Phase1: API doc for zotero flow

API 1: Login to Zotero and display the user's library items

API endpoint test server : https://58wk9x3q84.execute-api.us-east-1.amazonaws.com/test/api/v1/zotero/items

Method: 'POST'

Headers: *mandatory*

```
{
  "Content-Type": "application/json",
  "x-api-key": "DTLIRTOGIp2iUA1QnXp9S8qGwCCFs9vB58eEGq3E"
}
```

Status: 400 Bad Request

```
{
    "status": false,
    "message": "Missing required headers and/or parameters."
}
```

Request payload:

Request will be failed if any of the below keys/ values are missing.

```
{
  "user_type": "zotero",  # non-mandatory
  "user_data": {
    "user_id": "12345",
    "user_name": "xyz",
    "api_key": "123xyz"
  }
}
```

Status: 400 Bad Request

```
{
    "status": false,
    "message": "Missing key/value"
}
```

Response payload:

Status: 200 (success)

```
{
    "status": true,
    "message": "Job processed successfully",
    "fetch_id": "b1150867-10f2-4154-a38a-99db772645e4",
    "body": {
        "user_type": "zotero",
        "zotero_items_metadata": [
            {
                "key": "WTPTXZ28",
                "title": "Ejecta from the DART-produced active asteroid Dimorphos",
                "collection_name": [
                    "Demo"
                ],
                "collections": [
                    "F4875PR6"
                ]
            },
            {
                "key": "ZUFE76VD",
                "title": "Plagiarism Overview - Purdue OWL® - Purdue University",
                "collection_name": [
                    "Demo"
                ],
                "collections": [
                    "F4875PR6"
                ]
            }
        ]
    }
}
```


Status: 500


```
{
    "status": false,
    "message": 'Internal Server Error'
}
```


API 2: Generate Citations for the selected items

API endpoint test server: **https://58wk9x3q84.execute-api.us-east-1.amazonaws.com/test/api/v1/citations**

Method: 'POST'

Headers: *mandatory*


```
{
  "Content-Type": "application/json",
  "x-api-key": "DTLIRTOGIp2iUA1QnXp9S8qGwCCFs9vB58eEGq3E"
}
```


Status: 400 Bad Request


```
{
    "status": false,
```

```
    "message": "Missing required headers and/or parameters."
}
```

Request payload:

Request will be failed if any of the below keys/ values are missing.

```
{
  "citation_style": "apa",
  "user_data": {
        "user_id": "12345",
        "user_name": "xyz",
        "api_key": "123xyz"
    },
  "selected_items": [
      {
        "key": "EKCTMWTI"
      },
      {
        "key": "3VXZ9YP8"
      }
  ]
}
```

Status: 400 Bad Request

```
{
    "status": false,
    "message": "Missing key/value"
}
```

Response payload:

Status: 200 (success) for alphabetic citation style (APA)

```
{
    "status": true,
    "message": "Job processed successfully",
    "fetch_id": "3bb029c8-5a37-4f18-9c6e-68dcf5db5c31",
    "body": {
        "citation_style": "apa",
        "user_type": "",
        "citations_for_item_selected": [
            {
                "status": true,
                "key": "ITEM-1",
                "inline": "(Li et al., 2023)",
                "full": [
                    {
                        "data": "Li, J.-Y., Hirabayashi, M., Farnham, T. L., Sunshine, J. M., Knight, M. M., Tar
                    }
                ],
                "scite_badge_data": {
                    "status": true,
                    "message": "<div class=\"scite-badge\" data-doi=\"10.1038/s41586-023-05811-4\" data-layout=\
                }
            },
```

```
            {
                "status": true,
                "key": "ITEM-2",
                "inline": "(Quadrelli et al., 2015)",
                "full": [
                    {
                        "data": "Quadrelli, M. B., Wood, L. J., Riedel, J. E., McHenry, M. C., Aung, M., Cangahu
                    }
                ],
                "scite_badge_data": {
                    "status": false,
                    "message": "Citation stats unavailable."
                }
            }
        ]
    }
}
```


Status: 200 (success) for numeric citation style (IEEE)


```
{
    "status": true,
    "message": "Job processed successfully",
    "fetch_id": "9a80f6ed-5583-418d-826a-a6764c13869a",
    "body": {
        "citation_style": "ieee",
        "user_type": "",
        "citations_for_item_selected": [
            {
                "status": true,
                "key": "ITEM-1",
                "inline": "[1]",
                "full": [
                    {
                        "data_1": "[1]",
                        "data_2": "J.-Y. Li <i>et al.</i>, “Ejecta from the DART-produced active asteroid Dimorp
                    }
                ],
                "scite_badge_data": {
                    "status": true,
                    "message": "<div class=\"scite-badge\" data-doi=\"10.1038/s41586-023-05811-4\" data-layout=\
                }
            },
            {
                "status": true,
                "key": "ITEM-2",
                "inline": "[2]",
                "full": [
                    {
                        "data_1": "[2]",
                        "data_2": "M. B. Quadrelli <i>et al.</i>, “Guidance, navigation, and control technology
                    }
                ],
                "scite_badge_data": {
                    "status": false,
                    "message": "Citation stats unavailable."
                }
```

```
            }
        ]
    }
}
```

Status: 500

```
{
    "status": false,
    "message": 'Internal Server Error'
}
```

LIMITATIONS:

- AWS API gateway has an architectural time limit of max 30 secs after which it times out and we cannot increase this limit.
- Now if the user has a lot of items in its zotero library and the API endopint which fetches it takes more than 30 secs, the request will be failed throwing an internal server error.
- How big can a user library be?
  - Theoretically, a pro user can have 100+ or even 1000+ items in their library. In this extreme case, the user had 5428 items in their library, which slowed down the Zotero app itself.
- Also if there are only a couple of users whit such huge libraries we can fail the request for them but if there are a lot of users we might need to re think on the architecture.