Search Books:

API Address: <http://bookshop.sion-project.ir/api/search/searchBooks.php>

Method: POST, GET

Inputs:

Access Token:  
existence: required  
key: “access\_token”  
validate by: Access Token Validation Policy

Book IDs:  
existence: optional  
key: “book\_ids”  
validate by: Book ID Validation policy

Title:  
existence: optional  
key: “title”  
validate by: Title Validation policy

Price:  
existence: optional  
key: “price”  
validate by: Price Validation policy

Owner:  
existence: optional  
key: “owner”  
validate by: User ID Validation policy

Writer:  
existence: optional  
key: “writer”  
validate by: Writer Validation policy

Instructors:  
existence: optional  
key: “instructors”  
validate by: Instructor ID Validation policy  
value: [a JSON Array contains requested instructions books.]

Semesters:  
existence: optional  
key: “semesters”  
validate by: Semester Validation policy  
value: [a JSON Array contains requested semesters books.]

Courses:  
existence: optional  
key: “courses”  
validate by: Course ID Validation policy  
value: [a JSON Array contains requested courses books.]

Majors:  
existence: optional  
key: “majors”  
validate by: Major ID Validation policy  
value: [a JSON Array contains requested majores books.]

Start:  
existence: optional  
key: “start”  
validate by: Start Validation Policy  
value: [Default is 0]

Step:  
existence: optional  
key: “step”  
validate by: Step Validation Policy  
value: [Default is 30]

Success Output:

Code:  
existence: fix  
key: “code”  
value: 7898

Message:  
existence: fix  
key: “message”  
value: “Success!”

Results Number:  
existence: fix  
key: “size”  
value: [Number of all results.]

Books:  
existence: fix  
key: “books”  
value: [a JOSN Array contains search result books.]  
sample: [ {“book\_id”:213 , “title”:”Math” , “description”:”some description” , “price”: 12 , “owner”:315 , “writer”:”John Doe, Will Diaz” ,  
“files”:[“bookshop.com/x”, “bookshop.com/y”] ,  
“images”:[“bookshop.com/a”, “bookshop.com/b”],  
“instructors”:[“Sayyad Najafi”, “Nosratali Ashrafi”],  
“courses”:[“Database Systems”, “Advanced Programming”],  
“semesters”:[“Fall 2019”,”Winter 2020”],  
“majors”: [“Computer Engineering”, “Biology”],  
“created\_at”:”2020-02-15 21:42:50” , “updated\_at”:”2020-02-15 21:42:50”} ]

Failure Output:

Code:  
existence: fix  
key: “code”  
value: 107, 207, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 310, 311, 316, 317, 318, 319

Message:  
existence: fix  
key: “message”  
value:  
107: “Lack of Access Token Input.”  
207: “Invalid Access Token Input.”  
215: “Invalid Title Input.”  
216: “Invalid Price Input.”  
217: “Invalid Owner Input.”  
222: “Invalid Start Input.”  
223: “Invalid Step Input.”  
224: “Invalid Writer Input.”  
310: “Access Token has been expired.”  
311: “Wrong Access Token.”  
405: “Something Went Wrong.”