

Element to be clickable is overloaded method.
Add in element util.

```
389
390  /**
391  * An expectation for checking an element is visible and enabled such that you can click it.
392  * @param locator
393  * @param timeOut
394  */
395  public void clickWhenReady(By locator, int timeOut) {
396      WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeOut));
397      wait.until(ExpectedConditions.elementToBeClickable(locator)).click();
398 }
```

Wait for multiple elements-
Grab all footer links from the app.
Until now returns list of webelement.



The screenshot shows an IDE interface with two tabs: 'ElementUtil.java' and 'WaitForElements.java'. The 'WaitForElements.java' tab is active, displaying the following code:

```
1 package seleniumsessions;
2
3 import java.time.Duration;
4 import java.util.List;
5
6 import org.openqa.selenium.By;
7 import org.openqa.selenium.WebDriver;
8 import org.openqa.selenium.WebElement;
9 import org.openqa.selenium.chrome.ChromeDriver;
10 import org.openqa.selenium.support.ui.ExpectedConditions;
11 import org.openqa.selenium.support.ui.WebDriverWait;
12
13 public class WaitForElements {
14
15     public static void main(String[] args) {
16         WebDriver driver = new ChromeDriver();
17         driver.get("https://www.freshworks.com/");
18
19         By footerLinks = By.xpath("//footer//a");
20
21         WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
22         List<WebElement> footerList = wait.until(ExpectedConditions.presenceOfAllElementsLocatedBy(footerLinks));
23
24         System.out.println(footerList.size());
25     }
26
27 }
```

The output window below the code editor shows the result of the execution:

```
<terminated> WaitFor
Apr 02, 2024
WARNING: [REDACTED]
43
```

Another method added-
Visibility of all elements.

```
27
28     //An expectation for checking that all elements present on the web page that match the locator are visible.
29     //Visibility means that the elements are not only displayed but also have a height and width that is greater than 0.
30     List<WebElement> footerLinksList = wait.until(ExpectedConditions.visibilityOfAllElementsLocatedBy(footerLinks));
31 }
```

```
32
33     System.out.println(footerList.size());
34     System.out.println(footerLinksList.size());
```

A screenshot of a Java IDE interface. At the top, there are tabs for 'Console' and 'Problems'. Below them, the console output shows the following text:
<terminated> WaitForElements {4
Apr 02, 2025 8:4
WARNING: Unable
43
43

Create utility- Add in element util.

```
363
364  /**
365   * An expectation for checking that there is at least one element present on a web page.
366   * @param locator
367   * @param timeOut
368   * @return
369   */
370  public List<WebElement> waitForAllElementsPresence(By locator, int timeOut) {
371      WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeOut));
372      return wait.until(ExpectedConditions.presenceOfAllElementsLocatedBy(locator));
373  }
374

375  /**
376   * An expectation for checking that all elements present on the web page that match the locator are visible.
377   * Visibility means that the elements are not only displayed but also have a height and width that is greater than 0.
378   * @param locator
379   * @param timeOut
380   * @return
381   */
382  public List<WebElement> waitForAllElementsVisible(By locator, int timeOut) {
383      WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeOut));
384      return wait.until(ExpectedConditions.visibilityOfAllElementsLocatedBy(locator));
385  }
386
387
```

11.00

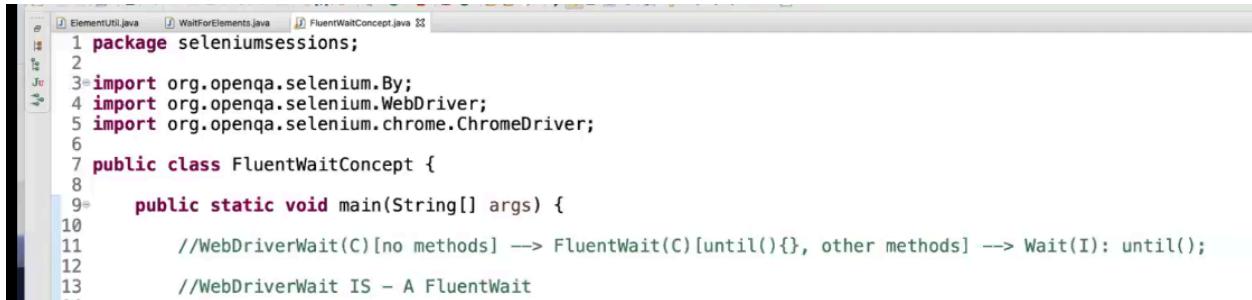
Second one is better because it guarantees the element is available before performing the actions. Performance wise visibility is slow as every element to be checked. Presence is faster because it just

waits for one element to be seen on webpage to work ahead.

Fluent wait-

Go to website.

Enter user name and password.



```
1 package seleniumsessions;
2
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.chrome.ChromeDriver;
6
7 public class FluentWaitConcept {
8
9     public static void main(String[] args) {
10
11         //WebDriverWait<C>[no methods] --> FluentWait<C>[until(){}], other methods] --> Wait<I>: until();
12
13         //WebDriverWait IS - A FluentWait
```

Fluent wait accepts any generics as argument.

```
81     */
82     public FluentWait<T input> {
83         this(input, Clock.systemDefaultZone(), Sleeper.SYSTEM_SLEEPER);
84     }
```

We will give webdriver.

Use top casting -

Interface = new class; line 23.

```
17     WebDriver driver = new ChromeDriver();
18
19     driver.get("https://classic.crmpro.com/");
20
21     By username = By.name("username");
22
23     Wait<WebDriver> wait = new FluentWait<WebDriver>(driver);
24
```

Until in fluent returns web element.

```
29     Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)
30         .withTimeout(Duration.ofSeconds(10))
31         .pollingEvery(Duration.ofSeconds(2))
32         .ignoring(NoSuchElementException.class)
33         .ignoring(StaleElementReferenceException.class)
34         .withMessage("====Element is not found====");
```

```

35
36     WebElement username_ele = wait.until(ExpectedConditions.visibilityOfElementLocated(username));
37     username_ele.sendKeys("testautomation");
38

```

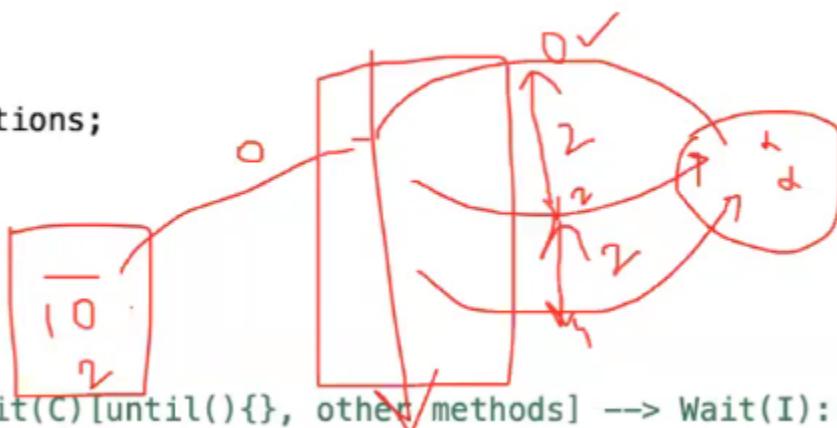


Polling time concept-

Polls every 2 seconds till the 10 second limit is reached. Script to server to browser.

~~EXCEPTION;~~

onditions;
t;



~~ntWait(C)[until(){}], other methods] --> Wait(I): until();~~

Pass wrong user name locator to see polling time in action-

```

26
27     By.username = By.name("username11");
28
29     Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)
30         .withTimeout(Duration.ofSeconds(10))
31         .pollingEvery(Duration.ofSeconds(2))
32         .ignoring(NoSuchElementException.class)
33         .ignoring(StaleElementReferenceException.class)
34         .withMessage("====Element is not found====");
35
36     WebElement username_ele = wait.until(ExpectedConditions.visibilityOfElementLocated(username));
37     username_ele.sendKeys("testautomation");
38

```

```

<terminated> FluentWaitConcept (9) [Java Application] /Users/naveenautomationlabs/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_14.0.2.v20200815-0932/re/bin/java [02-Apr-2025, 9:00:16 am - 9:00:20 am]
Apr 02, 2025 9:00:20 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 134, returning the closest version; found: 133; Please update your browser to the latest version
Exception in thread "main" org.openqa.selenium.TimeoutException: Expected condition failed: ==Element is not found== (tried for at least 1 second)
Build info: version: '4.29.0', revision: '5fc1ec94cb'
System info: os.name: 'Mac OS X', os.arch: 'x86_64', os.version: '10.16', java.version: '14.0.2'
Driver info: driver.version: unknown
    at org.openqa.selenium.support.ui.FluentWait.timeoutException(FluentWait.java:265)

```

Default polling time is 500 ms.

Remove polling time-

```

29     Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)
30         .withTimeout(Duration.ofSeconds(10))
31         .//.pollingEvery(Duration.ofSeconds(2))
32         .ignoring(NoSuchElementException.class)
33         .ignoring(StaleElementReferenceException.class)
34         .withMessage("====Element is not found====");
35
36     WebElement username_ele = wait.until(ExpectedConditions.visibilityOfElementLocated(username));
37     username_ele.sendKeys("testautomation");
38

```

Not entered. You will get in that it waited for 500 ms.
this is to prove that default polling time is 500 ms.



```

<terminated> FluentWaitConcept (9) [Java Application] /Users/naveenautomationlabs/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_14.0.2.v20200815-0932/re/bin/java [02-Apr-2025, 9:02:23 am - 9:02:45 am]
Apr 02, 2025 9:02:27 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 134, returning the closest version; found: 133; Please update your browser to the latest version
Exception in thread "main" org.openqa.selenium.TimeoutException: Expected condition failed: ==Element is not found== (tried for at least 1 second)
Build info: version: '4.29.0', revision: '5fc1ec94cb'
System info: os.name: 'Mac OS X', os.arch: 'x86_64', os.version: '10.16', java.version: '14.0.2'
Driver info: driver.version: unknown
    at org.openqa.selenium.support.ui.FluentWait.timeoutException(FluentWait.java:265)
    at org.openqa.selenium.support.ui.FluentWait.until(FluentWait.java:228)

```

We can ignore as many exceptions as we want.

Create utility-

Same class.

Added 16.

Updated 23.

```

13 import org.openqa.selenium.support.ui.WebDriverWait;
14
15 public class FluentWaitConcept {
16     static WebDriver driver;
17     public static void main(String[] args) {
18
19         //WebDriverWait(C)[no methods] --> FluentWait(C)[unt
20
21         //WebDriverWait IS - A FluentWait
22
23         driver = new ChromeDriver();
24
25         driver.get("https://classic.crmpro.com/");
26
36     public static WebElement waitForElementVisibleWithFluentWait(By locator, int timeOut, int pollingTime) {
37         Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)
38             .withTimeout(Duration.ofSeconds(timeOut))
39             .pollingEvery(Duration.ofSeconds(pollingTime))
40             .ignoring(NoSuchElementException.class)
41             .ignoring(StaleElementReferenceException.class)
42             .withMessage("====Element is not found====");
43
44         return wait.until(ExpectedConditions.visibilityOfElementLocated(locator));
45     }
46 }

```

Call utility from same class-

```

23     driver = new ChromeDriver();
24
25     driver.get("https://classic.crmpro.com/");
26
27     By username = By.name("username11");
28
29
30     waitForElementVisibleWithFluentWait(username, 10, 2).sendKeys("testautomation");
31
32 }
33

```

Updated to correct locator-

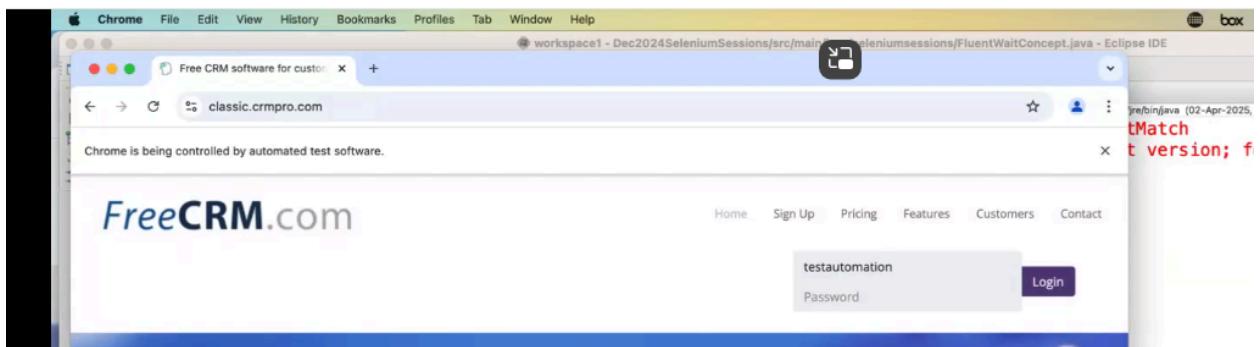


FreeCRM.com

Home Sign Up Pricing Features

testautomation

Password



Utility with presence of element located- Same class.

```

47
48  public static WebElement waitForElementPresenceWithFluentWait(By locator, int timeOut, int pollingTime) {
49      Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)
50          .withTimeout(Duration.ofSeconds(timeOut))
51          .pollingEvery(Duration.ofSeconds(pollingTime))| I
52          .ignoring(NoSuchElementException.class)
53          .ignoring(StaleElementReferenceException.class)
54          .withMessage("==Element is not found===");
55
56      return wait.until(ExpectedConditions.presenceOfElementLocated(locator));
57
58}

```

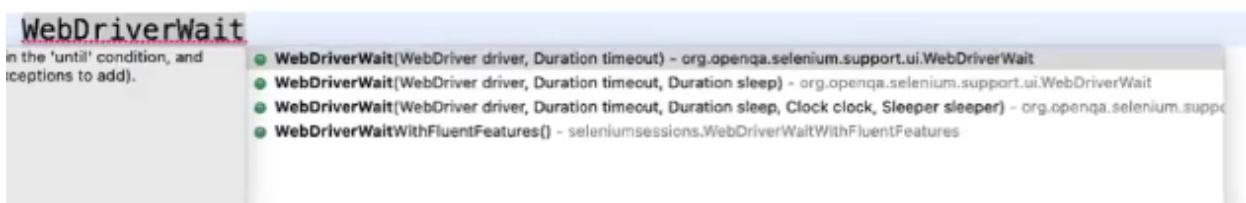
Move to element util-

```

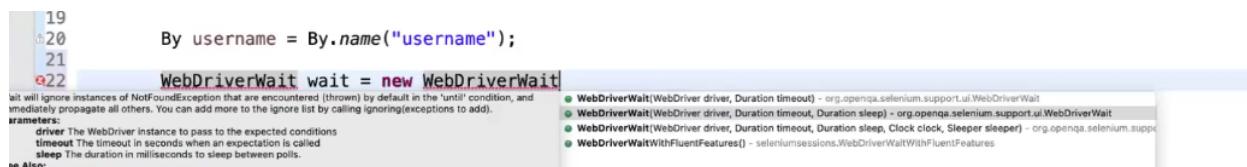
545
546  public WebElement waitForElementVisibleWithFluentWait(By locator, int timeOut, int pollingTime) {
547      Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)
548          .withTimeout(Duration.ofSeconds(timeOut))
549          .pollingEvery(Duration.ofSeconds(pollingTime))
550          .ignoring(NoSuchElementException.class)
551          .ignoring(StaleElementReferenceException.class)
552          .withMessage("==Element is not found===");
553
554      return wait.until(ExpectedConditions.visibilityOfElementLocated(locator));
555
556
557
558  public WebElement waitForElementPresenceWithFluentWait(By locator, int timeOut, int pollingTime) {
559      Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)
560          .withTimeout(Duration.ofSeconds(timeOut))
561          .pollingEvery(Duration.ofSeconds(pollingTime))
562          .ignoring(NoSuchElementException.class)
563          .ignoring(StaleElementReferenceException.class)
564          .withMessage("==Element is not found===");
565
566      return wait.until(ExpectedConditions.presenceOfElementLocated(locator));
567

```

So much constructors for web driver wait-



ocd-



```
19 By username = By.name("username");
20
21
22 WebDriverWait wait = new WebDriverWait
```

wait will ignore instances of NoSuchElementException that are encountered (thrown) by default in the 'until' condition, and immediately propagate all others. You can add more to the ignore list by calling ignoring(exceptions to add).

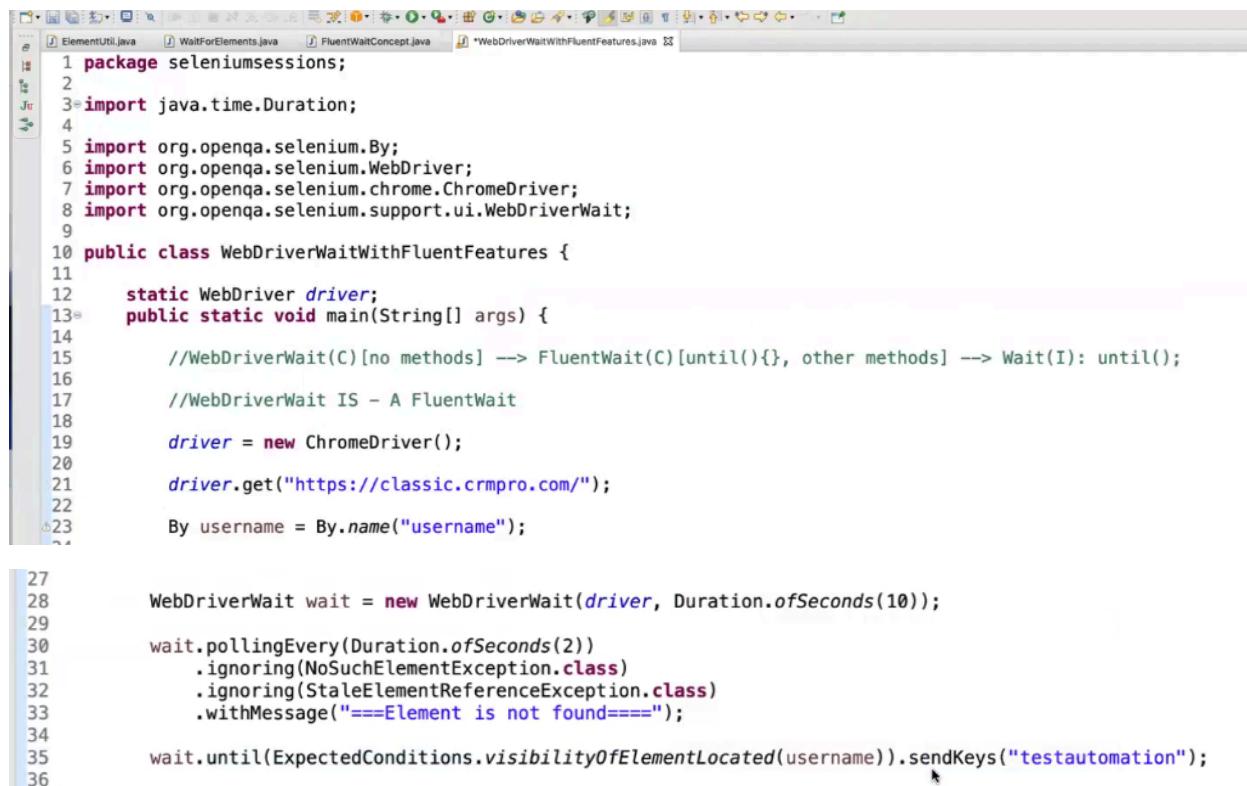
parameters:

- driver The WebDriver instance to pass to the expected conditions
- timeout The timeout in seconds when an expectation is called
- sleep The duration in milliseconds to sleep between polls.

be Also:

- WebDriverWait(WebDriver driver, Duration timeout) - org.openqa.selenium.support.ui.WebDriverWait
- WebDriverWait(WebDriver driver, Duration timeout, Duration sleep) - org.openqa.selenium.support.ui.WebDriverWait
- WebDriverWait(WebDriver driver, Duration timeout, Duration sleep, Clock clock, Sleeper sleeper) - org.openqa.selenium.support.ui.WebDriverWait
- WebDriverWaitWithFluentFeatures() - seleniumsessions.WebDriverWaitWithFluentFeatures

With web driver wait also the fluent wait can be achieved, as web driver wait class is a child of fluent wait class.



```
1 package seleniumsessions;
2
3 import java.time.Duration;
4
5 import org.openqa.selenium.By;
6 import org.openqa.selenium.WebDriver;
7 import org.openqa.selenium.chrome.ChromeDriver;
8 import org.openqa.selenium.support.ui.WebDriverWait;
9
10 public class WebDriverWaitWithFluentFeatures {
11
12     static WebDriver driver;
13     public static void main(String[] args) {
14
15         //WebDriverWait(C)[no methods] --> FluentWait(C)[until(){}, other methods] --> Wait(I): until();
16
17         //WebDriverWait IS - A FluentWait
18
19         driver = new ChromeDriver();
20
21         driver.get("https://classic.crmpro.com/");
22
23         By username = By.name("username");
24
25
26
27         WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
28
29         wait.pollingEvery(Duration.ofSeconds(2))
30             .ignoring(NoSuchElementException.class)
31             .ignoring(StaleElementReferenceException.class)
32             .withMessage("====Element is not found====");
33
34         wait.until(ExpectedConditions.visibilityOfElementLocated(username)).sendKeys("testautomation");
35
36     }
37 }
```

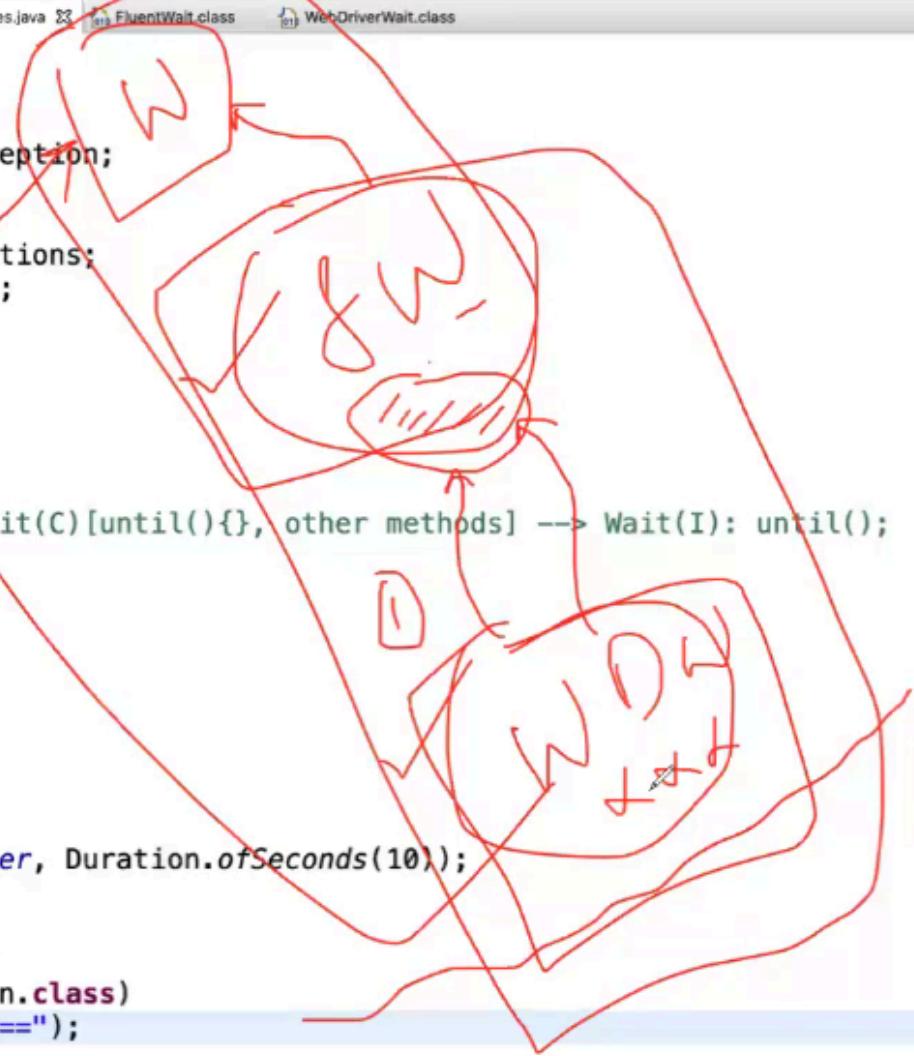


This proves no difference between web driver wait and fluent wait. Web driver wait is child of fluent wait. Both are classes.

Why fluent wait needed when webdriver wait does same thing -

47.00

Java selenium says, use the lowest level class whenever possible.



```
1 - Dec2024SeleniumSessions/src/main/java/sel sessions/WebDriverWaitWithFluentFeatures.java - Eclipse IDE
riverWaitWithFluentFeatures.java  FluentWait.class  WebDriverWait.class

:Exception;
referenceException;

:Driver;
expectedConditions;
eDriverWait;

tures {
}

js) {
--> FluentWait(C)[until(){}, other methods] --> Wait(I): until();

it

)ro.com/");
');

verWait(driver, Duration.ofSeconds(10));
seconds(2))
option.class)
enceException.class)
not found====");

isibilityOfElementLocated(username)).sendKeys("testautomation");
```

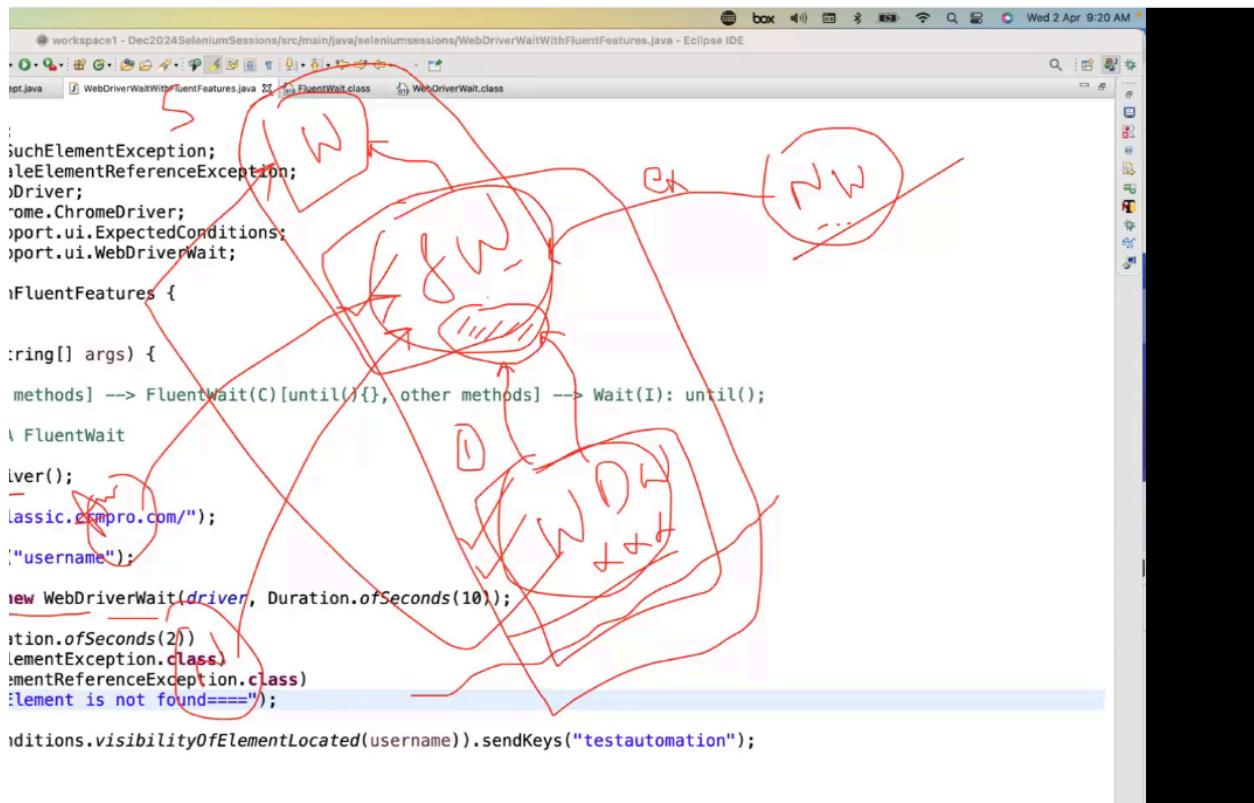
Element util class-
Updated alert method.
Used webdriver wait with fluent wait.

```

438
439 // *****wait for alert(JS POP)*****/
440 public Alert waitForAlert(int timeOut) {
441     WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeOut));
442     wait.pollingEvery(Duration.ofSeconds(1))
443         .ignoring(NoAlertPresentException.class)
444         .withMessage("==js alert not present");
445     return wait.until(ExpectedConditions.alertIsPresent());
446 }

```

Tomorrow i want to make my own wait-
I can extend fluent wait. So fluent wait acts as a
service layer. 48.00 used as library for external tools.



We can also supply polling time as third parameter
in one of the webdriver constructor-
Then no need of separate line for polling.

```

29     WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10), Duration.ofSeconds(2));
30     //WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
31
32     wait
33         .ignoring(NoSuchElementException.class)
34         .ignoring(StaleElementReferenceException.class)
35         .withMessage("==Element is not found==");
36

```

Ocd-



```

1 package seleniumsessions;
2
3 import java.time.Duration;
4
5 import org.openqa.selenium.By;
6 import org.openqa.selenium.NoSuchElementException;
7 import org.openqa.selenium.StaleElementReferenceException;
8 import org.openqa.selenium.WebDriver;
9 import org.openqa.selenium.chrome.ChromeDriver;
10 import org.openqa.selenium.support.ui.ExpectedConditions;
11 import org.openqa.selenium.support.ui.WebDriverWait;
12
13 public class WebDriverWaitWithFluentFeatures {
14
15     static WebDriver driver;
16     public static void main(String[] args) {
17
18         //WebDriverWait(C)[no methods] --> FluentWait(C)[until(){}, other methods] --> Wait(I): until();
19
20         //WebDriverWait IS - A FluentWait
21
22         driver = new ChromeDriver();
23
24         driver.get("https://classic.crmpro.com/");
25
26         By username = By.name("username");
27
28         WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10), Duration.ofSeconds(2));
29         //WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
30
31         wait
32             .ignoring(NoSuchElementException.class)
33             .ignoring(StaleElementReferenceException.class)
34             .withMessage("==Element is not found==");
35
36         wait.until(ExpectedConditions.visibilityOfElementLocated(username)).sendKeys("testautomation");
37
38     }
39
40
41 }
42
43 }
44

```

58 - naveen doesnt like screenshot and writing.

```
ElementUtil.java WaitForElements.java FluentWaitConcept.java WebDriverWaitWithFluentFeatures.java
1 package seleniumsessions;
2
3 public class TimeUtil {
4
5     private static final int SHORT_TIME = 5000;
6     private static final int MEDIUM_TIME = 10000;
7     private static final int LONG_TIME = 15000;
8     private static final int LONGER_TIME = 20000;
9
10    public static void shortWait() {
11        try {
12            Thread.sleep(SHORT_TIME);
13        } catch (InterruptedException e) {
14            e.printStackTrace();
15        }
16    }
17
18    public static void mediumWait() {
19        try {
20            Thread.sleep(SHORT_TIME);
21        } catch (InterruptedException e) {
22            e.printStackTrace();
23        }
24    }
25
26    public static void longWait() {
27        try {
28            Thread.sleep(SHORT_TIME);
29        } catch (InterruptedException e) {
30            e.printStackTrace();
31        }
32    }
33
34    public static void largerWait() {
35        try {
36            Thread.sleep(SHORT_TIME);
37        } catch (InterruptedException e) {
38            e.printStackTrace();
39        }
40    }
41}
```

Updated parameters of `Thread.sleep` for medium and larger-

```
17
18  public static void mediumWait() {
19      try {
20          Thread.sleep(MEDIUM_TIME);
21      } catch (InterruptedException e) {
22          e.printStackTrace();
23      }
24  }
25
26  public static void longWait() {
27      try {
28          Thread.sleep(SHORT_TIME);
29      } catch (InterruptedException e) {
30          e.printStackTrace();
31      }
32  }
33
34  public static void largerWait() {
35      try {
36          Thread.sleep(LONGER_TIME);
37      } catch (InterruptedException e) {
38          e.printStackTrace();
39      }
40 }
```

Updated parameter of `thread.sleep` for long wait-

```
26
27  public static void longWait() {
28      try {
29          Thread.sleep(LONG_TIME);
30      } catch (InterruptedException e) {
31          e.printStackTrace();
32      }
33 }
```

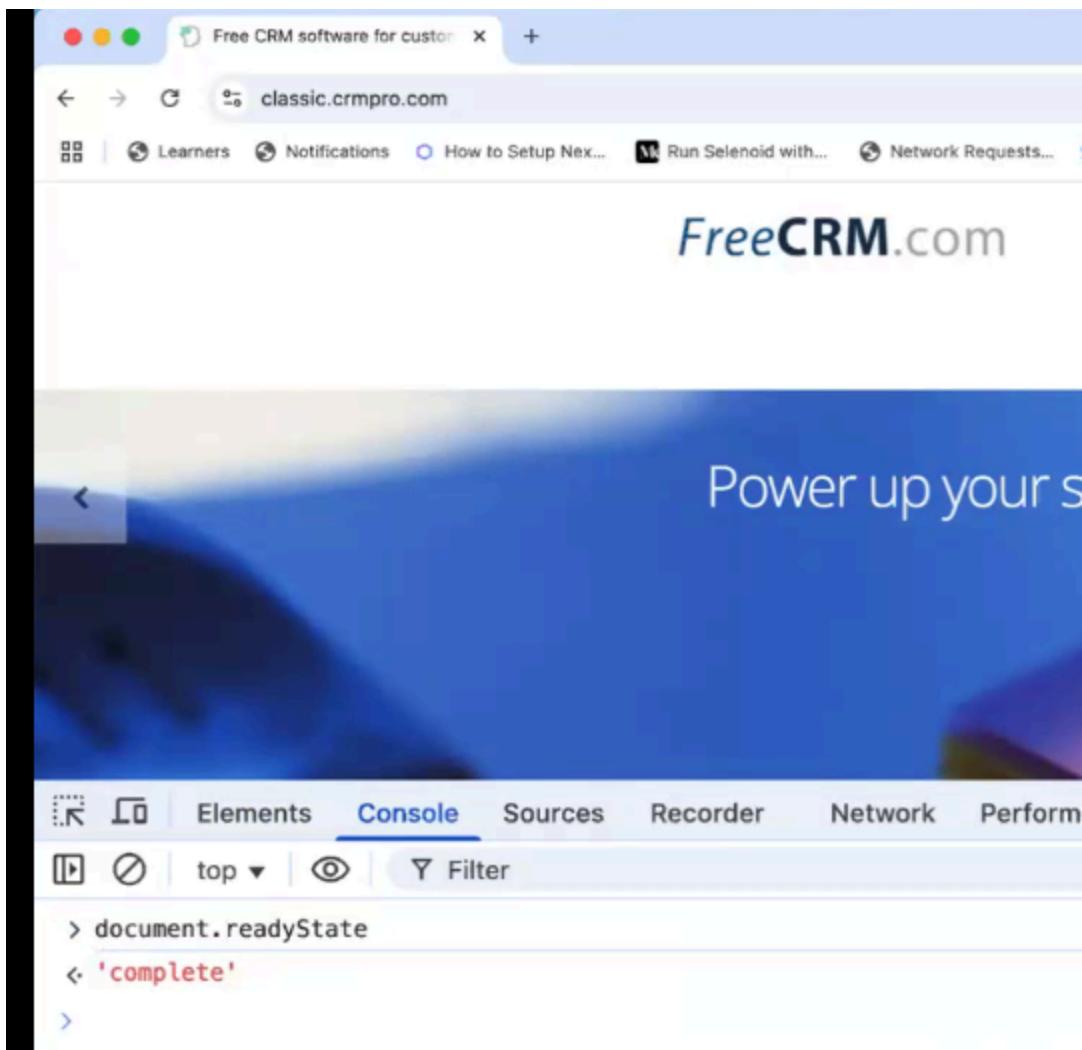
How to use in existing code-

```
2
3 import java.time.Duration;
4
5 import org.openqa.selenium.By;
6 import org.openqa.selenium.NoSuchElementException;
7 import org.openqa.selenium.StaleElementReferenceException;
8 import org.openqa.selenium.WebDriver;
9 import org.openqa.selenium.chrome.ChromeDriver;
10 import org.openqa.selenium.support.ui.ExpectedConditions;
11 import org.openqa.selenium.support.ui.WebDriverWait;
12
13 public class WebDriverWaitWithFluentFeatures {
14
15     static WebDriver driver;
16     public static void main(String[] args) {
17
18         //WebDriverWait(C)[no methods] --> FluentWait(C)[until(){}, other methods] --> Wait(I): until();
19
20         //WebDriverWait IS - A FluentWait
21
22         driver = new ChromeDriver();
23
24         driver.get("https://classic.crmpro.com/");
25         TimeUtil.shortWait();
```

Avoided thread.sleep, trycatch, throws.

If waits dont work then maintain thread.sleep as util like the above. Clever way of using thread.sleep. Its implementation is hidden in timeutils.

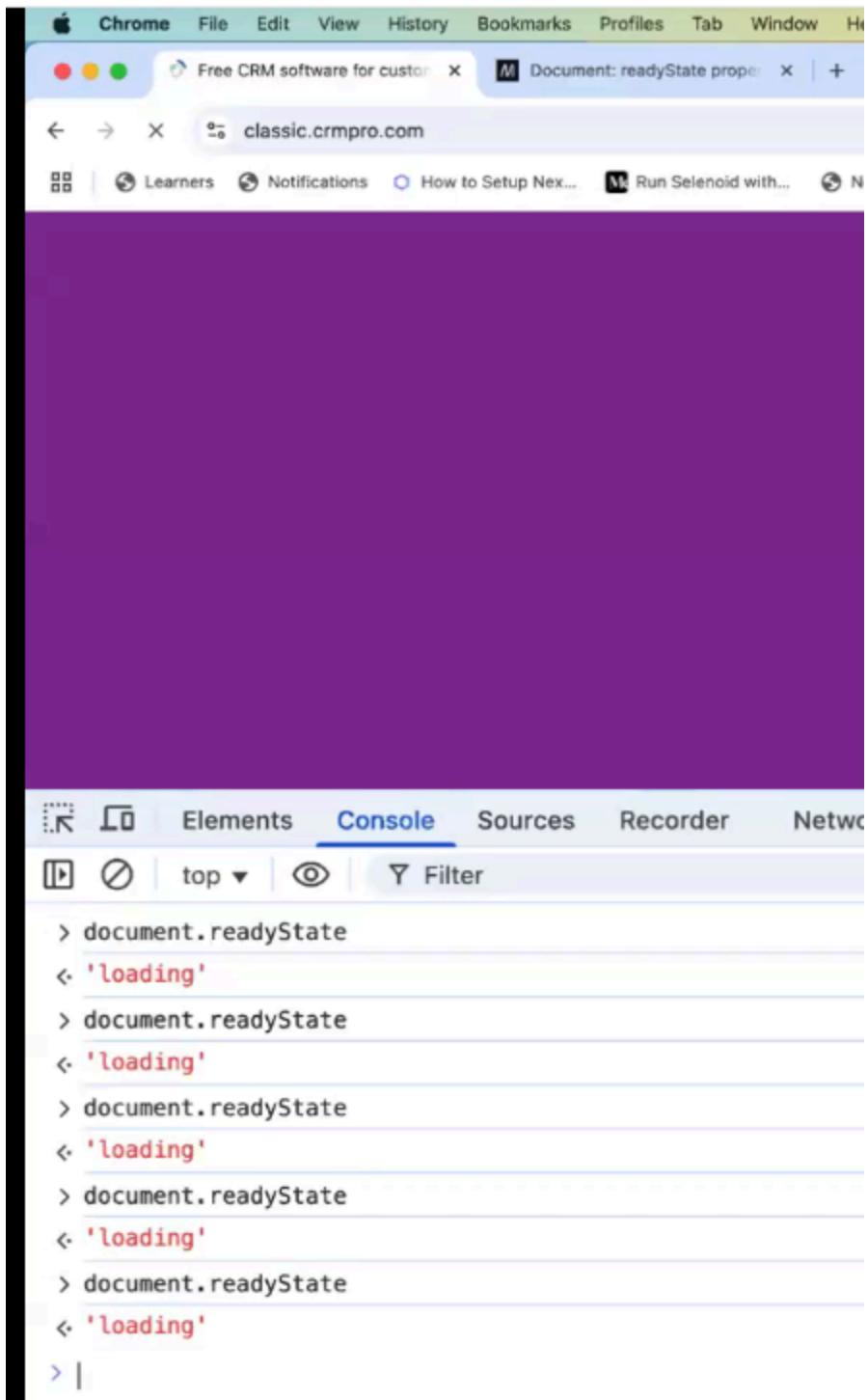
Wait for page loading-

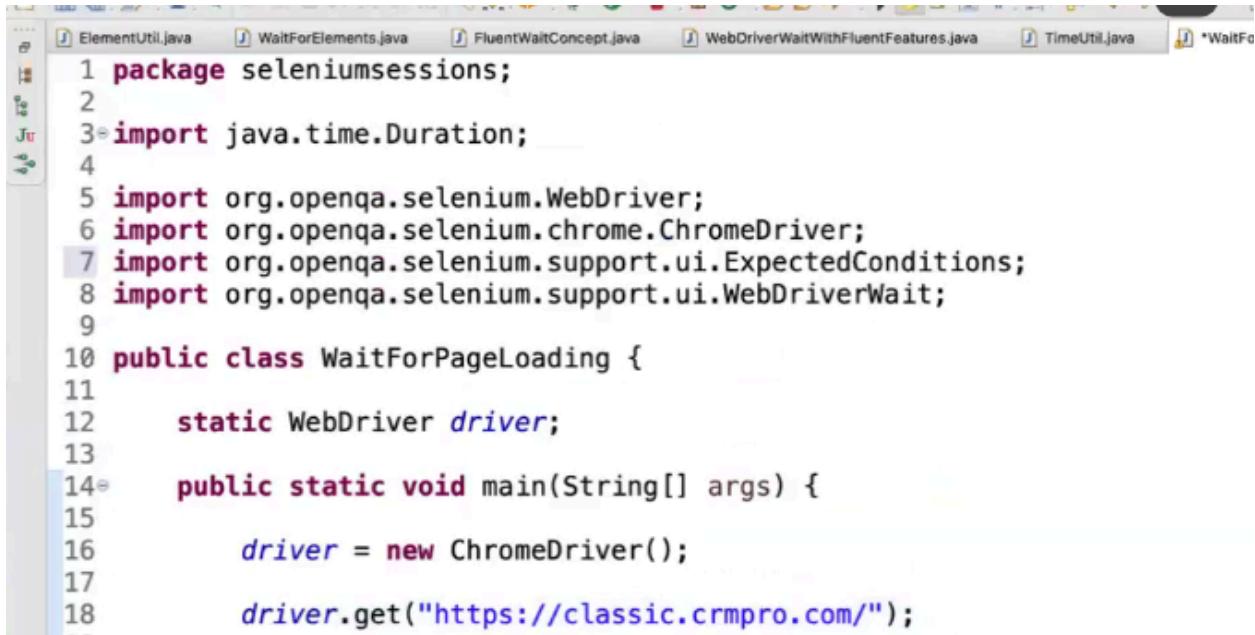


Three states-

The screenshot shows a browser window displaying the MDN Web Docs page for the `readyState` property of the `Document` interface. The URL in the address bar is `developer.mozilla.org/en-US/docs/Web/API/Document/readyState#loading`. The page navigation bar includes links for Learners, Notifications, How to Setup Next.js, Run Selenoid with..., Network Requests..., How to Remove Ar... (with a NEW badge), Learn Git Branching, macos - How to u..., nopCommerce de..., and Maven. The main content area has a "Theme" button. The breadcrumb navigation shows References > Web APIs > Document > readyState. A "Filter" button is available. The left sidebar lists properties: `loading`, `interactive`, `complete`, and others like `plugins`, `pointerLockElement`, `preferredStyleSheetSet`, `prerendering`, `referrer`, `rootElement`, and `scripts`. The `loading` state is highlighted with a purple background. The description for `loading` states: "The `document` is still loading." The description for `interactive` states: "The document has finished loading and the document has been parsed but sub-resources such as scripts, images, stylesheets and frames are still loading. The state indicates that the `DOMContentLoaded` event is about to fire." The description for `complete` states: "The document and all sub-resources have finished loading. The state indicates that the `load` event is about to fire."

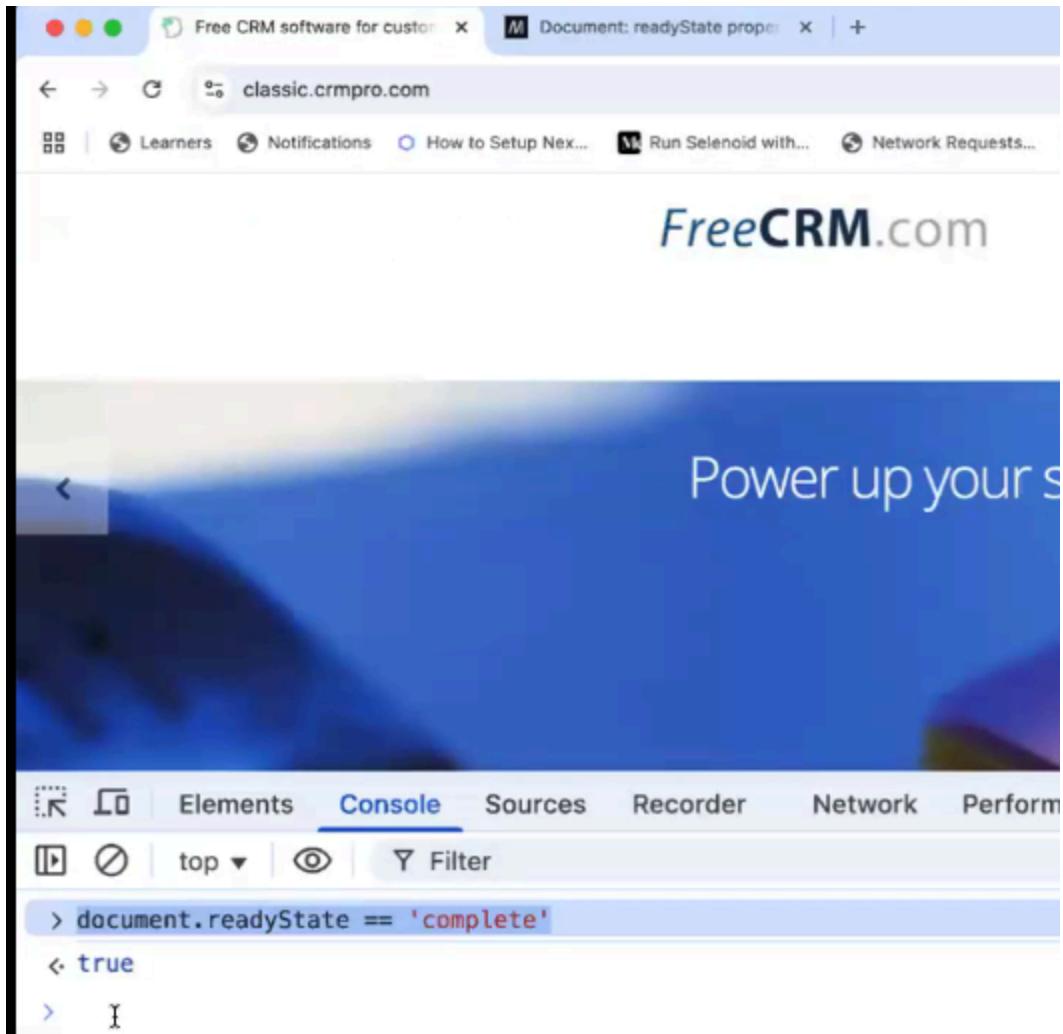
Refresh page and see state-





```
1 package seleniumsessions;
2
3 import java.time.Duration;
4
5 import org.openqa.selenium.WebDriver;
6 import org.openqa.selenium.chrome.ChromeDriver;
7 import org.openqa.selenium.support.ui.ExpectedConditions;
8 import org.openqa.selenium.support.ui.WebDriverWait;
9
10 public class WaitForPageLoading {
11
12     static WebDriver driver;
13
14     public static void main(String[] args) {
15
16         driver = new ChromeDriver();
17
18         driver.get("https://classic.crmpro.com/");
```

Wait for the complete page to load because sometimes some elements are loading and our scripts fail. Examples, frames.
Until returns object here.



Create utility-
Same class.
Convert to string as we want in java.

```
29+ public boolean isPageLoaded(int timeOut) {  
30     WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeOut));  
31     String flag = wait.until(ExpectedConditions.jsReturnsValue("return document.readyState === 'complete'")).toString();  
32     return Boolean.parseBoolean(flag); //true  
33 }  
34
```

Make function static.

```
29+ public static boolean isPageLoaded(int timeOut) {  
30     WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeOut));  
31     String flag = wait.until(ExpectedConditions.jsReturnsValue("return document.readyState === 'complete'")).toString();  
32     return Boolean.parseBoolean(flag); //true  
33 }  
34
```

Call from same class.

```

21     //wait.until(ExpectedConditions.jsReturnsValue("return document.readyState == 'complete'"));
22
23
24
25     if(isPageLoaded(5)) {
26         System.out.println("yes page is loaded completely");
27     }

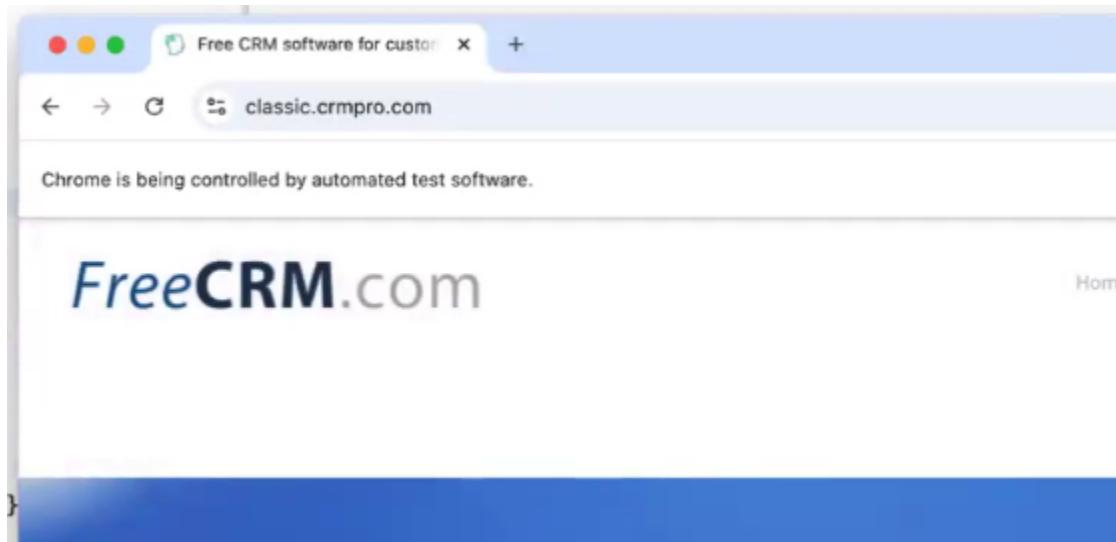
```

```

Console Problems Javadoc Declaration
terminated> WaitForPageLoading (2) [Java Application] /Users/j
Apr 02, 2025 9:43:08 AM org.openqa.selenium.support.ui.ExpectedConditions$1
WARNING: Unable to find an exact match for "yes page is loaded completely"

```

Page loaded fully.



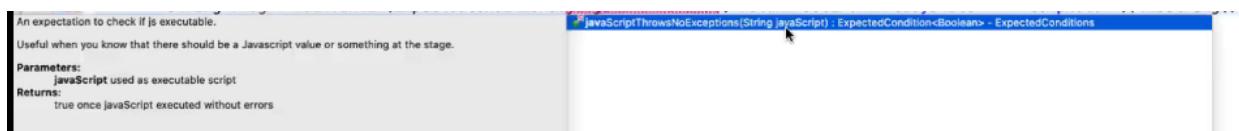
Add to element util-

```

574     public boolean isPageLoaded(int timeOut) {
575         WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeOut));
576         String flag = wait.until(ExpectedConditions.jsReturnsValue("return document.readyState === 'complete'")).toString();
577         return Boolean.parseBoolean(flag); //true
578     }

```

We can use javascript also and pass the js code-



You can write `document.title`, `document.url` etc and wait for it to return. Convert to string. 1.13