

Test Automation Using Telerik Test Studio

Script – AircraftFollow – Assign a Registration..tstest.cs

Assign a Registration..tstest.cs

```
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 Assign_a_Registration_ : 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
    }
}
```

```

int columnNumber;
int columnNumberRego;

// Add your test methods here...

[CodedStep(@"New Coded Step")]
public void Depart_and_Arrive_CodedStep2()
{
    System.Threading.Thread.Sleep(5000);
}

[CodedStep(@"Search Flight and update Rego")]
public void Depart_and_Arrive_CodedStep()
{
    bool flag = false;

    //int flightNumber=Utility.getFlightNumber();
    //string newFlightNumber = flightNumber.ToString();

    string newFlightNumber = "3770";           // Shakeel

    Log.WriteLine("ScheduleName:" + newFlightNumber);
    SilverlightApp app = ActiveBrowser.SilverlightApps()[0];
    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("Flight*"))
        {
            Log.WriteLine("Flight header found");
            columnNumber = headCount;
            Log.WriteLine("ColumnNumber : " + columnNumber);
            break;
        }
        headCount++;
    }

    int headCountRego = 1;
    foreach (DataGridColumnHeader header in headers)
    {
        Log.WriteLine("header column are : " + header.Text);
        if (header.Text.Equals("Rego"))
        {
            Log.WriteLine("Rego header found");
            columnNumberRego = headCountRego;
            Log.WriteLine("ColumnNumberRego : " + columnNumberRego);
            break;
        }
        headCountRego++;
    }
}

```

```

while (flag == false)
// for(int i=1;i<18;i++)
{
    DataGridRowsPresenter DGRP =
GR.Find.ByType<DataGridRowsPresenter>();
    DGRP.Refresh();
    System.Threading.Thread.Sleep(2000);
    DGRP.Refresh();
    System.Threading.Thread.Sleep(2000);
    IList<DataRow> rows = DGRP.Find.AllByType<DataRow>();
    foreach (DataRow row in rows)
    {
        int rowCounter = 1;
        //if(rowCounter==1)
        // {
        row.SetFocus();
        //}
        {
            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))
                    {

                        Log.WriteLine("Reached here to select from
Rego");
                        //TextBox TB =
cells[columnNumberATD-2].Find.ByType<TextBox>("ETDTime");
                        //Log.WriteLine("ETD Time is :" + TB.Text);
                        // string etdTime = TB.Text;
                        cells[columnNumberRego - 1].User.Click();
                        System.Threading.Thread.Sleep(1000);
                        cells[columnNumberRego - 1].User.Click();
                        System.Threading.Thread.Sleep(1000);
                        cells[columnNumberRego - 1].User.Click();
                        System.Threading.Thread.Sleep(1000);
                        ComboBox cb = cells[columnNumberRego -
1].Find.ByType<ComboBox>("uxEquipmentRegos");
                        cb.SelectedItemByIndex(true, 0);
                        System.Threading.Thread.Sleep(1000);

Pages.MerlotAero0.SilverlightApp.openSchedule_Image.User.Click();

```

```

        System.Threading.Thread.Sleep(2000);

Pages.MerlotAero0.SilverlightApp.openSchedule_Image.User.Click();
        System.Threading.Thread.Sleep(5000);

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

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

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

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

[CodedStep(@"Handle Equipment Rego change process")]
public void Assign_a_Registration_CodedStep()
{
    try
    {
        Pages.MerlotAero0.SilverlightApp.UxOkButton.User.Click();
        System.Threading.Thread.Sleep(2000);

Pages.MerlotAero0.SilverlightApp.openSchedule_Image.User.Click();
        System.Threading.Thread.Sleep(2000);
    }
    catch (Exception e)
    {
        Log.WriteLine("Equipment Rego change process Doesnot exists" +
e);
    }
}

[CodedStep(@"Verify if Rego is assigned")]
public void Assign_a_Registration_CodedStep1()
{
    bool flag = false;

    //int flightNumber = Utility.getFlightNumber();
    //string newFlightNumber = flightNumber.ToString();

```

```

Pages.MerlotAero0.SilverlightApp.ViewButton.User.Click();
System.Threading.Thread.Sleep(5000);

string newFlightNumber = "3770";    // Shakeel

Log.WriteLine("ScheduleName:" + newFlightNumber);
SilverlightApp app = ActiveBrowser.SilverlightApps()[0];
RadPaneGroup RPG = app.Find<RadPaneGroup>("uxRadPaneGroup");
Grid GR = RPG.Find<Grid>("uxMainGrid");
DataGridColumnHeadersPresenter DGCHP =
GR.Find<DataGridColumnHeadersPresenter>();
int headCount = 1;
IList<DataGridColumnHeader> headers =
DGCHP.Find<DataGridColumnHeader>();
foreach (DataGridColumnHeader header in headers)
{
    Log.WriteLine("header column are : " + header.Text);
    if (header.Text.Equals("Flight*"))
    {
        Log.WriteLine("Flight header found");
        columnNumber = headCount;
        Log.WriteLine("ColumnNumber :" + columnNumber);
        break;
    }
    headCount++;
}

int headCountRego = 1;
foreach (DataGridColumnHeader header in headers)
{
    Log.WriteLine("header column are : " + header.Text);
    if (header.Text.Equals("Rego"))
    {
        Log.WriteLine("Rego header found");
        columnNumberRego = headCountRego;
        Log.WriteLine("ColumnNumberRego :" + columnNumberRego);
        break;
    }
    headCountRego++;
}

while (flag == false)
// for(int i=1;i<18;i++)
{
    DataGridRowsPresenter DGRP =
GR.Find<DataGridRowsPresenter>();
DGRP.Refresh();
System.Threading.Thread.Sleep(2000);
DGRP.Refresh();
System.Threading.Thread.Sleep(2000);
IList<DataGridRow> rows = DGRP.Find<DataGridRow>();
foreach (DataGridRow row in rows)
{
    int rowCounter = 1;
    // if(rowCounter==1)

```

```

        //{
        row.SetFocus();
        // }
        {
            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))
                    {

                        Log.WriteLine("Reached here to select from
Rego");

                        //TextBox TB =
cells[columnNumberRego-1].Find.By<TextBox>("ETDTime");
                        //Log.WriteLine("ETD Time is :" + TB.Text);
                        // string etdTime = TB.Text;
                        cells[columnNumberRego - 1].User.Click();
                        Log.WriteLine("Rego selected is : " +
cells[columnNumberRego - 1].TextBlockContent);
                        if (cells[columnNumberRego -
1].TextBlockContent.Equals(null))
                        {
                            Log.WriteLine("Rego not assigned
successfully");
                        }
                        else
                        {
                            Log.WriteLine("Rego assigned
successfully");

                            // System.Threading.Thread.Sleep(5000);
                            //cells[columnNumberRego-1].User.Click();
                            // System.Threading.Thread.Sleep(5000);
                            // cells[columnNumberRego-1].User.Click();
                            // System.Threading.Thread.Sleep(5000);
                            // ComboBox cb =
cells[columnNumberRego-1].Find.By<ComboBox>("uxEquipmentRegos");
                            // cb.SelectItemByIndex(true,0);
                            // System.Threading.Thread.Sleep(5000);
                            //
Pages.MerlotAero0.SilverlightApp.openSchedule_Image.User.Click();
                            // System.Threading.Thread.Sleep(10000);

```

```

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

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

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

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

```

