AJAXによって書き換えられた後のHTMLを取得する

最近のWebアプリケーションではAJAXを使った動的な書き換えを~~ふんだんに~~至る所で当たり前のように使っています。

そのため、単純にURLを指定しHTMLを取得して必要なところを切り出して使う、ということができません。 取得したHTMLはJavaScript実行前のものなので。

何とかJavaScriptの実行結果が反映されたHTMLがほしくてがんばったのがこのプログラムになります。

2014/04/30
主な目的はWebアプリケーションのログイン画面を検査して少なくとも最初の画面が出ていること(曲がりなりにも動いていること)の確認です。
Webアプリの死活監視ですわな。

2014/04/20 更新
WebBrowser.Document プロパティで示す HTMLDocument に更新が入っても WebBrowser.DocumentText プロパティが更新されない環境があるようです。というか私の勤め先の環境がそうだった。
なのでWebBrowser.DocumentText プロパティではなくWebBrowser.Document.Body.OuterHTML プロパティを使う変更を行いました。
あとFormのクラスに居たファイル書き出し関連をProgram.csへ移動。

ダウンロード

hiddenbrowser.zip hiddenBrowser.exe 実行には.NET Framework4.0が必要です。

サンプル

C:\Work に hiddenBrowser.exeをおいて実行した例です。

```
C:\Work>hiddenbrowser http://www.bingco.jp bing.txt
hiddenBrowser http://www.bing.co.jp bing.txt

C:\Work>
```

C:\Work\bing.txtはこんな感じに。

```
今日はイースター！キリスト教圏の人々の春のお祭りです。
華やかに祝おう&amp;nbsp;»
イースターと言えばイースターエッグ。アヒルやニワトリのタマゴをカラフルに飾ります。
その作り方は？&amp;nbsp;»
ふわふわのアヒルのヒナ、小さな体でも動きは敏捷。
```

その敏捷さをいかした？彼らのかわいい遊びをご覧ください »
今日は、世界中の街角でイースターパレードが繰り広げられますが…。
なかでも有名なのはニューヨークのパレードです »Bingすべて表示日本語のページのみを表示ウェブ画像動画地図ニュースショッピングその他MSNOutlook.comログイン Bing.com in English|アヒルのヒナピヨピヨ春だよ！春が来たよ！今日はイースター！かわいいお菓子はいかが？話題の検索ワード皐月賞 · 村上春樹 新作発売 · 飛騨 古川祭 · 女子柔道選手権 · 富士芝桜まつり · 横須賀 カレー対決 · 浅草流鏑馬 2014·NHK ロング・グッドバイ© 2014 Microsoft | プライバシーと Cookies | 使用条件 | 広告掲載 | 広告について | ヘルプ | Bingへのご意見

やってること

やってることは

1. WebBrowserオブジェクトでターゲットページを表示。
2. 最初に指定したURLの内容が表示したときに発生する DocumentCompleted イベントを待つ。
3. DocumentCompleted イベントのタイミングで現在のHTMLを記録する。
4. Timerオブジェクトを使い0.2秒間隔でWebBrowserオブジェクトのHTMLを記録したHTMLと比較する。
5. 比較結果が異なれば、現在のWebBrowserオブジェクトのHTMLを記録する。
6. 2秒間HTMLに変化がなければ、現状のHTMLをファイルに出力する。
7. 30秒経過したらWebBrowserオブジェクトの現在のHTMLをファイルに出力する。

~~HTMLのエンコーディングが取れた場合はそれを使ってEncodingオブジェクトの取得を試み、だめだったらUTF-8決め打ち。~~
既に内部形式なんだからエンコーディングしなくて良いんだったorz

webBrowserオブジェクトはWindowsのフォームに配置された状態じゃないと動作しないので、Formオブジェクトに貼り付けています。 そのため、細工しないとこのFormがウインドウとして表示されてしまいます。

FormのShowIconプロパティ、ShowInTaskberプロパティ、を設定してタスクバーに姿を見せないようにした後、onShownイベントでForm.Hide()を実行しています。 Windows XPがサポートされなくなった今なら、Opacityプロパティを操作して透明なウインドウにしてしまうのも手かしらん。

Windowsフォームなアプリケーションで作って、最後にプロジェクトのプロパティにある『出力の種類』を「コンソールアプリケーション」に変更してあげればコマンドプロンプトで動作するプログラムに早変わり。

ソース

- hiddenBrowserForm.cs はWebBrowserオブジェクトを貼り付けたFormの定義。
- hiddenBrowserForm.Designer.cs はWebBrowserオブジェクト、Formオブジェクトのプロパティに関しての初期設定、VisualStudioさんが生成してくれる奴。
- Program.cs はプログラムのエントリポイントでFormの起動。

hiddenBrowserForm.cs

```
using System;
```

```
using System.Drawing;
using System.Windows.Forms;

namespace hiddenBrowser
{
    public partial class hiddenBrowserForm : Form
    {
        Uri Url;
        bool firstFlag = false;
        const int maxCount = 10;
        int updateCount = maxCount;

        public String firstString { get; set; }

        public hiddenBrowserForm(String url)
        {
            InitializeComponent();
            firstString = "";
            Url = new Uri(url);
            Hide();
        }

        private void Form1_Shown(object sender, EventArgs e)
        {
            webBrowser1.ScriptErrorsSuppressed = true;
            webBrowser1.Url = Url;
            firstFlag = true;
            timer2.Enabled = true;
        }

        private void webBrowser1_DocumentCompleted(object sender,
WebBrowserDocumentCompletedEventArgs e)
        {
            if (true == firstFlag)
            {
                firstString = webBrowser1.Document.Body.OuterHtml;
                timer1.Enabled = true;
            }
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            if ((null != webBrowser1.Document) && (null !=
webBrowser1.Document.Body))
            {
                String s = webBrowser1.Document.Body.OuterHtml;

                updateCount--;

                if (firstString != s)
                {
```

```
                    firstString = s;
                    updateCount = maxCount;

                }
            }

            if (0 == updateCount)
            {
                Close();
            }
        }

        private void timer2_Tick(object sender, EventArgs e)
        {
            if ((null != webBrowser1.Document) && (null !=
webBrowser1.Document.Body))
            {
                firstString = webBrowser1.Document.Body.OuterHtml;
            }
            Close();
        }

    }
}
```

hiddenBrowserForm.Designer.cs

```
namespace hiddenBrowser
{
    partial class hiddenBrowserForm
    {
        /// <summary>
        /// 必要なデザイナー変数です。
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// 使用中のリソースをすべてクリーンアップします。
        /// </summary>
        /// <param name="disposing">マネージ リソースが破棄される場合 true、破
棄されない場合は false です。</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
```

```
        #region Windows フォーム デザイナーで生成されたコード

        /// <summary>
        /// デザイナー サポートに必要なメソッドです。このメソッドの内容を
        /// コード エディターで変更しないでください。
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            this.webBrowser1 = new System.Windows.Forms.WebBrowser();
            this.timer1 = new
System.Windows.Forms.Timer(this.components);
            this.timer2 = new
System.Windows.Forms.Timer(this.components);
            this.SuspendLayout();
            //
            // webBrowser1
            //
            this.webBrowser1.Dock =
System.Windows.Forms.DockStyle.Fill;
            this.webBrowser1.Location = new System.Drawing.Point(0, 0);
            this.webBrowser1.MinimumSize = new System.Drawing.Size(20,
20);
            this.webBrowser1.Name = "webBrowser1";
            this.webBrowser1.Size = new System.Drawing.Size(284, 262);
            this.webBrowser1.TabIndex = 0;
            this.webBrowser1.DocumentCompleted += new
System.Windows.Forms.WebBrowserDocumentCompletedEventHandler(this.webBr
owser1_DocumentCompleted);
            //
            // timer1
            //
            this.timer1.Interval = 200;
            this.timer1.Tick += new
System.EventHandler(this.timer1_Tick);
            //
            // timer2
            //
            this.timer2.Interval = 30000;
            this.timer2.Tick += new
System.EventHandler(this.timer2_Tick);
            //
            // hiddenBrowserForm
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F,
12F);
            this.AutoScaleMode =
System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(284, 262);
            this.Controls.Add(this.webBrowser1);
            this.Name = "hiddenBrowserForm";
```

```
            this.ShowIcon = false;
            this.ShowInTaskbar = false;
            this.Text = "hiddenBrowser";
            this.WindowState =
System.Windows.Forms.FormWindowState.Minimized;
            this.Shown += new System.EventHandler(this.Form1_Shown);
            this.ResumeLayout(false);

        }

        #endregion

        private System.Windows.Forms.WebBrowser webBrowser1;
        private System.Windows.Forms.Timer timer1;
        private System.Windows.Forms.Timer timer2;
    }
}
```

Program.cs

```
using System;
using System.Windows.Forms;
using System.IO;
using System.Text;
using System.Text.RegularExpressions;

namespace hiddenBrowser
{
    static class Program
    {
        /// <summary>
        /// アプリケーションのメイン エントリ ポイントです。
        /// </summary>
        [STAThread]
        static void Main(String[] arg)
        {
            if (arg.Length != 2)
            {
                Console.WriteLine("usage:\n   hiddenBrowser URL
fileName");
                Application.Exit();
            }
            else
            {
                Console.WriteLine("hiddenBrowser {0}
{1}",arg[0],arg[1]);
                Application.EnableVisualStyles();
                Application.SetCompatibleTextRenderingDefault(false);
                hiddenBrowserForm f = new hiddenBrowserForm(arg[0]);
                Application.Run(f);
```

```
                outputContent(f.firstString, arg[1]);

            }
        }

        static public void outputContent(String firstString, String
FileName)
        {
            try
            {
                Regex regCrlf = new Regex(@"(\r\n|\r|\n)");
                Regex regTagStyle = new
Regex(@"<[sS][tT][yY][lL][eE].*?>.*?</[sS][tT][yY][lL][eE]>",
RegexOptions.Singleline);
                Regex regTagScript = new
Regex(@"<[sS][cC][rR][iI][pP][tT].*?>.*?</[sS][cC][rR][iI][pP][tT]>",
RegexOptions.Singleline);
                Regex regTagNoScript = new
Regex(@"<[nN][oO][sS][cC][rR][iI][pP][tT].*?>.*?</[nN][oO][sS][cC][rR][
iI][pP][tT]>", RegexOptions.Singleline);
                Regex regTagBr = new Regex(@"<[bB][rR].*?/*>");
                Regex regTagP = new Regex(@"</*[pP].*?/*>");
                Regex regTags = new Regex(@"<.+?>",
RegexOptions.Singleline);
                Regex regBlanks = new Regex(@"(□| |\t| ){1,}\n",
RegexOptions.Singleline);
                Regex regCrlf2 = new Regex(@"\n{2,}",
RegexOptions.Singleline);
                Regex regCrlf3 = new Regex(@"^\n{1,}",
RegexOptions.Singleline);

                String ans = firstString;

                ans = regCrlf.Replace(ans, "\n");
                ans = regTagStyle.Replace(ans, "");
                ans = regTagScript.Replace(ans, "");
                ans = regTagNoScript.Replace(ans, "");
                ans = regTagBr.Replace(ans, "\n");
                ans = regTagP.Replace(ans, "\n");
                ans = regTags.Replace(ans, "");
                ans = regBlanks.Replace(ans, "\n");
                ans = regCrlf2.Replace(ans, "\n");
                ans = regCrlf3.Replace(ans, "");


                FileStream fs = new FileStream(FileName,
FileMode.Create);
                Encoding ec = Encoding.UTF8;

                byte[] bf = ec.GetBytes(ans.ToCharArray());
```

```
                fs.Write(bf, 0, bf.Length);
                fs.Flush();
                fs.Close();
            }
            catch (Exception ee)
            {
                Console.WriteLine(ee.Message);
            }
        }
    }
}
```

Windows, www, IE, tool