Synchronization/wait: matching selenium test script speed with browser/application speed

**Wait-**

1.Static wait

2. Dynamic wait

**Static wait/java wait**🡪 Wait time is fixed, irrespective of condition. Script will be hold for mentioned seconds.

Eg: Thred.sleep(200);//200ms 10ms🡪190ms

**Dynamic wait/ Selenium🡪**

**1. Implicit wait:**

Applicable: complete webpage

1 parameter: time value (seconds)

//2000ms🡪 100ms🡪1900 release wait time

================================================================

**package** seleniumWait;

**import** java.time.Duration;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**public** **class** ImplicitWait {

**public** **static** **void** main(String[] args) {

WebDriver driver = **new** ChromeDriver();

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

driver.get("https://www.discoveryplus.in/home");

driver.manage().timeouts().implicitlyWait(Duration.*ofMillis*(6000));

// 6000--> page load 2000, 4000? save-->next execution

driver.findElement(By.*xpath*("(//h6[text()='Sign In'])[2]")).click();

}

}

//you are waiting for 100sec🡪 page loaded in 20 sec🡪 once page loaded🡪 release timeout(80sec)

==============================================================

2. Explicit wait:

Applicable: single element

1 parameter: time value(seconds)

2 parameter: Condition (isselected, isdisplayed, isenabled)

**package** seleniumWait;

**import** java.time.Duration;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.support.ui.ExpectedConditions;

**import** org.openqa.selenium.support.ui.WebDriverWait;

**public** **class** ExplicitWaitEg {

**public** **static** **void** main(String[] args) {

WebDriver driver = **new** ChromeDriver();

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

driver.get("https://www.discoveryplus.in/home");

WebDriverWait w = **new** WebDriverWait(driver, Duration.*ofMillis*(15000));

w.until(ExpectedConditions.*elementToBeClickable*(By.*xpath*("(//h6[text()='Sign In'])[2]")));

driver.findElement(By.*xpath*("(//h6[text()='Sign In'])[2]")).click();

}

}

===================================================================

3. Fluent wait:

Applicable: single element

1 parameter: time value(seconds)//1000ms

2 parameter: Condition (isselected, isdisplayed, isenabled)

3 frequency: time(time in sec)- 5 m sec🡪10ms

**package** seleniumWait;

**import** java.time.Duration;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.NoSuchElementException;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.support.ui.ExpectedConditions;

**import** org.openqa.selenium.support.ui.FluentWait;

**public** **class** FluentWaitEg {

**public** **static** **void** main(String[] args) {

WebDriver driver = **new** ChromeDriver();

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

driver.get("https://www.discoveryplus.in/home");

FluentWait<WebDriver> w = **new** FluentWait<WebDriver>(driver).withTimeout(Duration.*ofMillis*(15000))

.pollingEvery(Duration.*ofMillis*(1000)).ignoring(NoSuchElementException.**class**);

w.until(ExpectedConditions.*elementToBeClickable*(By.*xpath*("(//h6[text()='Sign In'])[2]")));

driver.findElement(By.*xpath*("(//h6[text()='Sign In'])[2]")).click();

}

}