CS4540 A6 Scraper

Ken Wang

Github Link

How to use

- The UI should be pretty simple to use. It splits into three parts
    1. The top left is the Find Enrollment section, where you input parameters and search for enrollments.
    - After you completed find search, the save button will be set active for you to click and will save as a csv file.
    2. The top right is the Find Course from catalog section, where you input parameters and search for a (or a list) of courses.
    - There will not be a save function for this. But it will print to console for reference.
    3. The last part is the Console at the bottom. It outputs all the neccessary info that the user might want to know. Including error messages.

Peer Review

- Name of Peer: XiaoChuang Huang , UID: u1067294

Time out value

- 6 seconds => that's more than enough for the school internet to access the pages we need. If the value is under 3 seconds, it sometimes will throw a exception before finding the element. And generally we don't need that much time so I set to 6.

Thoughts on Selenium

- Generally this is really simple and straight forward unlike other scraper that exists on lines.
- The easy part is that it's easy to find the element you want with the help of Chrome, where you just check the element and then copy the Xpath directly from Chrome and you are good to go.
- Then just simply interact with the element attributes, or text and store the values that we want.
- Selenium definitly wouldn't be hard for applying to test my LOT project. Since we now know the basic mechanics of it, all we need to do is the figure out the 'path' that we want to test, and the functionalities that we want to make sure it's correct, given the mocking data. The only hard part might be that if we're to change some interal model, like the Course instance vars, it might be really hard to adjust accordingly.

Above and beyond

- I handled most of the corner cases that I can think of, trying my best to think that the user is "dumb".
- The error message will be either shown via message box or in the console.
- Build a simple MV model to put the logic in a Scraper class.

- Made some multithreading so that the Scraper does not block the UI and the console will print things out simultainously.