

Genesys Process Automation Documentation

Python script that automates report downloading from Genesys

Installation

Libraries

- Undetected-chromedriver ([undetected-chromedriver · PyPI](#))
- Selenium ([Selenium with Python](#))

Browser

- Google Chrome ([Google Chrome Web Browser](#))
- SelectorsHub ([SelectorsHub - Chrome Web Store](#))

Setup

- Login to a chrome with a google account, then sign into Genesys under the chrome profile (this saves login credentials).
- Find information:
 - Profile Path (chrome://version/) – enter this link under chrome.
 - Path to download folder
- Enter information under appropriate sections in script.

Selenium

Calling selenium

```
element = driver.find_element(By.ID, "passwd-id")
element = driver.find_element(By.NAME, "passwd")
element = driver.find_element(By.XPATH, "///input[@id='passwd-id']")
element = driver.find_element(By.CSS_SELECTOR, "input#passwd-id")
element.click()
element.send_keys('Hello world')
element.send_keys(Keys.ENTER)
```

Common names in script

Website HTML (inspect element)

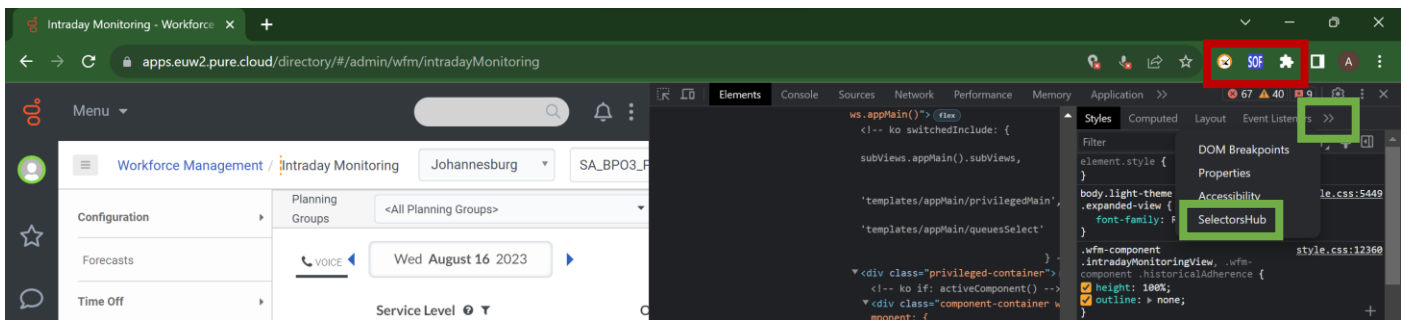
```
> <li class="breadcrumb-item-wfm">... </li> flex
v <li class="breadcrumb-item-intradayMonitoring"> flex
  ::before
  <span title="Intraday Monitoring" class="breadcrumb
  -text" xpath="1">Intraday Monitoring</span> == $0
</li>
<!-->
</ol>
> <div class="page-header-extras" style="padding-right:
```

SelectorsHub (chrome extension)

Selector	Selector Value
Rel cssSelector	span[title="Intraday Monitor...
Rel XPath	//span[@title="Intraday Mo...
index XPath	//span[@title="Intraday Mo...
testRigor Patl	"Intraday Monitoring"
jQuery	\$(span[title="Intraday Moni...

'Selector' and 'Selector Value' are referenced in defined functions.

Inspect Element



DO NOT USE EXTENSIONS TAB IN RED – SelectorsHub under extensions tab doesn't work.

Process

The script (as of 18/8/2023) takes 10 minutes 45 seconds to run, during this time it is advisable not to interact with the chrome browser the script is running on but a local machine can be lightly used. Before running the script I recommend opening Genesys once and ensuring that you are signed in and on a stable network connection. The script won't detect if the network cuts and will spit an error back out. Other points to note are:

- Keep logging in to Genesys through the Microsoft security key every month – Microsoft will automatically sign an account out after 30 days for security reasons.
- Clear temporary files – selenium creates large temporary files that can slow down a machine, every month I suggest clearing temporary files. On windows it can be done like so:
 - Windows + R
 - Type “%temp%”
 - Delete as many files the system allows within this folder
- The script can be run through the terminal or command prompt, this can be automated through the windows task scheduler.
 - <https://favtutor.com/blogs/run-python-file-terminal>
- Selenium doesn't rename files, two files with the name “test.csv”, the older one will be replaced not renamed.

Debugging

Debugging Tool

Most common IDEs come with a debugging tool. It is especially useful in this type of script as you can run the file line by line and properly isolate where the issue lies. Add breakpoints on lines of code that isn't running as expected and the file will stop at that line instead of continuing.

Tricks

- Stopping code at specific line: if you put an “input()” function, the code will stop at a line until enter key is hit in the console.
- Testing if element detection works in separate file. Sometimes the code trips up at detecting one specific button, if this is the case I like to run the code in a separate test file while only runs and detects one specific button to isolate the issue.
- Adding time, sometimes the code runs too fast and the webpage hasn't loaded the button yet, this leads to several glitches. Just adding some rest time between lines of code sometimes fixes the issue.

Common Error Messages

- **STALE_ELEMENT_REFERENCE** = [10, 'stale element reference']

Stale element means the selector value wasn't found, this could be because the wrong selector was used, changing selector method or ensuring that the element is in the right screen.

- **ELEMENT_NOT_VISIBLE** = [11, 'element not visible']

Selenium can only interact with buttons and elements that are available for the user to press, if it is hidden behind some other message or element, selenium won't be able to interact with it.

- **ELEMENT_IS_NOT_SELECTABLE** = [15, 'element not selectable']

This exception belongs to the same class as InvalidElementStateException. In such exceptions, the element is present on the web page, but the element cannot be selected. To handle this exception, the wait command must be used.

- **ELEMENT_NOT_INTERACTABLE** = [60, 'element not interactable']

Attempted to interact (click, send keys, etc.) to an uninteractable element, such as plain text. No input can be given to just plain text nor can it be clicked (hyperlinks are interactable, but not plain text).

Deliverables

Uploaded Files

test.py	File contains all libraries and defined functions from main.py, used to test specific buttons / web elements in isolation.
ostest.py	OS test was made to test the function used to rename files as two files can't have the same name.
main.py	Main python file, can run in isolation.
demo.py	Demo file made early in development. Presses and uses button under performance and workspace in Genesys.
databaseupload.py	File I modified from online resources made to upload files to the AWS server, incomplete and untested, but I left it here for convenience.