

Test Automation Using Telerik Test Studio

Script – CreweFollow – Search_Violation

```
// Search_Violation.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 Search_Violation : 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 columnNumber1;
int columnNumber4;
[CodedStep(@"LeftClick on HeaderButtonRadtogglebutton")]
public void _719_Reevaluate_Violations__CodedStep()
{

if(Pages.MerlotAero0.SilverlightApp.HeaderButtonRadtogglebutton1.IsChecked==true)
{
    Pages.MerlotAero0.SilverlightApp.HeaderButtonRadtogglebutton1.User.Click(ArtOfTest.WebAii.Core.MouseClickType.LeftClick, 1, 22,
    ArtOfTest.Common.OffsetReference.TopLeftCorner,
    ArtOfTest.Common.ActionPointUnitType.Percentage,
    ((System.Windows.Forms.Keys)(0)));
}
}

[CodedStep(@"LeftClick on HeaderButtonRadtogglebutton")]
public void Search_Violation_CodedStep()
{

if(Pages.MerlotAero0.SilverlightApp.HeaderButtonRadtogglebutton2.IsChecked==true)
{
    Pages.MerlotAero0.SilverlightApp.HeaderButtonRadtogglebutton2.User.Click(ArtOfTest.WebAii.Core.MouseClickType.LeftClick, 0, 50,
    ArtOfTest.Common.OffsetReference.TopLeftCorner,
    ArtOfTest.Common.ActionPointUnitType.Percentage,
    ((System.Windows.Forms.Keys)(0)));
}
}

[CodedStep(@"LeftClick on HeaderButtonRadtogglebutton")]
public void Search_Violation_CodedStep1()
{

if(Pages.MerlotAero0.SilverlightApp.HeaderButtonRadtogglebutton.IsChecked==false)
{
    Pages.MerlotAero0.SilverlightApp.HeaderButtonRadtogglebutton.User.Click(ArtOfTest.WebAii.Core.MouseClickType.LeftClick, 1, 38,
    ArtOfTest.Common.OffsetReference.TopLeftCorner,
    ArtOfTest.Common.ActionPointUnitType.Percentage,
    ((System.Windows.Forms.Keys)(0)));
}
}

```

```

}

[CodedStep(@"Extract data from violation tab")]
public void Search_Violation_CodedStep2()
{
    SilverlightApp app = ActiveBrowser.SilverlightApps()[0];
    RadPaneGroup RPG =
app.Find.ByName<RadPaneGroup>("uxGroupPairingExtras");
    ContentControl
cPresenter=RPG.Find.ByName<ContentControl>("uxExceptionsTabRegion");

    RadGridView
rgView=cPresenter.Find.ByName<RadGridView>("uxMonitViolationsGrid");
    GridViewHeaderRow
headersRow=rgView.Find.ByName<GridViewHeaderRow>("PART_HeaderRow");
    IList<GridViewHeaderCell>
headers=headersRow.Find.AllByType<GridViewHeaderCell>();
    int headCount=1;
    int headCount1=1;
    foreach(GridViewHeaderCell header in headers)
    {
        if(header.Text.Equals("Description"))
        {
            //Log.WriteLine("Description header found");
            columnNumber=headCount;
            // Log.WriteLine("ColumnNumber of Description :" +
columnNumber);
            break;
        }
        headCount++;
    }
    foreach(GridViewHeaderCell header in headers)
    {
        if(header.Text.Equals("Employee"))
        {
            //Log.WriteLine("Employee header found");
            columnNumber1=headCount1;
            //Log.WriteLine("ColumnNumber1 of Employee :" +
columnNumber1);
            break;
        }
        headCount1++;
    }
    GridViewVirtualizingPanel
grrp=rgView.Find.ByName<GridViewVirtualizingPanel>("PART_GridViewVirtualizingPa
nel");
    IList<GridViewRow> rows=grrp.Find.AllByType<GridViewRow>();
    //Log.WriteLine("number of rows"+rows.Count);
    int rowCount=1;
    foreach(GridViewRow row in rows)
    {
        int cellCount=1;
        if(rowCount==1)

```

```

    {
        bool hasDate=false;
        DateTime dateTime = new DateTime();
        ItemsPresenter ip=row.Find.ByType<ItemsPresenter>();
        IList<GridViewCell>
cells=ip.Find.AllByType<GridViewCell>();
        //Log.WriteLine("Total cell count is :"+cells.Count);
        foreach(GridViewCell cell in cells)
        {
            //Log.WriteLine("Cell Counter :" + cellCount);
            if(cellCount==columnNumber){

                Grid cellGrid=cell.Find.ByType<Grid>();
                ContentPresenter
cpre=cellGrid.Find.ByName<ContentPresenter>("PART_ContentPresenter");

                // Log.WriteLine("Name of
Subject--:"+cpre.TextBlockContent);

                string senT=cpre.TextBlockContent;

                string[] arrSen = senT.Split(new[] { "expiry"
}, StringSplitOptions.None);
                string allIndex=arrSen[0]+ ' '+arrSen[1];
                string courseName=arrSen[0];
                Log.WriteLine("Course
Name--:"+courseName);//store course name from here
                SetExtractedValue("courseName",courseName);

                string[] arrSenOn = senT.Split(new[] { "on" },
);
                string allIndex2=arrSenOn[0]+ ' '+arrSenOn[1];
                string secndIndexafterOn=arrSenOn[1];
                string []arrSen3=secndIndexafterOn.Split(' ');
                string courseDate=arrSen3[1];

                Log.WriteLine("Date--:"+courseDate); //store
course expiry date from here
                string crewDate=cpre.TextBlockContent;
                string dcrewDate=crewDate.ToString();
                // string
                currentDate=DateTime.Now.ToString("MM/dd/yyyy");

                // DateTime dt= Convert.ToDateTime(crewDate);
                // String
                updateddate=dt.ToString("MM/dd/yyyy");
                // Log.WriteLine("new date
formate--:"+updateddate);
                SetExtractedValue("dcrewDate",dcrewDate);

                System.Threading.Thread.Sleep(5000);
            }

            if(cellCount==columnNumber1){
                Grid cellGrid=cell.Find.ByType<Grid>();

```

```

ContentPresenter
cpre=cellGrid.FindByName<ContentPresenter>("PART_ContentPresenter");

        Log.WriteLine("Employee
Name--:"+cpre.TextBlockContent);
        string Employee=cpre.TextBlockContent;
        //string []splitEmployee=Employee.Split('(');
        // string employeeName=splitEmployee[0];
        string []splitEmployee=Employee.Split(' ');
        string employeeName=splitEmployee[0];
        Log.WriteLine("Crew Name---"+employeeName);
        SetExtractedValue("employeeName",employeeName);
    }
    cellCount++;
}
}
rowCount++;
}

}

[CodedStep(@"radmenuitem: 'Crew.Plan' click action.")]
public void datetest_CodedStep1()
{
    System.Threading.Thread.Sleep(20000);
    SilverlightApp
app=ActiveBrowser.SilverlightApps()[0];
    Telerik.WebAii.Controls.Xaml.RadMenu
rdmenu=app.Find.ByType< Telerik.WebAii.Controls.Xaml.RadMenu>();

    Telerik.WebAii.Controls.Xaml.RadMenuItem
item = Pages.MerlotAero0.SilverlightApp.CrewPlanRadmenuitem;
    Button
bt=item.FindByName<Button>("uxOpenInNewWindow");
    System.Threading.Thread.Sleep(20000);
    bt.User.Click();

    System.Threading.Thread.Sleep(20000);

}

[CodedStep(@"Type '' into TextTextbox28")]
public void Search_Violation_CodedStep3()
{
    System.Threading.Thread.Sleep(10000);
    string crewName=GetExtractedValue("employeeName").ToString();
    Pages.MerlotAero0.SilverlightApp.TextTextbox28.SetText(true,
crewName, 10, 100, false, true);
    Manager.Desktop.KeyBoard.SendString("{ENTER}");

}

[CodedStep(@"Modified Expiry Date")]
public void Search_Violation_CodedStep4()
{

```

```

System.Threading.Thread.Sleep(30000);
int extentHeight;
int rowHeight;
int rowCounter = 1;

RadGridView grid =
Pages.MerlotAero0.SilverlightApp.Course_Expiry_EntityDataGridRadgridview;
FrameworkElement VirtualizingPanel =
grid.Find.ByType("GridViewVirtualizingPanel");
extentHeight = (int)VirtualizingPanel.GetProperty(new
AutomationProperty("ExtentHeight", typeof(int)));
rowHeight = (int)grid.GetProperty(new
AutomationProperty("RowHeight", typeof(int)));
int totalPageRows = 0;
totalPageRows = (extentHeight / rowHeight);
Log.WriteLine("Total Rows : " + totalPageRows.ToString());
String getCourse=Utility.getCourseCode();
//Log.WriteLine(getCourse+"--have no rows to be update ");

extentHeight = (int)VirtualizingPanel.GetProperty(new
AutomationProperty("ExtentHeight", typeof(int)));
rowHeight = (int)grid.GetProperty(new
AutomationProperty("RowHeight", typeof(int)));
totalPageRows = 0;
totalPageRows = (extentHeight / rowHeight);
Log.WriteLine("Total Rows : " + totalPageRows.ToString());

int headCount=1;
IList<GridViewHeaderCell> headers = grid.HeaderRow.HeaderCells;
foreach(GridViewHeaderCell header in headers)
{
    if(header.Text.Equals("Expiry"))
    {
        Log.WriteLine("Expiry header found");
        columnNumber4=headCount;
        Log.WriteLine("ColumnNumber :" + columnNumber4);
        break;
    }
    headCount++;
}
System.Threading.Thread.Sleep(15000);
SilverlightApp app = ActiveBrowser.SilverlightApps()[0]; //Get
Silverlight app

Telerik.WebAii.Controls.Xaml.RadGridView rgv =
app.FindByName<Telerik.WebAii.Controls.Xaml.RadGridView>("uxEntityDataGrid");
//Get RadGrid

Log.WriteLine("Rows count:"+rgv.Rows.Count.ToString());
foreach(Telerik.WebAii.Controls.Xaml.GridViewRow gRow in rgv.Rows){

    Log.WriteLine("<-----Start of row
"+rowCounter.ToString()+"----->");
    if(rowCounter<2)

```

```
        {
            Log.WriteLine("rowCounter :" + rowCounter);
            int cellCounter = 1;
            foreach(Telerik.WebAii.Controls.Xaml.GridViewCell gCell
in gRow.Cells)
            {

                if(cellCounter==columnNumber4)
                {

                    gCell.User.Click();
                    System.Threading.Thread.Sleep(2000);
                    gCell.User.Click();

                    string dateOfCourse=Utility.getNextMonthDate(0);

Pages.MerlotAero0.SilverlightApp.MMDdYyyyMerlotdatepicker.SetText(true,
dateOfCourse, 10, 100);

                    System.Threading.Thread.Sleep(2000);

Manager.Desktop.KeyBoard.KeyPress(System.Windows.Forms.Keys.Escape);
                    System.Threading.Thread.Sleep(2000);
                }
                cellCounter++;
            }
            rowCounter++;
        }
    }
}
```