Java

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1  2  3  4  5  6  7 | WebDriver driver = new FirefoxDriver();    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);    driver.get("http://url\_that\_delays\_loading");    WebElement myDynamicElement = driver.findElement(By.id("myDynamicElement"));  **Fluent Wait**  Each **FluentWait** instance defines the maximum amount of time to wait for a condition, as well as the frequency with which to check the condition. Furthermore, the user may configure the wait to ignore specific types of exceptions whilst waiting, such as **NoSuchElementExceptions** when searching for an element on the page.   |  |  |  |  | | --- | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | // Waiting 30 seconds for an element to be present on the page, checking      // for its presence once every 5 seconds.      Wait wait = new FluentWait(driver)        .withTimeout(30, SECONDS)        .pollingEvery(5, SECONDS)        .ignoring(NoSuchElementException.class);      WebElement foo = wait.until(new Function() {        public WebElement apply(WebDriver driver) {        return driver.findElement(By.id("foo"));        }       });  **Explicit Wait**  It is more extendible in the means that you can set it up to wait for any condition you might like. Usually, you can use some of the prebuilt **ExpectedConditions** to wait for elements to become clickable, visible, invisible, etc.   |  |  | | --- | --- | | 1  2  3 | WebDriverWait wait = new WebDriverWait(driver, 10);    WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id("someid"))); | | |

## WebDriver Waits

What are WebDriver waits? What is the difference between implicit wait, explicit wait and fluent wait in WebDriver? More specifically, what is the relation between WebDriverWait and FluentWait?

Here are examples of using each wait method in WebDriver with Java.

### ****Implicit Wait****

An **implicit wait** is to tell WebDriver to poll the DOM for a certain amount of time when trying to find an element or elements if they are not immediately available. The default setting is 0. Once set, the implicit wait is set for the life of the WebDriver object instance.

* [**How to wait for a page to load in WebDriver**](https://www.testingexcellence.com/webdriver-wait-page-load-example-java/)
* [**WebDriver wait for AJAX call to complete**](https://www.testingexcellence.com/webdriver-wait-for-ajax-complete/)
* [**WebDriver – open new browser window with Javascript**](https://www.testingexcellence.com/webdriver-how-to-open-new-browser-window-with-javascript/)

Example of using implicit wait

WebDriver driver = new FirefoxDriver();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

driver.get("http://somedomain/slow\_loading\_url");

WebElement dynamicElement = driver.findElement(By.id("dynamicElement"));

**When should we use implicit waits?**

Normally, it is not recommended to use implicit waits, when we can use explicit waits or fluent waits.

### ****Explicit Wait****

An **explicit wait** is a code that you define to wait for a certain condition to occur before proceeding further in the code. WebDriverWait by default calls the ExpectedCondition every 500 milliseconds until it returns successfully.

Example of using explicit wait

WebDriver driver = new FirefoxDriver();

driver.get("http://somedomain/someurl");

WebElement dynamicElement = (new WebDriverWait(driver, 10))

.until(ExpectedConditions.presenceOfElementLocated(By.id("dynamicElement")));

**When should we use explicit waits?**

We would normally use explicit wait if an element takes a long time to load. We also used explicit wait to check CSS property of an element (presence, clickability. etc) which can change in Ajax applications.

### ****Fluent Wait****

When using the FluentWait instance, we can specify:

* The frequency with which FluentWait has to check the conditions defined.
* Ignore specific types of exception waiting such as NoSuchElementExceptions while searching for an element on the page.
* The maximum amount of time to wait for a condition

Example of using FluentWait

// Waiting 30 seconds for an element to be present on the page, checking

// for its presence once every 5 seconds.

Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)

.withTimeout(30, SECONDS)

.pollingEvery(5, SECONDS)

.ignoring(NoSuchElementException.class);

WebElement foo = wait.until(new Function<WebDriver, WebElement>()

{

public WebElement apply(WebDriver driver) {

return driver.findElement(By.id("foo"));

}

});

**When should we use FluentWait?**

When you try to test the presence of an element that may appear after every x seconds/minutes.

### ****Difference Between WebDriverWait and FluentWait****

WebDriverWait is a subclass of FluentWait. In FluentWait you have more options to configure, along with maximum wait time, like polling interval, exceptions to ignore etc.

So, instead of waiting and then using findElement:

WebDriverWait wait = new WebDriverWait(driver, 18);

wait.until(ExpectedConditions.elementToBeClickable(By.linkText("Account")));

WebElement element = driver.findElement(By.linkText("Account"));

element.sendKeys(Keys.CONTROL);

element.click();

we can use:

WebElement element = wait.until(

ExpectedConditions.elementToBeClickable(By.linkText("Account")));