**# This is a proposed "package" manifest of the tools/scripts module we use for collecting NY Times**

**manifestID**: 101

**type**: package

**namespace**: WE1S

**version**: 1.0

**resourceID**: []

**title**: New York Times collecting tools and scripts

**processingSequence**:

 **seq**: 1

 **accessMethod** :

 **api**: <http://api.nytimes.com/svc/search/v2/articlesearch>

 **description**: NY Times articles API, version 2; using various API keys

 **instructions**: be sure to use different API keys for multiple simultaneous collecting runs.

 **mechanisms**:

 **type**: script

 **label**: getTimesArticles\_fq.py

 **version**: 1.0

 **resourceLocation**: /volume1/homes/4Humanities/www/whatevery1says/workflows/scripts/python\_scripts/Scripts for NY Times

 **description**: Python script to search NY Times articles, collect JSON files about found articles, and create a .tsv (tab separated values) file summarizing the information in the JSONs.

 - **option**:

 **argument**: settings.cfg

 **value**: #EXAMPLE

 logfile = C:/workspace/times-testing.log

 json\_folder = C:/workspace/

 tsv\_file = C:/workspace/output.tsv

 [nytimes]

 query = "humanities"

 start\_year = 2014

 start\_month = 01

 start\_day = 1

 end\_year = 2014

 end\_month = 12

 end\_day = 31

 api\_key = [YOUR NY TIMES ARTICLES API KEY HERE]

 **instructions**: the settings.cfg file must reside in a subfolder of location of the getTimesArticles\_fq.py script. Open settings.cfg in a text editor and revise the output paths it includes to the local working folders or the end-destination folders you want. Also revise the search terms and search dates for your purposes. And enter your own API key, or a different key for each simultaneous search. Once you start the getTimesArticles\_fq.py script, it will serially search each day of the NYT for the date span you requested, returning JSON files for each day and also cumulatively adding to the summarizing .tsv file. This may take a long time depending on the date span you are searching. Searching for "humanities" over a decade, for example, will require hours to complete.

 **seq**: 2

 **mechanisms**:

 **type**: manual process

 **instructions**: The getTimesArticles\_fq.py script produces a set of JSON files containing information about each NYT article found in a search, plus a .tsv (tab separated values) file summarizing the JSONs that can be opened in a spreadsheet. After the script is done, (1) collect the JSON files in a subfolder for the collecting run, (2) open the .tsv file in a spreadsheet, (3) select the column in the spreadsheet for the article URLs and copy, (4) paste the URLs into a text file called urls.txt.

 **seq**: 3

 **mechanisms**:

 **type**: plugin

 **label**: DownloadThemAll

 **version**:

 **resourceLocation**: https://addons.mozilla.org/en-US/firefox/addon/downthemall/

 **description**: Firefox extension to download files in bulk

 **instructions**: Open the urls.txt file you created in the previous step in Firefox as a local file viewed in the browser. Then use DownloadThemAll to download and save all the files to a local folder or the end-destination folder of downloaded files you want.

 **seq**: 4

 **mechanisms**:

 **type**: script

 **label**: nyt\_scraper\_folder.py

 **version**: 1.0

 **resourceLocation**: /volume1/homes/4Humanities/www/whatevery1says/workflows/scripts/python\_scripts/Scripts for NY Times/nyt\_scraper\_folder.py

 **description**: Python script that uses Beautiful Soup 4 to scrape plain text from the titles and body articles of NY times articles.

 **instructions**: Customize path names in the script for your computer and working folders before running it. (Note: you must have installed the BeautifulSoup 4 package for Python for this script to work.) Then run the script on the folder of NYT articles you previously downloaded. The script will scrape the plain text from each article title and body text, and accumulate the text from all articles in an aggregate text file (with individual articles separated by a delimiter string of ten @ signs -- @@@@@@@@@@ .

 **seq**: 5

 **mechanisms**:

 **type**: script

 **label**: cut.py

 **version**: 1.0

 **description**: Python script to cut the above created aggregated text file at the designated delimiter into individual text files for each article.