Item Stock Tracker

Release v2.1.0

Ramya Sai Mullapudi, Rohan Prabhune, Arjun Madhusudan, Laks

Nov 04, 2021

PACKAGES AND MODULES:

"Item Stock Tracker" is a program designed to alert users when specific items from an online retailer are back in stock.

CHAPTER

ONE

INSTALLATION

1. Download the latest release from the repository based on your operating system. Releases
2. Extract the zip file downloaded.
3. Run ItemStockTracker.exe from the extracted folder to launch the application.

1.1 code

1.1.1 AmazonScraper module

class AmazonScraper.**AmazonScraper**(*url*)
Bases: object

This is the Scraper for Amazon. Takes in product url as input upon object creation. Method 'job' prints progress while method 'check_stock' obtains stock info and a string indicating cost of the product.

Amazon pages can have stock info in different ways. Following are the possible cases, interpretations, and return values of each case.

Case	Interpretation	Scraper return value
In Stock	Product is in stock	In Stock
Only x left Order soon	Product is in stock	In Stock
Currently unavailable	Product is out of stock	Out of Stock
In stock soon	Product is out of stock	Out of Stock
No stock info or captcha page	Stock information not available	No Stock Info

check_stock(*url*)
Obtains stock information from the given url.

Parameters **url** – URL of the product

Returns a string indicating the stock information and a string indicating cost of the product

job()
Prints the progress, and delegates the task to 'check_stock'.

Returns a string indicating the stock information and a string indicating cost of the product

1.1.2 BestBuyScraper module

class BestBuyScraper.**BestBuyScraper**(*url*)
Bases: object

This is the Scraper for BestBuy. Takes in product url as input upon object creation. Method 'job' prints progress while the method 'check_stock' obtains stock info and a string indicating cost of the product.

check_stock(*url*)
Obtains stock information from the given url.

Parameters **url** – URL of the product

Returns a string indicating the stock information and a string indicating cost of the product

job()
Prints the progress, and delegates the task to 'check_stock'.

Returns a string indicating the stock information and a string indicating cost of the product

1.1.3 Scraper module

class Scraper.**Scraper**
Bases: object

Scraper chooses which scraper to run. Method 'ChooseScraper' chooses the scraper.

ChooseScraper(*url*)
Chooses the scraper based on product url.

Parameters **url** – URL of the product

Returns the stock status information

1.1.4 SendEmail module

SendEmail.**sendEmail**(*receiver*, *itemName*, *url*)

Sends the notification email to the user. Takes in the user's email, product name, and product url as input. In order to send the email, please prepare an email account and fill in the gmail_user for email address and gmail_password for password.

Parameters

- **receiver** – The email address of the receiver(User)
- **itemName** – The name of the the product
- **url** – URL of the product

Returns 1 if the email sent successfully, 0 if not

1.1.5 Tracker module

class Tracker.**State**

Bases: object

The State class that holds all state data. Holds alert settings and the item list.

deleteAlert(*alert*)

Delete an alert.

Parameters **alert** – given the alert name which we want to delete

Returns

deleteEmail()

Clear the email address.

getStatus(*item*, *url*)

Get the item status for specific item.

Parameters

- **item** – given item name
- **url** – given item url

Returns the item status for given item

read_state(*s*)

Read in a state from file.

Parameters

- **filename** – given filename we want to read in
- **s** – is the sate we want add state data to

save_state(*s*)

Save the state to the file.

@param filename: the filename we want to save the state to @param s: is the state instance which stores the state data

updateAlert(*alert*)
Update the alert with given alert, add the new aler to the alert list.

Parameters **alert** – given alert

updateEmail(*email*)
Update the email with given email.

Parameters **email** – the given email

updateItem(*iturl*)
Update the item list with given item dict.

Parameters **iturl** – the given item dict,contains the item name, url, status and previous status

updateSetting(*setting*)
Update the setting with given setting.

Parameters **setting** – the given setting

updateStatus(*item*, *url*, *status*)
Update the item status.

Parameters

- **item** – given item name
- **url** – given item url
- **status** – new item status

1.1.6 WalmartScraper module

class WalmartScraper.**WalmartScraper**(*url*)
Bases: object

This is the Scraper for Walmart. Takes in product url as input upon object creation. Method 'job' prints progress while method 'check_stock' obtains stock info and a string indicating cost of the product.

Walmart pages can have stock info in different ways. Following are the possible cases, interpretations, and return values of each case.

Case	Interpretation	Scraper return value
Add to Cart	Product is in stock	In Stock
Out of Stock	Product is out of stock	Out of Stock

check_stock_price(*url*)
Obtains stock information from the given url.

Parameters **url** – URL of the product

Returns a string indicating the stock information and a string indicating cost of the product

job()
Prints the progress, and delegates the task to 'check_stock'.

Returns a string indicating the stock information and a string indicating cost of the product

1.1.7 utils module

Utils file contains the utility functions for enabling the additional features like launch on auto start.

utils.**become_persistent**(*filename*)
For Windows users we can launch the app during startup to start tracking the availability and send desktop notifications when available.

utils.**remove_startup**()
Disables auto launch during startup.

1.2 test

1.2.1 Amazon_cost_test module

Amazon_cost_test.**test_check_cost**()
Tests if the cost value is received correctly for each of the 5 stock status conditions (In Stock, Order Soon, Out of Stock, In Stock Soon, Invalid URL) on www.amazon.com.

1.2.2 AmazonScraper_test module

AmazonScraper_test.**test_check_stock**()
Tests if the stock status value is received correctly for each of the 5 stock status conditions (In Stock, Order Soon, Out of Stock, In Stock Soon, Invalid URL) on www.amazon.com.

AmazonScraper_test.**test_init**()
Tests if Amazon Scraper initializes properly.

AmazonScraper_test.**test_job**()
Tests if the Amazon scraper delegation and the console print messages are working fine for each of the 5 stock status conditions.

1.2.3 BestBuy_cost_test module

BestBuy_cost_test.**test_InStock_cost**()
Tests if cost value is received correctly when condition is In Stock on www.bestbuy.com.

BestBuy_cost_test.**test_OutOfStock_cost**()
Tests if cost value is received correctly when condition is Out of Stock on www.bestbuy.com.

1.2.4 BestBuyScraper_test module

BestBuyScraper_test.**test_InStock**()
Tests if the stock status value is received correctly for two stock status conditions (In Stock, Should be In Stock) on www.bestbuy.com.

BestBuyScraper_test.**test_OutOfStock**()
Tests if the stock status value is received correctly for two stock status conditions (Out of Stock, Should be Out of Stock) on www.bestbuy.com.

BestBuyScraper_test.**test_init**()
Tests if Bestbuy Scraper initializes properly.

1.2.5 Scraper_test module

Scraper_test.**test_ChooseScraper**()
Tests the scraper for Amazon, Bestbuy, and Walmart URLs.

Scraper_test.**test_init**()
Tests if Scraper initializes properly.

1.2.6 tracker_test module

tracker_test.**test_deleteAlert**()
Tests if alert gets removed properly.

tracker_test.**test_readState**()
Tests if the reading of the tracker.txt file is correct or not (to load preferences from previous run).

tracker_test.**test_updateStatus**()
Tests if the stock status change is updated properly.

1.2.7 Walmart_cost_test module

Walmart_cost_test.**test_InStock_cost**()
Tests if cost value is received correctly when condition is In Stock on www.beswalmarttbuy.com.

Walmart_cost_test.**test_OutOfStock_cost**()
Tests if cost value is received correctly when condition is Out of Stock on www.walmart.com.

1.2.8 WalmartScraper_test module

WalmartScraper_test.**test_InStock**()
Tests if the stock status value is received correctly for two stock status conditions (In Stock, Error Occurred) on www.walmart.com.

WalmartScraper_test.**test_OutOfStock**()
Tests if the stock status value is received correctly for two stock status conditions (Out of Stock, Error Occurred) on www.walmart.com.

WalmartScraper_test.**test_init**()
Tests if Walmart Scraper initializes properly.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex

PYTHON MODULE INDEX

INDEX

U

W