小试问题：

1. 提供一个代表性的网站，并给出要提取内容部分的一个范例
2. 数据存储格式是什么样比较好？（CSV/SQL文件/其他，最好有较好可读性）
3. 数据更新方式（只增加已有文件VS生成新的日期文件VS都要，邮件提醒类型和内容，涉及保存文件部分的命名等）

实际问题：

1. 详细需要提取信息的网站列表
2. 什么是需要的法规政策标准？是不是可能有很多类型的网页信息？
3. 就是怎么判断一个政策 是不是和你们相关 （答：有关键词 – 那需要一个关键词列表）
4. 网站是用什么写的？能否支持类似的格式提取？
5. 网站的相关信息格式（对应的关键字段），如果变动或者意思相同但是格式不同的兼容性
6. 网站是否包含大量文本（如新闻稿等）内容？
7. 网站是否有反爬虫系统
8. 爬虫拾取频次（每天一两次？）
9. 自动定时系统实现
10. 通知系统设置（邮件/钉钉/其他，哪个最为易行？）

SELECT:  
  
To interact with the dropdown menu on the website using Selenium in Python, you can utilize the Select class provided by the selenium.webdriver.support.ui module. This class offers methods to select options by their visible text, value, or index. Here's how you can modify your Python code to select the "全文" (fullText) option from the dropdown:

python

Copy

Edit

from selenium import webdriver

from selenium.webdriver.support.ui import Select

# Initialize the WebDriver (e.g., using Chrome)

driver = webdriver.Chrome()

# Navigate to the target website

driver.get('URL\_OF\_THE\_WEBSITE')

# Locate the dropdown element by its ID

dropdown = Select(driver.find\_element\_by\_id('titleOrfull'))

# Select the option with visible text "全文"

dropdown.select\_by\_visible\_text('全文')

# Continue with your data extraction logic here

# Close the WebDriver when done

driver.quit()

In this script:

Initialize the WebDriver: Ensure you have the appropriate WebDriver installed and configured for your browser. Replace 'URL\_OF\_THE\_WEBSITE' with the actual URL of the website you're targeting.

Locate the Dropdown Element: The find\_element\_by\_id method is used to locate the <select> element with the ID

You said:

Sorry I do not need you to provide the code, but to let me know how the "Select" function work.

So you are saying the select function is going to act as if you are clicking on it, but not like you are not re-coding the jsp file of the website, right?

ChatGPT said:

The Select class in Selenium is designed to interact with <select> elements in HTML, which are commonly used for dropdown menus. By using this class, you can simulate user interactions with dropdowns, such as selecting options, without directly modifying the website's code.

When you use the Select class to select an option from a dropdown, Selenium performs the following actions:

Locate the <select> element: Selenium identifies the dropdown element on the webpage.

Open the dropdown: It simulates the action of clicking on the dropdown to reveal the available options.

Select the desired option: Selenium selects the specified option, either by index, value, or visible text.

Trigger the onchange event: If the dropdown has an onchange event handler (as in your example with onchange="selectChange10(this)"), Selenium triggers this event to ensure that any associated JavaScript functions are executed.

By using the Select class, you can automate the process of interacting with dropdowns, ensuring that the website responds as it would to a real user selecting an option. This approach allows you to interact with the website's elements without the need to modify its underlying code.

For more detailed information on handling dropdowns with the Select class in Selenium, you can refer to the official Selenium documentation.

SELENIUM.DEV

Okay. WHat if it is not a dropdown menu? say something like this part of the code:

<!-- 主导航 -->

<div class="g\_newNav" id="daohang">

<div class="wrap">

<ul class="clearfix">

<li jsfl="" selectid="1">

<a href="#" class="">全部</a>

</li>

<li jsfl="xxgk" selectid="3">

<a href="#">信息公开</a>

</li>

<li jsfl="zxbs" selectid="4">

<a href="#">在线办事</a>

</li>

<li jsfl="hdjl" selectid="7">

<a href="#">互动交流</a>

</li>

<li jsfl="gayw" selectid="2">

<a href="#">公安要闻</a>

</li>

<li jsfl="gzdt" selectid="5">

<a href="#">工作动态</a>

</li>

<li jsfl="zcwj" selectid="9">

<a href="#">政策文件</a>

</li>

<li jsfl="zcjd" selectid="8">

<a href="#" class="current">政策解读</a>

</li>

</ul>

</div>

</div>

<!-- 主导航结束 -->

ChatGPT said:

To interact with a navigation menu like the one you've provided, where each item is represented by a <li> element containing an <a> tag, you can use Selenium to locate and click on the desired menu item. Here's how you can achieve this:

Locate the Menu Item: Identify the <a> tag within the <li> element that corresponds to the menu item you want to select.

Click the Menu Item: Use Selenium's click() method to simulate a user click on the identified menu item.

Here's an example in Python using Selenium:

python

Copy

Edit

from selenium import webdriver

# Initialize the WebDriver (ensure you have the appropriate driver installed)

driver = webdriver.Chrome()

# Navigate to the target website

driver.get('https://example.com')

# Locate the '政策解读' menu item by its text and click it

menu\_item = driver.find\_element\_by\_xpath("//a[text()='政策解读']")

menu\_item.click()

# Perform any additional actions or assertions here

# Close the browser

driver.quit()

Explanation:

driver.get('https://example.com'): Navigates to the target website.

driver.find\_element\_by\_xpath("//a[text()='政策解读']"): Locates the <a> tag with the exact text '政策解读'.

menu\_item.click(): Simulates a click on the located menu item.

Note: Ensure that the text inside the text() function matches exactly with the text in the <a> tag. If the text contains extra spaces or special characters, you might need to adjust the XPath accordingly.

For more advanced interactions, such as handling sub-menus or dynamic content, you might need to use Selenium's ActionChains class to perform mouse hover actions. This is particularly useful when dealing with menus that appear upon hovering.

TUTORIALSPOINT.COM

Additionally, if you encounter issues with elements not being clickable due to visibility or timing, consider using explicit waits to ensure the element is present and clickable before performing the click action.

Okay. Now I want to input some searching key words into its related search window. The jsp code is like below. What should we do? Say I want to type in "Electric Vehicle" and do the search.

<input type="text" class="s\_ipt fl" name="fullText" id="fullText" value="请输入您要搜索的内容" onblur="if(this.value==''){this.value='请输入您要搜索的内容';this.style.color='#c7cdd6';}" onfocus="if(this.value=='请输入您要搜索的内容'){this.value='';this.style.color='#333333';}" autocomplete="off" disableautocomplete="" style="color: rgb(51, 51, 51);">

ChatGPT said:

To automate the process of entering the search term "Electric Vehicle" into the input field and submitting the form using Selenium in Python, follow these steps:

Locate the Input Field: Identify the input field by its id attribute, which is fullText in this case.

Enter the Search Term: Use the send\_keys() method to input "Electric Vehicle" into the identified input field.

Submit the Form: After entering the search term, submit the form. This can be done by either pressing the Enter key or by locating and clicking the submit button.

Here's how you can implement this in Python using Selenium:

python

Copy

Edit

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

# Initialize the WebDriver (ensure you have the appropriate driver installed)

driver = webdriver.Chrome()

# Navigate to the target website

driver.get('https://example.com') # Replace with the actual URL

# Locate the input field by its ID

search\_input = driver.find\_element(By.ID, 'fullText')

# Clear any existing text in the input field

search\_input.clear()

# Enter the search term

search\_input.send\_keys('Electric Vehicle')

# Submit the form by pressing Enter

search\_input.send\_keys(Keys.RETURN)

# Alternatively, if there's a submit button, locate and click it

# submit\_button = driver.find\_element(By.ID, 'submit\_button\_id') # Replace with actual ID

# submit\_button.click()

# Perform any additional actions or assertions as needed

# Close the browser

driver.quit()

Explanation:

driver.find\_element(By.ID, 'fullText'): Locates the input field using its id attribute.

search\_input.clear(): Clears any existing text in the input field to ensure it's empty before entering new text.

search\_input.send\_keys('Electric Vehicle'): Inputs the search term into the field.

search\_input.send\_keys(Keys.RETURN): Simulates pressing the Enter key to submit the form. Alternatively, if there's a submit button, you can locate it by its id and click it to submit the form.

Ensure that the WebDriver (e.g., ChromeDriver) is correctly installed and that the path to the driver is set up properly.

For more detailed information on interacting with input fields and submitting forms using Selenium in Python, you can refer to this resource

Okay. Now it shows this part in jsp. Let us assume we are accessing the web using selenium. How can I get the title and links (URLs) to those results?

<div class="ListCon" id="info"> <div class="list item2"> <div class="clearfix"> <div class="txtRight fr"> <a href="http://www.mps.gov.cn/n6557563/c9386285/content.html" target="\_blank"><h3>一图看懂！公安部：全国<font color="red">新能源</font><font color="red">汽车</font>保有量已超过2000万辆</h3></a> <div class="extendAttr"> </div> <p></p> <span>所属栏目：政策解读 发布时间：2024-01-11</span> </div> </div> </div> <div class="list item2"> <div class="clearfix"> <div class="txtRight fr"> <a href="http://www.mps.gov.cn/n6557563/c7647931/content.html" target="\_blank"><h3>一图读懂：2020年全国新注册登记机动车3328万辆　<font color="red">新能源</font><font color="red">汽车</font>达492万辆</h3></a> <div class="extendAttr"> </div> <p></p> <span>所属栏目：政策解读 发布时间：2021-01-07</span> </div> </div> </div> <div class="list item2"> <div class="clearfix"> <div class="txtRight fr"> <a href="http://www.mps.gov.cn/n6557563/c5757122/content.html" target="\_blank"><h3>公安部交管局相关负责人就推广应用<font color="red">新能源</font><font color="red">汽车</font>专用号牌专访答记者问</h3></a> <div class="extendAttr"> </div> <p></p> <span>所属栏目：政策解读 发布时间：2017-08-14</span> </div> </div> </div> <div class="list item2"> <div class="clearfix"> <div class="txtRight fr"> <a href="http://www.mps.gov.cn/n6557563/c9297345/content.html" target="\_blank"><h3>《关于开展智能网联<font color="red">汽车</font>准入和上路通行试点工作的通知》解读</h3></a> <div class="extendAttr"> </div> <p></p> <span>所属栏目：政策解读 发布时间：2023-11-17</span> </div> </div> </div> </div>

ChatGPT said:

To retrieve the titles and URLs from the section you shared, here is how you can adjust the Python code using Selenium:

Locate the container: First, identify the div element with the ID info that contains all the search results.

Extract the links and titles: Each result is wrapped in an anchor (<a>) tag, where the title is in the <h3> tag, and the URL is in the href attribute of the anchor tag.

Here’s the Python code to extract the titles and URLs from the provided JSP structure:

python

Copy

Edit

from selenium import webdriver

from selenium.webdriver.common.by import By

# Initialize WebDriver (ensure you have the appropriate driver installed)

driver = webdriver.Chrome()

# Navigate to the target website

driver.get('http://your\_target\_website.com') # Replace with the actual URL

# Locate the container that holds the search results

results\_container = driver.find\_element(By.ID, 'info')

# Find all anchor tags within the results container

links = results\_container.find\_elements(By.TAG\_NAME, 'a')

# Extract the titles and URLs

for link in links:

title = link.find\_element(By.TAG\_NAME, 'h3').text # Get the title (inside <h3>)

url = link.get\_attribute('href') # Get the URL (href attribute)

print(f'Title: {title}')

print(f'URL: {url}')

print('---')

# Close the browser after retrieving the data

driver.quit()

How this works:

driver.find\_element(By.ID, 'info'): Locates the container (<div id="info">) that contains all the results.

find\_elements(By.TAG\_NAME, 'a'): Finds all anchor tags (<a>) inside the results\_container. These anchor tags contain the links to the articles.

link.find\_element(By.TAG\_NAME, 'h3').text: Extracts the text inside the <h3> tag, which is the title of the result.

link.get\_attribute('href'): Extracts the URL from the href attribute of the <a> tag.