Technically WebDriver is an **interface** in Java (element of selenium) that talks to the browser. At the beginning there was only an interface with static and default methods, and then classes that implemented that interface apierd, like ChromeDriver, FireFoxDriver… each of them responsible for working with a specific browser .

Per Selenium’s documentation, the methods in this interface fall into 3 categories:

1- Control of the browser itself

2- Selection of WebElements

3- Debugging aids

WebDriver - is also a webAPI. Why it is an API, because it consists of methods that we use in java automation script (client), and those methods of WebDriver (API) actually perform actions with use of browser that sends requests to AUT эют (**A**pplication **U**nder **T**est) it is has a role of Server.

API- s are designed to be used by programs they don't have a friendly interface like a website. A web API - is a set of rules and protocols that define how two systems can communicate with each other over the internet.

Just like the browser methods provided by WebDriver, we can also access the navigation methods

provided by WebDriver by typing driver.navigate().

Вебдрайвер имеет 5 групп методов.

1) - get methods

**get(url)**

**getTitle()**

**getCurrentUrl()**

**getPageSource**()

**getWindowHandle()**-вернёт id вкладки, на которой в моменте сфокусирован браузер

**getWindowHandles()** - вернёт сет id-шак

2) - conditional methods, returns true of false

**isDesplayed()** - элемент отоброжается на странице или нет

**isEnabled()** - (enabled- это про поведение, например включён или выключен)

**isSelected()** - for radioButtons, checkBoxes, dropDowns, т.е. не для всех веб элементов

3) - navigational methods

**driver.**

**navigate().to(String or url);** ----------------------- same as driver.get(only String)

**navigate().forward();** -------------------------равнозначно нажатию стрелки назад или вперёд в браузере

**navigate().back();**

**navigate().refresh()** —-----------------------обновить браузер

4) - browser methods

**close() -** закрывает **текущую вкладку** на которой сфокусирован драйвер (но сущность драйвера остаётся)

**quit()** - закрывает все вкладки (окна) браузера (которые он открывал, те что ты открыл вручную не будут закрыты, так как браузер их не открывал и ничего о них не знает)

5) - wait methods (are used to solve synchronization problems)

**Смотри Synchronization strategy.**[**https://drive.google.com/drive/u/0/folders/1QbYG97MEhkgYvbuu1q2nns1IogdtGQhq**](https://drive.google.com/drive/u/0/folders/1QbYG97MEhkgYvbuu1q2nns1IogdtGQhq)

-----------------------------------------------------------------------------------------------------------------------

0) Драйвер умеет проходить basic autorization на сайте или приложении(это всплывающее окно сразу при открытии сайта)

например у меня есть URL:regoffice.senla.eu/, тогда авторизация(username, password) ставится в сам URL

driver.get("**https://user:senlatest@regoffice.senla.eu/**");

-----------------------------------------------------------------------------------------------------------------------

1) driver.navigate().to("www.javatpoint.com"); In WebDriver, this method loads a new web page

in the existing browser window. It accepts String as parameter and returns void.

-----------------------------------------------------------------------------------------------------------------------

2) **driver.navigate().forward();** The respective command that takes you forward by one page

on the browser's history.It neither accepts anything nor returns anything.

-----------------------------------------------------------------------------------------------------------------------

3) **driver.navigate().back();** The respective command that takes you back by one page on

the browser's history.It neither accepts anything nor returns anything.

-----------------------------------------------------------------------------------------------------------------------

4) **driver.navigate().refresh();** this method refresh/reloads the current web page in the

existing browser window. It neither accepts anything nor returns anything.

-----------------------------------------------------------------------------------------------------------------------

5) **driver.navigate().equals()**;<- метод из класса Object(проверка на равенство).

-----------------------------------------------------------------------------------------------------------------------

6) **driver.getCurrentUrl();** возврасчаеет текущий адрес страницы(URL)

-----------------------------------------------------------------------------------------------------------------------

7) **driver.getPageSource();** возвращает источник строницы. html+всякие непонятные инструкции к нему

-----------------------------------------------------------------------------------------------------------------------

8) **driver.getTitle();** возвращает titel из текущего html документа/странцы

-----------------------------------------------------------------------------------------------------------------------

9) **String Text = driver.findElement(By.id("Text")).getText();** The command is used to retrieve

the inner text of the specified web element.The command doesn’t require any parameter and

returns a string value.<span>Текст который находится между</span>

-----------------------------------------------------------------------------------------------------------------------

10) **driver.findElement(By.id("someId")).getAttribute("название атрибута значение которого**

**хотим узнать");**The command requires a single string parameter that refers to an attribute

whose value we aspire to know and returns a string value as a result.

-----------------------------------------------------------------------------------------------------------------------

**driver.findElement(By.tagName("button"));** <--ищет элемент по названию тэга, если элементов с таким наз

ванием тэга несколько то вернёт первый

11)

driver.get("http://www.av.by");

String windowHandle=driver.getWindowHandle(); какбы записывает уникальный адрес окна браузера

**driver.findElement(By.cssSelector("div[id='adfox\_163240379968878578']>a")).click();** щёлкаем и переходим по ссылке в новое окно браузера

**driver.switchTo().window(windowHandle);** <--позволяет вернутся/переключится в предыдущее окно т.е.<http://www.av.by>

**driver.switchTo().alert().accept(); ←** will close the alert using ok button

**driver.switchTo().frame(webElement); ←** переключить фокус драйвера между фреймами

-----------------------------------------------------------------------------------------------------------------------

12)**driver.findElement(By.linkText(“Google”)).click()**;<--полное название ссылки

**driver.findElement(By.partialLinkText(“Goo”)).click();**<--частичное название

The command finds the element using link text and then click on that element and thus the

user would be redirected to the corresponding page.

-----------------------------------------------------------------------------------------------------------------------

Тег <select> отвечает за выподающее окно с вариантами на выбор.

13**)Select selectByValue = new Select(driver.findElement(By.xpath("//select[@name='country']")));**

selectByValue.selectByValue("BERMUDA");

selectByValue.deselectByVisibleText("BERMUDA");

Класс Select работает ТОЛЬКО с тегом <select> из DOM, и его экземпляр/обьект позволяет работать

с выпадающими <option> в данном <select> через методы(.selectByValye("...");

.selectByVisibleText("...");

.selectByIndex(2);<-х.з. чё за индекс такой

14)Работа с iframe. iframe-это вебстраница(типа рекламы) встроенная в текущую вебстраницу, т.е. HTML в HTML.

это выглядит

<html>

<body>

<iframe>

>#dokyment

<!DOCTIPE html>

<html>...</html>

</iframe>

<iframe>...</iframe>

<iframe>...</iframe>

</body>

</html>

чтобы обратится к элементу в каком-либо окне нужно сначала переключится на это окно (окно здесь-элемент типа как реклама)

driver.switchTo().frame(id/name), переключатся/switch можно с помощью атрибутов: id,name или index(порядковый номер iframe)

driver.switchTo().frame(“ID of the frame or name“);

Integer size = driver.findElements(By.tagName("iframe")).size();<-пощитать все <iframe> на странице

driver.switchTo().frame(0); переключится на iframe(0)-Index of the iframe starts with ‘0’.

driver.switchTo().frame(“frameName”); переключится на iframe с атрибутом name='frameName';

driver.switchTo().frame(“id of the element”);

driver.switchTo().defaultContent() driver вернётся в стартовое окно.

если в iframe нет id/name, можно использовать xpath

пример: WebElement webElement=driver.findElement(By.xpath("//iframe[@title='frame #3']"));

driver.switchTo().frame(webElement);

----------------------------------------------------------------------------------------------------------------

15) **driver.close();** закрывает текущее окно(вкладку)если при этом есть ещо открытые окна то при закрытии текущего,

фокус перемещается на другое открытое

----------------------------------------------------------------------------------------------------------------

16) **driver.quit();** завершает работу браузера при этом закрывает все вкладки.

----------------------------------------------------------------------------------------------------------------

17) В рамках использования webdrivera поддерживается работа с исключениями

WebElement saveButton = driver.findElement(By.id("Save"));

try{

if(saveButton.isDisplayed()){

saveButton.click();

}

}

catch(NoSuchElementException e){

e.printStackTrace();

}

18) boolean textBox = driver.findElement(By.xpath("//input[@name='textbox1']")).isEnabled();

Check Whether the Element is Enabled/доступен Or Disabled in the Selenium WebDriver.

---------------------------------------------------------------------------------------------------------------

19) driver.findElement(By.xpath("//input[@name='comments']")).submit();

Submit() is a better alternative to click() if the element to be clicked is a submit button.

---------------------------------------------------------------------------------------------------------------

20)Boolean checkIfElementPresent= driver.findElements(By.xpath("//input[@id='checkbox2']")).size()!= 0;

findElements(By, by) can be used to verify if an element is actually present in the webpage.

---------------------------------------------------------------------------------------------------------------

21)driver.manage().timeouts().implicitlyWait(1000, TimeUnit.SECONDS);

Implicitly/Неявно wait for 1000 seconds before executing the next line in the code.

...)webDriver.manage().deleteAllCookies(); delete all the cookies for the current domain

webDriver.manage().deleteCookieNamed("JSESSIONID"); удаляет выбранные coockies

...)webDriver.manage().window().fullscreen();<--открывает окно и разворачивает максимально для

просмотра плеер и автомотически включает.

...)webDriver.manage().window();<--открывает окно и разворачивает максимально

driver.manage().window().maximize(); <- - открывает сам браузер максимально (разворачивает).

22)WebDriverWait wait = new WebDriverWait(driver, 10); -wait until the element appears for 10sec.

WebElement element = wait.until(ExpectedConditions.visibilityOfElementLocated (By.xpath("//input[@id=’name’]")));

we can write a comment to wait until the element appears on the webpage.

---------------------------------------------------------------------------------------------------------------

23) WebDriverWait wait = new WebDriverWait(driver, 10);

WebElement element = wait.until(ExpectedConditions.alertIsPresent());

In some scenarios, we have to wait for alerts to continue the test. In this case, we use a command using until() method

from the WebdriverWait class and alertIsPresent() method from the ExpectedConditions class.

---------------------------------------------------------------------------------------------------------------

24) WebElement sourceLocator = driver.findElement(By.xpath("//\*[@id='image1']/a"));

WebElement destinationLocator = driver.findElement(By.xpath("//\*[@id='stage']/li"));

Actions actions=new Actions(driver);

actions.dragAndDrop(sourceLocator, destinationLocator).build().perform();

In the dragAndDrop method, we pass the two parameters, Source locator- the element we want to drag and Destination

locator- the element to which we want to drop.

---------------------------------------------------------------------------------------------------------------

25) alert - модальное окно. При всплывании модального окна оно не отоброжается в DOM, соответственно мы неможем

на него перейти с помощью driver.switch(), так как незашто зацепится и соответственно не можем ни элемент найти

ни кликнуть.

WebDriverWait wait = new WebDriverWait(driver, 5);

wait.until(ExpectedConditions.alertIsPresent()); <-ждать до появления alert (но небольше 5 сек.)

Alert alert = driver.switchTo().alert(); фокусируемся на alert

Assert.assertTrue(alert.getText().contains("ВЫ ДЕЙСТВИТЕЛЬНО ХОТИТЕ ПЕРЕЙТИ ПО ССЫЛКЕ?")); сравнение этой строки и

строки написанной на самой alert

alert.dismiss();/alert.acsept(); <-принять или отклонить модальное окно

\*\*\*\*\*во какой код нашол, намного понятней\*\*\*\*\*\*

driver.switchTo().alert().accept(); <--походу switch всё-таки работает))) только надо указывать что на alert

driver.switchTo().alert().getText();-To capture the alert message.

driver.switchTo().alert().sendKeys("Text");- send some data to alert box.

---------------------------------------------------------------------------------------------------------------

26) DriverManager.getConnection(URL, "username", "password" )Connect to the Database through URL and credentials.

---------------------------------------------------------------------------------------------------------------

27) How to handle <tables>...</tables> with Webdriver

Пример код HTML:

<table> <--контейнер для всей таблицы

<tbody> <--контейнер тела таблицы(можно и без него)

<tr> <--тег обозначающий ряд таблицы

<th> название первого столбца </th><--количество тегов <th> определяет количество столбцов

<th> название второго столбца </th><--в th прописывается как бы названия столбцов

<th> название третьего столбца </th><--сколько в одном tr, th, столько и колонок в таблице

</tr>

<tr>

<td>значение первого столбца </td> <--в этих тегах пишут значения ячейки

<td>значение второго столбца </td>

<td>значение третьего столбца </td>

</tr>

</tbody>

</table>

If a number of rows and columns are always constant, let’s say our HTML table will always have 5 rows and 3 columns.

approach will NOT BE a perfect solution for dynamically changing web tables.

for(int numberOfRows=1; numberOfRows<=5; numberOfRows++){

for(int numberOfRows=2; numberOfRows<=6; numberOfRows++){

for(int numberOfCol=1; numberOfCol <=3; numberOfCol++) {

WebElement webElement=driver.findElement(By.xpath("//table/tbody/tr["+numberOfRows+"]/td["+numberOfCol+"]"));

System.out.print(webElement.getText()+" "); выводим на экран содержимое каждой ячейки

}

System.out.println();

}

}

\*\*\*\*\*\*\*for dynamically changing web tables\*\*\*\*\*\*\*\*

WebElement htmltable=driver.findElement(By.xpath("//table/tbody"));

List<WebElement> rows=htmltable.findElements(By.tagName("tr"));//<--naxodim vse riadi (raws)

for(int rnum=1;rnum<rows.size();rnum++) { //<--rnum eto iterator, rows.size eto kolichestvo riadov

//tak-kak v tablice 5 riadov s <td> i odin s <th> to nado opredelitsia chto nam nado: <td> ili <th> ? Dopystim td.

List<WebElement> columns=rows.get(rnum).findElements(By.tagName("td"));//vozvraschaetsia chislo stolbcov v odnom riady

System.out.println("Number of columns in "+rnum+"columns.size()");

for(int cnum=0;cnum<columns.size();cnum++)

{

System.out.println(columns.get(cnum).getText());

}

}

Klychevim momentom iavliaetsia to chto

Columns sozdaetsia kazzdi raz novii s novim (ynikalnim kolichestvom) chislom konteinerov <td>...</td> i dalee v cikle

mi perebiraem cazzdi <td> po ocheredi, chitaia text tam napisanni(v iacheike).Poetomy programma ne ypadet esli kolichest-

vo stolbcov bydet meniatsia

------------------------------------------------------------------------------------------------------------------------

28)In few scenarios, element attributes change dynamically. It can be ‘id’, ’name’ etc.

Example: let’s say ‘id’ of a username field is ‘username\_123’ and the XPath will be

//\*[@id=’username\_123′] but when you open the page again the ‘id’ of ‘username’ field might have changed

and the new value may be ‘username\_234’. In this case, the test will fail because the selenium could not find the XPath

you have passed earlier as the id of the field has changed to some other value.

There are many approaches depending upon the type of problem:

#1- Absolute Path method

This is the easiest way to solve the issue. Absolute XPath is the path starting from the root. It will be something

like :/html[1]/body[1]/nav[1]/div[1]/form[1]/div[1]/div[1]/div[1]/artdeco-typeahead-deprecated[1]/artdeco-typeahead-

deprecated-input[1]/input[1]

But the risk with this method is, if something changes in the structure of your web page, your code will break.

So this is not a recommended method.

#2- Relative XPath using [contains@...] or [starts with@...] text.

Example: As in the above example, id value changes but few fields remains constant.

‘username\_123’ changed to ‘username\_234’ but ‘username’ always remained constant.т.е. часть значения осталось прежним

а именно - "username\_"

You can construct xpath as below:

driver.findElement(By.xpath(“//\*[contains(@id,’username’)]”)).sendKeys(“username”);

driver.findElement(By.xpath(“//\*[starts-with(@id,’user’)]”)).sendKeys(“username”);

‘contains’ is a java method which checks if id contains the substring 'username'.

starts-with() checks if any attribute starts with “user”.

#3- Use Multiple attributes to locate an element.

Xpath- //button[starts-with(@id, 'save') and contains(@class,'publish')]