

How to handle dynamic changing ID's In XPath?

Asked 3 years, 7 months ago Active 10 months ago Viewed 60k times



How to Handle Dynamic Changing IDs In XPath? for example :

17

```
//div[@id='post-body-3647323225296998740']/div[1]/form[1]/input[1]
```



In this xpath id "3647323225296998740" is changing every time when reloading the page.



Is there any way to handle this type of situation in Selenium?

9

automated-testing

selenium

selenium-webdriver

testng

xpath

edited Dec 23 '16 at 14:03



dzieciou

9,291

4

35

75

asked Apr 27 '16 at 14:08



Bharat Mane

6,003

7

29

57

How to Handle Dynamic Changing IDs? for example , ID

is "ss2j7z0k3x5r4y6g1rSR0MLLUG65N65Z75U72R61G6eG6dP73Y" – nancy smith Jul 21 '17 at 5:07

- 1 Actually dynamic IDs appearing on pages are a code smell that you should be querying. Particularly addressing the question, 'is it leaking information about the internals of the system?' In general dynamic ID should be avoided. Further more, if automated testing is non-functional requirement of the project, suitable ID should be provided as per a proper locator strategy. – Martin Spamer Oct 11 '18 at 16:43

5 Answers



At this situation We have lots of options-

16

Option 1 : Look for any other attribute which is not changing every time in that div node like name, class etc. So if this div node has class attribute then we can write xpath as below.



```
//div[@class='post-body entry-content']/div[1]/form[1]/input[1]
```

Option 2 : We can use absolute xpath (full xpath) where you not need to give any attribute names in xpath.

```
/html/body/div[3]/div[2]/div[2]/div[2]/div[2]/div[2]/div[2]/div/div[4]/div[1]/div/div/div/
```



Option 3 : We can use starts-with function. In this xpath's ID attribute, "post-body-" part remains the same every time.

```
//div[starts-with(@id, 'post-body-')]/div[1]/form[1]/input[1]
```

Option 4 : We can use contains function. Same way you can use contains function as bellow.

```
div[contains(@id, 'post-body-')]/div[1]/form[1]/input[1]
```

edited Dec 23 '16 at 14:03



dzieciou

9,291 4 35 75

answered Apr 28 '16 at 9:32



Bharat Mane

6,003 7 29 57

1 Imho, option 2 is not recommended, it is too brittle – olyv Nov 28 '16 at 6:16

Yes, maximum time we avoids to use Absolute x-path, but sometimes relative xpath is not working while locating the element, in that case we have to use absolute xpath: in which we have to design absolute path from starting node to descent element node. – Bharat Mane Dec 22 '16 at 9:05

Also, don't forget that you can check multiple parts of an id: constant1-12232-constant2: "/*[contains(@id,'constant1') and contains(@id,'constant2')]" – tjb May 3 at 9:13

Use contains() or starts-with() function in XPath:

Solution:

With contains() :

```
//div[contains(@id, 'post-body-')]/div[1]/form[1]/input[1]
```

With starts-with() :

```
//div[starts-with(@id, 'post-body-')]/div[1]/form[1]/input[1]
```

edited Jun 27 '17 at 13:13



alecxe ♦

10.6k 7 37 92

answered Apr 27 '16 at 14:11



Sagar007

1,150 2 7 21

Use other elements and attributes for partial and combination matches

Generally in these cases of database backed and changing ID's in web element selectors your options are to look for one or more of:

- a higher level selector such as the form name, e.g. `//form/input[@type='submit']`
- a combo of non-unique selectors that together are unique
e.g. `//form[@class='new_user']/input[@type='submit']`
- a data attribute such as `form[@data-form='new-user']//input`
- a partial match using starts-with or contains against post-body- e.g.
`//div[starts-with(@id, 'post-body-')]/div[1]/form[1]/input[1]`

edited Jul 21 '17 at 10:49

answered Nov 26 '16 at 22:07



Michael Durrant

18.2k 2 24 77



If the id is changing dynamically for a particular xpath then we can use the below xpath functions:

1

1. contains : xpath Syntax: `//html tag[contains(text())/@attribute, 'Expected Value']`
2. Following-Siblings:

Xpath Syntax : `//html tag/following-siblings::html tag`
ex: `//img[@src = 'image2.jpg']/following-siblings::input[@type = 'button']`

3. Preceding-Siblings: syntax will be same as following-Siblings
4. The descendant function which is used to identify the grand child node by skipping all sub, node from the current HTML tag.

edited Jul 21 '17 at 6:57



Bharat Mane

6,003 7 29 57

answered May 3 '16 at 6:50



Anubhav Mishra

13 7

1 I don't think "Siblings" will work in this situation. – [Bharat Mane](#) May 4 '16 at 6:05



You can use **regex** in this case, Instead of using XPath you can use some wild cards in **CSS Selector** to locate the same. Please [refer this link](#).

0

`input[id^='post-body-'][id$='id pattern']`



[Refer this link](#) for knowing how to use CSS selectors using wildcards.
