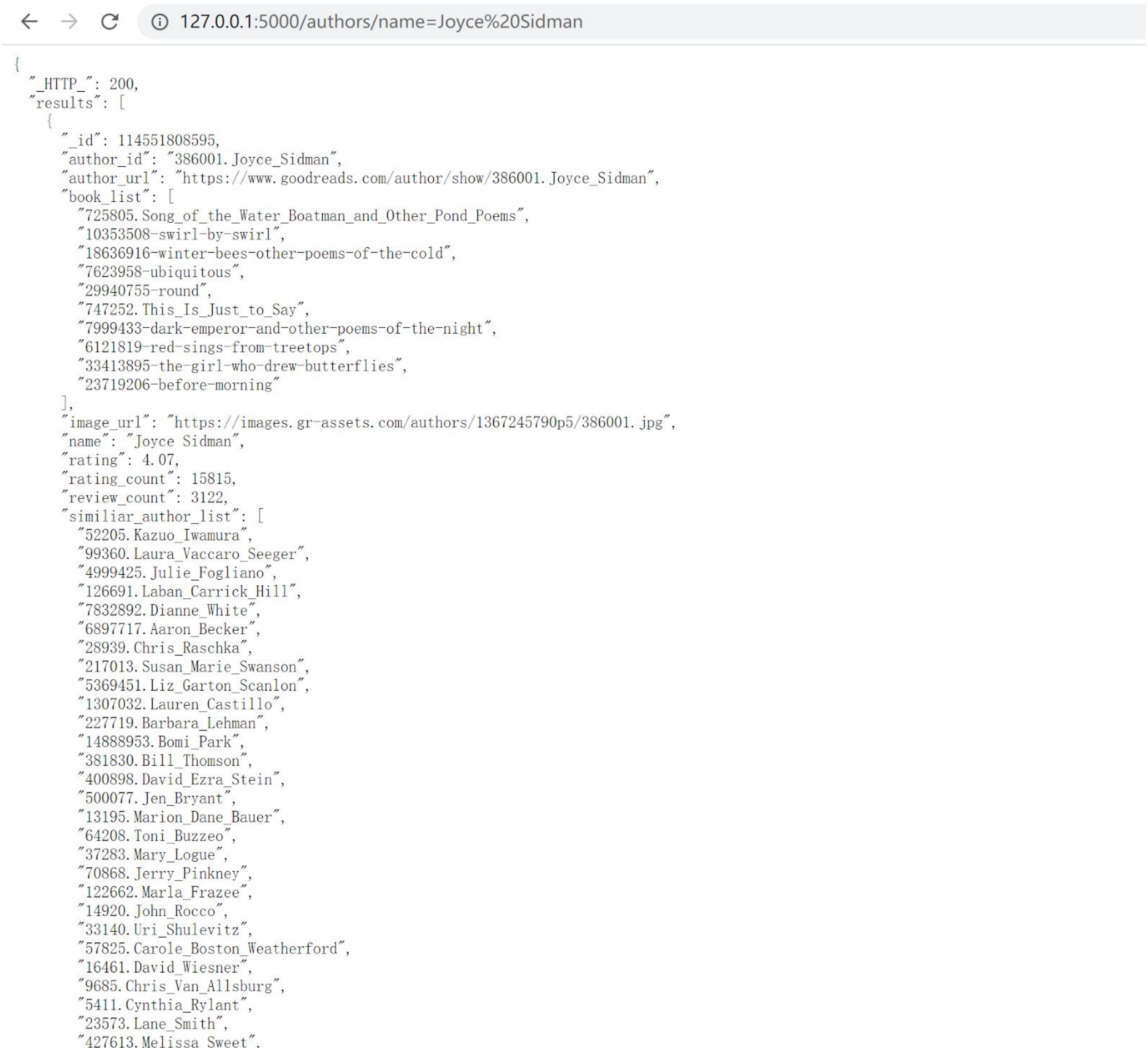


127.0.0.1:5000/books/title=Feather

```
{
  "_HTTP_": 200,
  "results": [
    {
      "_id": 98400944640,
      "author": null,
      "author_id": "74640.Jacqueline_Woodson",
      "author_url": "https://www.goodreads.com/author/show/74640.Jacqueline_Woodson",
      "book_id": "272334.Feathers",
      "book_url": "https://www.goodreads.com/book/show/272334.Feathers",
      "image_url": "https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1437018787l/272334._SY475_.jpg",
      "isbn": "9780399239892",
      "rating": 3.69,
      "rating_count": 6345,
      "review_count": 1136,
      "similiar_book_list": [
        "46285503-h-stpojkarna",
        "628218.Getting_Near_to_Baby",
        "44653684-sebbe-sa-nej",
        "42953235-s-j-vla-kallt",
        "44063819-h-lla-andan",
        "7999433-dark-emperor-and-other-poems-of-the-night",
        "89377.Penny_from_Heaven",
        "548609.Surviving_the_Applewhites",
        "6783155-the-song-of-the-whales",
        "49376276-flocken",
        "46130354-klubben",
        "49384582-inte-som-du",
        "46259585-bete-sig",
        "45134942-ber-ringen",
        "42873490-mitt-hj-rta-borde-sl-n-gon-annanstans",
        "26813380-when-friendship-followed-me-home",
        "522471.Hope_Was_Here",
        "1179370.Het_is_de_liefde_die_we_niet_begrijpen",
        "2871746-the-surrender-tree",
        "49238361-mellan-oss",
        "272334.Feathers",
        "45134859-mitt-storslagna-liv",
        "36487389-hotet",
        "45446502-tr-dhj-rta",
        "34104954-belle-prater-s-boy",
        "50903479-dyksommar",
        "43721199-kyla",
        "49392413-s-nderslagen",
        "48561138-v-nta-p-vind"
```

2) Find authors
Similar to find books, first we use /authors/ to search authors and use “search type”=”key word” to specify our search

For example http://127.0.0.1:5000/authors/name=Joyce Sidman



127.0.0.1:5000/authors/name=Joyce%20Sidman

```
{
  "_HTTP_": 200,
  "results": [
    {
      "_id": 114551808595,
      "author_id": "386001.Joyce_Sidman",
      "author_url": "https://www.goodreads.com/author/show/386001.Joyce_Sidman",
      "book_list": [
        "725805.Song_of_the_Water_Boatman_and_Other_Pond_Poems",
        "10353508-swirl-by-swirl",
        "18636916-winter-bees-other-poems-of-the-cold",
        "7623958-ubiquitous",
        "29940755-round",
        "747252.This_Is_Just_to_Say",
        "7999433-dark-emperor-and-other-poems-of-the-night",
        "6121819-red-sings-from-treetops",
        "33413895-the-girl-who-drew-butterflies",
        "23719206-before-morning"
      ],
      "image_url": "https://images.gr-assets.com/authors/1367245790p5/386001.jpg",
      "name": "Joyce Sidman",
      "rating": 4.07,
      "rating_count": 15815,
      "review_count": 3122,
      "similiar_author_list": [
        "52205.Kazuo_Iwamura",
        "99360.Laura_Vaccaro_Seeger",
        "4999425.Julie_Fogliano",
        "126691.Laban_Carrick_Hill",
        "7832892.Dianne_White",
        "6897717.Aaron_Becker",
        "28939.Chris_Raschka",
        "217013.Susan_Marie_Swanson",
        "5369451.Liz_Garton_Scanlon",
        "1307032.Lauren_Castillo",
        "227719.Barbara_Lehman",
        "14888953.Bomi_Park",
        "381830.Bill_Thomson",
        "400898.David_Ezra_Stein",
        "500077.Jen_Bryant",
        "13195.Marion_Dane_Bauer",
        "64208.Toni_Buzzeo",
        "37283.Mary_Logue",
        "70868.Jerry_Pinkney",
        "122662.Marla_Frazee",
        "14920.John_Rocco",
        "33140.Uri_Shulevitz",
        "57825.Carole_Boston_Weatherford",
        "16461.David_Wiesner",
        "9685.Chris_Van_Allsburg",
        "5411.Cynthia_Rylant",
        "23573.Lane_Smith",
        "427613.Melissa Sweet".
```

For debug, both finds authors and books will abort and return status code of 400 if

1) Wrong search type
2) No such information

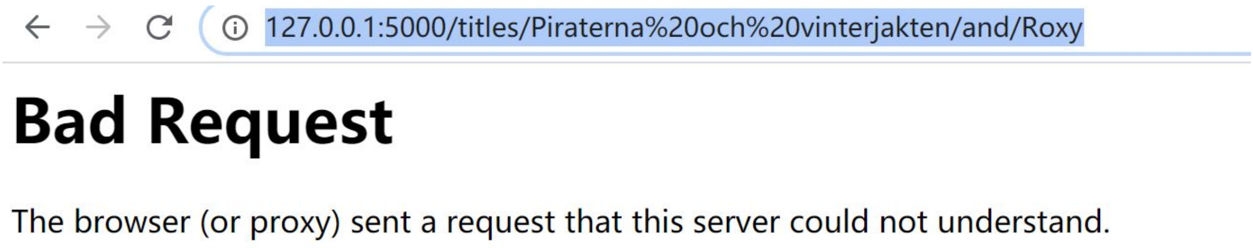
And return success code 200 if it finds at least one item.

Specific search for OR and AND

You can use following urls to search for books/authors that meet 1 requirement of 2 or meet both requirements. We suppose you will search books with titles and authors with name because those are primary keys.

And: "/titles/<string:attr>/and/<string:attr_val>" for example
http://127.0.0.1:5000/titles/Piraterna%20och%20vinterjakten/and/Roxy

Which return 400 because no such item

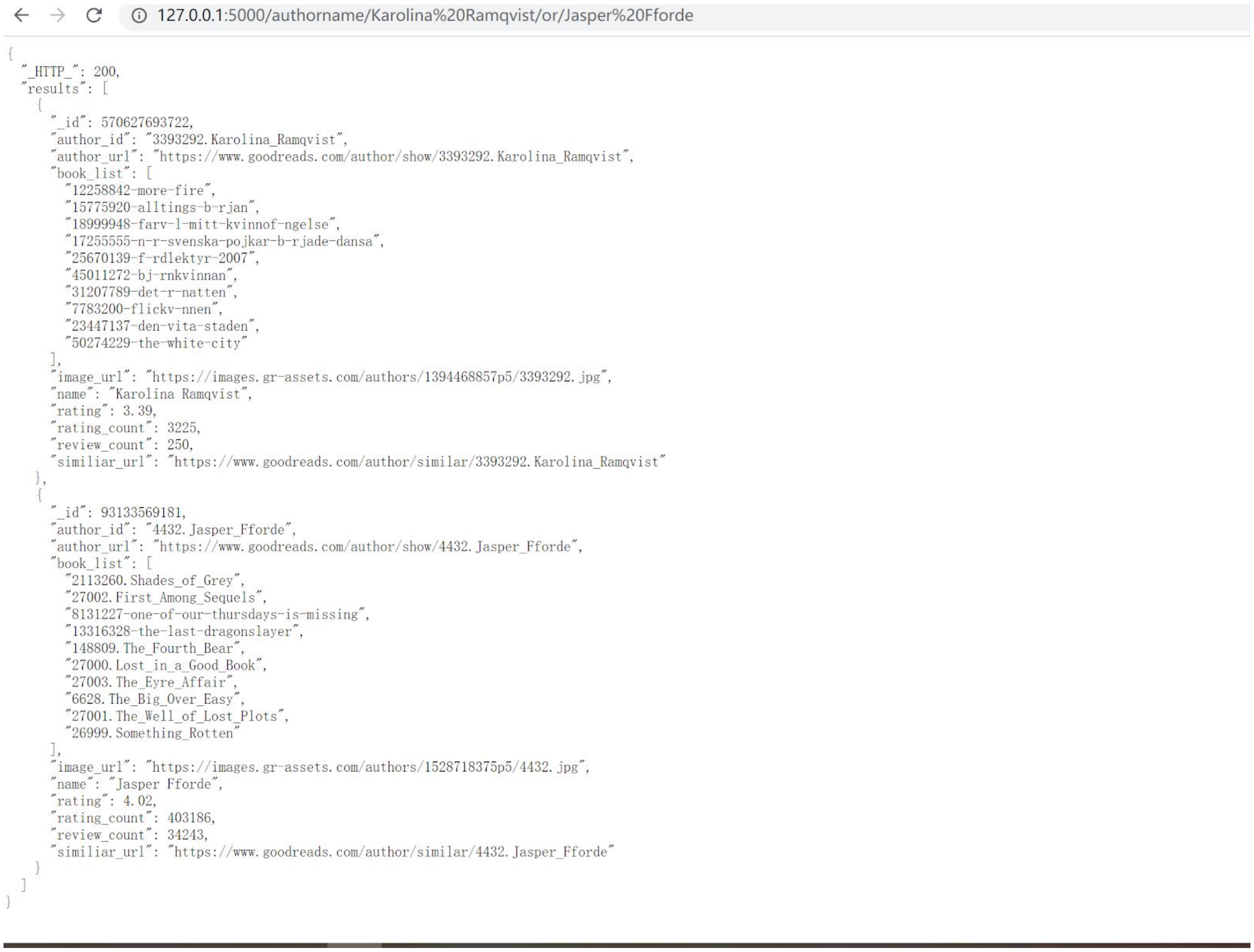


127.0.0.1:5000/titles/Piraterna%20och%20vinterjakten/and/Roxy

Bad Request

The browser (or proxy) sent a request that this server could not understand.

Or: for authors `/authorname/<string:attr>/or/<string:attr_val>`



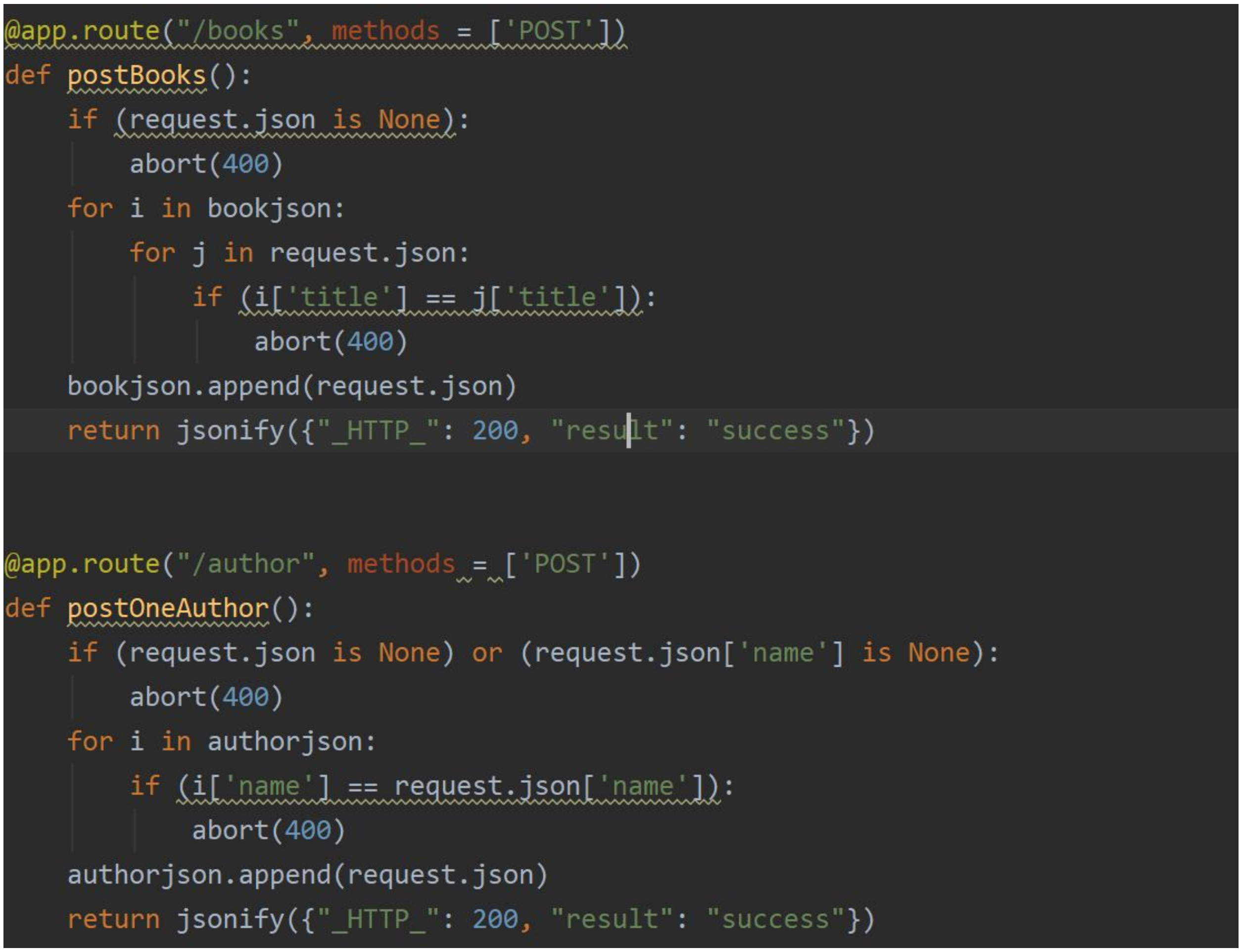
127.0.0.1:5000/authorname/Karolina%20Ramqvist/or/Jasper%20Fforde

```
{
  "_HTTP_": 200,
  "results": [
    {
      "_id": 570627693722,
      "author_id": "3393292.Karolina_Ramqvist",
      "author_url": "https://www.goodreads.com/author/show/3393292.Karolina_Ramqvist",
      "book_list": [
        "12258842-more-fire",
        "15775920-alltings-b-rjan",
        "18999948-farv-l-mitt-kvinnof-ngelse",
        "17255555-n-r-svenska-pojkar-b-rjade-dansa",
        "25670139-f-rdlektyr-2007",
        "45011272-bj-rnkvinnan",
        "31207789-det-r-natten",
        "7783200-flickv-nnen",
        "23447137-den-vita-staden",
        "50274229-the-white-city"
      ],
      "image_url": "https://images.gr-assets.com/authors/1394468857p5/3393292.jpg",
      "name": "Karolina Ramqvist",
      "rating": 3.39,
      "rating_count": 3225,
      "review_count": 250,
      "similiar_url": "https://www.goodreads.com/author/similar/3393292.Karolina_Ramqvist"
    },
    {
      "_id": 93133569181,
      "author_id": "4432.Jasper_Fforde",
      "author_url": "https://www.goodreads.com/author/show/4432.Jasper_Fforde",
      "book_list": [
        "2113260.Shades_of_Grey",
        "27002.First_Among_Sequels",
        "8131227-one-of-our-thursdays-is-missing",
        "13316328-the-last-dragonslayer",
        "148809.The_Fourth_Bear",
        "27000.Lost_in_a_Good_Book",
        "27003.The_Eyre_Affair",
        "6628.The_Big_Over_Easy",
        "27001.The_Well_of_Lost_Plots",
        "26999.Something_Rotten"
      ],
      "image_url": "https://images.gr-assets.com/authors/1528718375p5/4432.jpg",
      "name": "Jasper Fforde",
      "rating": 4.02,
      "rating_count": 403186,
      "review_count": 34243,
      "similiar_url": "https://www.goodreads.com/author/similar/4432.Jasper_Fforde"
    }
  ]
}
```

Will return information for two authors.

You can also put data to authors and books using "/author?put<string:name>"
It will request a json file and replace the author with name you type in url to your json file
It is similar for book "/books/puts<string:title>"

You can also simply add book/books, author/authors to database.
But be careful, you cannot upload existing file, or it will abort.
For multiple files , if one book is existing in database, all of posts will fail. So it is your responsibility to check info before: eg search them first.



```
@app.route("/books", methods = ['POST'])
def postBooks():
    if (request.json is None):
        abort(400)
    for i in bookjson:
        for j in request.json:
            if (i['title'] == j['title']):
                abort(400)
    bookjson.append(request.json)
    return jsonify({"_HTTP_": 200, "result": "success"})


@app.route("/author", methods = ['POST'])
def postOneAuthor():
    if (request.json is None) or (request.json['name'] is None):
        abort(400)
    for i in authorjson:
        if (i['name'] == request.json['name']):
            abort(400)
    authorjson.append(request.json)
    return jsonify({"_HTTP_": 200, "result": "success"})
```

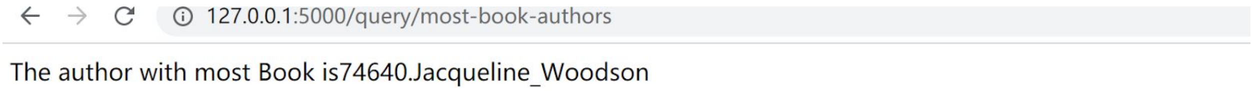
You can also delete them through titles and names:

Eg: "/book?<string:title>" for books
And /author?<string:name> for authors

Find specific data and visualization:

/query/most-book-authors: will give you the authors with most related books in database

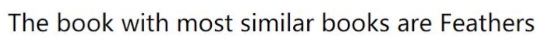
Which will show



127.0.0.1:5000/query/most-book-authors

```
The author with most Book is74640.Jacqueline_Woodson
```

/query/most-similar-book : will give books with most similar book in database

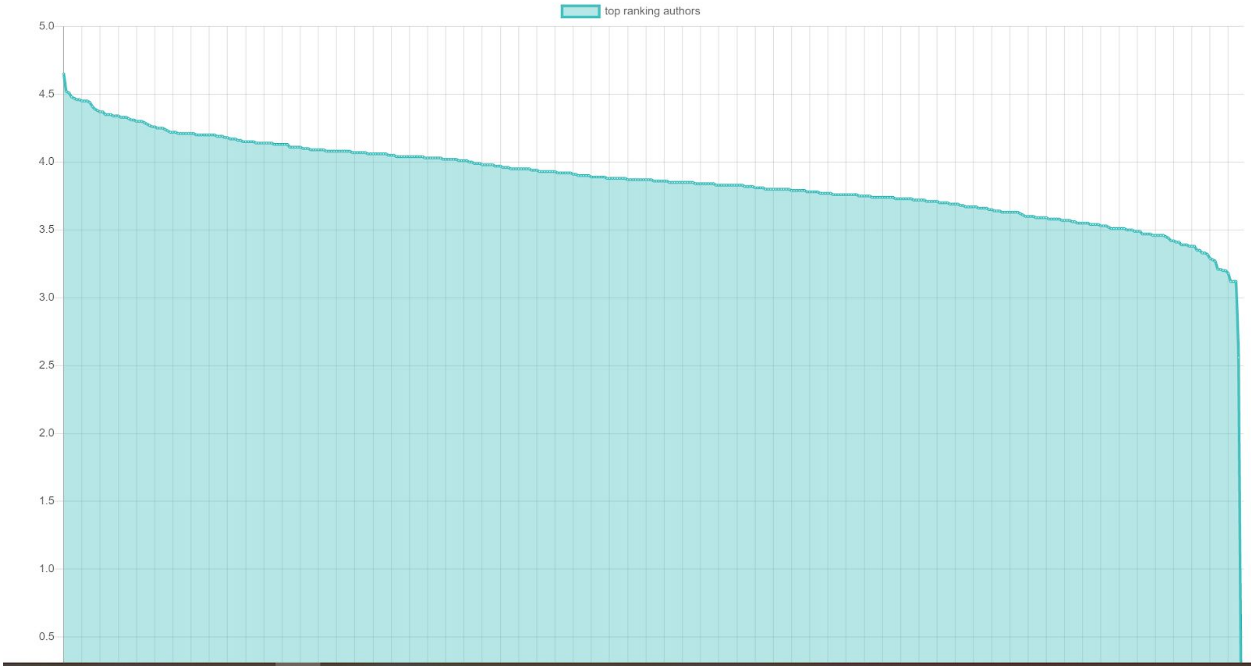


The book with most similar books are Feathers

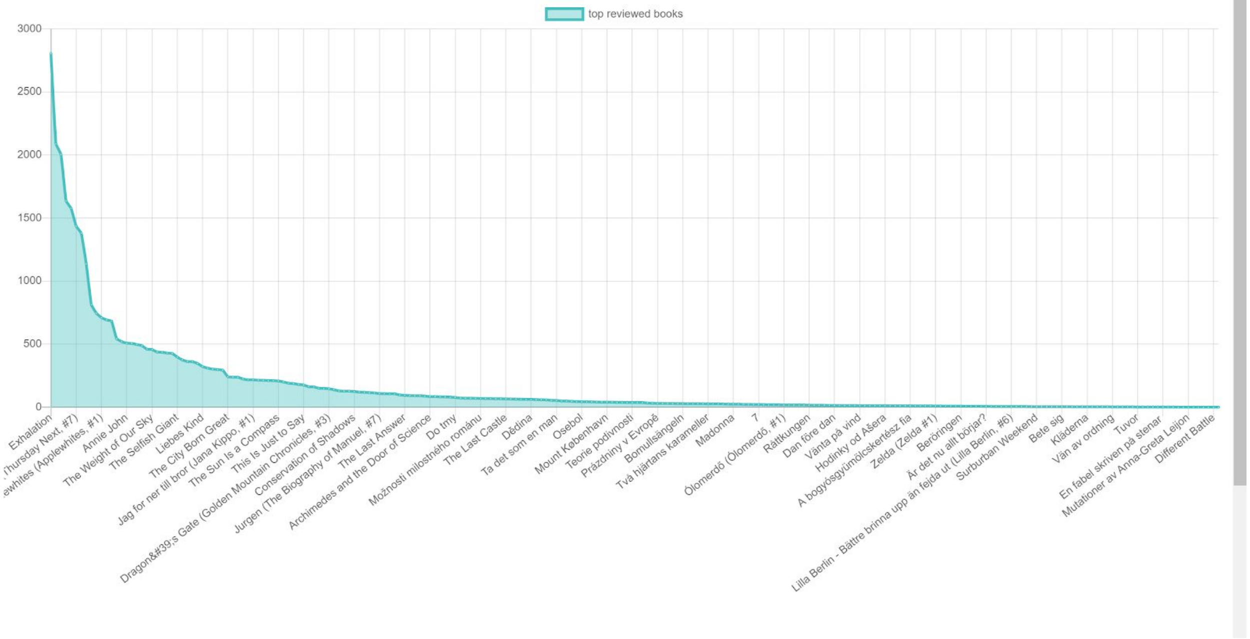
We can also visualize our data by ratings for author and book thorough

/author_chart and /book_chart

Rating Line Chart

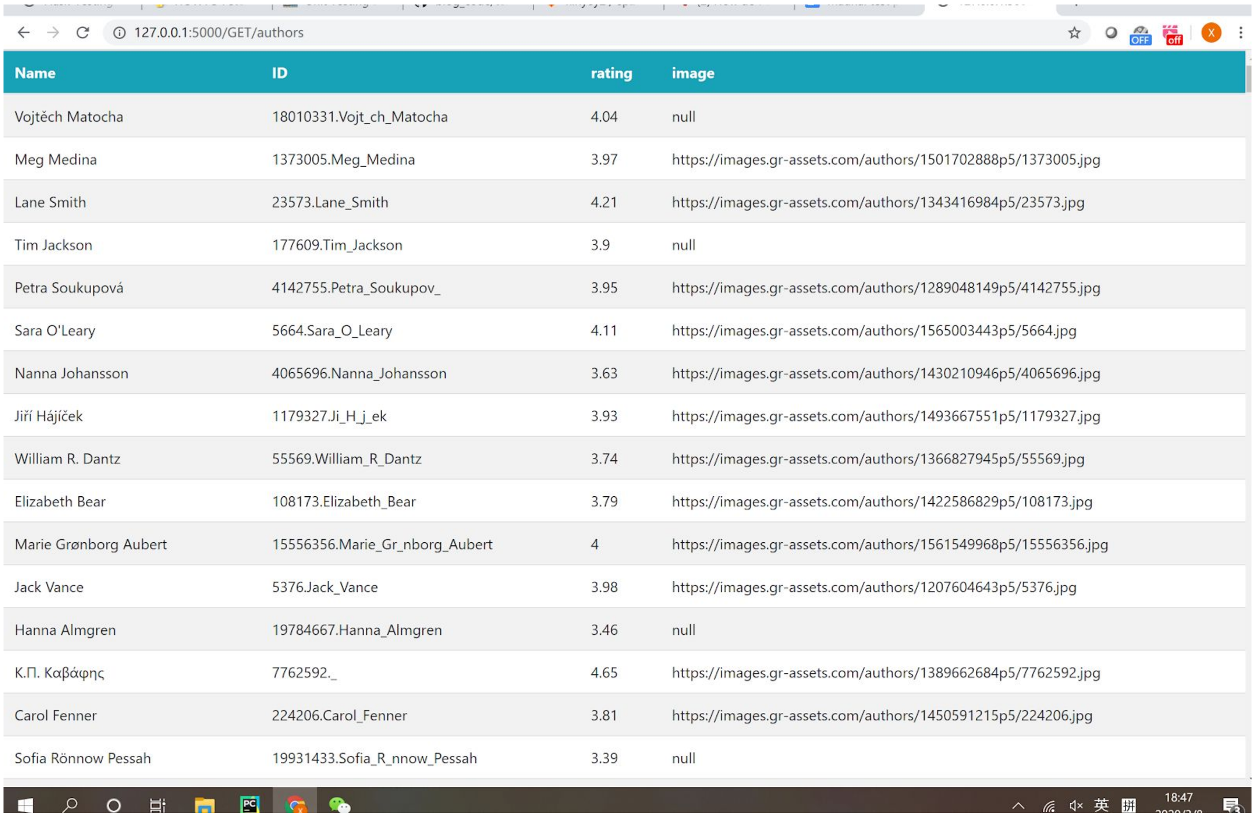


Rating Line Chart

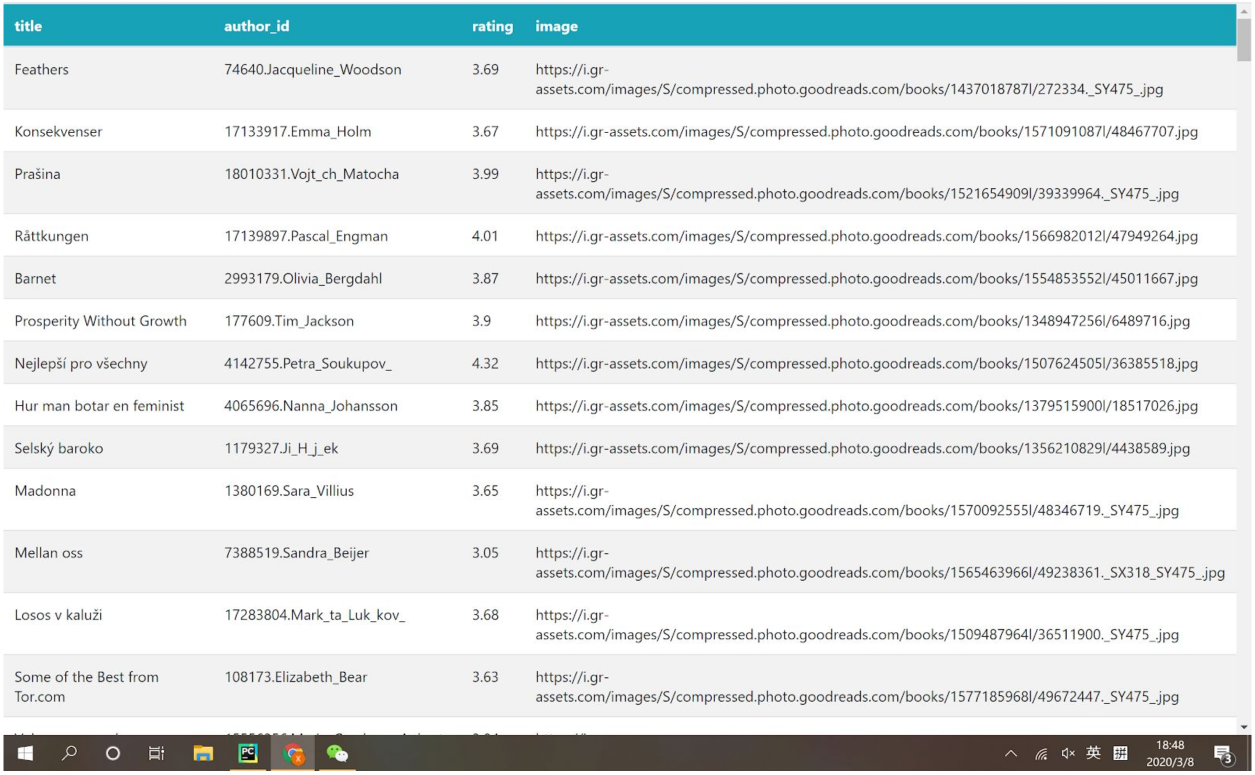


Finally, you can view all books and authors with flitered key information at any time thorough

/GET/authors and GET/books



Name	ID	rating	image
Vojtěch Matocha	18010331.Vojt_ch_Matocha	4.04	null
Meg Medina	1373005.Meg_Medina	3.97	https://images.gr-assets.com/authors/1501702888p5/1373005.jpg
Lane Smith	23573.Lane_Smith	4.21	https://images.gr-assets.com/authors/1343416984p5/23573.jpg
Tim Jackson	177609.Tim_Jackson	3.9	null
Petra Soukupová	4142755.Petra_Soukupov_	3.95	https://images.gr-assets.com/authors/1289048149p5/4142755.jpg
Sara O'Leary	5664.Sara_O_Leary	4.11	https://images.gr-assets.com/authors/1565003443p5/5664.jpg
Nanna Johansson	4065696.Nanna_Johansson	3.63	https://images.gr-assets.com/authors/1430210946p5/4065696.jpg
Jiří Hájíček	1179327.Ji_H_j_ek	3.93	https://images.gr-assets.com/authors/1493667551p5/1179327.jpg
William R. Dantz	55569.William_R_Dantz	3.74	https://images.gr-assets.com/authors/1366827945p5/55569.jpg
Elizabeth Bear	108173.Elizabeth_Bear	3.79	https://images.gr-assets.com/authors/1422586829p5/108173.jpg
Marie Grønborg Aubert	15556356.Marie_Gr_nborg_Aubert	4	https://images.gr-assets.com/authors/1561549968p5/15556356.jpg
Jack Vance	5376.Jack_Vance	3.98	https://images.gr-assets.com/authors/1207604643p5/5376.jpg
Hanna Almgren	19784667.Hanna_Almgren	3.46	null
Κ.Π. Καβάφης	7762592._	4.65	https://images.gr-assets.com/authors/1389662684p5/7762592.jpg
Carol Fenner	224206.Carol_Fenner	3.81	https://images.gr-assets.com/authors/1450591215p5/224206.jpg
Sofia Rönnow Pessah	19931433.Sofia_R_nnow_Pessah	3.39	null



title	author_id	rating	image
Feathers	74640.Jacqueline_Woodson	3.69	https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1437018787l/272334._SY475_.jpg
Konsekvenser	17133917.Emma_Holm	3.67	https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1571091087l/48467707.jpg
Prašina	18010331.Vojt_ch_Matocha	3.99	https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1521654909l/39339964._SY475_.jpg
Råttkungen	17139897.Pascal_Engman	4.01	https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1566982012l/47949264.jpg
Barnet	2993179.Olivia_Bergdahl	3.87	https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1554853552l/45011667.jpg
Prosperity Without Growth	177609.Tim_Jackson	3.9	https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1348947256l/6489716.jpg
Nejlepší pro všechny	4142755.Petra_Soukupov_	4.32	https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1507624505l/36385518.jpg
Hur man botar en feminist	4065696.Nanna_Johansson	3.85	https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1379515900l/18517026.jpg
Selský baroko	1179327.Ji_H_j_ek	3.69	https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1356210829l/4438589.jpg
Madonna	1380169.Sara_Villius	3.65	https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1570092555l/48346719._SY475_.jpg
Mellan oss	7388519.Sandra_Beijer	3.05	https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1565463966l/49238361._SX318_SY475_.jpg
Losos v kaluži	17283804.Mark_ta_Luk_kov_	3.68	https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1509487964l/36511900._SY475_.jpg
Some of the Best from Tor.com	108173.Elizabeth_Bear	3.63	https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1577185968l/49672447._SY475_.jpg

In test folder, there exists sufficient test, just run them one by one for testing.