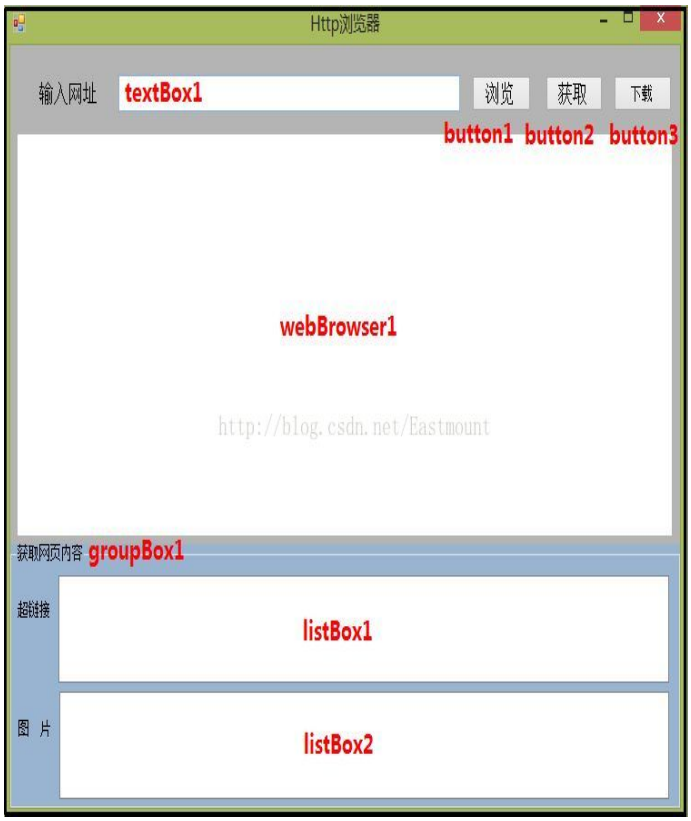
C# 网络编程之webBrowser获取网页url和下载网页中图片

该文章主要是通过C#网络编程的webBrowser获取网页中的url并简单的尝试下载网页中的图片,主要是为以后网络开发的基础学习.其中主要的通过应用程序结合网页知识、正则表达式实现浏览、获取url、下载图片三个功能.而且很清晰的解析了每一步都是以前一步为基础实现的.

一. 界面设计

界面设计如下图所示,添加控件如图,设置webBrowser1其Anchor属性为Top、Bottom、Left、Right,实现对话框缩放;设置groupBox1其Dock(定义要绑到容器控件的边框)为Buttom,实现当浏览器缩放时groupBox1始终在最下边;设置listBox其HorizontalScrollbar属性为True,显示水平滚动条.



二. 源代码

1.命名空间

```
//新添加命名空间
using System.Net;
using System.IO;
using System.Text.RegularExpressions;  //正则表达式
```

2.浏览

点击"浏览"按钮,生成button1_Click(object sender, EventArgs e)点击事件中添加如下代码,实现浏览网页:

```
private void button1_Click(object sender, EventArgs e)
{
    webBrowser1.Navigate(textBox1.Text.Trim());         //显示网页
}
```

调用webBrowser的Navigate方法将指定位置的文档加载到控件中,其中一种重载方法Navigate(urlString)将制定的统一资源定位符URL处的文档加载到WebBrowser控件中替换上一个文档.

3.获取

点击"获取"按钮,生成button2_Click(object sender, EventArgs e)点击事件中添加如下代码,通过获取"html.OuterHtml"当前网页的HTML内容,利用正则表达式获取网页中所有内容的URL超链接和图片的URL,并显示在listBox控件中.

```
<strong>//定义num记录listBox2中获取到的图片URL个数
public int num = 0;
//点击"获取"按钮
private void button2_Click(object sender, EventArgs e)
{
    HtmlElement html = webBrowser1.Document.Body;       //定义HTML元素
    string str = html.OuterHtml;                        //获取当前元素的HTML代码
    MatchCollection matches;                            //定义正则表达式匹配集合
    //清空
    listBox1.Items.Clear();
    listBox2.Items.Clear();
    //获取
    try
    {
        //正则表达式获取<a href></a>内容url
        matches = Regex.Matches(str, "<a href=\"([^\"]*?)\".*?>(.*?)</a>", RegexOptions.IgnoreCase);
        foreach (Match match in matches)
        {
            listBox1.Items.Add(match.Value.ToString());
        }
        //正则表达式获取<img src=>图片url
        matches = Regex.Matches(str, @"<img\b[^<>]*?\bsrc[\s\t\r\n]*=[\s\t\r\n]*[""']?[\s\t\r\n]*(?<imgUrl>
[^\s\t\r\n""'<>]*)[^<>]*?/?[\s\t\r\n]*>", RegexOptions.IgnoreCase);
        foreach (Match match in matches)         {
            listBox2.Items.Add(match.Value.ToString());
        }
        //记录图片总数
        num = listBox2.Items.Count;
    }
    catch (Exception msg)
    {
        MessageBox.Show(msg.Message);    //异常处理
    }
}</strong>
```

其中MatchCollection Regex.Matches(string input,string pattern,RegexOption options)表示使用指定的匹配选项pattern在输入的字符串中搜索指定正则表达式的所有结果.上面RegexOptions.IgnoreCase表示不区分大小写匹配.因为下载中我会显示下载成功结果到listBox2中,所以这里使用num先计算图片总数.

4.下载

在"获取"中我们已经获取到了所有网页内容的URL和图片的URL,这里想要下载图片,但它的格式通常是:“<img src="http://www.baidu.com/img/bdlogo.gif"width="270" height="129">”所以这里只需要获取src中的内容实现访问该图片,在调用文件相关知识实现简单下载图片.而获取src中

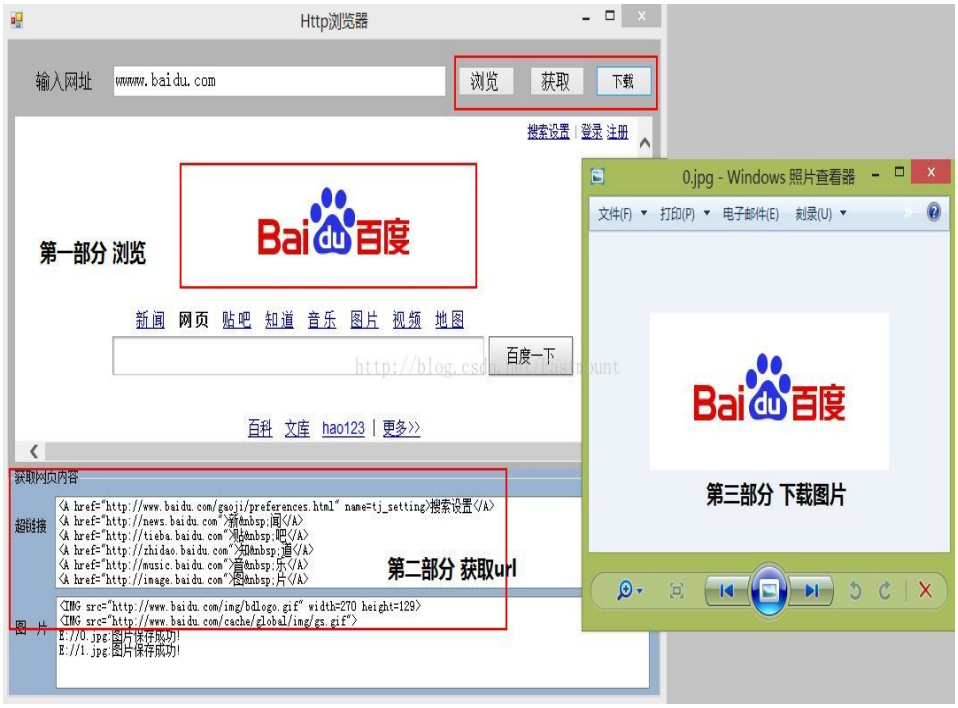
的值很显然也是通过正则表达式获取的.代码如下:

```
//点击"下载"实现下载图片
private void button3_Click(object sender, EventArgs e)
{
    string imgsrc = string.Empty;            //定义
    //循环下载
    for (int j = 0; j < num; j++)
    {
        string content = listBox2.Items[j].ToString();    //获取图片url
        Regex reg = new Regex(@"<img.*?src=""(?<src>[^""]*)""[^>]*>", RegexOptions.IgnoreCase);
        MatchCollection mc = reg.Matches(content);        //设定要查找的字符串
        foreach (Match m in mc)
        {
            try
            {
                WebRequest request = WebRequest.Create(m.Groups["src"].Value);//图片src内容
                WebResponse response = request.GetResponse();
                //文件流获取图片操作
                Stream reader = response.GetResponseStream();
                string path = "E://" + j.ToString() + ".jpg";        //图片路径命名
                FileStream writer = new FileStream(path, FileMode.OpenOrCreate, FileAccess.Write);
                byte[] buff = new byte[512];
                int c = 0;                                           //实际读取的字节数
                while ((c = reader.Read(buff, 0, buff.Length)) > 0)
                {
                    writer.Write(buff, 0, c);
                }
                //释放资源
                writer.Close();
                writer.Dispose();
                reader.Close();
                reader.Dispose();
                response.Close();
                //下载成功
                listBox2.Items.Add(path + ":图片保存成功!");
            }
            catch (Exception msg)
            {
                MessageBox.Show(msg.Message);
            }
        }
    }
}
```

该部分代码可能存在几个问题:
(1).获取图片格式不一定是jpg格式,这里主要想展示一种思想,具体的不同图片获取设置一下即可;
(2).采用该文件流的方法下载速度很慢,可以采用其他方法,WebClient.DownloadFile()等,因为我刚好研究了文件知识和网络爬虫,所以就采用了此基础方法;
(3).代码中的两层循环有点多余,但MatchCollection mc获取的是匹配集合,总体感觉此段还是有点乱;
(4).如果想批量下载图片,最好使用上线程等知识,同时采用一些优秀的算法(强调是算法),内存中获取,该程序只是基础知识.

三. 运行结果

运行结果如下图所示:点击"浏览"按钮可以实现浏览网页,点击"获取"可以获取网页的URL并显示在listBox控件中,最后点击"下载"把图片保存到E盘目录下,下面就是浏览百度时下载的logo图标.(如果图片没有源URL路径,需要自己去实现,如<img src="/sites/default/files/bitren_logo.gif">)



四. 网页基础知识

这里主要介绍HTML网页制作中的超链接和图片链接的基础知识,更好的方便大家理解这篇文章.(参考赵丰年的《网页制作教程》)

1.页面链接

网页中创建超链接需要使用A标记符,结束标记符为</A>.它的最基本属性是href,用于指定超链接的目标,通过href属性指定不同的值,可以创建不同类型的超链接.同时<A>和</A>之间可以用单击对象作为超链接的源(文字或图片).

如百度首页中的:“新 闻”.(锚点连接这里就不介绍)

2.插入图片

在HTML中使用IMG标记符向网页中插入图片,它的两个必要基本属性是src和alt.分别用于设置图像文件的位置和替换文本.

(1).src属性表示要插入图像的文件名,必须包含绝对路径或相对路径.

(2).alt属性表示图像的简单文本说明,用于不能显示图像的浏览器或显示时间过长时先替换显示.

如百度首页的logo图标图片“<img src="http://www.baidu.com/img/bdlogo.gif" width="270" height="129">”当直接访问该url时能访问图片,我们上面的程序主要就是通过这种方式下载网页中的图片的.如下图:

五. 正则表达式

正则表达式(Regular Expression)就是一个字符构成的串,它定义了一个用来搜索匹配字符串的模式.许多语言包括Perl、PHP、Python、JavaScript和JScript,都支持用正则表达式处理文本,一些文本编辑器用正则表达式实现高级“搜索-替换”功能.我所接触到的正则表达式一个是用户名密码设置和该网页知识中,所以我也还需要去学习该部分知识.这里主要用到3个正则表达式,其中下面两个代码非常有用:

1.获取HTML中所有图片的URL

(参考:http://blog.csdn.net/smeller/article/details/7108502)

```
/// <summary>
/// 取得HTML中所有图片的 URL
/// </summary>
/// <param name="sHtmlText">HTML代码</param>
/// <returns>图片的URL列表</returns>
public static string[] GetHtmlImageUrlList(string sHtmlText)
{
    // 定义正则表达式用来匹配 img 标签
    Regex regImg = new Regex(@"<img\b[^<>]*?\bsrc[\s\t\r\n]*=[\s\t\r\n]*[""']?[\s\t\r\n]*(?<imgUrl>[^\s\t
\r\n""'<>]*)[^<>]*?/?[\s\t\r\n]*>", RegexOptions.IgnoreCase);
    // 搜索匹配的字符串       MatchCollection matches = regImg.Matches(sHtmlText);
    int i = 0;
    string[] sUrlList = new string[matches.Count];
    // 取得匹配项列表
    foreach (Match match in matches)
    {
        sUrlList[i++] = match.Groups["imgUrl"].Value;
    }
    return sUrlList;
```

```
}
```

2.获得图片的src路径并保存

(参考:http://bbs.csdn.net/topics/320001867)

```
/// <summary>
/// 获得图片的路径并存放
/// </summary>
/// <param name="M_Content">要检索的内容</param>
/// <returns>IList</returns>
public static IList<string> GetPicPath(string M_Content)
{
    IList<string> im = new List<string>();//定义一个泛型字符类
    Regex reg = new Regex(@"<img.*?src=""(?<src>[^""]*)""[^>]*>", RegexOptions.IgnoreCase);
    MatchCollection mc = reg.Matches(M_Content); //设定要查找的字符串
    foreach (Match m in mc)
    {
        im.Add(m.Groups["src"].Value);
    }
    return im;
}
```

六. 总结

该文章主要是做C#网络知识中关于网络爬虫获取URL和简单下载图片的基础讲解,很清晰的讲述了首先要获取URL就需要浏览网页,至少要获取网页HTML内容,在通过简单的正则表达式获取<A href></A>内容;如果要下载图片就要获取图片的URL<img src="">获取src的网址,在下载该网址中的图片,获取方法还是使用正则表达式,下载方法可以使用很多,这里采用的是文件流,最好使用多线程等批量下载手段.

(免费下载地址:http://download.csdn.net/detail/eastmount/6355125)

主要通过该文件介绍一些基本的网络知识,同时我也在不断的学习研究,同时讲解正则表达式和网页基本的两个概念知识.最后感谢文章中那个网址的博主及一些人,希望该文章能够对大家有所帮助,同时如果文章中有错误或不足之处,还请大家海涵.

(By:Eastmount 2013-10-5 夜2点http://blog.csdn.net/eastmount)