**Webdriver Wait**

Genearl problem is when you use

driver.manage().timeouts.implictwait(10,Timeunit.seconds) driver for it’s life it will have the waiting time as 10 seconds what that means ?

even if you have the explicit wait like below your code still wait for 10seconds only

WebDriverWait wait=**new** WebDriverWait(driver, timeunit);

wait.until(**new** ExpectedCondition<Boolean> () {

**public** Boolean apply(WebDriver d) {

**return** *istextpresent*(d, b, text);

}

});

So to over come such situation what you should do ?

driver.manage().timeouts().implicitlyWait(0, TimeUnit.*SECONDS*); //set it zero

WebDriverWait wait=**new** WebDriverWait(driver, timeunit);

wait.until(**new** ExpectedCondition<Boolean> () {

**public** Boolean apply(WebDriver d) {

**return** *istextpresent*(d, b, text);

}

});

flag=*istextpresent*(driver, b, text);

driver.manage().timeouts().implicitlyWait(*Default\_wait\_page*, TimeUnit.*SECONDS*); // then update back

very important is to know that

Expectedconditions :class

Expectedcondition:Interface

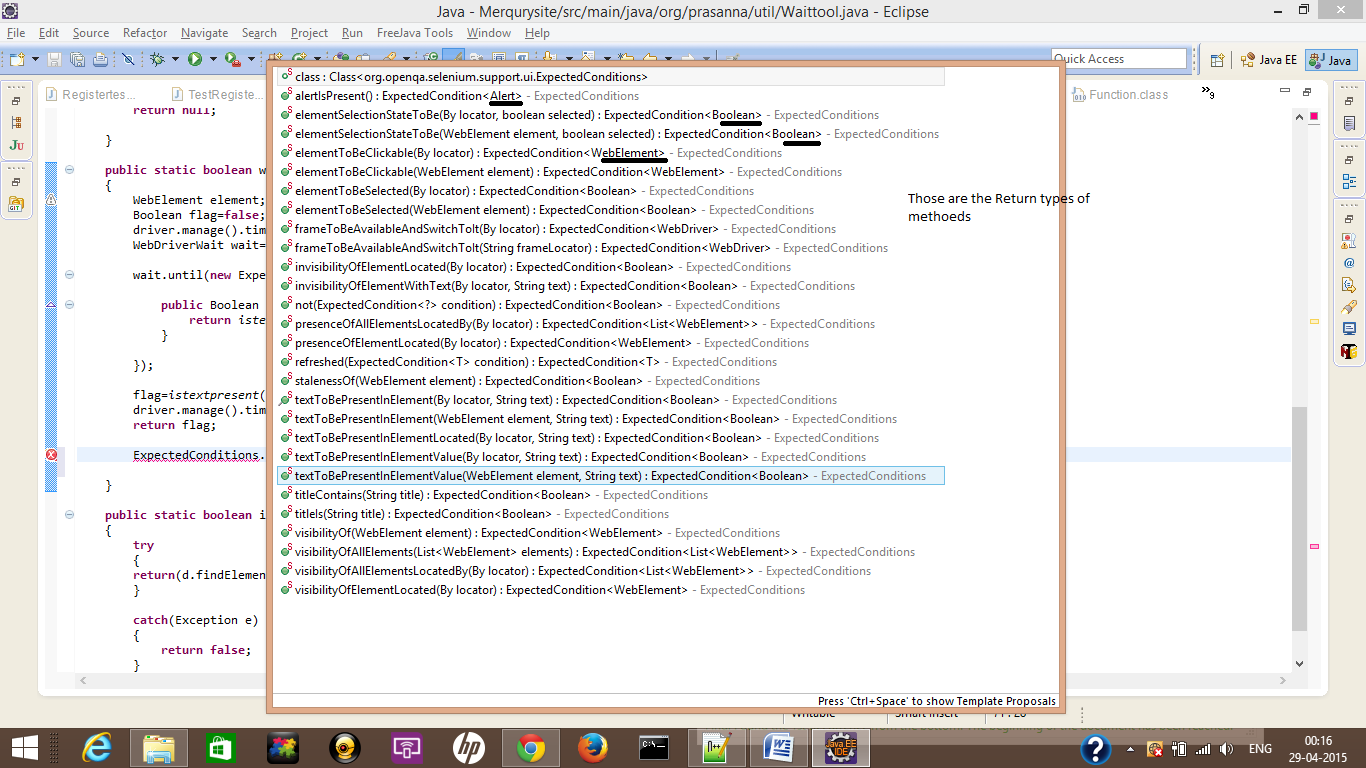
To use both first we need to create webdriverwait element so how to create it

Webdriver wait=new webdriverwait(driver,timeoutinseconds)

Then

Wait.unitl(Here either you can use Expectedconditions or Expectedcondition )

**Using the Expecetedcondtions class**

****

**Expectedcondition Interface**

Wait.unitl(Expectedcondition<Boolean>(){

@override

Public apply(Webdriver d)

{ any function that would generate Boolean

}

});

TO under stand the wait thing in here we should start to under stand the Expectedcondtion interface

So how is structure of the Expectedcondtion ?

public interface ExpectedCondition<webdriver> extends Function<WebDriver, T> {}

so this only extends the Function<Webdriver,T>

then we should look for Function<Webdriver,T> interface

this interface has

Apply and equals methoed

What is signature of the apply methoed ?

@Nullable T apply(@Nullable webdriver input);

So all the methoeds that are present in the Expectedcondtions class has return type as public static Expectedcondtion<T> titleis(String title)

{

Return New Expectedcondtion<T>{

@override

T Apply(Webdriver d)

{

Code to check the condtion

If(conditon)

{

Return(T)

}

}

}

}