

Citation Generation API Contract

Scope & UI:  Citation generator

Usecase:

To generate citations based on the below 4 options:

- Import from Zotero
- Free Search
- Add a new item to Zotero Library based on the free search
- Enter data manually - TBD
- Import From file - *will be picked up in the next phase*

API Templates:

Option 1: Import from Zotero

- Here the user will be redirected to the zotero login page
- After successful login, user will be asked to create a new API key which can have read/write access to the users library / collections
- An auth token will be generated to authenticate the user with the help of OAuth v1 flow.

API endpoint to generate auth token : BASE URL + ' /zotauth'

Method: 'POST'

Request Body:

```
1 {
2   "name": "zotero",
3   "consumer_key": "a3XXXXXXXXXX",
4   "consumer_secret": "1cXXXXXXXXXX",
5   "request_token_url": "https://www.zotero.org/oauth/request",
6   "access_token_url": "https://www.zotero.org/oauth/access",
7   "authorize_url": "https://www.zotero.org/oauth/authorize",
8   "base_url": "https://api.zotero.org"
9 }
10
```

Response Body:

```
1 {
2   "request_token": "<request_token>",
3   "request_token_secret": "<request_token_secret>",
4   "authorize_url": "<authorize_url>"
5 }
```

API endpoint to authenticate user : BASE URL + ' /verify'

Method: 'POST'

Request Body: To authenticate the user

```
1 {
2   "token": "<token>",
3   "request_token": "<request_token>",
4   "request_token_secret": "<request_token_secret>"
5 }
```

Response Body: Success

```
1 [
2   {
3     'key': 'TESF7ITX',
4     'version': 106,
5     'library': {
6       'type': 'user',
7       'id': 11222345,
8       'name': 'govindrs',
9       'links': {
10        'alternate': {
11          'href': 'https://www.zotero.org/govindrs',
12          'type': 'text/html'
13        }
14      }
15    },
16    'links': {
17      'self': {
18        'href': 'https://api.zotero.org/users/11222345/items/TESF7ITX',
19        'type': 'application/json'
20      },
21      'alternate': {
22        'href': 'https://www.zotero.org/govindrs/items/TESF7ITX',
23        'type': 'text/html'
24      }
25    },
26    'meta': {
27      'numChildren': 0
28    },
29    'data': {
30      'key': 'TESF7ITX',
31      'version': 106,
32      'itemType': 'webpage',
33      'title': 'Csl search by example',
34      'creators': [
35
36      ],
37      'abstractNote': '',
38      'websiteTitle': '',
39      'websiteType': '',
40      'date': '',
41      'shortTitle': '',
42      'url': 'https://editor.citationstyles.org/searchByExample/',
43      'accessDate': '2023-03-29T06:54:41Z',
44      'language': '',
45      'rights': '',
46      'extra': '',
47      'tags': [
48
```

```

49     ],
50     'collections': [
51         'E63G4WCI'
52     ],
53     'relations': {
54
55     },
56     'dateAdded': '2023-03-29T06:56:41Z',
57     'dateModified': '2023-03-29T06:56:41Z'
58 }
59 },
60 ]
61

```

Response Body: Failure

```

1 {
2   "message": "Error in Zotero OAuth token exchange"
3 }
4

```

Option 2: Free Search

- Here the user can generate citations by entering either a DOI, journal/book title or PMID (pubmed_id) in the search bar
- Based on the input we will search the data in our Dyson DB using Elastic Search query
- The user will need to specify a citation style here which is a mandatory parameter

API endpoint : BASE URL + ' /search'

Method: 'POST'

Headers: *mandatory*

```

1 {
2   Content-Type: application/json
3 }

```

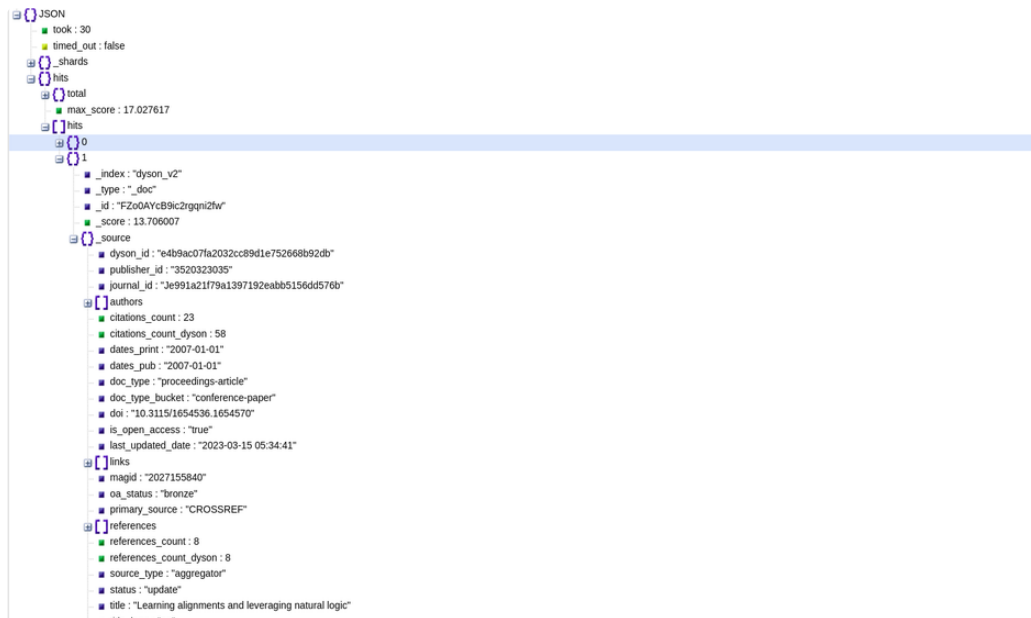
Request Body:

```

1 {
2   "index": "dyson_v2",
3   "payload": {
4     "query": {
5       "bool": {
6         "must": [
7           {
8             "match": {
9               "doi": "<doi>"
10            }
11          }
12        ]
13      }
14    }
15  }
16 }
17

```

Response Body: Since dyson success response is too big, have just added a screenshot



Option 3: Based on the free search a user can add a new item to its's Zotero library if required

API Endpoint: BASE URL + '/zotero/items/create'

Method: POST

Headers: *mandatory*

```
1 {
2   Authorization: <Zotero api_key generated in step 1>
3   Content-Type: application/json
4 }
```

Request Body:

```
1 {
2   "items": [
3     {
4       "itemType": "<itemType>",
5       "title": "<title>",
6       "creators": [
7         {
8           "firstName": "<firstName>",
9           "lastName": "<lastName>",
10          "creatorType": "<creatorType>"
11        }
12      ],
13       "publicationTitle": "<publicationTitle>",
14       "date": "<date>",
15       "volume": "<volume>",
16       "pages": "<pages>",
17       "DOI": "<doi>"
18     }
19   ]
20 }
```

Response Body:

```

1 {
2   "success": true,
3   "successful": [
4     {
5       "key": "<item_key>"
6     }
7   ],
8   "failed": []
9 }
10

```

Option 4: Enter data manually

- Here the user will enter the data in the form provided. Based on the different article types the form fields will vary.

API Endpoint: BASE URL + '/<article_type>_form'

Method: POST

Headers: *mandatory*

```

1 {
2   Authorization: <Zotero api_key generated in step 1>
3   Content-Type: application/json
4 }

```

Request Body: based on the article type the request body will have varying keys

```

1 {
2   "fetch_id": "",
3   "article_type": "journal/book/website"
4   "citation_style": "",
5   "title": "",
6   "author": "",
7   ...
8   ...
9   ...
10 }

```

Response body: Similar to the common response body defined below

Common Request Body - *TBD*

API endpoint: BASE URL + '/citation_generator'

Headers: *mandatory*

```

1 {
2   "Content-Type": "application/json",
3   "Authorization": "<static_bearer_token>"
4 }

```

Request Body:

```

1 {
2     "fetch_id": "",
3     "user_type": "zotero/non-zotero",      ## mandatory
4     "user_data": {                        ## this will be empty in case of non zotero users
5         "id": "11222345",
6         "name": "govindrs",
7         "item_key": "9UXLMG8V"
8         "api_key": "<api_key>"
9     },
10    "citation_style": "APA",    ## this is a mandatory key
11    "doi/title/pubmedis": "doi/title/35007113",    # pass any one of these
12 }

```

Citations styles

- Based on the citation style given by the user we will be using citproc python package to generate citations for the data received from zotero and dyson.
- We need to design the input in the below format for citeproc to generate citations for any given data/input

```

1 citeproc_input_json = ''
2 [
3     {
4         "author": [
5             {
6                 "family": "Schmidhuber",
7                 "given": "Jürgen"
8             }
9         ],
10        "id": "ITEM-4",
11        "issued": {
12            "date-parts": [
13                [2015]
14            ]
15        },
16        "title": "Deep learning in neural networks: An overview",
17        "container-title": "Neural Networks",
18        "volume": "61",
19        "page": "85-117",
20        "type": "article-journal",
21        "DOI": "10.1016/j.neunet.2014.09.003",
22        "URL": "https://doi.org/10.1016%2Fj.neunet.2014.09.003",
23        "publisher": "Elsevier {BV}"
24    },
25    {
26        "id": "ITEM-5",
27        "type": "book",
28        "call-number": "SB419.5 .D85 2008",
29        "edition": "Rev. and updated ed., [2nd ed.]",
30        "event-place": "Portland, Or",
31        "ISBN": "978-0-88192-911-9",
32        "number-of-pages": "328",
33        "publisher": "Timber Press",
34        "publisher-place": "Portland, Or",
35        "source": "Library of Congress ISBN",
36        "title": "Planting green roofs and living walls",
37        "author": [
38            {

```

```

39         "family": "Dunnett",
40         "given": "Nigel"
41     },
42     {
43         "family": "Kingsbury",
44         "given": "Noël"
45     }
46 ],
47 "issued": {
48     "date-parts": [
49         [
50             "2008"
51         ]
52     ]
53 }
54 ]
55 '''

```

Common Response Body

- Response Status 200

```

1  {
2      "status": "success",
3      "fetch_id": "",
4      "request_data": {
5          "user_type": "zotero/non-zotero",      ## mandatory
6          "user_data": {      ## this will be empty in case of non zotero users
7              "id": "11222345",
8              "name": "govindrs",
9              "item_key": "9UXLMG8V"
10             "api_key": "<api_key>"
11         },
12         "citation_style": "APA",      ## this is a mandatory key
13         "doi/title/pubmedis": "doi/title/35007113",      # pass any one of these
14     },
15     "response_data": {
16         "items": [{
17             "item_id": "1",
18             "item_data": {},
19             "citations": "",
20             "biblography": "",
21             "scite_badge": ""
22         },
23         {
24             "item_id": "2",
25             "item_data": {},
26             "citations": "",
27             "biblography": "",
28             "scite_badge": ""
29         }
30     ]
31 }
32 }
33

```

- Response Status 400

```
1 {
2   "status": "failed",
3   "message": "Invalid request id provided."
4 }
```

- Response Status 401

```
1 {
2   "status": "failed",
3   "message": "Authorization failed."
4 }
```

- Response Status 403

```
1 {
2   "status": "failed",
3   "message": "Missing Authorization token."
4 }
```

Scite Badge

 447  28  317  6

```
1 <div class="scite-badge"
2   data-doi="10.1016/j.biopsy.2005.08.012"
3   data-layout="horizontal"
4   data-show-zero="false"
5   data-small="false"
6   data-show-labels="false"
7   data-tally-show="true">
8 </div>
9 <script async type="application/javascript" src="https://cdn.scite.ai/badge/scite-badge-latest.min.js">
10 </script>
```

TO DO:

- ☐ Create compound citations
- ☐ Edit citations dynamically