



GUI Testing with Selenium WebDriver



Table of Content

1

Selenium IDE for Firefox

2

WebDriver in Java project

3

Browser Development console

4






Finding elements

5

Test framework using Page Object Pattern

Selenium IDE for Firefox (add-on)

The screenshot shows the Firefox Add-ons Manager interface. The browser's address bar displays 'about:addons'. The search bar at the top right contains the text 'selenium'. On the left sidebar, there are links for 'Search', 'Get Add-ons', 'Extensions', 'Themes', and 'Plugins'. The main content area displays a list of search results for Selenium IDE-related add-ons. The results are sorted by 'Best match'. The first result is 'Selenium IDE', which is highlighted. Below it are 'Kantu: Fresh Browser Automation plus Selenium IDE', 'Page Modeller', 'Katalon Recorder', and 'Wildfire'. Each result includes an icon, the add-on name, a brief description, the date it was last updated, and an 'Install' button. At the bottom of the list, there is a link to 'See all 333 results'.

| Name | Last Updated | Best match |
|---|------------------|-------------------------|
|  Selenium IDE Selenium IDE is an integrated development environment for Selenium tests. It is implemented as a Firefox extension, an... More | February 5, 2019 | Install |
|  Kantu: Fresh Browser Automation plus Selenium IDE Modern Open-Source Web Browser Macro Recorder and Selenium IDE. Use it for general web automation, web testing,... More | January 28, 2019 | Install |
|  Page Modeller Page Modeller enables developers to scan a web page and generate page object style code for various tools, languag... More | February 3, 2019 | Install |
|  Katalon Recorder Best automation recorder. Exports Selenium WebDriver code. Provides reports, logs, screenshots. Fast & extensible. More | March 14, 2018 | Install |
|  Wildfire Record browser actions then replay immediately. Craft your own custom automation workflows. More | May 30, 2017 | Install |

[See all 333 results](#)

Create first test project with random exploratory tests, playback

moz-extension://8c3f8c07-df93-4899-a25e-1efeefcfd9a3 - Selenium IDE - SeleniumIDETestProject* - Mozilla Firefox

Project: SeleniumIDETestProject*

Tests +

Search tests...

https://www.seleniumhq.org

| | Command | Target | Value |
|---|-----------------|---------------------------------|---------------|
| 1 | open | / | |
| 2 | set window size | 862x650 | |
| 3 | click | id=q | |
| 4 | type | id=q | webdriver |
| 5 | send keys | id=q | \${KEY_ENTER} |
| 6 | click | css=asc-results>asc-webResult a | |

Command //

Target

Value

Description

Log Reference

Tel Ran project



Name your new project

Please provide a name for your new project.

PROJECT NAME

You can change the name of your project at any time by clicking it and entering a new name.

OK

CANCEL

Set your project's base URL

Before you can start recording, you must specify a valid base URL for your project. Your tests will start by navigating to this URL.

BASE URL

START RECORDING

CANCEL

Task

- Create tests for several site's basic usage scenarios
- Run, observe results
- Describe bugs if any

Note: Please put comment “Ignore, for testing purposes only”

The screenshot shows the Selenium IDE interface within a Mozilla Firefox browser window. The title bar indicates the project is 'TelRan.de TestProject*'. The main interface is divided into several sections:

- Tests:** A sidebar on the left with a search bar and a list of test steps: 'ApplyForCourse' (highlighted), 'AskQuestion', 'Localization', and 'TelegramMessage'.
- Command Table:** A table with columns 'Command', 'Target', and 'Value'. It lists six steps:

| | Command | Target | Value |
|---|-----------------|-----------------------|-------|
| 2 | set window size | 1382x744 | |
| 3 | click | id=comp-jc94om9wlabel | |
| 4 | click | id=field1 | |
| 5 | click | id=field1 | |
| 6 | click | id=field1 | |
- Command Editor:** A section below the table with input fields for 'Command' (set to 'type'), 'Target' (set to 'id=comp-jo76d3p9fieldMessage'), 'Value' (set to 'Ignore, for test purposes'), and 'Description'.
- Log:** A bottom section with a 'Log' tab showing a list of executed steps and their results, such as '16. type on id=field3 OK' and '21. click on id=comp-jo76d3p9submit OK'. The final status is '**'ApplyForCourse' completed successfully**'.

1. Bug 1 Localization for main menu items fails

Steps:

1. Go to main page <https://www.tel-ran.de/>
2. Select De or En for language

Actual Result:

Some menu items are still in Russian, if click on them - content is displayed in Russian

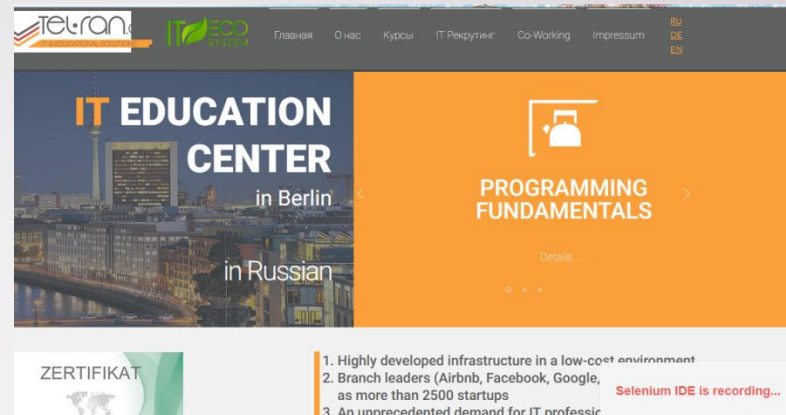
Expected Result:

All menu items and content according to selected language

Note: Reproducible for Chrome, FireFox

Priority: High (non professional site image)

Severity: Med (workaround exists)



2. Bug 2 Send Telegram Message does not work

Steps:

1. Go to main page <https://www.tel-ran.de/>
2. Press Message Using Telegram button
3. Press Send Message

Actual Result:

FireFox: Browser Error is displayed, looks like functionality is not implemented

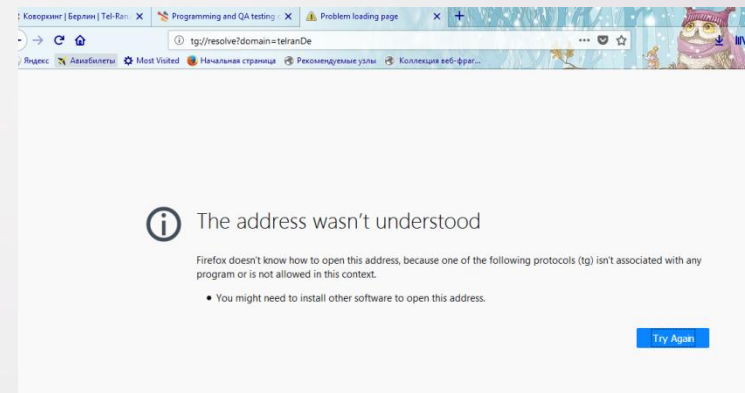
Chrome: Nothing happens

Expected Result:

Functionality is implemented or user-friendly temporary message is displayed

Priority: Normal

Severity: Med (workaround exists for sending message)



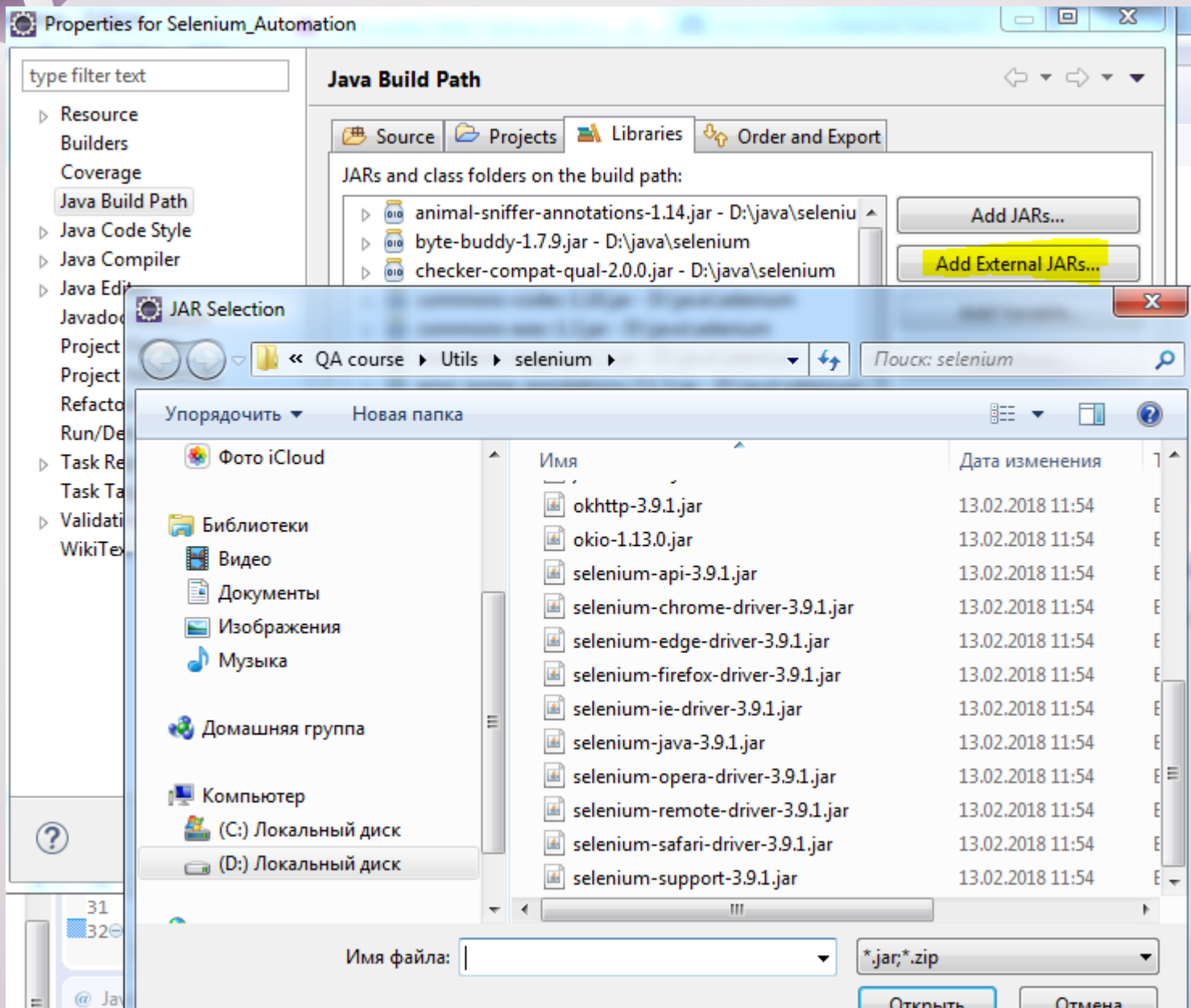
Selenium & WebDriver



Initially Selenium IDE and WebDriver were 2 products from different companies. Later on Selenium integrated WebDriver (Selenium WebDriver).

The idea - to take control over web browser elements directly from test script (Java, C#, Python etc.) via Selenium WebDriver APIs.

Configure Selenium java project



```
public class FirstSeleniumClass {

    WebDriver driver;
    String url1 = "https://www.wix.com/";
    String url2 = "https://www.amazon.in";
    String url3 = "https://www.facebook.com";
    String url4 = "https://www.flipkart.com";

    public void invokeBrowser() {

        try {
            System.setProperty("webdriver.chrome.driver", "D:\\java\\selenium\\drivers\\chromedriver.exe");
            driver = new ChromeDriver();
            // System.setProperty("webdriver.gecko.driver", "D:\\java\\selenium\\drivers\\geckodriver.exe");
            // driver = new FirefoxDriver();
            driver.manage().window().maximize();
            driver.manage().timeouts().implicitlyWait(60, TimeUnit.SECONDS);
            driver.get(url1);
            // WebElement temp = driver.switchTo().activeElement();

        } catch (Exception e) {
            e.printStackTrace();
        }

    }

    public void closeBrowser() {
        driver.close();
    }
}
```

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    FirstSeleniumClass test = new FirstSeleniumClass();
    test.invokeBrowser();
    test.closeBrowser();

}
```

Use browser development console

The screenshot shows the Amazon India homepage. The browser's address bar displays 'https://www.amazon.in'. The page features a navigation bar with links like 'Your Amazon.in', 'Today's Deals', 'Amazon Pay', 'Hello, Sign in Your Orders', 'Try Prime', 'Your Lists', and a 'Cart' icon. The main banner shows a couple with a gift box. Below the banner are sections for 'Meet the Echo family' and 'New On Prime Video'. The Chrome DevTools console is open on the right, showing the 'Elements' tab. The DOM tree highlights the <title> element, which contains the text: 'Online Shopping site in India: Shop Online for Mobiles, Books, Watches, Shoes and More - Amazon.in'.

```
assert("Online Shopping site in India: Shop Online for Mobiles, Books, Watches, Shoes and More - Amazon.in"
      .equals(driver.getTitle()));
```

Compatibility

Java

C#

Ruby

Python

WebDriver API

Firefox

Chrome

IE

HtmlUnit

Finding elements

ClassName

ID

Name

TagName

CssSelector

LinkText

PartialLinkText

XPath

```
▼<div class="nav-search-field ">
  <!-- DO NOT REMOVE: the text in the label are for screen reader, and it is
  not visible in the web page -->
  <label id="nav-search-label" for="twotabsearchtextbox" class="aok-offscreen
    Search
  </label>
  <input type="text" id="twotabsearchtextbox" value name="field-keywords"
    autocomplete="off" placeholder class="nav-input" dir="auto" tabindex="9">
</div>
<div id="nav-iss-attach"></div>
</div>
</form>
```

FindElement examples

```
public void findGoodsAmazon(String product) {  
    WebElement searchString = driver.findElement(By.id("twotabsearchtextbox"));  
    searchString.sendKeys(product);  
    searchString.submit();  
    // WebElement searchBtn = driver.findElement(By.cssSelector("#nav-search > form > div.nav-right > div > input"));  
    // searchBtn.click();  
}  
  
public void closeBrowser() {  
    driver.close();  
}  
  
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    FirstSeleniumClass test = new FirstSeleniumClass();  
    test.invokeBrowser();  
    test.findGoods("phones");  
  
    test.closeBrowser();  
}
```


WebElement interface methods

- Not all methods are applicable to particular WebElement objects, like ***clear or sendKeys*** is not applicable for button

| | |
|---------------------------|---|
| <code>clear</code> | |
| <code>click</code> | <code>org.openqa.selenium</code> |
| <code>findElement</code> | Interface WebElement |
| <code>findElements</code> | |
| <code>getAttribute</code> | All Superinterfaces: |
| <code>getCssValue</code> | <code>SearchContext, TakesScreenshot</code> |
| <code>sendKeys</code> | All Known Implementing Classes: |
| <code>submit</code> | <code>RemoteWebElement</code> |

WebElement interface properties

org.openqa.selenium

Interface WebElement

All Superinterfaces:

`SearchContext`, `TakesScreenshot`

All Known Implementing Classes:

`RemoteWebElement`

`isDisplayed`

`isEnabled`

`getLocation`

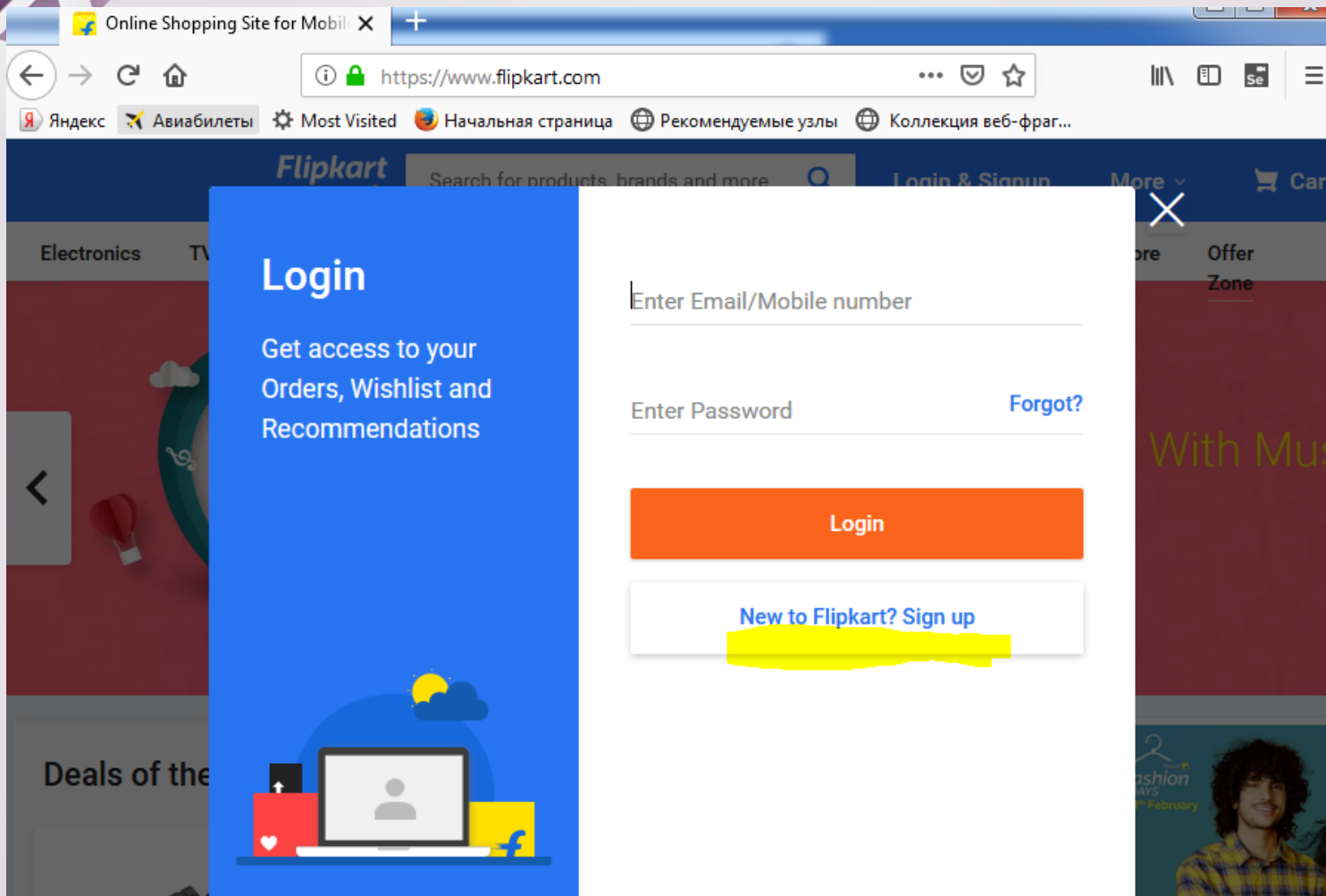
`isSelected`

`getSize`

`getTagName`

`getText`

Modal windows



```
public void signUpToFlipCart(){  
    WebElement temp = driver.switchTo().activeElement();  
    driver.findElement(By.xpath("/html/body/div[2]/div/div/div/div/div[2]/div/form/a/span")).click();  
}
```

Task “Find Jonny Depp”



```
public void findJonnyDepp() throws InterruptedException {
    WebElement searchBox = driver.findElement(By.name("q"));
    searchBox.sendKeys("Jonny Depp movies");
    searchBox.submit();
    Thread.sleep(1000);
    WebElement container = driver.findElement(By.id("hdtb-msb-vis"));
    ArrayList<WebElement> list = (ArrayList<WebElement>) container.findElements(By.tagName("a"));
    for (WebElement el : list) {
        if (el.getText().equals("Bilder")) {
            el.click();
            System.out.println("I am here");
            break;
        }
    }

    // driver.findElements(By.cssSelector("a[class = q gs]")).get(1).click();

    driver.findElement(By.id("5qa4q5HLc3e9sM:")).click();
}
```