

Test Automation Using Telerik Test Studio

Script – AircraftFollow – Add a Sector

```
using Telerik.TestingFramework.Controls.KendoUI;
using Telerik.WebAii.Controls.Html;
using Telerik.WebAii.Controls.Xaml;
using System;
using System.Collections.Generic;
using System.Text;
using System.Linq;

using ArtOfTest.Common.UnitTesting;
using ArtOfTest.WebAii.Core;
using ArtOfTest.WebAii.Controls.HtmlControls;
using ArtOfTest.WebAii.Controls.HtmlControls.HtmlAsserts;
using ArtOfTest.WebAii.Design;
using ArtOfTest.WebAii.Design.Execution;
using ArtOfTest.WebAii.ObjectModel;
using ArtOfTest.WebAii.Silverlight;
using ArtOfTest.WebAii.Silverlight.UI;

namespace MerlotAero_AutomationProject
{
    public class Add_a_sector : BaseWebAiiTest
    {
        #region [ Dynamic Pages Reference ]

        private Pages _pages;

        /// <summary>
        /// Gets the Pages object that has references
        /// to all the elements, frames or regions
        /// in this project.
        /// </summary>
        public Pages Pages
        {
            get
            {
                if (_pages == null)
                {
                    _pages = new Pages(Manager.Current);
                }
                return _pages;
            }
        }

        #endregion

        // Add your test methods here...
```

```

int columnNumber;
int columnNumberSTD;
int columnNumberSTA;
[CodedStep(@"New Coded Step")]
public void Depart_and_Arrive_CodedStep2()
{
    System.Threading.Thread.Sleep(4000);
}
int columnNumberFLN;
int columnNumberEquip;
int columnNumberOnwardFL;

[CodedStep(@"Search Route")]
public void Depart_and_Arrive_CodedStep()
{
    //ActiveBrowser.Window.Restore(); //18-Sep-2018: Shakeel
    //ActiveBrowser.Window.Maximize(); //

    bool flag = false;
    //int flightNumber=Utility.getFlightNumber();
    //string newFlightNumber = flightNumber.ToString();
    string newFlightNumber = "<<New route>>";
    Log.WriteLine("ScheduleName:" + newFlightNumber);
    SilverlightApp app = ActiveBrowser.SilverlightApps()[0];
    app.RefreshVisualTrees();
    RadPaneGroup RPG = app.Find.By<RadPaneGroup>("uxRadPaneGroup");
    Grid GR = RPG.Find.By<Grid>("uxMainGrid");
    DataGridColumnHeadersPresenter DGCHP =
GR.Find.By<DataGridColumnHeadersPresenter>();
    int headCount = 1;
    IList<DataGridColumnHeader> headers =
DGCHP.Find.AllBy<DataGridColumnHeader>();
    foreach (DataGridColumnHeader header in headers)
    {
        Log.WriteLine("header column are : " + header.Text);
        if (header.Text.Equals("Route"))
        {
            Log.WriteLine("Route header found");
            columnNumber = headCount;
            Log.WriteLine("ColumnNumber :" + columnNumber);
            break;
        }
        headCount++;
    }

    int headCountSTD = 1;
    foreach (DataGridColumnHeader header in headers)
    {
        Log.WriteLine("header column are : " + header.Text);
        if (header.Text.Equals("STD"))
        {
            Log.WriteLine("Flight header found");
            columnNumberSTD = headCountSTD;
            Log.WriteLine("ColumnNumberSTD :" + columnNumberSTD);
            break;
        }
        headCountSTD++;
    }
}

```

```

int headCountSTA = 1;
foreach (DataGridViewColumnHeader header in headers)
{
    Log.WriteLine("header column are : " + header.Text);
    if (header.Text.Equals("STA"))
    {
        Log.WriteLine("Flight header found");
        columnNumberSTA = headCountSTA;
        Log.WriteLine("ColumnNumberSTA :" + columnNumberSTA);
        break;
    }
    headCountSTA++;
}

int headCountFLN = 1;
foreach (DataGridViewColumnHeader header in headers)
{
    Log.WriteLine("header column are : " + header.Text);
    if (header.Text.Equals("Flight*"))
    {
        Log.WriteLine("Flight header found");
        columnNumberFLN = headCountFLN;
        Log.WriteLine("ColumnNumberFLN :" + columnNumberFLN);
        break;
    }
    headCountFLN++;
}

int headCountEquip = 1;
foreach (DataGridViewColumnHeader header in headers)
{
    Log.WriteLine("header column are : " + header.Text);
    if (header.Text.Equals("Equip"))
    {
        Log.WriteLine("Flight header found");
        columnNumberEquip = headCountEquip;
        Log.WriteLine("ColumnNumberEquip :" + columnNumberEquip);
        break;
    }
    headCountEquip++;
}

int headCountOnwardFL = 1;
foreach (DataGridViewColumnHeader header in headers)
{
    Log.WriteLine("header column are : " + header.Text);
    if (header.Text.Equals("Onward Flight*"))
    {
        Log.WriteLine("Flight header found");
        columnNumberOnwardFL = headCountOnwardFL;
        Log.WriteLine("ColumnNumberOnwardFL :" + columnNumberOnwardFL);
        break;
    }
    headCountOnwardFL++;
}

```



```

//##### Add Flight Number
#####
Log.WriteLine("Reached here to type data for
Flight Number");
cells[columnNumberFLN - 1].User.Click();
System.Threading.Thread.Sleep(2000);
cells[columnNumberFLN - 1].User.Click();
System.Threading.Thread.Sleep(2000);
TextBox TB = cells[columnNumberFLN -
1].Find.ByType<TextBox>();
TB.User.Click();
Utility.setFlightNumber();
int flightNumber = Utility.getFlightNumber();

Manager.Desktop.KeyBoard.KeyPress(System.Windows.Forms.Keys.Delete);

Manager.Desktop.KeyBoard.SendString(flightNumber.ToString());
System.Threading.Thread.Sleep(2000);
//##### Add Route
#####
Log.WriteLine("Reached here to type data for
Route");
cells[columnNumber - 1].User.Click();
System.Threading.Thread.Sleep(10000);
//cells[columnNumber-1].User.Click();
System.Threading.Thread.Sleep(10000);

Manager.Desktop.KeyBoard.TypeText((string)(System.Convert.ChangeType(Data["Route"],
typeof(string))), 5);
// Manager.Desktop.KeyBoard.TypeText("LASLAX");
System.Threading.Thread.Sleep(10000);
//##### Add
STD#####
Log.WriteLine("Reached here to type data for
STD");
cells[columnNumberSTD - 1].User.Click();
System.Threading.Thread.Sleep(2000);
cells[columnNumberSTD - 1].User.Click();
System.Threading.Thread.Sleep(2000);

Manager.Desktop.KeyBoard.SendString((string)(System.Convert.ChangeType(Data["STD"],
typeof(string))));
System.Threading.Thread.Sleep(5000);
//##### Add
STA#####
Log.WriteLine("Reached here to type data for
STA");
cells[columnNumberSTA - 1].User.Click();
System.Threading.Thread.Sleep(2000);

Pages.MerlotAero0.SilverlightApp.UxFirstButtonButton.User.Click(ArtOfTest.WebAii.Core.MouseClickType.LeftClick, 47, 6, ArtOfTest.Common.OffsetReference.TopLeftCorner,
ArtOfTest.Common.ActionPointUnitType.Pixel);
System.Threading.Thread.Sleep(2000);
cells[columnNumberSTA - 1].User.Click();
System.Threading.Thread.Sleep(2000);

```

```

Manager.Desktop.KeyBoard.SendString((string)(System.Convert.ChangeType(Data["STA"],
typeof(string)))));

                                System.Threading.Thread.Sleep(2000);
                                //##### Add Onward Flight

#####
                                // Log.WriteLine("Reached here to type data for
Onward Flight");
                                // cells[columnNumberOnwardFL-1].User.Click();
                                // System.Threading.Thread.Sleep(2000);
                                // cells[columnNumberOnwardFL-1].User.Click();
                                // System.Threading.Thread.Sleep(2000);
                                // int onwardFlightNumber=
Utility.getOnwardFlightNumber();
                                //
Manager.Desktop.KeyBoard.TypeText(onwardFlightNumber.ToString(),5);
                                // System.Threading.Thread.Sleep(2000);
                                //
Manager.Desktop.KeyBoard.KeyPress(System.Windows.Forms.Keys.Escape);
                                flag = true;
                                break;
                                }
                                //cell.User.Click();
                                }
                                cellCounter++;
                                }
                                Log.WriteLine("Current row is : " + rowCounter);
                                }
                                rowCounter++;
                                }

//Pages.MerlotAero1.SilverlightApp.VerticalLargeIncreaseRepeatbutton.UserPress(1174)
;

//Pages.MerlotAero0.SilverlightApp.VerticalLargeIncreaseRepeatbutton1.UserPress(172)
;

Manager.Desktop.KeyBoard.KeyPress(System.Windows.Forms.Keys.PageDown);
    }
}

[CodedStep(@"Click UxFirstButtonButton")]
public void Add_a_sector_CodedStep()
{
    // Click UxFirstButtonButton

Pages.MerlotAero0.SilverlightApp.UxFirstButtonButton.User.Click(ArtOfTest.WebAii.Core.MouseClickType.LeftClick, 47, 6, ArtOfTest.Common.OffsetReference.TopLeftCorner,
ArtOfTest.Common.ActionPointUnitType.Pixel);

    }

[CodedStep(@"New Coded Step")]
public void Add_a_sector_CodedStep1()
{
    bool flag = false;
    //int flightNumber=Utility.getOnwardFlightNumber;

```

```

//string newFlightNumber = flightNumber.ToString();
string newFlightNumber = null;
Log.WriteLine("FlightName:" + newFlightNumber);
SilverlightApp app = ActiveBrowser.SilverlightApps()[0];
app.RefreshVisualTrees();
RadPaneGroup RPG = app.Find.ByName<RadPaneGroup>("uxRadPaneGroup");
Grid GR = RPG.Find.ByName<Grid>("uxMainGrid");
DataGridColumnHeadersPresenter DGCHP =
GR.Find.ByType<DataGridColumnHeadersPresenter>();
int headCount = 1;
IList<DataGridColumnHeader> headers =
DGCHP.Find.AllByType<DataGridColumnHeader>();
foreach (DataGridColumnHeader header in headers)
{
    Log.WriteLine("header column are : " + header.Text);
    if (header.Text.Equals("Route"))
    {
        Log.WriteLine("Route header found");
        columnNumber = headCount;
        Log.WriteLine("ColumnNumber :" + columnNumber);
        break;
    }
    headCount++;
}

int headCountSTD = 1;
foreach (DataGridColumnHeader header in headers)
{
    Log.WriteLine("header column are : " + header.Text);
    if (header.Text.Equals("STD"))
    {
        Log.WriteLine("Flight header found");
        columnNumberSTD = headCountSTD;
        Log.WriteLine("ColumnNumberSTD :" + columnNumberSTD);
        break;
    }
    headCountSTD++;
}

int headCountSTA = 1;
foreach (DataGridColumnHeader header in headers)
{
    Log.WriteLine("header column are : " + header.Text);
    if (header.Text.Equals("STA"))
    {
        Log.WriteLine("Flight header found");
        columnNumberSTA = headCountSTA;
        Log.WriteLine("ColumnNumberSTA :" + columnNumberSTA);
        break;
    }
    headCountSTA++;
}

int headCountFLN = 1;
foreach (DataGridColumnHeader header in headers)
{
    Log.WriteLine("header column are : " + header.Text);

```

```

        if (header.Text.Equals("Flight*"))
        {
            Log.WriteLine("Flight header found");
            columnNumberFLN = headCountFLN;
            Log.WriteLine("ColumnNumberFLN :" + columnNumberFLN);
            break;
        }
        headCountFLN++;
    }
}

```

```

int headCountEquip = 1;
foreach (DataGridColumnHeader header in headers)
{
    Log.WriteLine("header column are : " + header.Text);
    if (header.Text.Equals("Equip"))
    {
        Log.WriteLine("Flight header found");
        columnNumberEquip = headCountEquip;
        Log.WriteLine("ColumnNumberEquip :" + columnNumberEquip);
        break;
    }
    headCountEquip++;
}

```

```

int headCountOnwardFL = 1;
foreach (DataGridColumnHeader header in headers)
{
    Log.WriteLine("header column are : " + header.Text);
    if (header.Text.Equals("Onward Flight*"))
    {
        Log.WriteLine("Flight header found");
        columnNumberOnwardFL = headCountOnwardFL;
        Log.WriteLine("ColumnNumberOnwardFL :" + columnNumberOnwardFL);
        break;
    }
    headCountOnwardFL++;
}

```

```

while (flag == false)
// for(int i=1;i<18;i++)
{
    DataGridViewsPresenter DGRP =
GR.Find.ByType<DataGridViewsPresenter>();
    DGRP.Refresh();
    System.Threading.Thread.Sleep(2000);
    DGRP.Refresh();
    System.Threading.Thread.Sleep(2000);
    IList<DataRow> rows = DGRP.Find.AllByType<DataRow>();
    Log.WriteLine("Total rows in second phase are " + rows.Count);
    foreach (DataRow row in rows)
    {
        //row.Refresh();
    }
}

```



```

        int rowCounter = 1;
        //if(rowCounter==4)
        //{
        //row.User.Click();

        // }

        {
            int cellCounter = 1;
            IList<DataGridCell> cells =
row.Find.AllByType<DataGridCell>();
            foreach (DataGridCell cell in cells)
            {
                //if(rowCounter==1&&cellCounter==2)
                // {
                // cell.User.Click();
                // }
                //System.Threading.Thread.Sleep(5000);
                //Log.WriteLine("clicked on first cell");

                if (cellCounter == columnNumber)
                {
                    Log.WriteLine("Cell data is :" +
cell.TextBlockContent);
                    if (cell.TextBlockContent.Equals(newFlightNumber))
                    {
                        //##### Add Equip
                        Number#####
                        Log.WriteLine("Reached here to type data for
Flight Number");

                        cells[columnNumberEquip - 1].User.Click();
                        System.Threading.Thread.Sleep(2000);
                        cells[columnNumberEquip - 1].User.Click();
                        System.Threading.Thread.Sleep(2000);

                        ComboBox CB = cells[columnNumberEquip -
1].Find.ByType<ComboBox>();
                        //CB.User.Click();

                        Manager.Desktop.KeyBoard.TypeText((string)(System.Convert.ChangeType(Data["Equipment
Type"], typeof(string))), 5);
                        System.Threading.Thread.Sleep(2000);
                        //##### Add Flight
                        Number#####
                        Log.WriteLine("Reached here to type data for
Flight Number");

                        cells[columnNumberFLN - 1].User.Click();
                        System.Threading.Thread.Sleep(2000);
                        cells[columnNumberFLN - 1].User.Click();
                        System.Threading.Thread.Sleep(2000);
                        TextBox TB = cells[columnNumberFLN -
1].Find.ByType<TextBox>();
                        TB.User.Click();
                        Utility.setFlightNumber();
                        int flightNumber = Utility.getFlightNumber();

                        Manager.Desktop.KeyBoard.KeyPress(System.Windows.Forms.Keys.Delete);

```

```

Manager.Desktop.KeyBoard.SendString(flightNumber.ToString());
System.Threading.Thread.Sleep(2000);
//##### Add Route
#####
Route");
Log.WriteLine("Reached here to type data for
cells[columnNumber - 1].User.Click();
System.Threading.Thread.Sleep(2000);
//cells[columnNumber-1].User.Click();
System.Threading.Thread.Sleep(2000);

Manager.Desktop.KeyBoard.TypeText((string)(System.Convert.ChangeType(Data["Route"],
typeof(string))), 5);
// Manager.Desktop.KeyBoard.TypeText("LASLAX");
System.Threading.Thread.Sleep(2000);
//##### Add
STD#####
STD");
Log.WriteLine("Reached here to type data for
cells[columnNumberSTD - 1].User.Click();
System.Threading.Thread.Sleep(2000);
cells[columnNumberSTD - 1].User.Click();
System.Threading.Thread.Sleep(2000);

Manager.Desktop.KeyBoard.SendString((string)(System.Convert.ChangeType(Data["STD"],
typeof(string))));
System.Threading.Thread.Sleep(5000);
//##### Add
STA#####
STA");
Log.WriteLine("Reached here to type data for
cells[columnNumberSTA - 1].User.Click();
System.Threading.Thread.Sleep(2000);

Pages.MerlotAero0.SilverlightApp.UxFirstButtonButton.User.Click(ArtOfTest.WebAii.Core.MouseClickType.LeftClick, 47, 6, ArtOfTest.Common.OffsetReference.TopLeftCorner,
ArtOfTest.Common.ActionPointUnitType.Pixel);
System.Threading.Thread.Sleep(2000);
cells[columnNumberSTA - 1].User.Click();
System.Threading.Thread.Sleep(2000);

Manager.Desktop.KeyBoard.SendString((string)(System.Convert.ChangeType(Data["STA"],
typeof(string))));
System.Threading.Thread.Sleep(2000);
//##### Add Onward Flight
#####
// Log.WriteLine("Reached here to type data for
Onward Flight");
// cells[columnNumberOnwardFL-1].User.Click();
// System.Threading.Thread.Sleep(2000);
// cells[columnNumberOnwardFL-1].User.Click();
// System.Threading.Thread.Sleep(2000);
// int onwardFlightNumber=
Utility.getOnwardFlightNumber();
//
Manager.Desktop.KeyBoard.TypeText(onwardFlightNumber.ToString(),5);
// System.Threading.Thread.Sleep(2000);

```

```

//
Manager.Desktop.KeyBoard.KeyPress(System.Windows.Forms.Keys.Escape);
    flag = true;
    break;
}
//cell.User.Click();
}
cellCounter++;
}
Log.WriteLine("Current row is : " + rowCounter);
}
rowCounter++;
}

//Pages.MerlotAero1.SilverlightApp.VerticalLargeIncreaseRepeatbutton.UserPress(1174)
;

//Pages.MerlotAero0.SilverlightApp.VerticalLargeIncreaseRepeatbutton1.UserPress(172)
;

Manager.Desktop.KeyBoard.KeyPress(System.Windows.Forms.Keys.PageDown);
    }
}
}
}
}

```

