

工作在浏览器上人-YangBobin

知识不在广泛，在于精通。知识不在积累，在于消化。 学习不在激情，在于坚持。书不在多，一两本真正看懂就行。书读百遍，其义自现。

随笔 - 892, 文章 - 1, 评论 - 68, 阅读 - 123万

目录
导航

导航

博客园
首页
订阅
管理

搜索

找找看

我的标签

设计模式(24)
C#语言新特性(10)
DevExpress(9)
并行编程(9)
报表(5)
git(4)
Socket(4)
WCF(3)
ADO.NET(3)
VS Code(3)
更多

随笔分类 (938)

00 常用知识(5)
01 HTML(17)
02 CSS(24)
03 JavaScript(47)
04 TypeScript(24)
05 Vue(7)
06 JQuery(38)
08 Echarts(12)
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)

C#爬虫（05）：AngleSharp解析html文档

目录

- 一、简介
- 二、使用AngleSharp生成自动缩进格式化的html方法
 - 1、操作DOM示例
 - 2、更改标签属性
 - 3、使用AngleSharp生成html代码自动缩进格式化
 - 4、使用AngleSharp下载获取html代码
 - 5、爬取豆瓣美女图片

一、简介

AngleSharp:<https://github.com/AngleSharp/AngleSharp>

AngleSharp中文官方文档

- <https://www.cnblogs.com/cgzl/p/8970582.html>
- https://blog.csdn.net/qq_36051316/article/details/80002931
- <https://www.cnblogs.com/rbzz/p/10037055.html>

AngleSharp是一个.NET库，使您能够解析基于尖括号的超文本，例如HTML，SVG和MathML，该库还支持未经验证的XML，AngleSharp的一个重要方面是CSS也可以解析。

AngleSharp与类似的库（例如HtmlAgilityPack）相比的优势在于：

- 公开的DOM使用的是W3C官方指定的API，即，甚至在AngleSharp中也可以使用querySelectorAll之类的东西。
- 解析器还使用HTML 5.1规范，该规范定义了错误处理和元素校正。

- 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)
- 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)

AngleSharp库专注于标准合规性，交互性和可扩展性。因此，它为使用C # 的Web开发人员提供了从在任何现代浏览器中使用DOM所获得的所有可能性。

🖥️ AngleSharp

🤖 The ultimate angle brackets parser library parsing HTML5, MathML, SVG and CSS to construct a DOM based on the official W3C specifications.

● C# ☆ 3.3k 🍷 426

🖥️ AngleSharp.Css

🤖 Library to enable support for cascading stylesheets in AngleSharp.

● C# ☆ 21 🍷 14

🖥️ AngleSharp.Js

🤖 Extends AngleSharp with a .NET-based JavaScript engine.

● C# ☆ 62 🍷 21

🖥️ AngleSharp.io

🤖 IO libraries for the AngleSharp project.

● C# ☆ 19 🍷 9

🖥️ AngleSharp.Xml

🤖 Library to add XML and DTD parsing capabilities to AngleSharp.

● C# ☆ 8 🍷 1

🖥️ AngleSharp.XPath

🤖 Adds XPath support to AngleSharp as an alternative to CSS selectors.

● C# ☆ 28 🍷 6

官方实例：<https://github.com/AngleSharp/AngleSharp.Samples>

这个简单的示例将使用Wikipedia的网站进行数据检索。

```
var config = Configuration.Default.WithDefaultLoader();
var address = "https://en.wikipedia.org/wiki/List_of_The_Big_Bang_Theory_episodes";
var context = BrowsingContext.New(config);
var document = await context.OpenAsync(address);
var cellSelector = "tr.vevent td:nth-child(3)";
var cells = document.QuerySelectorAll(cellSelector);
var titles = cells.Select(m => m.TextContent);
```

二、使用AngleSharp生成自动缩进格式化的html方法

1、操作DOM示例

```
//创建一个（可重用）解析器前端
var parser = new HtmlParser();
//html DOM节点
var source = "
<h1>Some example source</h1>
<p>This is a paragraph element</p>
";

//解析源文件
var document = parser.Parse(source);
//创建P标签
var p = document.CreateElement("p");
```

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](#)[w3school在线教程\(老\)](#)[我的码云gitee](#)[我的GitHub \(慢\)](#)

```
p.TextContent = "This is another paragraph.";
//添加到DOM
document.Body.AppendChild(p);
//返回完整html
var html = document.DocumentElement.OuterHtml;
ViewData["html"] = html;
```

效果展示

Some example source

This is a paragraph element

This is another paragraph.

;

2、更改标签属性

给标签添加自定义属性

```
var parser = new HtmlParser();
//为以下源代码生成HTML DOM
var document = parser.Parse("
<ul>
<li>First element</li>
<li>Second element</li>
<li>third</li>
<li class='bla'>Last</li>
</ul>
");
//获取所有li元素并将test属性设置为值测试
var elements = document.QuerySelectorAll("li").Attr("test", "test");
//元素仍然包含所有li元素
ViewData["html"] = document.DocumentElement.OuterHtml;
```

效果展示

- First element
- Second element
- third
- Last

```
::before
▼<ul>
  <li test="test">First element</li>
  <li test="test">Second element</li>
  <li test="test">third</li>
  <li class="bla" test="test">Last</li>
</ul>
"
```

3、使用AngleSharp生成html代码自动缩进格式化

```
var parser = new HtmlParser();
var document = parser.ParseDocument(text);
```

```
using (var writer = new StringWriter())
{
    document.ToHtml(writer, new PrettyMarkupFormatter
    {
        Indentation = "\t",
        NewLine = "\n"
    });
    var indentedText = writer.ToString();
}
```

4、使用AngleSharp下载获取html代码

```
var requester = new DefaultHttpRequester("Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome
requester.Headers.Add("Accept", "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8");
requester.Headers.Add("Referer", "");
requester.Headers.Add("Accept-Language", "zh-Hans-CN,zh-Hans;q=0.8,en-US;q=0.5,en;q=0.3");
var context = BrowsingContext.New(Configuration.Default.WithLocaleBasedEncoding().WithDefaultLoader().WithDefaultCookies()).
//根据虚拟请求/响应模式创建文档
var document = context.OpenAsync(url).Result;
using (var writer = new StringWriter())
{
    document.ToHtml(writer, new PrettyMarkupFormatter
    {
        Indentation = "\t",
        NewLine = "\n"
    });
    var indentedText = writer.ToString();
}
```

5、爬取豆瓣美女图片

新建一个Belle类用于保存获取的图片信息

```
///
/// 解析html
///
public class Belle
{
    ///
    /// 标题
    ///
    public string Title { get; set; }
    ///
    /// 图片地址
    ///
    public string ImageUrl { get; set; }
}
```

获取html并解析

```
// 设置配置以支持文档加载
var config = Configuration.Default.WithDefaultLoader();
// 豆瓣地址
var address = "https://www.dbmeinv.com/dbgroup/show.htm?cid=4";
// 请求豆瓣网
var document = BrowsingContext.New(config).OpenAsync(address);
// 根据class获取html元素
var cells = document.Result.QuerySelectorAll(".panel-body li");
// We are only interested in the text - select it with LINQ
List list = new List();
foreach (var item in cells)
{
    var belle = new Belle
    {
        Title= item.QuerySelector("img").GetAttribute("title"),
        ImageUrl= item.QuerySelector("img").GetAttribute("src")
    };
    list.Add(belle);
}
ViewData["html"] = list;
```

效果如下



分类: 12 C#语言基础, 15 WinForm

好文要顶

关注我

收藏该文

springsnow

粉丝 - 96 关注 - 5

+加关注

« 上一篇: [C#爬虫 \(04\) : HtmlAgilityPack解析html文档](#)

» 下一篇: [将本地已有项目添加到 git管理](#)

posted on 2020-07-10 10:53 springsnow 阅读(1211) 评论(1) 编辑 收藏 举报

目录
导航

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) 博客园首页

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...

Powered by:

博客园

Copyright © 2022 springsnow

Powered by .NET 6 on Kubernetes