# C#爬虫（01）：HttpClient网络HTTP请求和相应

## 目录

---

HttpClient 类 (System.Net.Http) | Microsoft Docs

C# HttpClient设置cookies的两种办法 - 深入学习ing - 博客园 (cnblogs.com)

C# 使用HttpClient获取cookie_Stay Hungry-CSDN博客

## 一、概述

Net4.5以上的提供基本类，用于发送 HTTP 请求和接收来自通过 URI 确认的资源的 HTTP 响应。

HttpClient是一个高级 API，用于包装其运行的每个平台上可用的较低级别功能。

```
// HttpClient is intended to be instantiated once per application, rather than
static readonly HttpClient client = new HttpClient();
```

09 BootStrap(33)
12 C#语言基础(130)
13 Visual Studio(40)
15 WinForm(38)
16 ASP.NET(26)
17 ASP.NET Core(19)
18 EF Core(10)
19 .NET设计模式(29)
20 三方控件(65)
21 Oracle(32)
22 SqlServer(55)
23 MongoDB(11)
30 ABP(54)
31 Python(91)
40 GitHub项目(1)
50 Office(19)
51 SoftWare分享(43)
53 Linux系统(24)
55 非技术(22)
88 微信开发(12)
99 English Learning(10)

**随笔档案** (891)

2022年5月(22)
2022年4月(2)
2022年2月(5)
2021年11月(4)
2021年9月(1)
2021年8月(2)
2021年7月(9)
2021年6月(3)
2021年5月(2)
2021年4月(1)
2021年3月(2)
2021年1月(1)
2020年12月(6)
2020年11月(8)

```
static async Task Main()
{
  // Call asynchronous network methods in a try/catch block to handle exception
  try
  {
     HttpResponseMessage response = await client.GetAsync("http://www.contoso.c
     response.EnsureSuccessStatusCode();
     string responseBody = await response.Content.ReadAsStringAsync();
     // Above three lines can be replaced with new helper method below
     // string responseBody = await client.GetStringAsync(uri);

     Console.WriteLine(responseBody);
  }
  catch(HttpRequestException e)
  {
     Console.WriteLine("\nException Caught!");
     Console.WriteLine("Message :{0} ",e.Message);
  }
}
```

目录
导航

## 二、HttpClient的使用

1.使用HttpClient调用Oauth的授权接口获取access_token

1）OAuth使用的密码式

2）获取到access_token后才进行下一步

2.带着access_token调用接口

1）hearder上添加bearer方式的access_token

2）调用接口确保成功获取到返回的结果

2020年10月(61)
2020年9月(22)
2020年8月(6)
2020年7月(39)
2020年6月(80)
2020年5月(54)
2020年4月(9)
2020年3月(40)
2020年2月(27)
2020年1月(55)
2019年12月(32)
2019年11月(33)
2019年10月(30)
2019年9月(17)
2019年8月(27)
2019年7月(8)
2019年6月(5)
2019年3月(25)
2019年2月(18)
2019年1月(24)
2018年12月(14)
2018年11月(5)
2018年10月(23)
2018年9月(9)
2018年8月(118)
2018年7月(34)
更多

**文章档案** (1)

2010年5月(1)

**工具网站**

在线正则表达式测试工具
csdn博客
DevExpress中文帮助文档
w3cschool.cn
菜鸟教程
在线正则表达式测试器

```
try
{
    string host = ConfigurationManager.AppSettings["api_host"];
    string username = ConfigurationManager.AppSettings["api_username"];
    string password = ConfigurationManager.AppSettings["api_password"];

    HttpClient httpClient = new HttpClient();

    // 设置请求头信息
    httpClient.DefaultRequestHeaders.Add("Host", host);
    httpClient.DefaultRequestHeaders.Add("Method", "Post");
    httpClient.DefaultRequestHeaders.Add("KeepAlive", "false");   // HTTP KeepA
    httpClient.DefaultRequestHeaders.Add("UserAgent","Mozilla/5.0 (Windows NT 6

    //获取token
    var tokenResponse = httpClient.PostAsync("http://" + host + "/token", new F
                {"grant_type","password"},
                {"username", username},
                {"password", password}
            }));
    tokenResponse.Wait();
    tokenResponse.Result.EnsureSuccessStatusCode();
    var tokenRes = tokenResponse.Result.Content.ReadAsStringAsync();
    tokenRes.Wait();
    var token = Newtonsoft.Json.Linq.JObject.Parse(tokenRes.Result);
    var access_token = token["access_token"].ToString();

    // 调用接口发起POST请求
    var authenticationHeaderValue = new AuthenticationHeaderValue("bearer", acc
    httpClient.DefaultRequestHeaders.Authorization = authenticationHeaderValue;
```

```
var content = new StringContent(parameter);
    content.Headers.ContentType = new MediaTypeHeaderValue("application/json");
    var response = httpClient.PostAsync("http://" + host + "/" + api_address, c

    response.Wait();
    response.Result.EnsureSuccessStatusCode();
    var res = response.Result.Content.ReadAsStringAsync();
    res.Wait();return Newtonsoft.Json.JsonConvert.DeserializeObject(res.Result)
```

目录导航

w3school在线教程(老)
我的码云gitee
我的GitHub（慢）

目录
导航

```
}
catch (Exception ex)
{

    return ResultEx.Init(ex.Message);
}
```

## HttpClient 获取图片并保存到本机

```
class Program
{
    static void Main()
    {
        //图片路径：https://img.infinitynewtab.com/wallpaper/1.jpg
        string imgSourceURL = "https://img.infinitynewtab.com/wallpaper/";
        DownloadImags(imgSourceURL).Wait();
    }
    private static async Task DownloadImags(string url)
    {
        var client = new HttpClient();
        System.IO.FileStream fs;
        int a = 1;
        //文件名：序号+.jpg。可指定范围，以下是获取100.jpg~500.jpg.
        for (int i = 100; i <= 500; i++)
        {
            var uri = new Uri(Uri.EscapeUriString(url+i.ToString()+".jpg"));
            byte[] urlContents = await client.GetByteArrayAsync(uri);
            fs = new System.IO.FileStream(AppDomain.CurrentDomain.BaseDirectory
            fs.Write(urlContents, 0, urlContents.Length);
            Console.WriteLine(a++);
        }
    }
}
```

## 以下为封装的类库

```
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Text;
using System.Threading.Tasks;
using System.Xml.Serialization;


public class HttpClientHelpClass
{
    ///
    /// get请求
    ///
    ///
    ///
    public static string GetResponse(string url, out string statusCode)
    {
        if (url.StartsWith("https"))
            System.Net.ServicePointManager.SecurityProtocol = SecurityProtocolT

        var httpClient = new HttpClient();
        httpClient.DefaultRequestHeaders.Accept.Add(      new MediaTypeWithQual
        HttpResponseMessage response = httpClient.GetAsync(url).Result;
        statusCode = response.StatusCode.ToString();
        if (response.IsSuccessStatusCode)
        {
            string result = response.Content.ReadAsStringAsync().Result;
            return result;
        }
        return null;
    }

    public static string RestfulGet(string url)
    {
        HttpWebRequest request = WebRequest.Create(url) as HttpWebRequest;
        // Get response
        using (HttpWebResponse response = request.GetResponse() as HttpWebRespo
        {
            // Get the response stream
            StreamReader reader = new StreamReader(response.GetResponseStream()
            // Console application output
            return reader.ReadToEnd();
        }
    }

    public static T GetResponse(string url)
       where T : class, new()
    {
        if (url.StartsWith("https"))
            System.Net.ServicePointManager.SecurityProtocol = SecurityProtocolT

       var httpClient = new HttpClient();
        httpClient.DefaultRequestHeaders.Accept.Add(   new MediaTypeWithQuality
        HttpResponseMessage response = httpClient.GetAsync(url).Result;

        T result = default(T);

        if (response.IsSuccessStatusCode)
        {
            Task<string> t = response.Content.ReadAsStringAsync();
            string s = t.Result;

            result = JsonConvert.DeserializeObject(s);
        }
        return result;
    }

    ///
    /// post请求
    ///
    ///
    /// post数据
    ///
    public static string PostResponse(string url, string postData, out string s
```

目录导航

```
{
    if (url.StartsWith("https"))
        System.Net.ServicePointManager.SecurityProtocol = SecurityProtocolT

    HttpContent httpContent = new StringContent(postData);
    httpContent.Headers.ContentType = new MediaTypeHeaderValue("application
    httpContent.Headers.ContentType.CharSet = "utf-8";

    HttpClient httpClient = new HttpClient();
    //httpClient..setParameter(HttpMethodParams.HTTP_CONTENT_CHARSET, "utf-

    HttpResponseMessage response = httpClient.PostAsync(url, httpContent).R

    statusCode = response.StatusCode.ToString();
    if (response.IsSuccessStatusCode)
    {
        string result = response.Content.ReadAsStringAsync().Result;
        return result;
    }

    return null;
}

///
/// 发起post请求
///
///
/// url
/// post数据
///
public static T PostResponse(string url, string postData)
    where T : class, new()
{
    if (url.StartsWith("https"))
        System.Net.ServicePointManager.SecurityProtocol = SecurityProtocolT

    HttpContent httpContent = new StringContent(postData);
    httpContent.Headers.ContentType = new MediaTypeHeaderValue("application
    HttpClient httpClient = new HttpClient();
```

目录
导航

```
        T result = default(T);

        HttpResponseMessage response = httpClient.PostAsync(url, httpContent).R

        if (response.IsSuccessStatusCode)
        {
            Task<string> t = response.Content.ReadAsStringAsync();
            string s = t.Result;

            result = JsonConvert.DeserializeObject(s);
        }
        return result;
    }


    ///
    /// 反序列化Xml
    ///
    ///
    ///
    ///
    public static T XmlDeserialize(string xmlString)
        where T : class, new()
    {
        try
        {
            XmlSerializer ser = new XmlSerializer(typeof(T));
            using (StringReader reader = new StringReader(xmlString))
            {
                return (T)ser.Deserialize(reader);
            }
        }
        catch (Exception ex)
        {
            throw new Exception("XmlDeserialize发生异常：xmlString:" + xmlString
        }

    }

    public static string PostResponse(string url, string postData, string token
```

目录
导航

```
{
    if (url.StartsWith("https"))
        System.Net.ServicePointManager.SecurityProtocol = SecurityProtocolT

    HttpContent httpContent = new StringContent(postData);
    httpContent.Headers.ContentType = new MediaTypeHeaderValue("application
    httpContent.Headers.ContentType.CharSet = "utf-8";

    httpContent.Headers.Add("token", token);
    httpContent.Headers.Add("appId", appId);
    httpContent.Headers.Add("serviceURL", serviceURL);


    HttpClient httpClient = new HttpClient();
    //httpClient..setParameter(HttpMethodParams.HTTP_CONTENT_CHARSET, "utf-

    HttpResponseMessage response = httpClient.PostAsync(url, httpContent).R

    statusCode = response.StatusCode.ToString();
    if (response.IsSuccessStatusCode)
    {
        string result = response.Content.ReadAsStringAsync().Result;
        return result;
    }

    return null;
}

///
/// 修改API
///
///
///
///
public static string KongPatchResponse(string url, string postData)
{
    var httpWebRequest = (HttpWebRequest)WebRequest.Create(url);
    httpWebRequest.ContentType = "application/x-www-form-urlencoded";
    httpWebRequest.Method = "PATCH";
```

目录
导航

```
        byte[] btBodys = Encoding.UTF8.GetBytes(postData);
        httpWebRequest.ContentLength = btBodys.Length;
        httpWebRequest.GetRequestStream().Write(btBodys, 0, btBodys.Length);

        HttpWebResponse httpWebResponse = (HttpWebResponse)httpWebRequest.GetRe
        var streamReader = new StreamReader(httpWebResponse.GetResponseStream()
        string responseContent = streamReader.ReadToEnd();

        httpWebResponse.Close();
        streamReader.Close();
        httpWebRequest.Abort();
        httpWebResponse.Close();

        return responseContent;
    }

    ///
    /// 创建API
    ///
    ///
    ///
    ///
    public static string KongAddResponse(string url, string postData)
    {
        if (url.StartsWith("https"))
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls;
        HttpContent httpContent = new StringContent(postData);
        httpContent.Headers.ContentType = new MediaTypeHeaderValue("application
        var httpClient = new HttpClient();
        HttpResponseMessage response = httpClient.PostAsync(url, httpContent).R
        if (response.IsSuccessStatusCode)
        {
            string result = response.Content.ReadAsStringAsync().Result;
            return result;
        }
        return null;
    }

    ///
    /// 删除API
```

目录
导航

```
///
///
///
public static bool KongDeleteResponse(string url)
{
    if (url.StartsWith("https"))
        ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls;

    var httpClient = new HttpClient();
    HttpResponseMessage response = httpClient.DeleteAsync(url).Result;
    return response.IsSuccessStatusCode;
}

///
/// 修改或者更改API
///
///
///
///
public static string KongPutResponse(string url, string postData)
{
    if (url.StartsWith("https"))
        ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls;

    HttpContent httpContent = new StringContent(postData);
    httpContent.Headers.ContentType = new MediaTypeHeaderValue("application

    var httpClient = new HttpClient();
    HttpResponseMessage response = httpClient.PutAsync(url, httpContent).Re
    if (response.IsSuccessStatusCode)
    {
        string result = response.Content.ReadAsStringAsync().Result;
        return result;
    }
    return null;
}

///
/// 检索API
///
```

目录
导航

```
        ///
        ///
        public static string KongSerchResponse(string url)
        {
            if (url.StartsWith("https"))
                ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls;

           var httpClient = new HttpClient();
            HttpResponseMessage response = httpClient.GetAsync(url).Result;
            if (response.IsSuccessStatusCode)
            {
                string result = response.Content.ReadAsStringAsync().Result;
                return result;
            }
            return null;
        }
    }
```

目录
导航

**编辑推荐:**
· 斗鱼 H5 直播原理解析，它是如何省了 80% 的 CDN 流量?
· 超强的纯 CSS 鼠标点击拖拽效果
· 新零售SaaS架构：中央库存系统架构设计
· 不安装运行时运行 .NET 程序 - NativeAOT
· 从 C# 崩溃异常 中研究页堆布局

**最新新闻**:
· 微软秋季发布会：5G版Surface Pro亮相 加深与苹果生态融合
· 扎克伯格谈新款万元VR头显：成本价，我们不像苹果那样定高价
· 比亚迪×奔驰的火爆新车，让我开到半夜不回家
· 抖音集团上线新 Logo
· 腾讯视频否认将接入 88VIP
» 更多新闻...