# Урок 03. Веб драйвер

## Цели

научиться конфигурировать драйвер при старте,

настраивать его под специфичные задачи;

научиться задавать настройки, специфичные для конкретного браузера.

## План

WebDriver

- Конфигурирование драйвера

--Capabilities

DesiredCapabilities caps = new DesiredCapabilities()

setCapability()

Shared capabilities

Driver specific capabilities

--Options

ChromeOptions chromeOptions = new ChromeOptions()

FirefoxOptions firefoxOptions = new FirefoxOptions()

addArguments()

Browser Arguments:

–headless To open browser in headless mode. Works in both Chrome and Firefox browser

–start-maximized To start browser maximized to screen. Requires only for Chrome browser. Firefox by default starts maximized

–incognito To open private chrome browser

–disable-notifications To disable notifications, works Only in Chrome browser

setPageLoadStrategy()

PageLoadStrategy.NORMAL

PageLoadStrategy.EAGER

PageLoadStrategy.NONE

- Запуск и остановка

driver.get("https://devqa.io");

driver.getCurrentUrl();

driver.getTitle();

driver.getPageSource();

driver.close();

driver.quit();

- Запуск разных браузеров

WebDriver driver = new ChromeDriver();

WebDriver driver = new FirefoxDriver();

WebDriver driver = new EdgeDriver();

WebDriver driver = new InternetExplorerDriver();

## Результаты

Проект готовый к запуску тестов на разных браузерах и с разными настройками.

# Урок 04. Локаторы

## Цели

проанализировать все типы локаторов;

научиться строить сложные локаторы;

проанализировать какие возникают ошибки при работе с элементами;

рассмотреть подходы к хранению локаторов.

## План

- Команды поиска

driver.findElement()

driver.findElements()

Find Element From Element

Find Elements From Element

- Стратегии поиска

By.id

By.className

By.name

By.tagName

By.linkText

By.partialLinkText

By.xpath

By.cssSelector

- Виды локаторов

CSS,

XPATH,

JQUERY

сравнение CSS и xPath;

- Правила

валидация локаторов в браузере;

подходы к хранению локаторов в проекте.

## Результаты

Оптимальные локаторы элементов на странице.

# Урок 05. Веб элементы

## Цели

проанализировать какие свойства элементов страницы доступны для чтения средствами автоматизации и какие проверки можно на них строить;

объяснить какие элементы Selenium считает interactable и clickable.

## План

Webelement

click()

sendKeys()

clear()

submit()

isDisplayed()

isEnabled()

isSelected()

getText()

getTagName()

getCssValue()

getSize()

getRect()

getLocation()

getAttribute(“”)

driver.switchTo().activeElement()

- Текстбоксы

- Кнопки

- Чекбоксы

- Радиобаттоны

- Выпадающие списки

Select select = new Select(<WebElement object>);

selectByVisibleText()

select.selectByIndex()

selectByValue()

deselectByVisibleText()

deselectByIndex()

deselectByValue()

deselectAll()

getOptions()

getAllSelectedOptions()

getFirstSelectedOption()

- Таблицы

## Результаты

Умение работать с элементами

# Урок 06. Ожидания

## Цели

применять явные и неявные ожидания для повышения стабильности тестов и осуществления сложных проверок.

## План

- Implicit Waits Неявные ожидания

driver.manage().timeouts().implicitlyWait()

- Explicit Waits Явные ожидания

WebDriverWait wait = new WebDriverWait()

wait.until()

- Expected Conditions

ExpectedConditions:

alertIsPresent()

elementSelectionStateToBe()

elementToBeClickable()

elementToBeSelected()

frameToBeAvaliableAndSwitchToIt()

invisibilityOfTheElementLocated()

invisibilityOfElementWithText()

presenceOfAllElementsLocatedBy()

presenceOfElementLocated()

textToBePresentInElement()

textToBePresentInElementLocated()

textToBePresentInElementValue()

titleIs()

titleContains()

visibilityOf()

visibilityOfAllElements()

visibilityOfAllElementsLocatedBy()

visibilityOfElementLocated()

- Fluent Waits

Wait wait = new FluentWait()

- Thread.sleep Задержка

## Результаты

Умение работать с ожиданиями

# Урок 07. Окна

## Цели

поработать с нативными элементами браузера.

## План

-Переходы

driver.navigate().to("https://devqa.io");

driver.navigate().back();

driver.navigate().forward();

driver.navigate().refresh();

- Размеры окна

driver.manage().window().getSize().getWidth()

driver.manage().window().getSize().getHeight()

driver.manage().window().setSize()

driver.manage().window().getPosition().getX()

driver.manage().window().getPosition().getY()

driver.manage().window().setPosition()

driver.manage().window().maximize()

driver.manage().window().fullscreen()

- Alert

- Prompt,

- Confirm

driver.switchTO().alert.getText()

driver.switchTO().alert.accept()

driver.switchTO().alert.dismiss()

driver.switchTO().alert.sendKeys(“Text”)

- iFrame

driver.switchTo.frame()

driver.switchTo().defaultContent();

- Tabs

getWindowHandle()

getWindowHandles()

driver.switchTo().window()

// Opens a new tab and switches to new tab

driver.switchTo().newWindow(WindowType.TAB);

// Opens a new window and switches to new window

driver.switchTo().newWindow(WindowType.WINDOW);

- BasicAuth

Переключение

Размеры окна

## Результаты

Умение работать с окнами

# Урок 08. Куки, логи, скриншоты

## Цели

## План

- Куки

addCookie(arg)

getCookies()

getCookieNamed()

deleteCookieNamed()

deleteCookie()

deleteAllCookies()

- Логирование

driver().manage().logs().getAvailableLogTypes()

-- Типы логов

browser Логи от javascript консоли браузера

client Логи от клиентской части протокола Webdriver (например, Java bindings)

driver Внутренние логи драйвера (например, логи FirefoxDriver)

performance Логи относящиеся к производительности на странице (тайминги загрузки ресурсов)

server Логи от Selenium сервера

-- Уровни логгирования

OFF: Логгирование отключено

SEVERE: Сообщения об ошибках. К примеру, при неизвестной команде.

WARNING: Предупреждения о том, что могло быть неверным, хоть ситуация и было успешно обработано. Например, перехваченное исключение.

INFO: Сообщения информативного характера. Например, о полученных командах.

DEBUG: Сообщения для дебаггинга. Например, информация о состоянии драйвера.

ALL: Все сообщения. Это способ получить все сообщения независимо от его уровня.

-- Получение логов

Logs logs = driver.manage().logs();

LogEntries logEntries = logs.get(LogType.DRIVER);

for (LogEntry logEntry : logEntries) {

System.out.println(logEntry.getMessage());

}

-- Конфигурация логгирования

- Скриншоты

TakesScreenshot screenShot =(TakesScreenshot)driver;

FileHandler.copy(screenShot.getScreenshotAs(OutputType.FILE), new File("path/to/destination/folder/screenshot.png"));

- Chrome Devtools

## Результаты

Умение работать с куками, логами браузера, скриншоты

# Урок 09. Нестандартные задачи

## Цели

объяснить, как решать нестандартные задачи.

## План

- Цвета

private final Color HEX\_COLOUR = Color.fromString("#2F7ED8");

private final Color RGB\_COLOUR = Color.fromString("rgb(255, 255, 255)");

private final Color RGB\_COLOUR = Color.fromString("rgb(40%, 20%, 40%)");

private final Color RGBA\_COLOUR = Color.fromString("rgba(255, 255, 255, 0.5)");

private final Color RGBA\_COLOUR = Color.fromString("rgba(40%, 20%, 40%, 0.5)");

private final Color HSL\_COLOUR = Color.fromString("hsl(100, 0%, 50%)");

private final Color HSLA\_COLOUR = Color.fromString("hsla(100, 0%, 50%, 0.5)");

private final Color TRANSPARENT = Color.fromString("transparent");

- Действия мыши и клавы

Actions

keyboard events

keyDown()

keyUp()

sendKeys()

mouse events

clickAndHold()

contextClick()

doubleClick()

dragAndDrop(source,target)

dragAndDropBy(source,xOffset,yOffset)

moveByOffset(xOffset,yOffset)

moveByElement()

release()

build()

perform()

Канвас элемент

Перетягивание объекта

Скроллинг

- Выполнение JS

executeAsyncScript()

executeScript()

Скроллинг

Загрузка файлов

Видеоплеер

Веб сторадж

## Результаты

Умение решать нестандартные задачи