**Background and guideline:**

* Using Scrapy framework to scrape data off 5 websites;
* Please keep in mind of concurrency and scalability. Certain web site can handle lots of request, some only can handle single thread with few seconds delay.
* Able to adjust scan frequency for each web site:
  + High frequency with proxy (Tor/Paid/un-paid)
  + Minutes, Daily, Weekly, Monthly
* Multi threading:
  + able to check the thread status, terminate it gracefully once it completes its task.
  + Also allow a TTL for each thread, and terminate it after certain time.
  + Reuse thread slot after previous task is completed (terminated gracefully / timeout)
  + These parameters will be read from MySQL table
* Able to schedule the crawler for a particular web site during certain time of the day.
* Download delay or time delay for each request can be given as an input and there has to be a provision for the same in the code (Not to overload the websites)
* Taking input parameters from MySQL (Simple select) and output back to MySQL (stored procedure)
* A portion of output will be saved in CSV locally
* Able to use proxy (Free/Paid/Tor)
* Robust xpath and css links in response.xpath & response.css and or .html or .json path whatsoever relevant used in the project
* Code should be self-explanatory with relevant comments and explanations in the code delivery
* Initial step would be validation of crawled data with that of available on the websites
* A Demo/ Document explaining the code execution is needed
* Few 1 hour support session after the code is appreciated
* There will be a lot of details to follow after the project is awarded
* Detail is a main key in this project. If this project is done properly, and up to satisfaction, at least 3 projects will follow
* We will provide all select statements, table schema and stored procedures.

**Amazon Product Price**

Sample URL: <https://www.amazon.ca/gp/offer-listing/199957950X?ie=UTF8&qid=1554396712&sr=8-1>

**Required Data:**

ISBN10: this is the input;

ItemPrice:

ShippingPrice : Shipping fee: Free Shipping=0;

ItemCondition:

SellerName: Amazon.ca pic = amazon.ca

Seller Rating:

* starsRating: 0-5
* positiveRating: 0-100
* numRating: #### (whatever number on there, minus the comma sign)

ArrivalDate

* minArrivalDate: 25 (Example: April 29, 2019 – use this date minus the current scan date April 4, 2019)
* maxArrivalDate: 47, (Example: May 21, 2019)

ShippingLocation:

* United Kingdom, IA United States,

DeliveryBoxText: We want the whole ul text for other data analytic. Please use a separator to separate them and put it inside the database.

* **Example**:

|  |
| --- |
| * **Arrives between** April 30 - May 22. * Ships from United Kingdom. [Learn more](https://www.amazon.ca/gp/help/customer/display.html/ref=olp_intl_help?ie=UTF8&nodeId=918748) about import fees and international shipping time. * [Shipping rates](https://www.amazon.ca/gp/aag/details/ref=olp_merch_ship_2?ie=UTF8&asin=199957950X&seller=A28ACMVQIZOEMD&sshmPath=shipping-rates#aag_shipping) and [return policy](https://www.amazon.ca/gp/aag/details/ref=olp_merch_return_2?ie=UTF8&asin=199957950X&seller=A28ACMVQIZOEMD&sshmPath=returns#aag_returns). |

**Database Output Sample:**

* Only the lowest price / first record goes into the database

|  |  |
| --- | --- |
| ISBN10 | 199957950X |
| itemPrice | 50.88 |
| shippingPrice | - |
| itemCondition | - |
| seller | amazon.ca |
| starRating | - |
| positiverating | - |
| minArrivalDate | - |
| maxArrivalDate | - |
| shippingLocation | - |
| lastScanTime |  |
| lastUpdatedWSID |  |

**Sample 2:**

|  |  |
| --- | --- |
| ISBN10 | 199957950X |
| itemPrice | 63.66 |
| shippingPrice | 7.5 |
| itemCondition | 11 |
| seller | \_distribution\_ |
| starRating | 4.5 |
| positiverating | 85 |
| minArrivalDate | 25 |
| maxArrivalDate | 47 |
| shippingLocation | United Kingdom |

**CSV output:**

* The database output only required the first record, but CSV output will required all the offers for future analysis
* Please include DeliveryBoxText inside the CSV as well

**Condition Chart:**

|  |  |  |
| --- | --- | --- |
| **ABE Descrption** | **AMZ Index** | AMZ Descrption |
| Brand New | 11 | New |
| Like New | 1 | Used, Like New |
| As New | 1 |  |
| New | 1 |  |
| Very Good | 2 | Used, Very Good |
| Good | 3 | Used, Good |
| Acceptable | 4 | Used, Acceptable |
| Fine | 4 |  |

**Abebooks.com Product Prices:**

URL: <https://www.abebooks.com/servlet/SearchResults?cm_sp=SearchF-_-topnav-_-Results&ds=20&isbn=0669326356&sts=t>

**Requirement:**

* Please set the destination to Canada. Before there was a parameter we can set

**Required Data:**

**Price in USD**: 21.66

**Shipping Cost to Canada:**

**ShippingCountry: U.S.A or United Kingdom**

**Condition**: Used or New

**Cover**: whatever the text inside the div

RateAndSpeed:

* Standard:
  + firstPrice
  + additional price,
  + minSpeed: 21
  + maxSpeed:33
* Expediate:
  + firstPrice
  + additional price,
  + minSpeed: 5
  + maxSpeed:10

**Quantity Available**:

**SellerName**:

**SellerLocation**: example: Philadelphia, PA, U.S.A.

**SellerRating**: 0-5

**DescrptionText**: be careful with this text, there are characters not accepted by MySQL;

* Example: Houghton Mifflin Harcourt. Hardcover. Condition: Fine. 1328879984 Remainder mark. Seller Inventory # Z1328879984Z1

**Sample Output:**

|  |  |
| --- | --- |
| ISBN10 | 1328879984 |
| itemPrice | 21.66 |
| shippingPrice | 3.49 |
| itemConditionLabel | Used |
| coverLabel | Hardcover |
| qtyAvailable | 2 |
| seller | Booklot COM LLC |
| sellerLocation | Philadelphia, PA, U.S.A. |
| starRating | 4 |
| shippingStandardFirst | 6.21 |
| shippingStandardAdditional | 5 |
| shippingStandardMinSpeed | 21 |
| shippingStandardMaxSpeed | 33 |
| shippingExpediateFirst | 13 |
| shippingExpediateAdditional | 8 |
| shippingExpediateMinSpeed | 5 |
| shippingExpediateMaxSpeed | 10 |

**CSV output:**

* The database output only required the first record, but CSV output will required all the offers for future analysis
* Please include Item description inside the CSV as well

**Condition Chart:**

|  |  |  |
| --- | --- | --- |
| **ABE Descrption** | **AMZ Index** | AMZ Descrption |
| Brand New | 11 | New |
| Like New | 1 | Used, Like New |
| As New | 1 |  |
| New | 1 |  |
| Very Good | 2 | Used, Very Good |
| Good | 3 | Used, Good |
| Acceptable | 4 | Used, Acceptable |
| Fine | 4 |  |

**Bound & Copy Scraper**

**URL**: <http://pc8.su.ucalgary.ca/search/index.cfm?formreturn=yes>

**Search Input #1:** ISBN: 100-999

**Search Input #2**: Select statement from MySQL

Note: There are 2 type of search input, the reason being we want to scrape off most data with few calls. That’s why we are using a counter from 100 – 999. This will update most records inside the table. Second round we select those ISBNs didn’t get updated and scan them individually. We will provide the MySQL query and table schema later.

**Concurrency/Speed**: Single thread ONLY; please only use single thread and request one after another one as this company only has one server. Multiple request will bring the server down. We are planning to run this during night time or few hours.

**Frequency**: Daily at night

**Output**: MySQL only

**U of C – Textbook list**

**URL**: <https://www.calgarybookstore.ca/buy_courselisting.asp>

**Input**: none

**Concurrency/Speed:** Single thread

**Frequency**: <10 times each year

**Output**: MySQL

**Sample Output:**

|  |  |
| --- | --- |
| batchIndex | 1 |
| termID | 36|188 |
| term | U Calgary - Winter 2019 |
| departmentID | 3091 |
| department | ACCT-Accounting |
| courseID | 609159 |
| course | 217 |
| sectionID | 275113 |
| section | 2-Luminita Enache |
| title | Pkg Financial Accounting: Tools For Business Decision-Making W/ Wileyplus |
| authors | Kimmel |
| ISBN | 9781119368458 |
| copyright | Copyright 17 |
| edition | Edition 7C |
| binding | Looseleaf |
| newPrice | 141.25 |
| usedPrice |  |
| req | Required / Recommend / Optional |
| lastUpdated |  |

**Note**: batchIndex is each time we restart the scraper, we want a new batch index. The reason is we want to keep track how many records we received /scraped in each batch. In case if the program stopped, we know how many rows were inserted.

[SellMyTextbooks](http://www.sellmytextbooks.org/members/191/index.cfm?index=QUOTE)

**Sample ISBN Input:**

0321811054

0495391328

1118345002

1118875869

0470501979

0023364505

1138954365

0205579353

1464188041

**Input**: ISBN10 from text file / select from database

**Speed:** Single thread

**Frequency**: on-going scan; in the past, they have blocked us because of scanning frequency was too high; We need to limit this to 1 scan for every 30 seconds

**Output**: MySQL

**Sample output:**

|  |  |
| --- | --- |
| ISBN10 | 1464188041 |
| title | INTRO.TO GENETIC ANALYSIS (LOOSELEAF) |
| author | GRIFFITHS |
| location | Mount Royal University BookStore |
| price | 24 |
| lastUpdated |  |