# Homework

# Selenium IDE

## Fix the XPath

* Go to the following web page [www.dir.bg](http://www.dir.bg) and try to find the elements by their XPath.

Below there are some XPaths, to use them you have to fix them a bit. Make the necessary changes and locate the elements.

//div[@id=lenta-butoni-left]/ul/li[3]/a[@href='http://www.dir.bg']

//html/body/div/div/div/div[4]/ul/li[3]/a

C:\Users\ageorgieva\Desktop\dirbg.jpg

//div[@id='lenta-search']/form[@class='searchForm']/fieldset/div/input

//\*[@id="s"]

//ul[@id='lenta-butoni']/li[2]/

/html/body/div/div/div/ul/li[2]/a

As a result of this task you have to prepare Selenium Test case with the three correct XPaths.

## Create Automated Test

Below are some sample workflows. Create Selenium Test cases for them.

* **Yahoo! Test Case**

1. Open Firefox Web Browser
2. Open Selenium IDE (and minimize)
3. In the address bar, type <http://www.yahoo.com>
4. AssertTitle Yahoo!
5. In the search input button, type "qa testing"
6. Click on the "Web Search" submit button
7. Wait for Search Results to come on "http:/search.yahoo.com"
8. Assert the browsers title has the value "qa testing- Yahoo! Search Results"
9. Verify "qa testing" text is present anywhere in the search results.

After finishing your recording:

1. Stop recording
2. Save the test case
3. Run the test case
4. Review the test case
5. Fix any errors
6. Save the updated test case

* **Google Test Case**

1. Open Firefox Browser
2. Open Selenium IDE (and minimize)
3. In the address bar, type <http://www.google.com/>
4. AssertTitle Google
5. In the search input button, Type "QA Testing"
6. Click on the "Google Search" submit button
7. Wait for Search Results to be shown in the next page
8. Assert the browsers title has the value "qa testing - Google Търсене"
9. Verify "QA Testing" text is present anywhere in the search results

After finishing your recording:

1. Stop recording
2. Save the test case
3. Run the test case
4. Review the test case and fix any errors
   1. Option 1
5. Fast to slow
   1. Option 2
6. setSpeed
   1. Option 3
7. pause
   1. Option 4
8. Change AssertTitle to waitForTitle
9. Change VerifyTextPresent to waitForTextPresent
10. Save the reviewed test case

* **Take a Cake Test Case**

1. Open <http://www.take-a-cake.eu/>
2. Verify the “Вход” link button is present and click on it
3. Wait for the next page to be shown
4. Verify the name for forgotten password button is exactly “Забравена парола?” and click on it
5. Assert that the browser’s title has the value "Forgot Your Password - Take A Cake"
6. Click the “Продължи“ button
7. Verify that the user is prompted to fill in the empty field
8. Click on the „Към Вход“ link
9. Verify the h1 text is “Вход или Създаване на профил”

* **Make a record Test Case**

1. Go to <http://demos.kendoui.com/web/listview/editing.html>. Click the “Add new record” button. Save the record without entering values in the Product Name and Unit Price fields. Verify that the user is prompted to fill in the empty fields.
2. Open <http://demos.kendoui.com/web/listview/editing.html>. Think about appropriate test case and try to create an automated test for it.

* **Slider Test Case**

1. Open <http://demos.kendoui.com/web/slider/api.html>. In the input with id=”newValue” write “4” and click on “Set Value”. Verify that the correct value is shown on the slider.
2. Open <http://demos.kendoui.com/web/slider/api.html>. Think about appropriate test case and try to create an automated test for it.

* **QueueMetrics Test Case**

Open <http://demo.loway.ch/queuemetrics-livedemo/frontPage.jsp>. Enter "demo" for username and "demo" for password. In the drop down list for Agent report select the option to filter by **Mara (301).** Click on “Today” link to see her Quick reports. If the Period start date is matching the current date, alert a message “Matching”, otherwise alert “Not Matching”.

Tips: The start hour always should be 00:00

* **Store.bg Test Cases**

1. Open <http://www.book.store.bg>. In the textbox with name “srchstr” write “JavaScript” and click on “Търси”. Verify that the results are for JavaScript.
2. Open <http://www.book.store.bg>. In the textbox write “JavaScript”. Verify that the radio button „всички раздели” is selected and select „само в книги“. Click on “Търси” and verify that above the search results the correct search options are shown.
3. Open <http://www.book.store.bg>. In the textbox write “JavaScript”. Verify that the radio button „всички раздели” is selected and select „само в книги“. Click on “Търси” and verify that the first book in the search results is about JavaScript (use regular expressions). Store the title of the book in a variable. Follow the link to the book (the title of the book). Then click to see the window for calculation of a delivery for another town („щракнете тук“). Verify that the popup that opens contains the title of the book. Use the stored variable for the verification.
4. Open <http://www.book.store.bg>. If the radio button “всички раздели” is clicked search for “Melanie Delon”. If the other radio button is clicked, first click on “всички раздели” and then do the same search. Verify the search results.
5. Try to create an automated test by writing it yourself without using the record option.
6. Create a Test Suite containing all those tests.

* **Etsy Test Cases**

1. Open <http://www.Etsy.com>. Navigate to Category – Furniture -> Chair and click on the first menu item in the result list. Verify the item details page is displayed. Assert the Price of the product shown on this page.
2. Open <http://www.Etsy.com>. Click on Kids section on left, on Kids page click on tweet button, switch to new window, sign in with wrong username and password to twitter and check the error.

## Exercise for Experts

### Regular Expressions Results

If you use the following regular expressions what will be a possible result?



|  |  |
| --- | --- |
| Regular Expression | Result (Xpath) |
| [A-z0-9.\_%+-]+@[A-z0-9.-]+\.[A-z]{2,4} | //html/body/div/div[3]  //html/body/div/div[5]/div/a |
| \(+[0-9]{2}+\)+\s+[0-9]{3}+\s+[0-9]{4} | //html/body/div/div[3] |
| \d{4}\s\d{3}\s\d{2} | //html/body/div/div/div[2]/div[2]/strong |
| \d{2}-\d{2}-\d{4} | //html/body/div/div[7] |
| \w+\,+\s+\d{1,2}\s+\w+\s+\d{4}\,+\s+\d{2}\:\d{2}\:\d{2} | //html/body/div/div/div/p |

* Use these expressions to verify that text is present in the html file Index.html. Prepare Selenium Test Case with verifications.