Data scraping using .NET Framework Classes + Fizzler

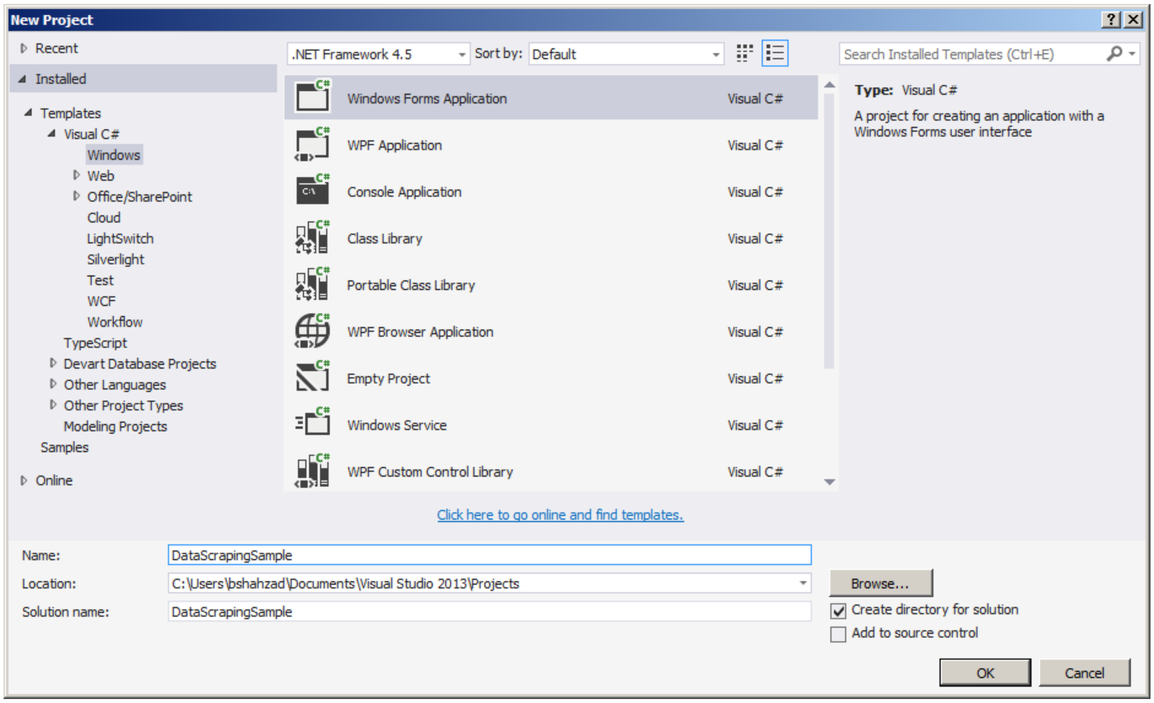
This guide will explain how to use .NET classes to get HTML response against a request. Once HTML is received then you can use different libraries (e.g. Fizzler, CsQuery). Here we've used Fizzler +HTMLAgilityPack. This tutorial will guide you about adding these packages using NuGet. It explains how to extract information from a web page. This tutorial is only for learning purposes.

Version	**Last updated**	Comments	Modified By
V1.0	22-04-2016		Bilal Shahzad

Here we are going to build a simple windows application which will extract basic job information from “rozee.pk” site.

Walkthrough

1- Create a new “Windows Forms” application with .NET Framework 4.5.



New Project

.NET Framework 4.5 Sort by: Default

Type: Visual C#

A project for creating an application with a Windows Forms user interface

Name: DataScrapingSample

Location: C:\Users\bshahzad\Documents\Visual Studio 2013\Projects

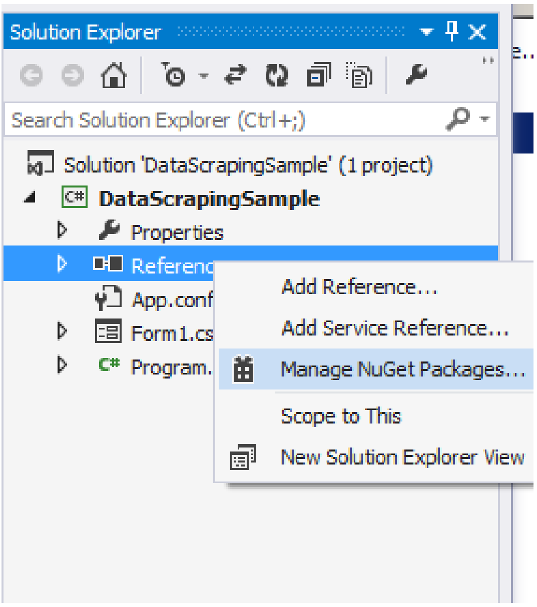
Solution name: DataScrapingSample

☑ Create directory for solution

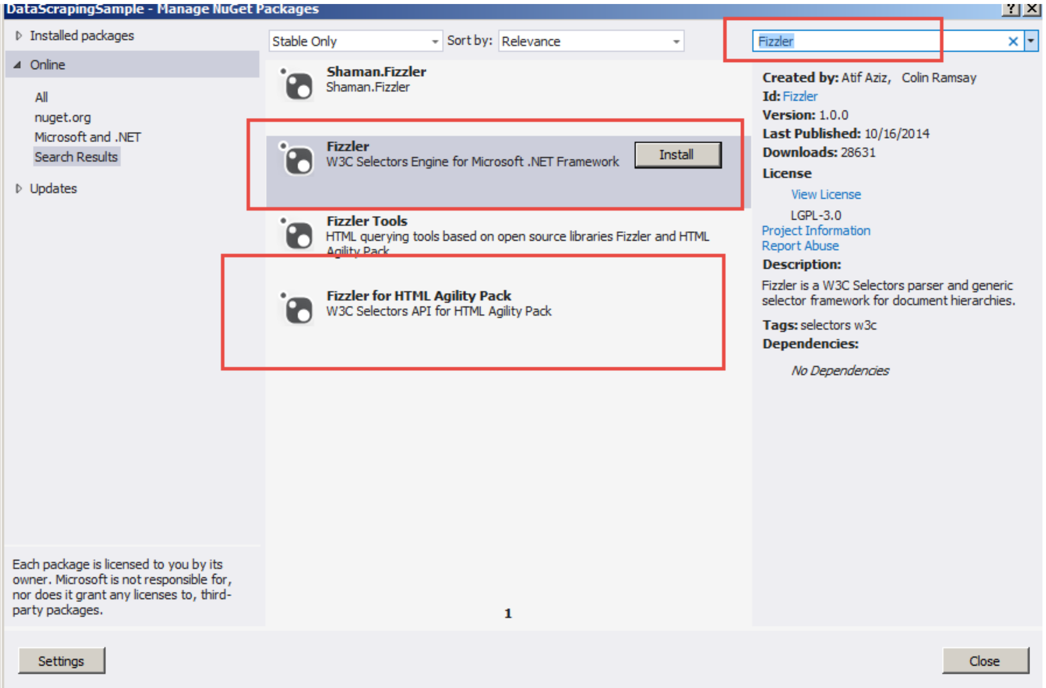
☐ Add to source control

Recent
Installed
Templates
Visual C#
Windows
Web
Office/SharePoint
Cloud
LightSwitch
Silverlight
Test
WCF
Workflow
TypeScript
Devart Database Projects
Other Languages
Other Project Types
Modeling Projects
Samples
Online
Search Installed Templates (Ctrl+E)
Windows Forms Application Visual C#
WPF Application Visual C#
Console Application Visual C#
Class Library Visual C#
Portable Class Library Visual C#
WPF Browser Application Visual C#
Empty Project Visual C#
Windows Service Visual C#
WPF Custom Control Library Visual C#
Click here to go online and find templates.
Browse...
OK
Cancel

2- Now we need to add some references to access “Fizzler + HtmlAgilityPack” APIs. Right click on “References” and select “Manage NuGet Packages…”



3- In search box type “Fizzler” and Install the packages as highlighted in following screenshot.



Shaman.Fizzler
Shaman.Fizzler

Fizzler
W3C Selectors Engine for Microsoft .NET Framework

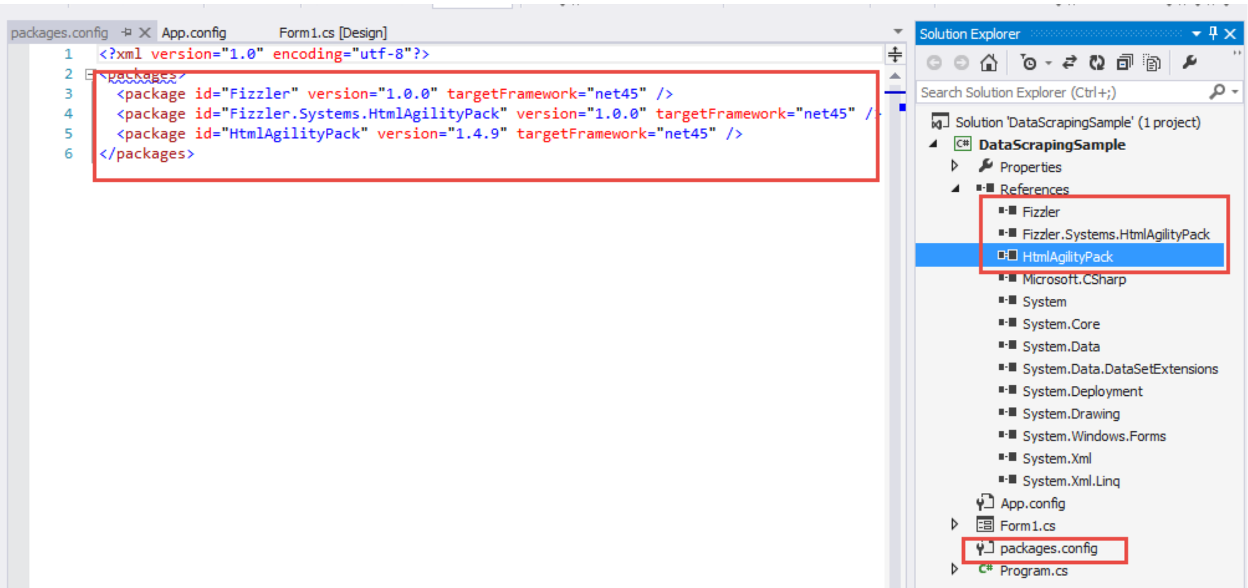
Fizzler Tools
HTML querying tools based on open source libraries Fizzler and HTML Agility Pack

Fizzler for HTML Agility Pack
W3C Selectors API for HTML Agility Pack

Created by: Atif Aziz, Colin Ramsay
Id: Fizzler
Version: 1.0.0
Last Published: 10/16/2014
Downloads: 28631
License
View License
LGPL-3.0
Project Information
Report Abuse
Description:
Fizzler is a W3C Selectors parser and generic selector framework for document hierarchies.
Tags: selectors w3c
Dependencies:
No Dependencies

Each package is licensed to you by its owner. Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.

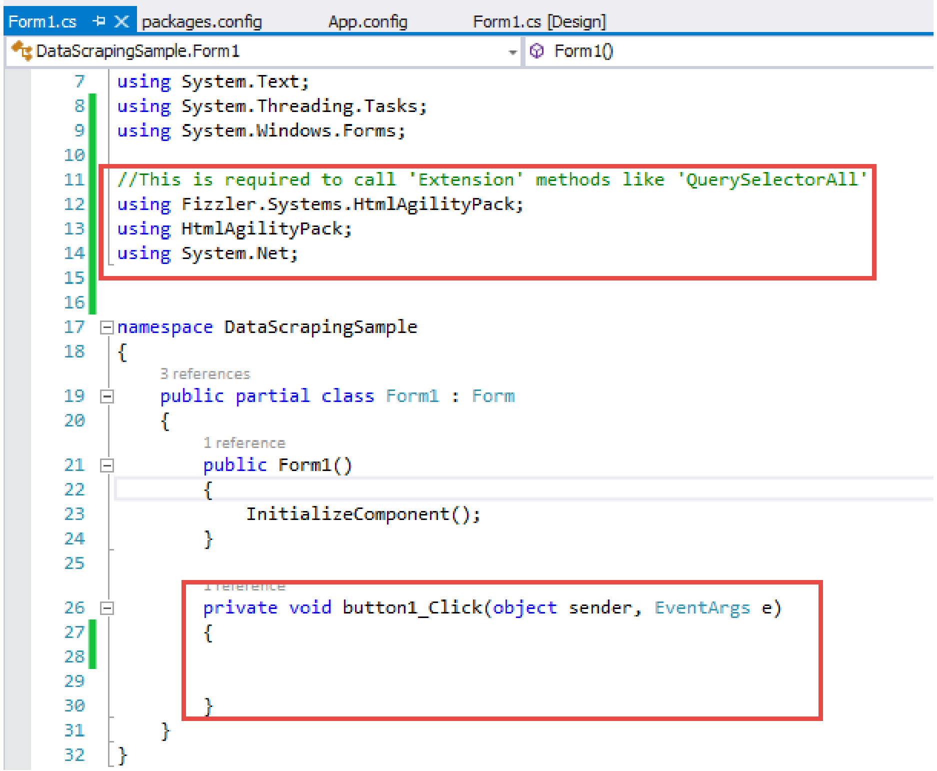
4- In “Packages.config”, you will see following packages are added. Also check the reference of DLLs shown in right side panel.



```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="Fizzler" version="1.0.0" targetFramework="net45" />
  <package id="Fizzler.Systems.HtmlAgilityPack" version="1.0.0" targetFramework="net45" /
  <package id="HtmlAgilityPack" version="1.4.9" targetFramework="net45" />
</packages>
```

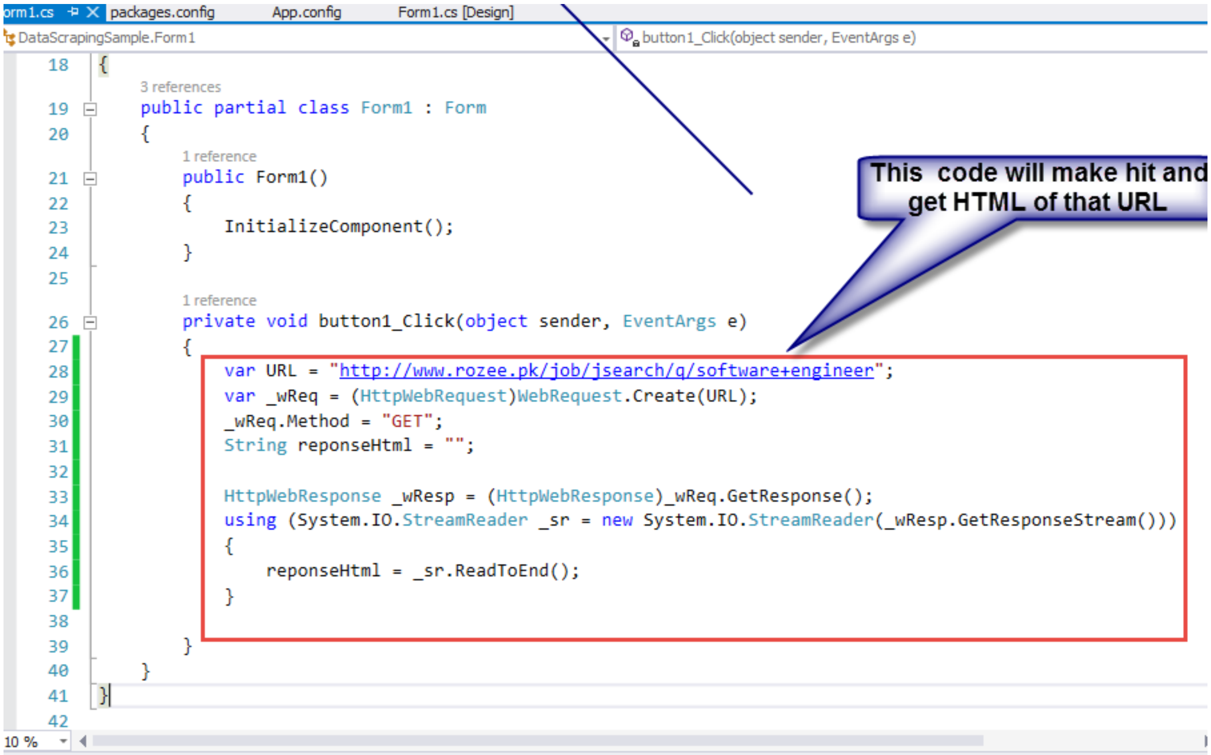
Solution 'DataScrapingSample' (1 project)
- DataScrapingSample
  - Properties
  - References
    - Fizzler
    - Fizzler.Systems.HtmlAgilityPack
    - HtmlAgilityPack
    - Microsoft.CSharp
    - System
    - System.Core
    - System.Data
    - System.Data.DataSetExtensions
    - System.Deployment
    - System.Drawing
    - System.Windows.Forms
    - System.Xml
    - System.Xml.Linq
  - App.config
  - Form1.cs
  - packages.config
  - Program.cs

5- Now add a button on a form and generate click event of it. In Code, add the reference of namespaces as shown below.



```
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

//This is required to call 'Extension' methods like 'QuerySelectorAll'
using Fizzler.Systems.HtmlAgilityPack;
using HtmlAgilityPack;
using System.Net;


namespace DataScrapingSample
{
    3 references
    public partial class Form1 : Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();
        }

        1 reference
        private void button1_Click(object sender, EventArgs e)
        {


        }
    }
}
```

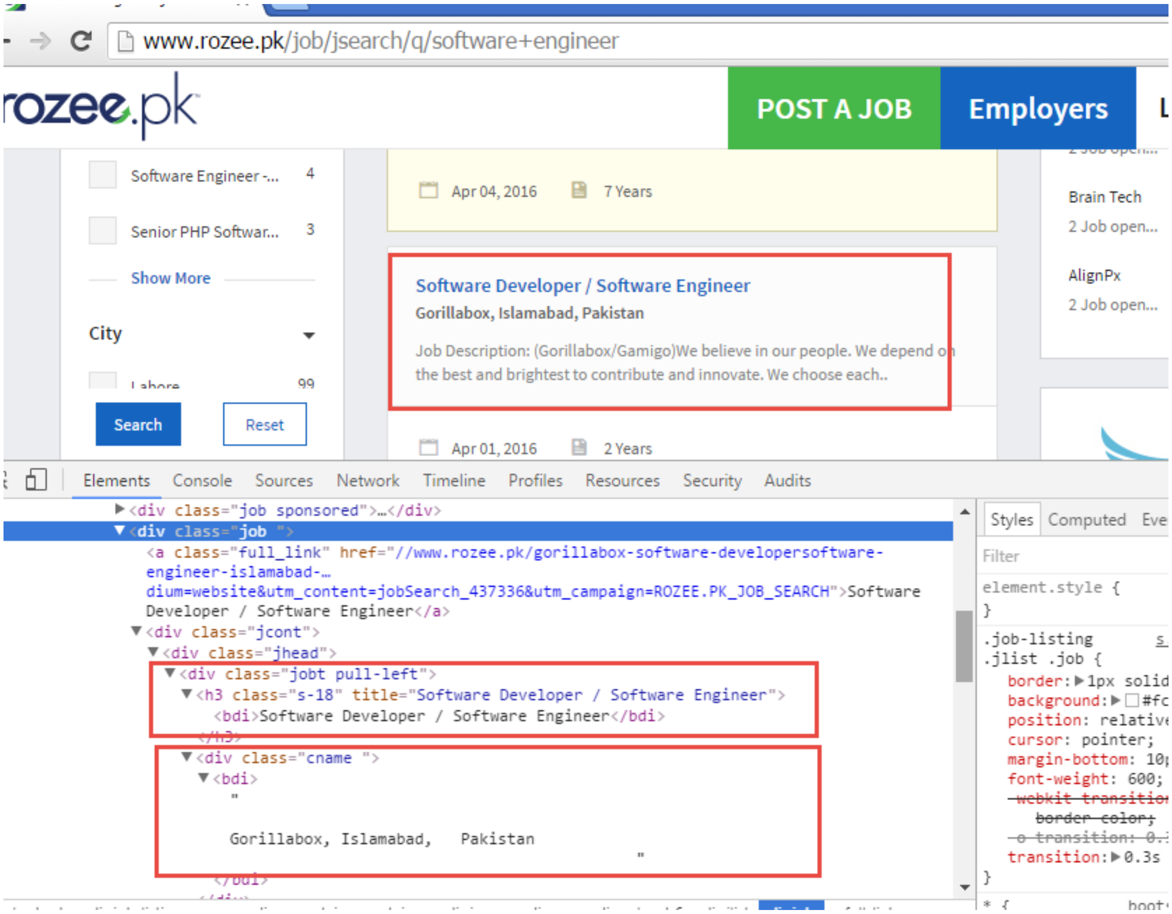
6- We are going to scrap data from following http://www.rozee.pk/job/jsearch/q/all. If you type "Software Engineer" in highlighted textbox below and press highlighted search button, page will be redirected to http://www.rozee.pk/job/jsearch/q/software+engineer. We want to scrap this page.



This code will make hit and get HTML of that URL

```
{
    3 references
    public partial class Form1 : Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();
        }

        1 reference
        private void button1_Click(object sender, EventArgs e)
        {
            var URL = "http://www.rozee.pk/job/jsearch/q/software+engineer";
            var _wReq = (HttpWebRequest)WebRequest.Create(URL);
            _wReq.Method = "GET";
            String reponseHtml = "";

            HttpWebResponse _wResp = (HttpWebResponse)_wReq.GetResponse();
            using (System.IO.StreamReader _sr = new System.IO.StreamReader(_wResp.GetResponseStream()))
            {
                reponseHtml = _sr.ReadToEnd();
            }

        }
    }
}
```

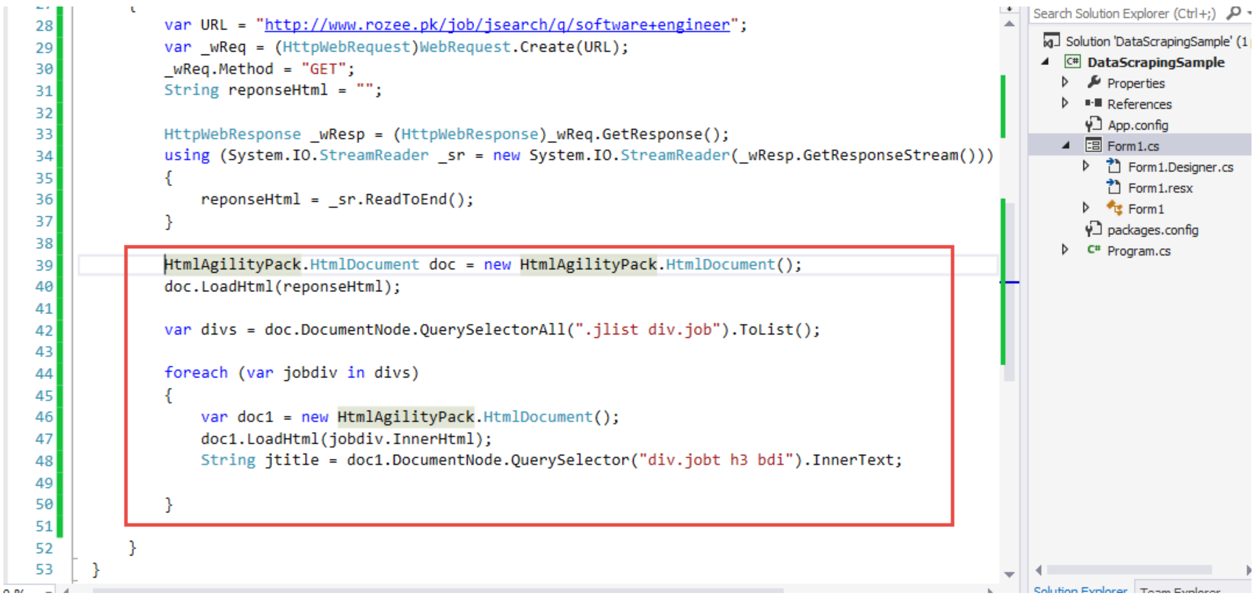
7- If we open above link in browser, we can investigate the HTML of our results. Each result is in “div.job”. Inside div, we can see the selectors of Title and company and so on.



```
▶<div class="job sponsored">…</div>
▼<div class="job ">
   <a class="full_link" href="//www.rozee.pk/gorillabox-software-developersoftware-
   engineer-islamabad-…
   dium=website&utm_content=jobSearch_437336&utm_campaign=ROZEE.PK_JOB_SEARCH">Software
   Developer / Software Engineer</a>
  ▼<div class="jcont">
    ▼<div class="jhead">
      ▼<div class="jobt pull-left">
        ▼<h3 class="s-18" title="Software Developer / Software Engineer">
           <bdi>Software Developer / Software Engineer</bdi>
         </h3>
        ▼<div class="cname ">
          ▼<bdi>
             "

             Gorillabox, Islamabad,    Pakistan
                                                       "
           </bdi>
```

```
element.style {
}
.job-listing
.jlist .job {
  border:▶1px solid
  background:▶□#fc
  position: relative
  cursor: pointer;
  margin-bottom: 10
  font-weight: 600;
  ~~-webkit-transition~~
    ~~border-color;~~
  ~~-o-transition: 0.~~
  transition:▶0.3s
}
```

8- Her is the code to find all DIVs and then iterating those divs to extract information from it.



```
            var URL = "http://www.rozee.pk/job/jsearch/q/software+engineer";
            var _wReq = (HttpWebRequest)WebRequest.Create(URL);
            _wReq.Method = "GET";
            String reponseHtml = "";

            HttpWebResponse _wResp = (HttpWebResponse)_wReq.GetResponse();
            using (System.IO.StreamReader _sr = new System.IO.StreamReader(_wResp.GetResponseStream()))
            {
                reponseHtml = _sr.ReadToEnd();
            }

            HtmlAgilityPack.HtmlDocument doc = new HtmlAgilityPack.HtmlDocument();
            doc.LoadHtml(reponseHtml);

            var divs = doc.DocumentNode.QuerySelectorAll(".jlist div.job").ToList();

            foreach (var jobdiv in divs)
            {
                var doc1 = new HtmlAgilityPack.HtmlDocument();
                doc1.LoadHtml(jobdiv.InnerHtml);
                String jtitle = doc1.DocumentNode.QuerySelector("div.jobt h3 bdi").InnerText;

            }

        }
    }
```

Useful Links

http://stackoverflow.com/questions/846994/how-to-use-html-agility-pack

http://www.mikesdotnetting.com/article/273/using-the-htmlagilitypack-to-parse-html-in-asp-net

http://stackoverflow.com/questions/22092208/parsing-html-with-csquery

https://github.com/jamietre/CsQuery