

# TestApp by Unicus

## Tutorial för testutvecklare

Författare: Joakim Odermalm

Dokumentversion: 1.1, 190510, Lade till try/catch för testdatahämtning i exempel 2  
1.0, 190410, Skapad

### Innehåll

|   |   |
|---|---|
| <b>API</b>                                      | 2 |
| <b>Indata</b>                                   | 2 |
| <b>Utdata</b>                                   | 2 |
| <b>Interface</b>                                | 2 |
| <b>Konstruktor-metoder för TestResult</b>       | 2 |
| <b>Förklaring</b>                               | 3 |
| <b>Förberedelser</b>                            | 3 |
| <b>Iterativa steg</b>                           | 3 |
| <b>Passerat test</b>                            | 3 |
| <b>Oförutsägbara fel</b>                        | 3 |
| <b>Sammanfattning och kontrollera dig själv</b> | 3 |
| <b>Enkelt exempel</b>                           | 4 |
| <b>Realistiskt exempel</b>                      | 5 |

# API

## Indata

Typ av webdriver, från globalt användarval i UI.

Testdata i array av textsträngar, från inmatning per test i UI.

## Utdata

Ett objekt **TestResult** med

En obligatorisk **Outcome** - Pass, Fail, Warning eller NotRun

Samt tre valfria informationsdetaljer:

Ett teststeg (nummer) - för att meddela vilket steg ett test misslyckades på

Ett meddelande (textsträng) - meddela något om vad som gick fel, varning, vad som utförts eller liknande.

En Exception - om en exception uppstått under körningen kan information om denna returneras

Ovan väljer testutvecklaren hur den vill använda, med ett TestResult returneras också automatiskt körningstiden och (i förekommande fall) vilken webdriver som användes vid körningen.

Utdatan landar sedan i motsvarande fält i UI:t, se illustration. [Kommande illustration]

## Interface

```
public interface IWebTest
{
    TestResult Test(WebDriverType webDriverType, string[] testData);
}
public interface ITest
{
    TestResult Test(string[] testData);
    TestKind Kind { get; } // { Other, Web, AD, Exchange, ADO };
}
```

## Konstruktor-metoder för TestResult

```
TestResult(Outcome o);
TestResult(Outcome o, int step)
TestResult(Outcome o, string message)
TestResult(Outcome o, Exception ex)
TestResult(Outcome o, int step, string message)
TestResult(Outcome o, int step, Exception ex)
TestResult(Outcome o, string message, Exception ex)
TestResult(Outcome o, int step, string message, Exception ex)
```

# Förklaring

I exemplet nedan är markerat med **överstrykning** hur API-funktionerna används.

## Förberedelser

Allra överst importeras **TestRunnerLib**, klassen deklarerar interfacet **IWebTest** och vi definierar metoden **Test** som tar parametrarna **WebDriverType** samt en **string[]** med testdata och returnerar ett **TestResult**.

Härnäst förbereder vi data som kommer att användas i testet och initialiserar en stegräknare **step** samt en meddelandesträng **message**.

Vi använder sedan hjälpmetoden **WebDriver.Get** för att hämta den webdriver som användaren valt.

## Iterativa steg

Genom hela testet gör vi sedan tre olika saker:

- När ett nytt steg i testet påbörjas räknas **step** upp. Varför detta är viktigt framgår lite senare.
- När något händer som är värt att anmärka men inte behöver faila testet lägger vi till anmärkningen till **message**. (I exemplet t.ex. när flera företag el. avdelningar med samma namn hittas).
- När något vi testar specifikt efter misslyckas avbryts testet genom att skicka ett **TestResult med Outcome.Fail**, teststeg och ett meddelande om vad felet var.

## Passerat test

Om hela testet passerats skickar vi ett **TestResult med Outcome.Pass** och strängen **message** som innehåller ev. icke-kritiska meddelanden som uppkommit.

## Oförutsägbara fel

Sist har vi uppfångandet av undantag (exceptions). Vi har tidigare testat efter specifika fel, men här fångas oförutsägbara fel upp - Timeout, NoSuchElementException, etc. Dessa kan fångas upp från var som helst i testkörningen, men genom att vi har räknat upp **step** för varje teststeg enl. ovan kommer det att framgå av vårt **TestResult** under vilket steg felet uppstod.

Lägg märke till att **WebDriverException** som fångar denna typ av undantag (NoSuchElement mfl.) returnerar **Outcome.Fail** precis som vid de specifika felen ovan, men vid **DriverNotFoundException** och den allmänna **Exception** returnerar vi istället **Outcome.Warning** (visas som en gul varningstriangel i appen) för att markera att något gått fel med testkörningen i sig - i motsats till funnet fel i testobjektet.

## Sammanfattning och kontrollera dig själv

Nedan följer först ett enkelt introducerande exempel, och sedan ett mer realistiskt exempel där de iterativa stegen är tydligare (med fler felpunkter och användning av **message**.)

Förslagsvis kan man gå igenom exemplen punkt för punkt med denna förklaring tillgänglig och kontrollera att man förstår:

- De förberedande delarna, **överstrukna i blått**.
- De iterativa delarna: **step**, lägga till information till **message** och kontroll av förutsägbara fel som leder till **Outcome.Fail**.
- Avslutning med passerat test - **Outcome.Pass** med **message**.
- Uppfångande av oförutsägbara fel med olika typer av **Exception**.

## Enkelt exempel

```
namespace Esumit
```

```
{
    // TestRunnerApp
    using TestRunnerLib;
    // Selenium
    using OpenQA.Selenium;
    using OpenQA.Selenium.Support.UI;

    public class SetUser : IWebTest
    {
        public TestResult Test(WebDriverType webDriverType, string[] testdata)
        {
            // Test values
            string loginUser = testdata[1];

            int step = 0;
            string message = string.Empty;

            try
            {
                using (IWebDriver driver = WebDriver.Get(webDriverType))
                {
                    driver.Navigate().GoToUrl("https://demo2.coreit.se/esumit");
                    var wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

                    step = 1;
                    // Klicka "byt"
                    IWebElement e = wait.Until(c => c.FindElement(By.Id("ctl00_lblDemo")));
                    e.Click();

                    step = 2;
                    // Välj användare
                    IWebElement combo = wait.Until(c => c.FindElement(By.Id("ctl00_main_selUser")));
                    var select = new SelectElement(combo);
                    select.SelectByValue(loginUser);

                    step = 3;
                    // Klicka OK-knapp
                    e = wait.Until(c => c.FindElement(By.Id("ctl00_main_Button1")));
                    e.Click();

                    step = 4;
                    // Kontrollera användarnamn i toppmenyn
                    e = wait.Until(c => c.FindElement(By.Id("ctl00_lblUser")));
                    if (e.Text == loginUser)
                        return new TestResult(Outcome.Pass, "Logged in: " + e.Text);
                    else
                        return new TestResult(Outcome.Fail, $"Lyckades inte byta till användare: {loginUser}");
                }
            }

            // Web driver error
            catch (OpenQA.Selenium.DriverServiceNotFoundException ex)
            {
                return new TestResult(Outcome.Warning, step, ex);
            }
            // Selenium exception during action
            catch (OpenQA.Selenium.WebDriverException ex)
            {
                return new TestResult(Outcome.Fail, step, message, ex);
            }
            // Other errors
            catch (Exception ex)
            {
                return new TestResult(Outcome.Warning, step, message, ex);
            }
        }
    }
}
```

## Realistiskt exempel

```
namespace Esumit
```

```
{  
    // TestRunnerApp  
    using TestRunnerLib;  
    // Selenium  
    using OpenQA.Selenium;  
    using OpenQA.Selenium.Support.UI;  
  
    public class Test16 : IWebTest  
  
    /* ID: 16  
     * Rubrik: Skapande av avdelning  
     * Systemdel: Inställningar -> Organisation  
     * Förberedelser: Testfall 15  
     */  
  
    {  
        int step = 0;  
        string message = string.Empty;  
  
        public TestResult Test(WebDriverType webDriverType, string[] testdata)  
        {  
            // Set testing values, fall back to defaults  
            try  
            {  
                string companyName = string.IsNullOrEmpty(testdata[1]) ? "Regressiva AB" : testdata[1];  
                string deptName = string.IsNullOrEmpty(testdata[2]) ? "Riskanalys" : testdata[2];  
                string ou = string.IsNullOrEmpty(testdata[3]) ? "EN:Groups" : testdata[3];  
                string loginUser = string.IsNullOrEmpty(testdata[4]) ? "Unicus eSumIT Admin" : testdata[4];  
                string expectedOURow = $"OU=esumitDemo,DC=syslab,DC=local";  
            }  
            catch (IndexOutOfRangeException ex)  
            {  
                return new TestResult(Outcome.Warning, step, "Testdata saknas", ex);  
            }  
            try  
            {  
                using (IWebDriver driver = WebDriver.Get(webDriverType))  
                {  
                    driver.Navigate().GoToUrl("https://demo2.coreit.se/esumit");  
                    var wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));  
  
                    // Byt användare till parameter  
                    IWebElement loginBtn = wait.Until(c => c.FindElement(By.Id("ctl00_lblDemo")));  
                    loginBtn.Click();  
                    IWebElement combo = wait.Until(c => c.FindElement(By.Id("ctl00_main_selUser")));  
                    var select = new SelectElement(combo);  
                    select.SelectByValue(loginUser);  
                    IWebElement okBtn = wait.Until(c => c.FindElement(By.Id("ctl00_main_Button1")));  
                    okBtn.Click();  
                    IWebElement userLabel = wait.Until(c => c.FindElement(By.Id("ctl00_lblUser")));  
                    if (userLabel.Text != loginUser)  
                        return new TestResult(Outcome.Fail, step, $"Lyckades inte byta till användare:  
{loginUser}");  
  
                    // Förberedelser klara  
  
                    step = 1;  
                    // Klicka inställningar  
                    IWebElement e = wait.Until(c => c.FindElement(By.Id("ctl00_lblSettings")));  
                    e.Click();  
  
                    // Klicka organisation-knapp  
                    e = wait.Until(c => c.FindElement(By.Id("ctl00_main_settingMenu1_lblDepartment")));  
                    e.Click();  
  
                    step = 2;  
                    // Hitta organisation i träd  
                    var companyList = new List<IWebElement>();  
                    companyList.AddRange(driver.FindElements(By.XPath("//div[@id=\"ctl00_main_RadTreeViewDep\"]/descendant::span[text()=\"\" + companyName + \"\"]")));  
                    if (companyList.Count == 0)
```

```

        return new TestResult(Outcome.Fail, step, "Hittade inte företag " + companyName);
    if (companyList.Count > 0)
        message += $"Hittade {companyList.Count} företag med namn: {companyName}, skapar avd i
första.";

```

```

// klicka vald organisation
companyList[0].Click();

```

```

// klicka "skapa ny avd"-knapp
e = wait.Until(c => c.FindElement(By.Id("ctl00_main_btnAdd")));
e.Click();

```

```

step = 3;
// Skriv in avd.namn
e = wait.Until(c => c.FindElement(By.Id("ctl00_main_frmEdit_nameLabel")));
e.SendKeys(deptName);

```

```

step = 4;
// Klicka AD-knappen & byt iframe
e = wait.Until(c => c.FindElement(By.Id("ctl00_main_frmEdit_adImg")));
e.Click();
driver.SwitchTo().Frame("modalFrame");

```

```

// Verifiera att fönstret med AD-träd visas
e = wait.Until(c => c.FindElement(By.Id("adTree")));
if (!e.Displayed)
    return new TestResult(Outcome.Fail, step, "AD-träd visades ej");

```

```

step = 5;
// Hitta OU-element från sträng
// expander: "/*div[@id='adTree']/descendant::span[text()='EN']/../span[@class='rtPlus']"
// next node:
"/*div[@id='adTree']/descendant::span[text()='EN']/../span[@class='rtPlus']"

```

```

string[] ouArray = ou.Split(':');
string xpathString = "/*div[@id='adTree']";
xpathString += $"/descendant::span[text()='{{ouArray[0]}}']";
var toClick = new List<string>();
toClick.Add(xpathString);
for (int i = 1; i < ouArray.Length; i++)
{
    xpathString += $"/../span[@class='rtPlus']";
    toClick.Add(xpathString);
}

```

```

for (int i = 0; i < ouArray.Length; i++)
{
    var ouList = new List<IWebElement>();
    ouList.AddRange(driver.FindElements(By.XPath(toClick[i])));
    if (ouList.Count == 0)
        return new TestResult(Outcome.Fail, step, "Hittade inte OU med namn: " + ouArray[i] + ".
XPath: " + toClick[i]);

```

```

    if (ouList.Count > 1)
        message += "Hittade " + ouList.Count + " OU med namn " + ou + ", väljer första. ";
    if (i < ouArray.Length - 1)
    {
        string expanderPath = toClick[i] + "/../span[@class='rtPlus']";
        var expander = wait.Until(c => c.FindElement(By.XPath(expanderPath)));
        expander.Click();
    }
    else
        ouList[0].Click();
}

```

```

// Klicka OK-knapp
e = wait.Until(c => c.FindElement(By.Id("btnSave")));
e.Click();

```

```

// Tillbaka till tidigare iframe
driver.SwitchTo().DefaultContent();

```

```

step = 6;
// Bygg förväntad jämförelserad
string ouRow = string.Empty;

```

```

for (int i = ouArray.Length - 1; i >= 0; i--)
{
    ouRow += "OU=" + ouArray[i] + ",";
}
ouRow += expectedOURow;

e = wait.Until(c => c.FindElement(By.Id("txtOU")));
string txtOU = e.GetAttribute("value");
if (txtOU != ouRow)
    return new TestResult(Outcome.Fail, step, $"Fel i OU-rad: {txtOU}{Environment.NewLine}
Expected: {ouRow}");

```

```

step = 7;
// Spara

```

```

// *** Sidan har flera knappar med samma id men bara en synlig / klickbar ***
// Detta funkar därför ej:
//     e = wait.Until(c => c.FindElement(By.Id("ctl00_main_frmEdit_InsertButton")));
//     e.Click();
// Istället:
var btnList = new List<IWebElement>();
btnList.AddRange(driver.FindElements(By.Id("ctl00_main_frmEdit_InsertButton")));
if (btnList.Count == 0)
    return new TestResult(Outcome.Fail, step, "Hittade inte knapp");
bool btnClicked = false;
foreach (var btn in btnList)
{
    if (btn.Displayed)
    {
        btnClicked = true;
        btn.Click();
        break;
    }
}
if (!btnClicked)
    return new TestResult(Outcome.Fail, step, "No btn visible");

```

```

// Hitta avd i träd - först företag
string orgPath = "'//div[@id='ctl00_main_RadTreeViewDep']/descendant::span[text()='\" +
companyName + \"']";
var orgList = new List<IWebElement>();
orgList.AddRange(driver.FindElements(By.XPath(orgPath)));
if (orgList.Count == 0)
    return new TestResult(Outcome.Fail, step, $"Hittade inte org: {companyName}");
if (orgList.Count > 0)
    message += $"Hittade {orgList.Count} org med namn: {companyName}. Söker avd i första.";
// Click expander
string path = orgPath + "$/../span[@class='rtPlus']";
e = wait.Until(c => c.FindElement(By.XPath(path)));
e.Click();
// Hitta avd i träd - nu avd
string deptPath = orgPath + "$/../following-sibling::ul/descendant::span[text()='\" + {deptName} \"]";
var deptList = new List<IWebElement>();
deptList.AddRange(driver.FindElements(By.XPath(deptPath)));
if (deptList.Count == 0)
    return new TestResult(Outcome.Fail, step, $"Hittade inte avd: {companyName}/{deptName}");
if (deptList.Count > 0)
    message += $"Hittade {deptList.Count} avd med namn: {companyName}/{deptName}.";

// Return result, possibly with message
return string.IsNullOrEmpty(message) ? new TestResult(Outcome.Pass) : new
TestResult(Outcome.Pass, message);
}
}

```

```

// Web driver error
catch (OpenQA.Selenium.DriverServiceNotFoundException ex)
{
    return new TestResult(Outcome.Warning, step, message, ex);
}
// Selenium exception during action
catch (OpenQA.Selenium.WebDriverException ex)
{
    return new TestResult(Outcome.Fail, step, message, ex);
}

```

```
}  
// Other errors  
catch (Exception ex)  
{  
    return new TestResult(Outcome.Warning, step, message, ex);  
}  
}  
}  
}
```