

# **Selenium Commands**

By Swapnil Koshti

# POM – Page Object Model

- POM – Design pattern
- One Page = One Java Class
- Centralized Object Repository
- Page Object methods = user actions
- Business-Level Methods
- Improved Maintainability
- Reduces Code Duplication

# POM Annotations

- `@FindBy(id = "username")`
- `@FindBys({  
    @FindBy(tagName = "input"),  
    @FindBy(className = "form-control")  
})`

AND Logic

- `@FindAll({  
    @FindBy(id = "submitBtn"),  
    @FindBy(css = ".btn-submit"),  
    @FindBy(xpath = "//button[text()='Submit']")  
})`

OR Logic

- `@CacheLookup`

# POM Page Factory

- PageFactory.initElements(WebDriver, PageClass Object) - void
  - PageFactory.initElements(WebDriver, PageClass Type) - Class Type Object
  - PageFactory.initElements(SearchContext, PageClass Object) - void
- 
- WebDriver and WebElement both implement SearchContext

# Exceptions

- NoSuchElementException
- NoSuchWindowException
- NoSuchFrameException
- NoSuchCookieException
- NoAlertPresentException
- NoSuchSessionException
- StaleElementReferenceException
- ElementNotInteractableException
- ElementClickInterceptedException
- ElementNotSelectableException

# Headless Browser

A headless browser is a normal browser (Chrome, Firefox, etc.) that runs without opening any visible window.

- Same rendering engine as full browser
- Executes JavaScript, CSS, network calls the same way
- Just no GUI, so you don't see it on screen
- Faster (no time spent drawing UI)
- Good for CI/CD servers (Jenkins, GitLab, etc.) without display
- Multiple tests can run in parallel on same machine
- Less resource usage (CPU/RAM)

# Browser Configuration

- ChromeOptions
- FirefoxOptions
- EdgeOptions

# Common Options

- `--headless=new` – start chrome in headless mode
- `download.default_directory` – Set Download directory
- `download.prompt_for_download` – disable download prompt
- `--allow-insecure-localhost` – allow to open insecure websites
- `--ignore-certificate-errors` – Skip certificate check