1) File List

----------------

I have writen this code using jetbrains pyCharm which is an IDE for python using python version 3.5.1 and for creating .EXE file I used py2exe in python version 3.3.3 , this crawler code contains python files which are described below.

i) main.py ----> It contains a Tkinter\_GUI Class with functions like user\_interface, file\_open, read\_file, connect\_userinputs, start\_crawling, create\_threads, job and create\_jobs.

ii) htmlTagExtractor.py ----> here I used Beautifulsoup python library for pulling data from HTML pages and extracting URL links which can be extracted from "href" attribute under <a> </a> tags , this file can be modified according to our need like what kind of information we want to exteact from a page , as Beautifulsoup provide many options to extract and modify HTML and XML pages.

iii) bot.py ----> It create shared variables and files so that it can be shared by all the threads, it contains functions like updateDirFiles, update\_files, crawl\_page, gather\_links and add\_links\_to\_queue

iv) utilityFunctions.py ----> It contains all useful functions which will be used by thread on regular basis , it contains functions like add\_repository, create\_repository\_files, append\_file , delete\_file\_contents, dump\_to\_set, dump\_to\_file , find\_domain, Netloc

v) setup.py ----> this is created for converting python program to .exe file so that it can run easily by just clicking the main.exe which is under dist\main

it can be done by typing “python setup.py py2exe" in terminal

2) Python Libraries used

-------------------------------------

Beautifulsoup(bs4) , tkinter, threading, urllib, os, py2exe etc

3) General description

----------------------------------

I tried my best to make this crawler to run efficiently and quickly I used the functionality of multithreading, I have tested this crawler in my Windows 8 machine and its working fine, this is a basic crawler loaded with Tkinter GUI and .EXE file .

At User Interface user has to enter 3 information.

i) Enter Repository Name:

Here user has to enter a repository/folder name.

Where all the crawled links which are already crawled will be stored automatically in "crawledLinks.txt" , links to be crawled( waiting links) in "waitingToCrawlLinks.txt" and out of scope links in "outOfScopeLinks.txt"which stored links that are not related to domain name of entered URL will be saved, it could be an ip address, blank URL , links directing to some other domain etc.

ii) Enter URL:

Here user has to enter a starting URL such as (http://python.org) from where the crawling should begin for which all the checks and conditions has been written on htmlTagExtractor.py , bot.py , utilityFunctions.py.

iii) Enter No of links to Crawl:

It will crawl no of links which are specified in "No of links to crawl Entry box".

After which click "Start Crawling Button " and crawling will start and links will be stored in respective text files then wait for some time and then close the GUI.

Start the program again for another task :)

4) How to run

----------------------

i) I have created an .EXE version of this crawler so that you dont need any other softwares to run this crawler (in windows machine), the .EXE file is in path (Web\_Crawler\_RB\dist\main) just click this main.exe file and you are ready to go :)

The text files will be saved under "Web\_Crawler\_RB\dist\RepostioryName\_EnteredByYou.

ii) The normal way to run this crawler is to install python in your machine save all python file in a folder and update your Python with python libraries described in point no 2 and then compile main.py , a Tkinter GUI will pop up and fill the information and then click "start Crawling".

5) Conclusion

--------------------

I have written this code in seperate files which are interrelated, I tried my best to write a clean code so that more functionalities can be added easily without messing up whole code in future.

At last I would say that so many other functionalities can be added in this python code and both User Interface and User Experience can be improved further.