**Author: Long Nguyen Duc Hoang - Worcester Polytechnic Institute – Class of 2016**

Email: [lhnguyenduc@wpi.edu](mailto:lhnguyenduc@wpi.edu) or [hlong290494@gmail.com](mailto:hlong290494@gmail.com)

**IPEDS Wrapper Maintenance Guideline**

1. **Overview**

* I wrote this wrapper as a template to access various resources and custom data from the NCES’ IPEDS Data Center. At the time of this writing, IPEDS hadn’t provided a direct API for programmers to access their data. Therefore, I have to write a wrapper that simulate a user’s web browser in order to download the data from IPEDS.
* This writing serves as an annual maintenance for the wrapper class IPEDSDownload.java

1. **How does IPEDS Datacenter work**

* IPEDS Datacenter is a web service that provides extraction of data from the National Center for Education Statistics. A user must navigate through the website, picking Institutions and interested Variables. After around 2-3 steps, the user will be able to reach the download page. The process is time-consuming, and the interface is not really user-friendly.
* After each querying step, IPEDS Datacenter webpage will send a POST request to a query builder. The query builder is intelligent enough to prevent non-legitimate access with cookie and request validation. However, this can be circumvented because I detected some pattern in the POST requests.
* Therefore, I used Firefox plugins such as Firebug (the Net function) and Tamper Data to monitor the flow of data between the client and server. Then, I will simulate the web browser by sending a correct POST request to the appropriate webpage location. Here are some key properties of the request we must consider:

|  |
| --- |
| * For the POST request   + POST requests for every page changes (the browser goes to another page from pressing Continue/Submit button)   + POST requests has the *\_\_VIEWSTATE* parameter (a must)   + POST requests has *\_\_VIEWSTATEGENERATOR* parameter (there are some cases that the request doesn’t contain this)   + POST requests must contain form data from our selection. * Request headers   + The headers must have Referrer parameter. A correct parameter value for each request is necessary.   + The headers should contain standard web browser headers (Host, User-Agent, Connection, etc.)   + Set of cookies data accumulated from the querying steps. This is very important. |

* An engineer maintaining this wrapper should pull those request parameters by manually navigating through the IPEDS Datacenter website. Every time the browser changes the page, the engineer should investigate the POST request send **at the moment of button press/page change**. The engineer will need to pull out EVERY request parameters and request headers. Cookies can be temporarily ignored because the functions in a Java library, **JSoup**, can handle that.
* At the end, IPEDS Datacenter will render a Download page which will give you the option to download a CSV file. Please note that, it is zipped.

Now, I will explain carefully the code I write:

1. **The wrapper class**

At the time of this writing, we are interested in the Talent Development Metric/Graduation Rates from IPEDS. Here are the steps to download the data manually:

|  |
| --- |
| **Link:**  The Link for IPEDS:  [http://nces.ed.gov/ipeds/datacenter/](https://exchange.wpi.edu/owa/redir.aspx?C=eMSOoV3lpkivh4zpJ10Qta0Jul6999AIAyZwf8m3TmDKuffdz_nlUt06I03t5wNE9teeE2MJNvM.&URL=http%3a%2f%2fnces.ed.gov%2fipeds%2fdatacenter%2f)   1. Select “DOWNLOAD CUSTOM DATA FILES” 2. Use final release data and press Continue 3. Select Institutions By Variables and then select Browse/Search Variables 4. Select “Frequently used/Derived variables” 5. Select Institutions 6. Select Year and in List of variables select “State Abbreviation” and “Degree granting status” 7. Press Continue 8. Press Continue 9. Click State abbreviation to select the state. ( Can select multiple states but in end would get multiple files) 10. Click Save 11. Click “Degree granting status” and select “Degree granting” 12. Click Save 13. Press Submit 14. Click Continue 15. In Select variables, select Year and then select Completions 16. In Select Qualifying Variables     1. Select CIP Code- 2010 Classification: select CIP two digit codes (11,14,15, 27, 40)     2. In Award Level Code select all from Associate degree to “Doctor’s degree”, Save. 17. In Select from List of Variables select “Grand Total” 18. Click Continue 19. Save the file   **Data Format:**  10 columns: where 9th column contains the total for institution    **General Notes:**  We can select multiple states and multiple years when selecting state abbreviation and year and in end each separate file will be generated for each state and year.  **Reference: File “Graduation Rate 2012 MA.csv”** |

Now, please look at the code, I will explain what it will do for each step

From line 32 to 51, we will do step 1-2 above – Logging in the IPEDS Datacenter system. As you can see from line 31 to 48, I have customized the POST request with headers and data parameters to send to <http://nces.ed.gov/ipeds/datacenter/login.aspx>. As a result, IPEDS Datacenter will return some session cookies data that we need to save (line 51.) Please note that every POST data and header parameters must be completed in order to simulate a real browser request. Otherwise, IPEDS Datacenter will reject your request and send back a starting HTML page – which means you had done something wrong. **One tip: Write the HTML into a separate file and track those files step-by-step to make sure that the process goes smoothly.**

In the POST request, parameters such as *\_\_VIEWSTATE* and *\_\_EVENTVALIDATION* are unchanged, while *ibtnLoginLevelOne.x is* not. The engineer should investigate multiple time to determine the constants and focus on making them right. The non-constants do not seem to affect the overall result.

The easiest way to do this is to make the request exactly the same as those of the browser.

This wrapper covers 6 crucial steps – with the last step involve downloading the zipped file. For each step, the engineer need to simulate the browser’s request just like I explained above. However, be extremely careful about **COOKIES**.

IPEDS Datacenter website tells the browser to save cookie data differently on each query pages, so we must build the cookie data step-by-step. On line 72, there is a **cookieMerge** operation that merges the response cookie assignment from Step 2 to those of Step 1. The same principle is applied to each step. The engineers should also track the cookies for each step, validate to see if the cookie returned by **JSoup library** is sufficient and correct (JSoup.Response.cookies() method.)

Finally, step 6, IPEDS Datacenter will render a page that contains the link to download a zipped CSV file, JSoup also support jQuery-style element. The engineer should investigate the source of the download page to pick out the most suitable CSS selector, and grab the correct download link. After that, use JSoup to download the file, ***don’t forget the cookies and the request headers***!

*.ignoreContentType(true) – line 174* is also necessary for JSoup to download the zip file. By default, JSoup will only parse HTML/XML content. That method will tells JSoup to ignore and save the content as Bytes. Finally, use FileOutputStream to write the JSoup’s Reponse Body in bytes from the last step.

Extract the download zip file with the UnZip class. And voila!

I hope this documentation is clear enough for you – the maintainer – to modify and improve this wrapper. Please do not hesitate to contact me if you have any problem with it.