|  |  |  |
| --- | --- | --- |
| q | Full-text query string. | * When creating a query, list search terms separated by a '+', in the form q=term1+term2\_term3. (Alternatively, you can separate them with a space, but as with all of the query parameter values, the spaces must then be URL encoded.) The API returns all entries that match all of the search terms (like using AND between terms). Like Google's web search, the API searches on complete words (and related words with the same stem), not substrings. * To search for an exact phrase, enclose the phrase in quotation marks: q="exact phrase". * To exclude entries that match a given term, use the form q=-term. * The search terms are case-insensitive. * Example: to search for all entries that contain the exact phrase "Elizabeth Bennet" and the word "Darcy" but don't contain the word "Austen", use the following query parameter value: q="Elizabeth+Bennet"+Darcy-Austen * There are special (case-sensitive) keywords you can specify in the search terms to search in particular fields, such as:   + intitle: Returns results where the text following this keyword is found in the title.   + inauthor: Returns results where the text following this keyword is found in the author.   + inpublisher: Returns results where the text following this keyword is found in the publisher.   + subject: Returns results where the text following this keyword is listed in the category list of the volume.   + isbn: Returns results where the text following this keyword is the ISBN number.   + lccn: Returns results where the text following this keyword is the Library of Congress Control Number.   + oclc: Returns results where the text following this keyword is the Online Computer Library Center number. |

Here is an example of searching for Daniel Keyes' "Flowers for Algernon":

GET https://www.googleapis.com/books/v1/volumes?q=flowers+inauthor:keyes&[**key**](https://developers.google.com/books/docs/v1/using#key)=yourAPIKey

**data**:

**items**: array

0

**volumeInfo**:

**authors**: Array(1)

**imageLinks**:

**smallThumbnail**:

**previewLink**:

**title**:

{*/\* {bookResults.map((result) => {*

*return<div key = {result.id}>*

*<h1>{bookResults.kind}</h1>*

*{bookResults.items.map((item,index) => {*

*return <span key= {index}> {item.volumeInfo}</span>*

*})}*

*</div>*

*})} \*/*}