

C# 网络编程之网页简单下载实现

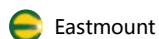
原创 Eastmount 最后发布于2013-07-30 16:29:34 阅读数 6190 ☆ 收藏

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法...



¥9.90

去订阅

这是根据《C#网络编程实例教程》中学到的知识实现的一个C#网页简单下载器，其中涉及到的知识主要是HTTP协议编程中相关类：[HttpWebRequest类](#)、[HttpWebResponse类](#)、[WebRequest类](#)、[WebResponse类](#)、[Uri类](#)、[WebClient类](#)。通过它们相应的方法实现,这里涉及到了4个新接触的知识点我想重点讲述：

(1).ComboBox控件：这是一个下拉列表的可编辑的文本框，右键该控件在显示的"编辑项"中添加网址，可见实例中的下拉控件；

(2).saveFileDialog控件：这是一个保存文件位置对话框，通过该控件可以保存文本文件在自定义位置，还有一个相对应的打开控件OpenFileDialog；

(3).HttpWebResponse类属性：该类常用属性如下：每个类都有相应的属性，这里只详细说明HttpWebResponse类的属性：

CharacterSet [Ⓜ]	获取响应的字符集 [Ⓜ]
ContentEncoding [Ⓜ]	获取用于对响应体进行编码的方法 [Ⓜ]
ContentLength [Ⓜ]	获取请求返回的内容长度 [Ⓜ]
contentType [Ⓜ]	获取响应的内容类型 [Ⓜ]
Cookies [Ⓜ]	获取或设置与此请求关联的 Cookie [Ⓜ]
Headers [Ⓜ]	获取与来自服务器的响关联的标头 [Ⓜ]
LastModified [Ⓜ]	获取最后一次修改响应内容的日期和时间 [Ⓜ]
Method [Ⓜ]	获取用于返回响应的方法 [Ⓜ]
ProtocolVersion [Ⓜ]	获取响应中使用的 HTTP 协议的版本 [Ⓜ]
ResponseUri [Ⓜ]	获取响应请求的 Internet 资源的 URI [Ⓜ]
Server [Ⓜ]	获取发送响应的服务器的名称 [Ⓜ]
StatusCode [Ⓜ]	获取响应的状态 [Ⓜ]
StatusDescription [Ⓜ]	获取与响应一起返回的状态说明 [Ⓜ]

在下列实例中我使用的是HttpWebResponse response.ContentLength获取应答内容长度。

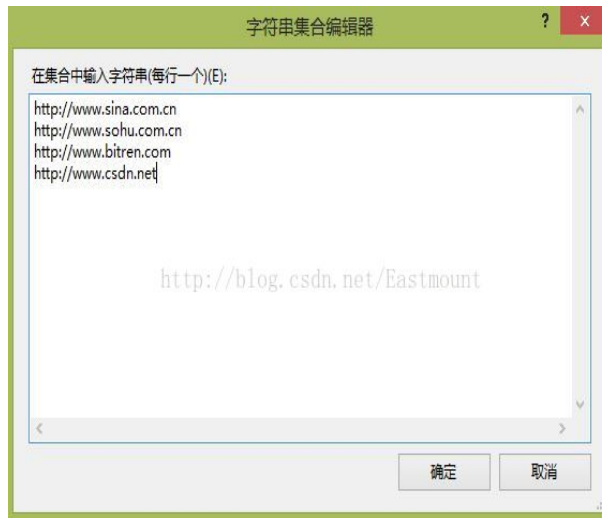
(4).try..catch..finally：用于处理异常情况，try块中使用引发异常语句，try块中发生异常后跳转到catch异常处理程序，catch用于异常处理程序，即使引发异常，finally块中的代码也会执行，从程序中释放资源。

一.界面设计



添加saveFileDialog控件，该控件的Filter(文件筛选器)属性设置为"Web文件|*.htm"

添加一个ComboBox控件设置Text属性为 "http://www.baidu.com"，右键该控件Items属性添加如下几条网址：



二.源代码

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

// 添加命名空间
using System.Net;
using System.IO;

namespace Http_Web_Request
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        // 双击"开始下载" 添加Click事件
        private void button1_Click(object sender, EventArgs e)
        {
            string uri = comboBox1.Text;

            //HttpWebRequest对象实例: 该类用于获取和操作HTTP请求 创建WebRequest对象
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(uri);

            //HttpWebResponse对象实例: 该类用于获取和操作HTTP应答
            HttpWebResponse response = (HttpWebResponse)request.GetResponse(); // 获取答复

            // 构造数据流、文件流对象实例
            Stream stream = null;
            FileStream filestream = null;

            try
            {
                // 获取应答流赋值
```

```

        stream = response.GetResponseStream();
        // 确定应答内容的大小
        long size = response.ContentLength;
        // 定义"另存为对话框"文件类型
        saveFileDialog1.Filter = "网页文件(*.htm)|*.htm|ASP.NET文件(*.aspx)|*.aspx";

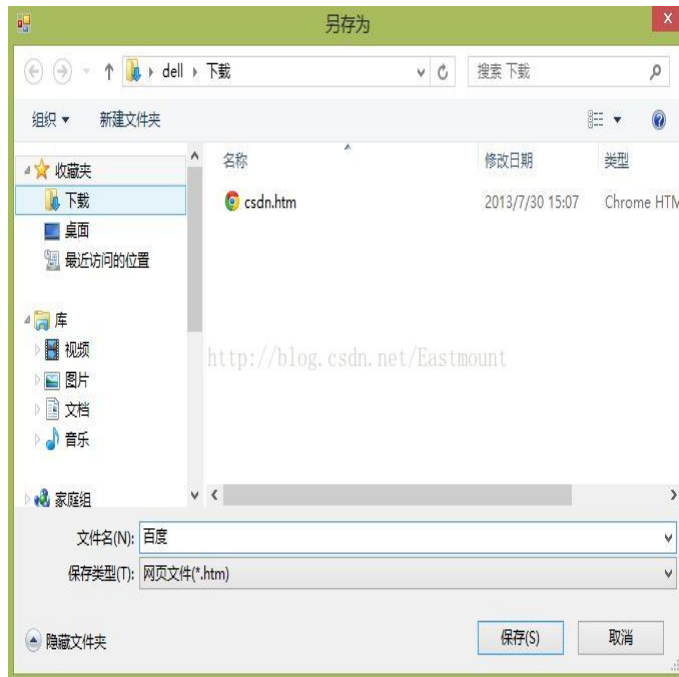
        // 打开保存对话框 点击"确定"按钮执行
        if (saveFileDialog1.ShowDialog() == DialogResult.OK)
        {
            // 为文件流对象实例赋值
            filestream = new FileStream(saveFileDialog1.FileName, FileMode.OpenOrCreate,
FileAccess.Write);
            int length = 1024;
            byte[] Num = new byte[1025]; // 构造数组
            int i = 0;

            while ((i = stream.Read(Num, 0, length)) > 0) // 流不为空写流
            {
                filestream.Write(Num, 0, i); // 写流
            }
        }
        catch (Exception msg)
        {
            MessageBox.Show(msg.Message); // 异常处理
        }
        finally
        {
            stream.Close(); // 关闭流
            filestream.Close();
        }
    }
}
}

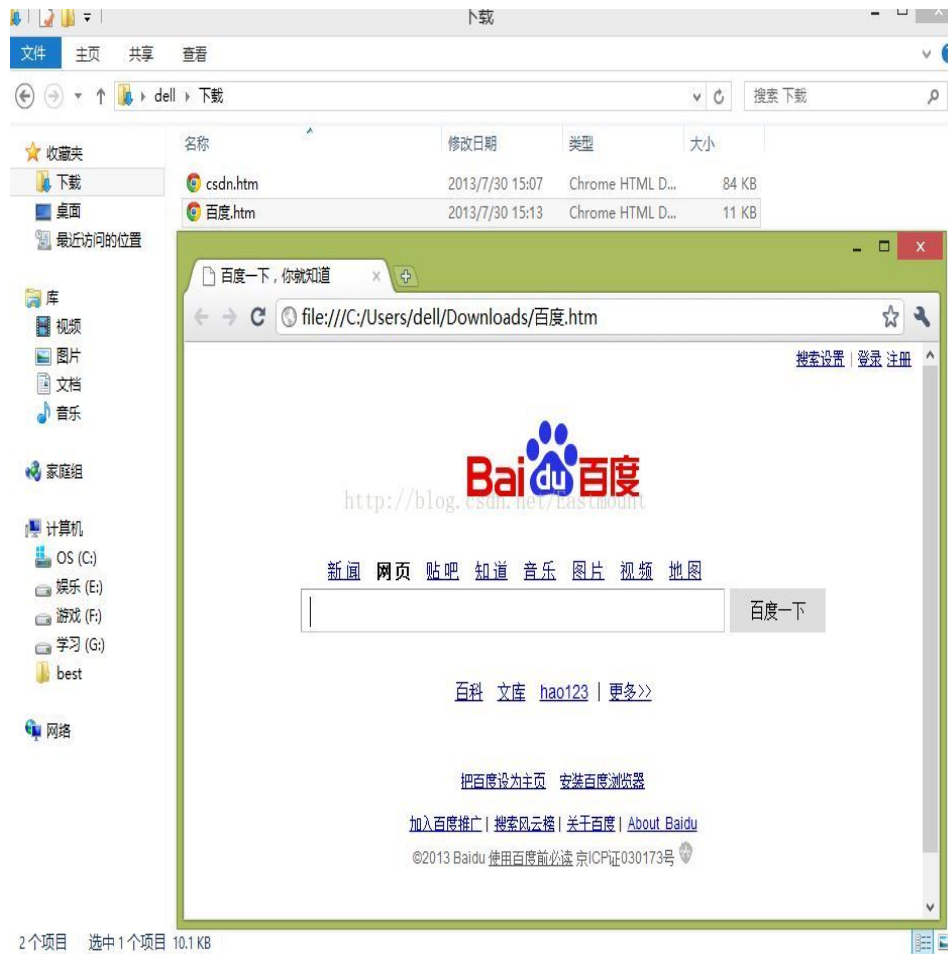
```

三.运行结果

运行程序在“下拉”菜单中选择“百度”，点击“开始下载”后弹出的保存对话框中选择下载地址自定义命名保存，如下图所示：



保存后显示下载后的静态网址如下图所示：



同时也在保存有的静态网页时会产生乱码，如csdn网站，它下载的内容不像我们说看到的正常形式保存的。

四.总结

通过这个例子关于Http网络编程知识我想主要表达saveFileDialog控件的使用、ComboBox下拉控件的使用、类中常

用的属性的熟悉及了解、try.catch.finally语法的应用，感谢我看的书籍作者、一些博客及自己，这篇文章加深了我自己对相关知识的了解，同时也希望大家！有错误或不足之处见谅！

(By:Eastmount 7-30-下午4点 <http://blog.csdn.net/eastmount/>)

👍 点赞 3 ☆ 收藏 ➦ 分享 ...



Eastmount 博客专家

发布了446 篇原创文章 · 获赞 6052 · 访问量 489万+

他的留言板

关注