Book library API list

Table of Contents

[GET BOOK INFORMATION RELATED APIs 1](#_Toc175689112)

[1. Get book information by book name. 1](#_Toc175689113)

[2. Search book by contains. 2](#_Toc175689114)

[3. ~~Search book by StartWith~~ 2](#_Toc175689115)

[3. Fuzzy Search book 2](#_Toc175689116)

[~~4. Find has similar book name~~ 3](#_Toc175689117)

[~~5. Get book by category~~ 3](#_Toc175689118)

[6. Get books by author 3](#_Toc175689119)

[~~6. Get books by publisher~~ 4](#_Toc175689120)

[~~7. Get books by published year~~ 4](#_Toc175689121)

[8. Get books by pages and limits 4](#_Toc175689122)

[9. Get books counts info 4](#_Toc175689123)

[10. Delete book node 4](#_Toc175689124)

[11. Add books 5](#_Toc175689125)

[12. Modify book 5](#_Toc175689126)

[Not find error. 6](#_Toc175689127)

[GET AUTHORS RELATED APIs 6](#_Toc175689128)

[1. Get authors counts 6](#_Toc175689129)

[2. Get author info 6](#_Toc175689130)

[~~3. Get similar authors~~ 6](#_Toc175689131)

[4. Get authors by contains 6](#_Toc175689132)

[~~5. Get authors by startwith~~ 7](#_Toc175689133)

[~~OTHER APIs~~ 7](#_Toc175689134)

[~~1. Get all categories~~ 7](#_Toc175689135)

[~~2. Get all published year~~ 7](#_Toc175689136)

# GET BOOK INFORMATION RELATED APIs

## 1. Get book information by book name.

GET: http://127.0.0.1:8000/api/books/?action=get-by-name&name=Bruce Lee: The Incomparable Fighter

REPLY:

{

"title": "Bruce Lee: The Incomparable Fighter",

"authors": [

{

"name": "M. Uyehara",

"link": "/authors/M. Uyehara"

}

]

}

## 2. Search book by contains.

GET: http://127.0.0.1:8000/api/books/?action=get-by-contain-name&name=Bruce lee

REPLY:

{

"books": [

{

"title": "Bruce Lee: The Incomparable Fighter",

"authors": [

{

"name": "M. Uyehara",

"link": "/authors/M. Uyehara"

}

]

},

{

"title": "The Bruce Lee Story",

"authors": [

{

"name": "Mike Lee",

"link": "/authors/Mike Lee"

},

{

"name": "Linda Lee",

"link": "/authors/Linda Lee"

}

]

}

]

}

## 3. ~~Search book by StartWith~~

~~GET: http://127.0.0.1:8000/api/books/?action=get-book-by-start-with&name=The Flying&ignorecase=false~~

~~REPLY:~~

## 3. Fuzzy Search book

Introduction: get all books, filter book tile by contain, filter author by contain

GET: http://127.0.0.1:8000/api/books/?action=fuzzy-search-book&author=Uyehara&book=Bruce

REPLY:

{

"books": [

{

"title": "Bruce Lee: The Incomparable Fighter",

"authors": [

{

"name": "M. Uyehara",

"link": "/authors/M. Uyehara"

}

]

}

]

}

## ~~4. Find has similar book name~~

~~Introduction: if cannot any book, please use this API (need consider the threshold )~~

~~GET: http://127.0.0.1:8000/api/books/?action=find-similar-books&name=Bruce Lee The Incomparable&threshold=50~~

~~REPLY:~~

## ~~5. Get book by category~~

~~Introduction: the category name must correct.~~

~~GET:~~ [~~http://127.0.0.1:8000/api/books/?action=get-book-by-link&link=category&category-name=Architecture~~](http://127.0.0.1:8000/api/books/?action=get-book-by-link&link=category&category-name=Architecture)

~~REPLY:~~

## 6. Get books by author

Introduction: the author name must correct.

GET: http://127.0.0.1:8000/api/books/?action=get-book-by-author&author-name=M. Uyehara

REPLY:

{

"books": [

{

"title": "Bruce Lee: The Incomparable Fighter",

"authors": [

{

"name": "M. Uyehara",

"link": "/authors/M. Uyehara"

}

]

}

]

}

## ~~6. Get books by publisher~~

~~Introduction: the publisher name must correct.~~

~~GET: http://127.0.0.1:8000/api/books/?action=get-book-by-link&link=publisher&publisher-name=Tehuti Research Foundation~~

REPLY:

## ~~7. Get books by published year~~

~~Introduction: If the start year < end year, return to ascending order. If the start year > end year, return to descending order.~~

~~GET: http://127.0.0.1:8000/api/books/?action=get-book-by-link&link=publishedyear&end-year=1986&start-year=1987~~

REPLY:

## 8. Get books by pages and limits

~~Introduction: order-by total support 3 types [‘title’, ‘ratingCounts’, ‘publishDate’]~~

GET: http://127.0.0.1:8000/api/books/?action=get-all-books&page=1&limit=5

REPLY:

{

"books": [

{

"title": "The Church of Christ: A Biblical Ecclesiology for Today",

"authors": [

{

"name": "Everett Ferguson",

"link": "/authors/Everett Ferguson"

}

]

},

{

"title": "The Overbury affair (Avon)",

"authors": [

{

"name": "Miriam Allen De Ford",

"link": "/authors/Miriam Allen De Ford"

}

]

},

{

"title": "A Walk in the Woods: a Play in Two Acts",

"authors": [

{

"name": "Lee Blessing",

"link": "/authors/Lee Blessing"

}

]

},

{

"title": "Saint Hyacinth of Poland",

"authors": [

{

"name": "Mary Fabyan Windeatt",

"link": "/authors/Mary Fabyan Windeatt"

}

]

},

{

"title": "Rising Sons and Daughters: Life Among Japan's New Young",

"authors": [

{

"name": "Steven Wardell",

"link": "/authors/Steven Wardell"

}

]

}

]

}

## 9. Get books counts info

GET: http://127.0.0.1:8000/api/books/?action=get-books-counts

REPLY:

{

"count": 4000

}

## 10. Delete book node

POST (example): curl --location 'http://127.0.0.1:8000/api/books/?action=delete-book' \

--header 'Content-Type: application/json' \

--**data** '{

"book-title": "The Bruce Lee Story"

}'

REPLY:

{

"error": 0

}

## 11. Add books

POST (example): curl --location 'http://127.0.0.1:8000/api/books/?action=add-books' \

--header 'Content-Type: application/json' \

--**data** ' {

"add-books":

[{

"title": "Bruce Lee: The Incomparable Fighter",

"authors": [ "M. Uyehara"]

},

{

"title": "The Bruce Lee Story",

"authors": ["Mike Lee","Linda Lee"]

}]

}'

Reply: success 🡺

{

"error": 0

}

Fail 🡪

{

"error": 1,

"exist-books": [

{

"title": "Bruce Lee: The Incomparable Fighter",

"authors": "M. Uyehara"

},

{

"title": "The Bruce Lee Story",

"authors": "Mike Lee, Linda Lee"

}

]

}

## 12. Modify book

POST: curl --location 'http://127.0.0.1:8000/api/books/?action=update-book' \

--header 'Content-Type: application/json' \

--**data** ' {

"book-title": "Bruce Lee: The Incomparable Fighter",

"authors": [ "H. Uyehara"]

}'

REPLY:

{

"error": 0

}

## Not find error.

REPLY:

{

"error": "Cannot find any book"

}

# GET AUTHORS RELATED APIs

## 1. Get authors counts

GET: http://127.0.0.1:8000/api/authors/?action=get-authors-count

REPLY:

{

"count": 2398

}

## 2. Get author info

GET: http://127.0.0.1:8000/api/authors/?action=get-author-info&name=Lee Blessing

REPLY:

{

"author": {

"name": "Lee Blessing",

"similar\_authors": [],

"books": [

"A Walk in the Woods: a Play in Two Acts"

]

}

}

## ~~3. Get similar authors~~

~~GET: http://127.0.0.1:8000/api/others/?action=get-similar-authors&name=Richard Overy&threshold=75~~

REPLY:

## 4. Get authors by contains

GET: http://127.0.0.1:8000/api/others/?action=get-by-contain-name&name=Mike

REPLY:

{

"authors": [

{

"name": "Mike Lee",

"similar\_authors": [],

"books": [

"The Bruce Lee Story"

]

},

{

"name": "Mike Murdock",

"similar\_authors": [],

"books": [

"The Leadership Secrets of Jesus"

]

}

]

}

## ~~5. Get authors by startwith~~

~~GET: http://127.0.0.1:8000/api/others/?action=get-by-start-with-name&name=Richard B&ignorecase=false~~

REPLY:

# ~~OTHER APIs~~

## ~~1. Get all categories~~

GET: http://127.0.0.1:8000/api/others/?action=get-all-categories

REPLY:

## ~~2. Get all published year~~

~~GET: http://127.0.0.1:8000/api/others/?action=get-all-published-year~~

REPLY: