This is a **space-time web crawler** which extracts links from the content of a downloaded webpage and Save the details that have been downloaded in your own machine.

**Input**: raw\_content\_obj (an object of Type “UrlResponse” declared at the beginning of

datamodel/search/server\_datamodel.py. For simplicity it’s given as ICS, UCI domain.)

**Output**: list of URLs in string form. Each URL should be in absolute form. The frontier takes care of ignoring duplicates.

The function **is\_valid** (applications/search/crawler\_frame.py), returns True or False based on whether a URL is valid and must be downloaded or not.

**Input**: URL is the URL of a web page in string form

**Output**: True if URL is valid, False if the URL otherwise. It filters out crawler traps (e.g. the ICS calendar, dynamic URL’s, etc.) and double‐check that the URL is valid and in absolute form. Returning False on a URL does not let that URL to enter your frontier. Robot rules and duplication rules are checked separately.

**Implementing your Project**

Step 1 Getting the project

*git clone the project*

Step 2 Installing the dependencies

Make sure you do not have conflicting libraries by issuing the command.

*python ‐m spacetime ‐v*

You should see the following output”

*Spacetime Version is 2.0*

*Rtypes Version is 2.0*

If the outputs do not match, or if it returns an error “unrecognized argument: version”, please

uninstall the old spacetime, and rtypes by issuing the commands.

*python ‐m pip uninstall spacetime*

*python ‐m pip uninstall rtypes*

Get the latest repository of spacetime‐crawler to get the latest version of spacetime and rtypes.

Both packages are included with the assignment.

Step 3 Running the crawler

**Execute the following commands**

To run the crawler.

*python applications/search/crawler.py -a amazon.ics.uci.edu -p 9400*

To check frontier status.

*python applications/search/check\_frontier.py -a amazon.ics.uci.edu -p 9400*

To reset frontier progress.

*python applications/search/reset\_frontier.py -a amazon.ics.uci.edu -p 9400*

To clean only those urls from frontier that no longer satisfy your is\_valid function.

*python applications/search/delete\_invalids\_from\_frontier.py -a amazon.ics.uci.edu -p 9400*

Note that resetting the frontier may take a while depending on how big

the frontier is.

Step 4 Monitoring the crawler

The invalid links can be seen at

[*http://amazon.ics.uci.edu:9400/<Your\_crawler\_id>/invalid*](http://amazon.ics.uci.edu:9400/%3cYour_crawler_id%3e/invalid)

This crawler keeps a track of subdomains that it visited, and count how many different URLs it has processed from each of those subdomains and write it out to a file at the end. Also it writes the page with the most out links (the number of links that are present on a particular webpage), invalid links, crawler traps encountered, and so on.