

< Teach
Me
Skills />

Selenium WebDriver. Локаторы



Локаторы

Локатор - это строка, уникально идентифицирующая UI-элемент. Когда вы делаете клик мышкой, ввод текста и прочие действия, вы эти действия выполняете над вполне конкретным объектом. Селениум поступает так же. Но поскольку он не умеет читать ваши мысли, то ему надо четко указать объект, для которого надо применить то или иное действие



Id

- **By.id("<SOME_ID>")**- соответствует элементу, у которого атрибут id равен значению SOME_ID. Например, у нас есть элемент, который в HTML записывается так:

```
<input type='text' id='some_input_id' name='some_input_name' value="" /> В
```

этом случае локатор будет иметь вид: **By.id("some_input_id")**.

Также следует отметить, что данный вид локаторов является одним из самых быстрых в нахождении и одним из самых уникальных. Это связано с тем, что в DOM-структуре ссылки на элементы, у которых задан ID, хранятся в отдельной таблице и через JS (собственно именно через него осуществляется доступ к элементам на конечном уровне) обращение к элементам по ID идет достаточно короткой инструкцией, наподобие SOME_ID.

Name

- **By.name("SOME_NAME")** - соответствует элементу, у которого атрибут name равен значению SOME_NAME. Эффективно применяется при работе с полями ввода формы (кнопки, текстовые поля, выпадающие списки). Если брать предыдущий пример:

<input type=text id='some_input_id' name='some_input_name' value="" /> то данный элемент может быть также идентифицирован локатором вида

By.name("some_input_name").

Данный тип локаторов тоже является достаточно быстрым в нахождении, но менее уникальным, так как на странице может быть несколько форм, у которых могут быть элементы с одинаковым именем.

LinkText и PartialLinkText

By.linkText("SOME TEXT") или **By.partialLinkText("SOME")** - специально для ссылок используется отдельно зарезервированный тип локаторов, который находит нужную ссылку по ее тексту. Это сделано отчасти потому, что ссылки как правило не имеют таких атрибутов как ID или name. Соответственно, ссылка, которая в HTML записывается так:

Link Text 2345 в Селениуме идентифицируется локатором **By.linkText("Link Text 2345")** или по части слова **By.partialLinkText("2345")**

ВАЖНО: Работает только с тегами **<a>**

TagName

By.tagName("SOME_TAG") - для поиска по названию тега используется локатор **tagName**. Не рекомендуется к использованию так как в большинстве случаев различные теги представлены в больших количествах, но порой бывают довольно редкие, например **textarea** или **iframe**

<div>some div</div> в Селениуме идентифицируется локатором **By.tagName("div")**



ClassName

By.className("SOME_CLASS") - соответствует элементу, у которого атрибут class содержит значение SOME_CLASS. Проблема локатора в том, что поиск может осуществляться **только по одному классу**. Например для элемента:

```
<input class="gLfyf gsfi" />
```

валиден следующий локатор: **By.className("gLfyf")**, а вот полный список классов **By.className("gLfyf gsfi")** выдаст NoSuchElementException.

CSS - еще больше



```
5 .padding {  
6   padding: 0.5em;  
7 }  
17 .padding-H-05em {  
18   padding-left: 0.5em;  
19   padding-right: 0.5em;  
20 }  
21 .padding-V-05em {  
22   padding-top: 0.5em;  
23   padding-bottom: 0.5em;  
24 }  
25 )
```

- Поиск по Id: **By.cssSelector("#id");**
- Поиск по class: **By.cssSelector(".class");**
- Получение любого тега по значению атрибута:
By.cssSelector("a[target=_blank]")
- **Синтаксис CSS-селекторов**

CssSelector

By.cssSelector("CSS_PATH") - данный тип локаторов основан на описаниях таблиц стилей (CSS), соответственно и синтаксис такой же. В отличие от локаторов по ID, по имени или по тексту ссылки, данный тип локаторов может учитывать иерархию объектов, а также значения атрибутов, что делает его ближайшим аналогом XPath. А в силу того, что объект находится по данному локатору быстрее, чем XPath, рекомендуется прибегать к помощи CSS вместо XPath.

XPath

By.xpath("SOME_XPATH") - наиболее универсальный тип локаторов. HTML, как и его более обобщенная форма - XML, представляет собой различное сочетание тегов, которые могут содержать вложенные теги, а те в свою очередь тоже могут содержать теги и т.д. То есть, выстраивается определенная иерархия, наподобие структуры каталогов в файловой системе. И задача XPath - отразить подобный путь к нужному элементу, с учетом иерархии. Например, XPath вида:

A/B/C/D указывает на некоторый элемент с тегом D, который находится внутри тега C, а тот в свою очередь - внутри тега B, который находится внутри тега A, который находится на самом верхнем уровне иерархии.

XPath

Если брать использование XPath в Селениуме, то там зачастую полный путь указывать не нужно, более того, вредно, особенно, если вложенность тега нужного элемента достаточно высока. Как правило, удобно указывать путь, начиная с некоторого промежуточного элемента, пропуская теги более высокого порядка.

Например, такой XPath: `//table/tbody/tr/td/a` ссылается на первую ссылку в первой строке тела первой таблицы. Обратите внимание на начало данной записи. Строка `'//'` означает, что поиск элемента начинается с некоторого произвольного места.

XPath

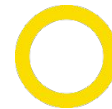
У XPath есть много удобств, но есть и основной недостаток - **низкая скорость нахождения объекта**. В частности, с подобной проблемой можно столкнуться при работе с IE, так как при работе с XPath под IE используются JScript-библиотеки, которые не отличаются высокой скоростью выполнения. В таких случаях рекомендуется воспользоваться CSS-локаторами (см. ниже), но в некоторых случаях от XPath уйти не получится.



XPath - основы

- ▶ `//div` - искать внутри всей страницы тег `div`
- ▶ `//div/a` - найти тег `div` и ТОЛЬКО внутри него искать тег `a`
- ▶ `//*[a]` - найти любой тег, внутри него искать тег `a`
- ▶ `//form/..` - выход на один уровень вверх (как с директориями на компьютере)
- ▶ `(//div)[5]` - получение пятого элемента, найденного по xpath, указанному в круглых скобках

XPath - еще больше



- Получение любого тега по любому атрибуту:
`By.xpath("//tag[@attribute='value']");`
- Получение любого тега по тексту:
`By.xpath("//tag[text()='text']");`
- Получение любого тега по частичному совпадению текста/атрибута:
`By.xpath("//tag[contains(text(),'text')]");`
- Синтаксис и дополнительные фичи
- Xpath Axes