Data Communication and Networks	Project 3: Simple HTTP Crawler
CSCI-351 Fall 2018	November 8, 2018

This project is due at 11:59:59pm on November 22, 2018 and is worth 30% of your project scores. You must complete it with a partner. You may only complete it alone or in a group of three if you have the instructor's explicit permission to do so for this project.

Note that there is no milestone deadline for this project.

1 Description

HTTP is one of the most important application level protocols, which is widely used today. This project is to help you understand how to fetch HTTP content that you see everyday and parse the information that you are interested in, and finally store it to the database for the effective management.

2 Requirements

Your will write a simple HTTP Crawler program, which will:

- Fetch the conference list from WikiCFP (www.wikicfp.com)
- Parse the (1) event, (2) when, (3) where, and (4) deadline information from the html page.
- Construct a database to store it into the local disk as a database table.
- Be able to answer for simple queries.

3 Your cralwer program

For this project, you may use *any* languages that you want and *any* third party libraries that help you (1) fetch, (2) parse, and (3) store the webpages. However, if your code does not compile or run on the glados server then it is your fault.

3.1 Input and output

The command line syntax for your crawler is given below. The program takes upto two command line arguments. The syntax for launching your program is therefore:

```
./351crawler <dbmake|search> <optional-argument-1> <optional-argument-2>
```

`dbmake` (Required) It (1) fetches the first five pages from the WikiCFP[1]. If you send five different queries to fetch the five webpages, you have to be careful that you issue the HTTP requests no frequent than every five seconds (See the Warnings subsection); (2) parses the Event,

[1] http://wikicfp.com/cfp/call?conference=computer%20science

When, Where, and Deadline information; (3) stores the information in a database by creating a table using a database connector such as SQLite3[2][3]. This commands will generate a database file (e.g., .db file if you used SQLite3) in the same directory of the `351crawler`.

`all` (Required) It prints out all the event, when, where, and deadline information that you have fetched.

`search` (Required) It takes two arguments for <year> and <month>. Your program will look up the database to find the events that matches with the year and month.

3.2 Warnings

If you send too many queries to WikiCFP server within a short period of time, your IP could be blocked. WikiCFP gratefully allows you to issue an query at most once every five seconds, thus please strictly follow the guidlines.[4] See Section 7 (Advice) not to be blocked from the WikiCFP.

3.3 Development

You should develop your client program on the Glados Linux machines. You are welcome to use your own Linux/OS X machines, but you are responsible for getting your code working, and your code *must* work when graded on the glados Linux machines. If you do not have a glados account, you should get one ASAP in order to complete the project.

4 Submitting your project

4.1 Registering your team

You should pick out a team name (no spaces or non-alphanumeric characters). One of team members should send me an email (the title is [CSCI351] Registering a team) with team name, your name, your RIT ID, partner name, partner's RIT ID.

You must register your team by 11:59:59pm on November 13th, 2018.

4.2 Milestone submission

There is NO milestone deadline

4.3 Final submission

For the final submission, you should submit your (thoroughly documented) code along with a plain-text (no Word or PDF) README file. In this file, you should describe your high-level approach, the challenges you faced, a list of properties/features of your design that you think is good, and an overview of how you tested your code. You MUST submit a "shell" `runme.sh` script

[2] https://www.sqlite.org/index.html

[3] SQLite3 is not currently installed at Glados. You should consider setting up a virtual environment first (e.g., `virtualenv` for python)

[4] http://wikicfp.com/cfp/data.jsp

that generates the executable file `351crawler`: you choose your language so you have to prepare it. You should submit your project to Project3 folder in the Mycourses Dropbox. Specifically, place all of your code and README files into one folder (Project3) and zip it (TEAMNAME.zip) and upload it to the Dropbox.

You must submit your project by 11:59:59pm on November 22, 2018. Enjoy Thanksgiving.

5 Grading

The grading in this project will consist of

70% Program functionality

15% Error handling

15% Style and documentation

6 Advice

A few pointers that you may find useful while working on this project:

- To test your code that parses the html pages, I strongly recommend to first download the whole webpage as a file and begin to test your parsing code with the downloaded page as an input. By doing so, your code will not send multiple same HTTP requests to the server, which saves your time and does not put heavy burdens on the WikiCFP server.
- This course is not a database class; however, I believe you will need to use a database at some point in your future career. SQLite3 will be a good starting point to familiarize yourself with handling a database. Please see the SQLite3 tutorial[5] to familarize yourself with basic database commands. More specifically, you will need to learn (1) `CREATE Database`, (2) `CREATE Table`, (3) `SELECT` command (`WHERE` and `ORDER` options).
- As the SQLite is not installed on Glados, you may consider setting up your working space using virtual environment builders such as virtualenv for python. If you used virtual environment builders, you also need to put the instructions for setting up the environments in the `runme.sh`.
- Check the Mycourses for question and clarifications. You should post project-specific questions there first, before emailing the professor.
- Finally, get started early and come to the instructor's office hours. You are welcome to come to the lab and work, and ask the instructor any questions you may have.

[5] https://www.tutorialspoint.com/sqlite/sqlite_select_query.htm