1 - Open http://seleniumpractise.blogspot.com/2016/08/how-to-use-explicit-wait-in-selenium.html

Click on timer

Wait for text "WebDriver"

Use 2 conditions of ExpectedCondition

Solution:

**public** **class** TimerElementCatch {

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

// Create a new instance of the Chrome driver

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

// Navigate to the website

driver.get("http://seleniumpractise.blogspot.com/2016/08/how-to-use-explicit-wait-in-selenium.html");

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

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

// click on Timer

Thread.*sleep*(3000);

driver.findElement(By.*xpath*("//button[normalize-space()='Click me to start timer']")).click();

// wait for text 'WebDriver'

WebDriverWait wait = **new** WebDriverWait(driver, Duration.*ofSeconds*(20));

WebElement AutomationTools = driver.findElement(By.*xpath*("//p[@id='demo']"));

wait.until(ExpectedConditions.*textToBePresentInElement*(AutomationTools, "WebDriver"));

System.***out***.println(AutomationTools.getText());

}

}

2- Create a method in Utility class named as "captureScreenshotOfWebElement"

will capture the screenshot of element

will store element screenshot in element screenshot folder

all screenshot should have date time stamp

Solution:

**public** **class** Utility {

**static** WebDriver *driver*;

**public** **static** **void** captureScreenshotOfWebElement(WebDriver driver ) {

// TakesScreenshot ts = (TakesScreenshot) driver;

// File src = ts.getScreenshotAs(OutputType.FILE);

// File destination = new File("./MyScreenshot.png");

**try**

{

FileHandler.*copy*(((TakesScreenshot) driver).getScreenshotAs(OutputType.***FILE***), **new** File("./screenshots/Screenshot\_"+*getCurrentTime*()+".png"));

}

**catch** (IOException e)

{

System.***out***.println("Something went wrong" +e.getMessage());

}

}

**public** **static** String getCurrentTime() {

String date = **new** SimpleDateFormat("HH:mm:ss\_dd\_MM\_yyyy").format(**new** Date());

**return** date;

}

Create a method in Utility class named as "waitForElement" (dont use any waits)

method should except locator and timeout in seconds

will wait until element is not enabled

add sleep of 1 second

handle all exception while retrying

Solution:

**public** **class** Utility {

**static** WebDriver *driver*;

**public** **static** WebElement waitForElement(By locator, **int** timeout) {

WebElement element = **null**;

**long** endTime = System.*currentTimeMillis*() + (timeout \* 1000);

**while** (System.*currentTimeMillis*() < endTime) {

**try** {

element = *driver*.findElement(locator);

**if** (element.isEnabled()) {

**return** element;

}

} **catch** (NoSuchElementException e) {

// handle all exceptions while retrying

}

**try** {

Thread.*sleep*(1000); // sleep for 1 second

} **catch** (InterruptedException e) {

Thread.*currentThread*().interrupt();

}

}

**throw** **new** TimeoutException("Timed out waiting for element to be enabled: " + locator);

}

}