Bulk ILL requests

Script file names:

- bulk_ill_requests.ipynb
- bulk_ill_requests.ipy

Script details:

The script was originally written as a python notebook using Jupyter Notebook packaged by Anaconda. This Python Notebook file (.pynb) is available as well as a Python file (.py) version however the following libraries may need to be installed in your environment:

For Python and Notebook files:

- pandas
- requests
- xlsxwriter

Notebook files only:

- ipywidgets

The python notebook file contains a progress bar which is commented out in the python file as this requires importing extra libraries that would probably need to be installed and the progress bar will not show correctly outside of a notebook.

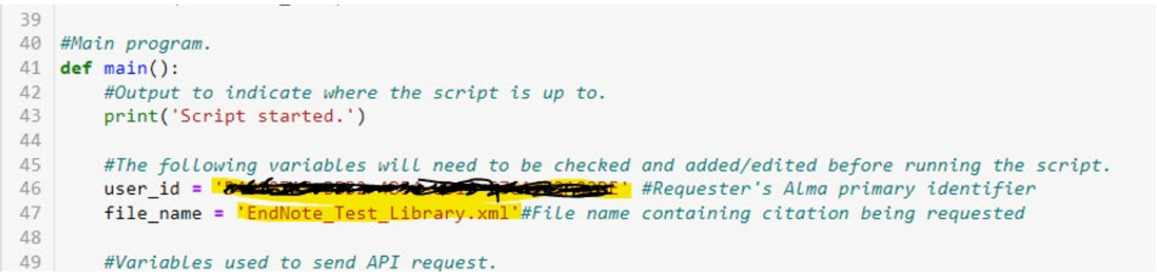
Accepted request file formats:

Bulk ILL requests currently support EndNote, Mendeley and Zotero XML exports. Other reference manager XML exports may also work but are untested.

Procedure:

1. Save the request XML file in the same location as the script.
2. Look up the users Primary ID in Alma.
3. Open the script for editing in your preferred program (e.g. Notepad++, visual studio, Jupyter Notebooks etc).
4. Scroll down to the main function and set the **user_id** and **file_name** variables to match the new request.

e.g.



```

#Main program.
def main():
    #Output to indicate where the script is up to.
    print('Script started.')

    #The following variables will need to be checked and added/edited before running the script.
    user_id = [redacted] #Requester's Alma primary identifier
    file_name = 'EndNote_Test_Library.xml'#File name containing citation being requested

    #Variables used to send API request.
```

Note: In the Python file the variables are on lines 49 and 50, in the notebook file 48 and 49.

5. Run the script using your preffered program (e.g. command line, Python IDLE, Jupyter Notebooks etc.)

6. Once the script has run an ouput file called Bulk_ill_request_response.xlsx will be created. Check the file to confirm the script ran correctly. The responses column will contain whether requets were placed, if there was a reason the request was not placed or any error messages.

Note: The response file has two sheets detailed and counts. Detailed provides a list of all the requests with relevant response. Counts shows the count of each type of resposne.

7. If ran successfully Email the requester the excel file and confirm the requests have been processed, copy in *our ill email*. Otherwise troubleshoot and communicate with the requester accordingly.
8. Delete the excel, csv and XML files.

Note: a csv file is also created, this file is written as each request is placed. This file is created in case the script is suddenly interrupted and halted.