BROWSER AUTOMATION WITH SELENIUM #!pip install selenium In [1]: from selenium import webdriver import pandas as pd import time from selenium.webdriver.support.ui import Select In [2]: ## for opening a specific website by automated test software, chromedriver driver = webdriver.Chrome(executable_path= r'C:\Users\sunan\Downloads\chromedriver.exe') driver.get('https://www.imdb.com/') #To access any website # maximize window driver.maximize_window() # dropdown dropdown = driver.find_element_by_class_name('ipc-icon--arrow-drop-down') dropdown.click() ##automatically opens a dropdown menu time.sleep(1) #after navigating a dropdown bar will pause for 1second # accesses the linked page of advanced search from dropdown menu of IMDB website element = driver.find_element_by_link_text('Advanced Search') element.click() # click on avanced title search adv_title = driver.find_element_by_link_text('Advanced Title Search') adv_title.click() # selecting feature film feature_film = driver.find_element_by_id('title_type-1') feature_film.click() # select tv movie tv_movie = driver.find_element_by_id('title_type-12') tv_movie.click() # min date min_date = driver.find_element_by_name('release_date-min') min_date.click() min_date.send_keys('1990') ## starting time range # max date max_date = driver.find_element_by_name('release_date-max') max_date.click() max_date.send_keys('2020') ## last time range # rating min rating_min = driver.find_element_by_name('user_rating-min') rating_min.click() dropdown_2 = Select(rating_min) dropdown_2.select_by_visible_text('1.0') ##select_by_visible_text has been used for selecting exactly what is shown in dropdown menu # rating max rating_max = driver.find_element_by_name('user_rating-max') rating_max.click() dropdown_3 = Select(rating_max) dropdown_3.select_by_visible_text('10') # oscar Winning oscar_nominated = driver.find_element_by_id('groups-4') oscar_nominated.click() # color info color = driver.find_element_by_id('colors-3') color.click() # language language = driver.find_element_by_name('languages') dropdown_4 = Select(language) dropdown_4.select_by_visible_text('English') # 250 results results_count = driver.find_element_by_id('search-count') dropdown_5 = Select(results_count) dropdown_5.select_by_index(2) # submit submit = driver.find_element_by_xpath('(//button[@type="submit"])[2]') ##Xpath Expresssion for button tag submit.click() # current current_url = driver.current_url <ipython-input-2-0220e84c50e7>:3: DeprecationWarning: executable_path has been deprecated, please pass in a Service object driver = webdriver.Chrome(executable_path= r'C:\Users\sunan\Downloads\chromedriver.exe') <ipython-input-2-0220e84c50e7>:11: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead dropdown = driver.find_element_by_class_name('ipc-icon--arrow-drop-down') <ipython-input-2-0220e84c50e7>:16: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead element = driver.find_element_by_link_text('Advanced Search') <ipython-input-2-0220e84c50e7>:21: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead adv_title = driver.find_element_by_link_text('Advanced Title Search') <ipython-input-2-0220e84c50e7>:26: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead feature_film = driver.find_element_by_id('title_type-1') <ipython-input-2-0220e84c50e7>:30: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead tv_movie = driver.find_element_by_id('title_type-12') <ipython-input-2-0220e84c50e7>:35: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead min_date = driver.find_element_by_name('release_date-min') <ipython-input-2-0220e84c50e7>:40: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead max_date = driver.find_element_by_name('release_date-max') <ipython-input-2-0220e84c50e7>:46: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead rating_min = driver.find_element_by_name('user_rating-min') <ipython-input-2-0220e84c50e7>:53: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead rating_max = driver.find_element_by_name('user_rating-max') <ipython-input-2-0220e84c50e7>:59: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead oscar_nominated = driver.find_element_by_id('groups-4') <ipython-input-2-0220e84c50e7>:63: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead color = driver.find_element_by_id('colors-3') <ipython-input-2-0220e84c50e7>:67: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead language = driver.find_element_by_name('languages') <ipython-input-2-0220e84c50e7>:72: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead results_count = driver.find_element_by_id('search-count') <ipython-input-2-0220e84c50e7>:78: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead submit = driver.find_element_by_xpath('(//button[@type="submit"])[2]') ##Xpath Expresssion for button tag Breaking down all the steps In []: ## for opening a specific website by automated test software, chromedriver driver = webdriver.Chrome(executable_path= r'C:\Users\sunan\Downloads\chromedriver.exe') driver.get('https://www.imdb.com/') #To access any website In []: ## accessing another website driver = webdriver.Chrome(executable_path= r'C:\Users\sunan\Downloads\chromedriver.exe') driver.get('https://www.pexels.com/') going step by step..... In []: ## for opening a specific page by automated test software, chromedriver driver = webdriver.Chrome(executable_path= r'C:\Users\sunan\Downloads\chromedriver.exe') driver.get('https://www.imdb.com/') #To access any website dropdown = driver.find_element_by_class_name('ipc-icon--arrow-drop-down') dropdown.click() ##automatically opens a dropdown menu time.sleep(1) #after navigating a dropdown bar will pause for 1second In []: # advanced search from dropdown menu of IMDB website element = driver.find_element_by_link_text('Advanced Search') element.click() Opens IMDB website ----- Clicks on Dropdown menu, ALL ----- goes to the linked advanced serach page ## for opening a specific website by automated test software, chromedriver driver = webdriver.Chrome(executable_path= r'C:\Users\sunan\Downloads\chromedriver.exe') driver.get('https://www.imdb.com/') #To access any website # dropdown dropdown = driver.find_element_by_class_name('ipc-icon--arrow-drop-down') dropdown.click() ##automatically opens a dropdown menu time.sleep(1) #after navigating a dropdown bar will pause for 1second # accesses the linked page of advanced search from dropdown menu of IMDB website element = driver.find_element_by_link_text('Advanced Search') element.click() Opens IMDB website ----- Clicks on Dropdown menu, ALL ----- advanced serach page ----- advanced title search In []: ## for opening a specific website by automated test software, chromedriver driver = webdriver.Chrome(executable_path= r'C:\Users\sunan\Downloads\chromedriver.exe') driver.get('https://www.imdb.com/') #To access any website # maximize window driver.maximize_window() # dropdown dropdown = driver.find_element_by_class_name('ipc-icon--arrow-drop-down') dropdown.click() ##automatically opens a dropdown menu time.sleep(1) #after navigating a dropdown bar will pause for 1second # accesses the linked page of advanced search from dropdown menu of IMDB website element = driver.find_element_by_link_text('Advanced Search') element.click() # click on avanced title search adv_title = driver.find_element_by_link_text('Advanced Title Search') adv_title.click() Opens IMDB website --- Clicks on Dropdown menu, ALL --- advanced serach page --- advanced title search --- select: feature film, Podcast Series Checkboxes ## for opening a specific website by automated test software, chromedriver driver = webdriver.Chrome(executable_path= r'C:\Users\sunan\Downloads\chromedriver.exe') driver.get('https://www.imdb.com/') #To access any website # maximize window driver.maximize_window() # dropdown dropdown = driver.find_element_by_class_name('ipc-icon--arrow-drop-down') dropdown.click() ##automatically opens a dropdown menu time.sleep(1) #after navigating a dropdown bar will pause for 1second # accesses the linked page of advanced search from dropdown menu of IMDB website element = driver.find_element_by_link_text('Advanced Search') element.click() # click on avanced title search adv_title = driver.find_element_by_link_text('Advanced Title Search') adv_title.click() # selecting feature film feature_film = driver.find_element_by_id('title_type-1') feature_film.click() # select tv movie tv_movie = driver.find_element_by_id('title_type-12') tv_movie.click() Opens IMDB website --- Clicks on Dropdown menu, ALL --- advanced serach page --- advanced title search --- select: feature film, Podcast Series --- Release date Selecting time range ## for opening a specific website by automated test software, chromedriver driver = webdriver.Chrome(executable_path= r'C:\Users\sunan\Downloads\chromedriver.exe') driver.get('https://www.imdb.com/') #To access any website # maximize window driver.maximize_window() # dropdown dropdown = driver.find_element_by_class_name('ipc-icon--arrow-drop-down') dropdown.click() ##automatically opens a dropdown menu time.sleep(1) #after navigating a dropdown bar will pause for 1second # accesses the linked page of advanced search from dropdown menu of IMDB website element = driver.find_element_by_link_text('Advanced Search') element.click() # click on avanced title search adv_title = driver.find_element_by_link_text('Advanced Title Search') adv_title.click() # selecting feature film feature_film = driver.find_element_by_id('title_type-1') feature_film.click() # select tv movie tv_movie = driver.find_element_by_id('title_type-12') tv_movie.click() # min date min_date = driver.find_element_by_name('release_date-min') min_date.click() min_date.send_keys('1990') ## starting time range # max date max_date = driver.find_element_by_name('release_date-max') max_date.click() max_date.send_keys('2020') ## last time range Opens IMDB website --- Clicks on Dropdown menu, ALL --- advanced serach page --- advanced title search --- select: feature film, Podcast Series --- Release date --- dropdown: User rating Dropdown ## for opening a specific website by automated test software, chromedriver driver = webdriver.Chrome(executable_path= r'C:\Users\sunan\Downloads\chromedriver.exe') driver.get('https://www.imdb.com/') #To access any website # maximize window driver.maximize_window() # dropdown dropdown = driver.find_element_by_class_name('ipc-icon--arrow-drop-down') dropdown.click() ##automatically opens a dropdown menu time.sleep(1) #after navigating a dropdown bar will pause for 1second # accesses the linked page of advanced search from dropdown menu of IMDB website element = driver.find_element_by_link_text('Advanced Search') element.click() # click on avanced title search adv_title = driver.find_element_by_link_text('Advanced Title Search') adv_title.click() # selecting feature film feature_film = driver.find_element_by_id('title_type-1') feature_film.click() # select tv movie tv_movie = driver.find_element_by_id('title_type-12') tv_movie.click() # min date min_date = driver.find_element_by_name('release_date-min') min date.click() min_date.send_keys('1990') ## starting time range # max date max_date = driver.find_element_by_name('release_date-max') max_date.click() max_date.send_keys('2020') ## last time range # rating min rating_min = driver.find_element_by_name('user_rating-min') rating_min.click() dropdown_2 = Select(rating_min) dropdown_2.select_by_visible_text('1.0') ##select_by_visible_text has been used for selecting exactly what is shown in dropdown menu # rating max rating_max = driver.find_element_by_name('user_rating-max') rating max.click() dropdown_3 = Select(rating_max) dropdown_3.select_by_visible_text('10') Opens IMDB website --- Clicks on Dropdown menu, ALL --- advanced serach page --- advanced title search --- select: feature film, Podcast Series --- Release date --- dropdown: User rating --- checkbox Title Groups: Oscar Winning --- Color info : Colorized ## for opening a specific website by automated test software, chromedriver driver = webdriver.Chrome(executable_path= r'C:\Users\sunan\Downloads\chromedriver.exe') driver.get('https://www.imdb.com/') #To access any website # maximize window driver.maximize_window() # dropdown dropdown = driver.find_element_by_class_name('ipc-icon--arrow-drop-down') dropdown.click() ##automatically opens a dropdown menu time.sleep(1) #after navigating a dropdown bar will pause for 1second # accesses the linked page of advanced search from dropdown menu of IMDB website element = driver.find_element_by_link_text('Advanced Search') element.click() # click on avanced title search adv_title = driver.find_element_by_link_text('Advanced Title Search') adv_title.click() # selecting feature film feature_film = driver.find_element_by_id('title_type-1') feature_film.click() # select tv movie tv_movie = driver.find_element_by_id('title_type-12') tv_movie.click() # min date min_date = driver.find_element_by_name('release_date-min') min_date.click() min_date.send_keys('1990') ## starting time range # max date max_date = driver.find_element_by_name('release_date-max') max_date.click() max_date.send_keys('2020') ## last time range # rating min rating_min = driver.find_element_by_name('user_rating-min') rating_min.click() dropdown_2 = Select(rating_min) dropdown_2.select_by_visible_text('1.0') ##select_by_visible_text has been used for selecting exactly what is shown in dropdown menu # rating max rating_max = driver.find_element_by_name('user_rating-max') rating_max.click() dropdown_3 = Select(rating_max) dropdown_3.select_by_visible_text('10') # oscar Winning oscar_nominated = driver.find_element_by_id('groups-4') oscar_nominated.click() # color info color = driver.find_element_by_id('colors-3') color.click() Opens IMDB website --- Clicks on Dropdown menu, ALL --- advanced serach page --- advanced title search --- select: feature film, Podcast Series --- Release date --- dropdown: User rating --- checkbox Title Groups: Oscar Winning --- Color info : Colorized --- Language selection: by visible text --- Dropdown: Display options (accessing by index) ## for opening a specific website by automated test software, chromedriver driver = webdriver.Chrome(executable_path= r'C:\Users\sunan\Downloads\chromedriver.exe') driver.get('https://www.imdb.com/') #To access any website # maximize window driver.maximize_window() dropdown = driver.find_element_by_class_name('ipc-icon--arrow-drop-down') dropdown.click() ##automatically opens a dropdown menu time.sleep(1) #after navigating a dropdown bar will pause for 1second # accesses the linked page of advanced search from dropdown menu of IMDB website element = driver.find_element_by_link_text('Advanced Search') element.click() # click on avanced title search adv_title = driver.find_element_by_link_text('Advanced Title Search') adv_title.click() # selecting feature film feature_film = driver.find_element_by_id('title_type-1') feature_film.click() # select tv movie tv_movie = driver.find_element_by_id('title_type-12') tv_movie.click() # min date min_date = driver.find_element_by_name('release_date-min') min_date.click() min_date.send_keys('1990') ## starting time range max_date = driver.find_element_by_name('release_date-max') max_date.click() max_date.send_keys('2020') ## last time range # rating min rating_min = driver.find_element_by_name('user_rating-min') rating_min.click() dropdown_2 = Select(rating_min) dropdown_2.select_by_visible_text('1.0') ##select_by_visible_text has been used for selecting exactly what is shown in dropdown menu # rating max rating_max = driver.find_element_by_name('user_rating-max') rating_max.click() dropdown_3 = Select(rating_max) dropdown_3.select_by_visible_text('10') # oscar Winning oscar_nominated = driver.find_element_by_id('groups-4') oscar_nominated.click() # color info color = driver.find_element_by_id('colors-3') color.click() # language language = driver.find_element_by_name('languages') dropdown_4 = Select(language) dropdown_4.select_by_visible_text('English') # 250 results results_count = driver.find_element_by_id('search-count') dropdown_5 = Select(results_count) dropdown_5.select_by_index(2) In []: ## for opening a specific website by automated test software, chromedriver driver = webdriver.Chrome(executable_path= r'C:\Users\sunan\Downloads\chromedriver.exe') driver.get('https://www.imdb.com/') #To access any website # maximize window driver.maximize_window() # dropdown dropdown = driver.find_element_by_class_name('ipc-icon--arrow-drop-down') dropdown.click() ##automatically opens a dropdown menu time.sleep(1) #after navigating a dropdown bar will pause for 1second # accesses the linked page of advanced search from dropdown menu of IMDB website element = driver.find_element_by_link_text('Advanced Search') element.click() # click on avanced title search adv_title = driver.find_element_by_link_text('Advanced Title Search') adv_title.click() # selecting feature film feature_film = driver.find_element_by_id('title_type-1') feature_film.click() # select tv movie tv_movie = driver.find_element_by_id('title_type-12') tv_movie.click() # min date min_date = driver.find_element_by_name('release_date-min') min_date.click() min_date.send_keys('1990') ## starting time range # max date max_date = driver.find_element_by_name('release_date-max') max_date.click() max_date.send_keys('2020') ## last time range # rating min rating_min = driver.find_element_by_name('user_rating-min') rating_min.click() dropdown_2 = Select(rating_min) dropdown_2.select_by_visible_text('1.0') ##select_by_visible_text has been used for selecting exactly what is shown in dropdown menu # rating max rating_max = driver.find_element_by_name('user_rating-max') rating_max.click() dropdown_3 = Select(rating_max) dropdown_3.select_by_visible_text('10') # oscar Winning oscar_nominated = driver.find_element_by_id('groups-4') oscar_nominated.click() # color info color = driver.find_element_by_id('colors-3') color.click() # language language = driver.find_element_by_name('languages') dropdown_4 = Select(language) dropdown_4.select_by_visible_text('English') # 250 results results_count = driver.find_element_by_id('search-count') dropdown_5 = Select(results_count) dropdown_5.select_by_index(2) # submit submit = driver.find_element_by_xpath('(//button[@type="submit"])[2]') ##Xpath Expresssion for button tag submit.click() # current current_url = driver.current_url Supportive Materials Xpath Expresssion : https://www.lambdatest.com/blog/complete-guide-for-using-xpath-in-selenium-with-examples/ https://www.ibm.com/docs/en/app-connect/11.0.0?topic=xpath-creating-expressions