**Selenium Python Demo2:**

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

import csv

C\_Service = webdriver.ChromeService(executable\_path="/usr/bin/chromedriver")

driver=webdriver.Chrome(service=C\_Service)

**# Ask user for the number of pages to scrape**

user\_input = input("\nPlease enter number total page (Higher page counts will take longer): ")

time.sleep(3)

**# Function to scrape titles from a each single page**

def scrape\_titles\_from\_page(page\_number): driver.get(f'https://edition.cnn.com/search?q=&from={page\_number}0&size=10&page={page\_number}&sort=newest&types=all&section=')

WebDriverWait(driver,10).until(

EC.presence\_of\_element\_located((By.CLASS\_NAME,'container\_\_headline-text')))

titles = driver.find\_elements(By.CLASS\_NAME, 'container\_\_headline-text')

return [title.text for title in titles]

**# Function to save titles to a CSV file**

def save\_titles\_to\_csv(titles, csv\_file):

with open(csv\_file, mode='a', newline='', encoding='utf-8') as file:

writer = csv.writer(file)

writer.writerow([f'\nPage{current\_page}: '])

for title in titles:

writer.writerow([title])

**# Setup the CSV file**

csv\_file = 'cnnnews.csv'

max\_pages = int(user\_input)

current\_page = 1

**# Initialize the CSV file with headers**

with open(csv\_file, mode='w', newline='', encoding='utf-8') as file:

writer = csv.writer(file)

writer.writerow(['======Title Scraping Record======'])

writer.writerow(['News Titles: '])

**# Loop through each page and scrape the titles on terminal**

while current\_page <= max\_pages:

print(f"Scraping news processing......({current\_page}/{user\_input})")

titles = scrape\_titles\_from\_page(current\_page)

save\_titles\_to\_csv(titles, csv\_file)

current\_page += 1

time.sleep(3)

**# Print completion message and close the browser**

print("\n======================================================")

print("Total page already scraping: ", user\_input)

print("CSV record for all pages complete, pls check csv file.")

driver.quit()

Output screen:

文本

描述已自动生成

Record CSV file:

文本

描述已自动生成

Omit multiple pages……

文本

描述已自动生成

**Explain Code**

from selenium import **webdriver**

from selenium.webdriver.common.by import **By**

from selenium.webdriver.support.ui import **WebDriverWait**

from selenium.webdriver.support import **expected\_conditions** as EC

import **time**

import **csv**

**C\_Service** = webdriver.ChromeService(executable\_path="/usr/bin/chromedriver")

**drive**r=webdriver.Chrome(service=C\_Service)

* Requitment for selenium import:
  + **webdriver**- Basic selenium python requitment

*from selenium import webdriver*

* + **By**- Syntax to finding specific website html element,

*titles = driver.find\_elements(By.CLASS\_NAME, 'container\_\_headline-text')*

* + **WebDriverWait**- Syntax to operation delay for webdriver until next action

*WebDriverWait(driver,10).until(*

*EC.presence\_of\_element\_located((By.CLASS\_NAME,'container\_\_headline-text’))*

*)*

* + **expected\_condition**- Syntax to set condition for automated requitment

*EC.presence\_of\_element\_located((By.CLASS\_NAME,'container\_\_headline-text’))*

* + **time**- Syntax to manage time delay with secord to webdriver

*time.sleep (3)*

* + **csv**- Syntax to manage csv file to record selenium log

*def save\_titles\_to\_csv(titles, csv\_file):*

*with open(csv\_file, mode='a', newline='', encoding='utf-8') as file:*

*writer = csv.writer(file)*

*writer.writerow([f'\nPage{current\_page}: '])*

*for title in titles:*

*writer.writerow([title])*

*# Setup the CSV file*

*csv\_file = 'cnnnews.csv'*

* ***C\_service* and driver**
* location for internal executable path for chromedriver to make sure webdriver can running functionality

*C\_Service = webdriver.ChromeService(executable\_path="/usr/bin/chromedriver")*

*driver=webdriver.Chrome(service=C\_Service)*

**# Ask user for the number of pages to scrape**

user\_input = input("\nPlease enter number total page (Higher page counts will take longer): ")

time.sleep(3)

* They will ask user to enter number in terminal, number can enter custom self example I want scraping 14 pages titile for website CNN news:

文本

描述已自动生成

**# Function to scrape titles from a each single page**

def scrape\_titles\_from\_page(page\_number): driver.get(f'https://edition.cnn.com/search?q=&from={page\_number}0&size=10&page={page\_number}&sort=newest&types=all&section=')

WebDriverWait(driver,10).until(

EC.presence\_of\_element\_located((By.CLASS\_NAME,'container\_\_headline-text')))

titles = driver.find\_elements(By.CLASS\_NAME, 'container\_\_headline-text')

return [title.text for title in titles]

* They will driver get to website CNN news wait until class name for 'container\_\_headline-text' appear (to prevent network delays from causing code errors).
* Next, they will find element with class name ‘container\_\_headline-text’
* After than, they will record title for each page using variable ‘titles’ represent
* They will continue looping to doing that

**# Function to save titles to a CSV file**

def save\_titles\_to\_csv(titles, csv\_file):

with open(csv\_file, mode='a', newline='', encoding='utf-8') as file:

writer = csv.writer(file)

writer.writerow([f'\nPage{current\_page}: '])

for title in titles:

writer.writerow([title])

* They will save titles in single csv file, also create heading for each page and titles continue lopping

**# Setup the CSV file**

csv\_file = 'cnnnews.csv'

max\_pages = int(user\_input)

current\_page = 1

* Setup csv file name with ‘cnnnews.csv’
* Setup maxixum pages with user input number for 14
* Setup current\_page to heading

**# Initialize the CSV file with headers**

with open(csv\_file, mode='w', newline='', encoding='utf-8') as file:

writer = csv.writer(file)

writer.writerow(['======Title Scraping Record======'])

writer.writerow(['News Titles: '])

**# Loop through each page and scrape the titles on terminal**

while current\_page <= max\_pages:

print(f"Scraping news processing......({current\_page}/{user\_input})")

titles = scrape\_titles\_from\_page(current\_page)

save\_titles\_to\_csv(titles, csv\_file)

current\_page += 1

time.sleep(3)

* They will intialize the csv headers and lopping scraping news process on terminal until maximum 14 pages and stop

文本

描述已自动生成

**# Print completion message and close the browser**

print("\n======================================================")

print("Total page already scraping: ", user\_input)

print("CSV record for all pages complete, pls check csv file.")

driver.quit()

* After stop scraping, they will popup result scraping and ending in terminal

文本

描述已自动生成

* You can check result scraping at csv file