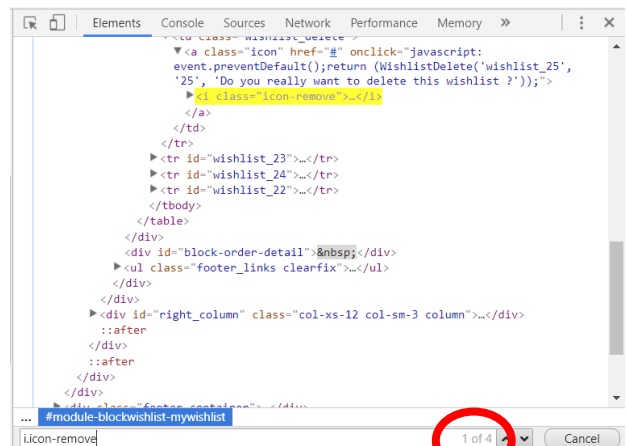


Huiswerk Opdracht

Overtime!

WebTables!

Name	Qty	Viewed	Created	Direct Link	Default	Delete
Cardio Grind	0	0	2017-07-06	View	<input type="checkbox"/>	<input type="checkbox"/>
Hiit List	0	0	2017-07-06	View	<input type="checkbox"/>	<input type="checkbox"/>
Leg Day	0	0	2017-07-06	View	<input type="checkbox"/>	<input type="checkbox"/>
No Pain No Gain	0	0	2017-07-06	View	<input checked="" type="checkbox"/>	<input type="checkbox"/>



En nu?

WebTables!

1 →

2 ↓

Name	Qty	Viewed	Created	Direct Link	Default	Delete
No Pain No Gain	0	0	2017-07-06	View	<input checked="" type="checkbox"/>	✕

3 →

Hoe kom ik veilig bij dit element?

```

<table class="table table-bordered">
  <thead> == $0
    <tr>
      <th class="first_item">Name</th>
      <th class="item mywishlist_first">Qty</th>
      <th class="item mywishlist_first">Viewed</th>
      <th class="item mywishlist_second">Created</th>
      <th class="item mywishlist_second">Direct Link</th>
      <th class="item mywishlist_second">Default</th>
      <th class="last_item mywishlist_first">Delete</th>
    </tr>
  </thead>
  <tbody>...</tbody>
</table>

```

```

<table class="table table-bordered">
  <thead>...</thead>
  <tbody> == $0
    <tr id="wishlist_22">
      <td style="width:200px;">...</td>
      <td class="bold align_center"> 0</td>
      <td>0</td>
      <td>2017-07-06</td>
      <td>...</td>
      <td class="wishlist_default">...</td>
      <td class="wishlist_delete">...</td>
    </tr>
  </tbody>
</table>

```

3



Testcase Home – Delete een wishlist

- Opdrachten bijlage -> HomeWork
- Denk aan validatie!
- Kijk goed naar je referentie materiaal
- Ja, er zit nog een valkuil in.
- Ja, het kan echt met Selenium ;)

4



Testcase Home – Delete een wishlist

```

20      @Test
21      public void deleteWishList() {
22          WebDriverWait wait = new WebDriverWait(driver, 10);
23
24          driver.manage().window().maximize();
25
26          // Open the website
27          driver.get("https://techblog.polteq.com/testshop/index.php");
28
29          // Click on the login link
30          driver.findElement(By.className("login")).click();
31
32          // Enter username
33          driver.findElement(By.id("email")).sendKeys("bootcamper@feelthepain.com");
34
35          // Enter password
36          driver.findElement(By.id("passwd")).sendKeys("1qazxsw2");
37
38          // Click the Log in button
39          driver.findElement(By.id("SubmitLogin")).click();
40
41          //Click the wishlistButton
42          driver.findElement(By.cssSelector("i.icon-heart")).click();

```

5



Testcase Home – Delete een wishlist

```

44      //Step 1: Locate the webtable.
45      WebElement wishListTable = driver.findElement(By.cssSelector(".table.table-bordered"));
46
47      //Step 2: Locate rows(horizontal) of the webtable within the webtable itself.
48      List<WebElement> rowsInTable = wishListTable.findElements(By.tagName("tr"));
49
50      //Step 2b: Calculate no of rows In the webtable.
51      int rowCount = rowsInTable.size();
52      System.out.println(rowsInTable.size());
53
54      //Step 4: Get the first column(vertical) from the first row which has all the header titles
55      // th -> table header
56      List<WebElement> headerRowInTable = rowsInTable.get(0).findElements(By.tagName("th"));
57
58      //Step 4b: Calculate no of columns present
59      int columnsCount = headerRowInTable.size();
60      System.out.println(columnsCount);
61

```

6



Testcase Home – Delete een wishlist

```

62 //Step 5: Get the index for the Name column and Delete column. We achieve this by looping
63 // over the List from the headerRowInTable containing the columns
64 int nameColumn = -1;
65 int deleteColumn = -1;
66 for (int i = 0; i < headerRowInTable.size(); i++) {
67     System.out.println(headerRowInTable.get(i).getText());
68     if (headerRowInTable.get(i).getText().equals("Name")) {
69         nameColumn = i;
70     } else if (headerRowInTable.get(i).getText().equals("Delete")) {
71         deleteColumn = i;
72     }
73 }
74
75 //Step 6: Assert that both Name and Delete have been found
76 Assertions.assertThat(nameColumn > -1 && deleteColumn > -1)
77     .as("Check if Name and Delete were found in the headerRow")
78     .isTrue();
79
80 System.out.println(nameColumn);
81 System.out.println(deleteColumn);
82

```

7



Testcase Home – Delete een wishlist

```

83 //Step 7: Find the correct row which contains the wishlist we are looking for
84 // SKIP HEADER ROW
85 String listToDelete = "Leg Day";
86
87 //-----Method building
88 Assertions.assertThat(checkWebTableForName(rowsInTable, nameColumn, listToDelete))
89     .as("Check if name can be found in the webtable")
90     .isTrue();
91 //-----Method building
92

```

8



Testcase Home – Delete een wishlist

```

93 boolean wishListFound = false;
94 for (int i = 1; i < rowsInTable.size(); i++) {
95     //Step 7b: System.out.println("Row" + i + ":" );
96     // Grab one of the rows in the webtable by using the index (i), the get all the columns
97     // and place them in a new List. Now wishListColumn represents all the columns in row(i)
98     List<WebElement> wishListColumn = rowsInTable.get(i).findElements(By.tagName("td"));
99     for(WebElement webElement : wishListColumn ){
100         System.out.println(webElement.getText());
101     }
102     // Step 7c: Since we now which column number holds the names of the wishlist (step5 nameColumn) we can
103     // directly get that column number and check the contents of this cell
104     if (wishListColumn.get(nameColumn).getText().equals(listToDelete)) {
105         //System.out.println(wishListColumn.get(deleteColumn).getAttribute("class"));
106         //Step 7d: We have the correct row which has the name of the wishlist as first column
107         // Now we can click the delete icon, again a column number we already now (step5 deleteColumn)
108         wishListColumn.get(deleteColumn).findElement(By.cssSelector("a.icon")).click();
109         wishListFound = true;
110         //Step 7e: We break the loop otherwise java continue but the website is trying to process the delete
111         break;
112     }
113 }
114 //Step 7f: Assert we found the wishlist
115 Assertions.assertThat(wishListFound).as("Could not find the given wishlist").isTrue();

```

9



Testcase Home – Delete een wishlist

```

117 //Step 8: Accept the Javascript alerts
118 try {
119     //Wait 10 seconds till alert is present
120     Alert alert = wait.until(ExpectedConditions.alertIsPresent());
121     //Accepting alert.
122     alert.accept();
123     System.out.println("Accepted the alert successfully.");
124 } catch (Throwable e) {
125     System.err.println("Error came while waiting for the alert popup. " + e.getMessage());
126 }
127
128
129

```

10



Testcase Home – Delete een wishlist

```

130 //Step 9: Refresh the WebTable, element has become stale meaning the webtable in memory of the code
131 //      is not the same anymore as the webtable on the site (- 1 wishlist)
132 //      We wait for the "Feel the pain" text to be gone by inverting the TextToPresentInElement
133
134 wait.until(ExpectedConditions.not(ExpectedConditions.textToBePresentInElement(wishListTable, listToDelete)));
135 wishListTable = driver.findElement(By.cssSelector(".table.table-bordered"));
136 rowsInTable = wishListTable.findElements(By.tagName("tr"));
137
138 //Step 10: Check if the deletion was successfull. We can reuse code by looking for the name and place it in a method
139 //      called checkWebTableForName
140 Assertions.assertThat(checkWebTableForName(rowsInTable, nameColumn, listToDelete))
141     .as("Check if delete was successfull")
142     .isFalse();
143
144 //Step 11: Add the wishlist back so the can is 100% re-runable with out any code or function adjustments
145 driver.findElement(By.cssSelector("input#name")).sendKeys(listToDelete);
146 driver.findElement(By.cssSelector("button#submitWishlist")).click();

```

11



Testcase Home – Delete een wishlist

```

148 //Step 12: Refresh WebTable, element has become stale. A wishlist has been added. We wait for the
149 //      Webtable to reload after adding the list back and set the variable wishListTable again
150 wait.until(ExpectedConditions.elementToBeClickable(By.cssSelector(".table.table-bordered")));
151 wishListTable = driver.findElement(By.cssSelector(".table.table-bordered"));
152 rowsInTable = wishListTable.findElements(By.tagName("tr"));
153
154 //Step 13: Assert if the newly added wishlist can be found
155 //-----Method building
156 Assertions.assertThat(checkWebTableForName(rowsInTable, nameColumn, listToDelete))
157     .as("Check if adding the list back was successfull")
158     .isTrue();
159 //-----Method building
160
161 driver.quit();
162 }

```

12



Testcase Home – Delete een wishlist

```
165 public boolean checkWebTableForName(List<WebElement> rowsInWebTable, int columnToCheck, String nameToFind){
166     for(int i = 1; i < rowsInWebTable.size(); i++){
167         List<WebElement> wishListColumn = rowsInWebTable.get(i).findElements(By.tagName("td"));
168
169         if (wishListColumn.get(columnToCheck).getText().equals(nameToFind)){
170             return true;
171         }
172     }
173     return false;
174 }
```