Find Controls on a Form

Use the API to find controls on a form.

```csharp
using System;
using System.Collections.Generic;
using System.Web.UI;
using PNMsoft.Sequence.Forms.Web.UI;

namespace CustomCodeBehind
{
    ///

    /// Extending the Control element
    ///

    public static class ControlExtender
    {
        public static T FindControl(this Control control, string id) where T :Control
        {
            T result = default(T);

            if (string.Equals(control.ID, id, StringComparison.OrdinalIgnoreCase))
            {
                if (control is T)
                {
                    return (T)control;
                }
                return result;
            }
            try
            {
                foreach (Control c in control.Controls)
                {
                    T ctr = c.FindControl(id);

                    if (ctr != default(T))
                    {
                        return ctr;
                    }

                }
            }
            catch (Exception)
            {
            }


            return result;
        }

        ///

        /// Find all control of type T recursively
        ///

        ///
        ///
        ///
        internal static void FindControl(this Control control, List controls)where T : Control
        {
            if (controls == null) controls = new List();
```

```csharp
            if (controls == null) controls = new List();

            if (control is T)
            {
                controls.Add((T)control);
            }

            foreach (Control c in control.Controls)
            {
                FindControl(c, controls);
            }
        }

        ///
        /// Add a recursively search for contained control by its name
        ///

        /// The container
        /// The control id
        /// The searched control
        public static Control FindControlByIdRecursive(this Control currentCtrl,string controlId)
        {
            if (currentCtrl == null)
                return null;

            Control ctrl = currentCtrl.FindControl(controlId);
            if (ctrl == null)
                foreach (Control subCtrl in currentCtrl.Controls)
                {
                    ctrl = subCtrl.FindControlByIdRecursive(controlId);
                    if (ctrl != null)
                        break;
                }
            return ctrl;
        }

        ///
        /// Get a list of controls from the page based on their type
        ///

        /// The parent control to look in it
        /// The type of the controls to be retrieved.
        /// A list of controls selected by their type
        public static List FindControlsByTypeRecursive(this ControlcurrentCtrl, string controlType
)
        {
            List ctrls = new List();
            Type type = Type.GetType(controlType);
            if (currentCtrl.GetType() == type)
                ctrls.Add(currentCtrl);
            else
                foreach (Control subCtrl in currentCtrl.Controls)
                {
                    ctrls.AddRange(subCtrl.FindControlsByTypeRecursive(controlType));
                }

            return ctrls;
        }

        ///

        /// Returns the parent Form control of a control. In case there is no parent form, returns
```

```csharp
     null
        ///

        /// The current control to start from
        ///
        public static FormControl GetParentFormCtrl(this Control currentCtrl)
        {
            if (currentCtrl == null)
                return null;

            var ctrl = currentCtrl.Parent;
            if (ctrl is FormControl)
                return (ctrl as FormControl);
            return ctrl.GetParentFormCtrl();
        }

        public static T GetParentForm(this Control ctrl)
            where T : Control
        {
            if (!(ctrl is T))
                return GetParentForm(ctrl.Parent);

            return (T)ctrl;
        }


        ///

        /// Find a parent control (container) by its id. Returns null when no parent is found
        ///

        /// The current control to start fro
        /// The parent id to look for
        ///
        public static Control GetParentControlById(this Control currentCtrl, stringcontrolId)
        {
            if (currentCtrl == null)
                return null;

            var ctrl = currentCtrl.Parent;
            return ctrl.ID == controlId ? ctrl : ctrl.GetParentControlById(controlId);
        }
    }
}
```