# IronWebScraper Namespace

## Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommonUserAgents</td>
<td>Static helper class which lists common web-browser user-agent strings.</td>
</tr>
<tr>
<td>HtmlNode</td>
<td>The HtmlNode class represents a single DOM element in an HTML or XML document.</td>
</tr>
<tr>
<td>HtmlNodeExtensions</td>
<td>Extension methods for finding elements within an IEnumerable&lt;HtmlNode&gt;.</td>
</tr>
<tr>
<td>HttpIdentity</td>
<td>A class defining the browsing 'identity' to be used to fetch a given URL with UserAgent and Http Header information.</td>
</tr>
<tr>
<td>License</td>
<td>A public static class used to apply license keys to IronWebScraper.</td>
</tr>
<tr>
<td>MetaData</td>
<td>A flexible dictionary of object values or objects to any Request. Meta can contain instances of classes, List and Dictionaries. This meta-data can then be accessed while parsing the Response and even passed forward to the next Request. E.g:</td>
</tr>
</tbody>
</table>

## Examples

```
Request["page-number"] = int pageNum
```

- Request
  - Represents a http request to be made by IronWebScraper
- Response
  - Represents a http response made by IronWebScraper
ScrapedData

A flexible dictionary of object values used to conveniently store scraped data of any type, in a key-value dictionary which can be saved as JSON using the Yield method.

E.g:

```csharp
var Data = new ScrapedData();
Data['title'] = DateTime.Now;
Data['date'] = DateTime.Now;
```

WebScraper

A base class which developers can extend to build custom web-scraping applications.

Enumerations

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebScraperLogLevel</td>
<td>Level of WebScraper logging to the Console. Because this Enum is a Flag type options can be combined using a pipe: e.g. LogLevel.Critical</td>
</tr>
<tr>
<td>WebScraperThrottle</td>
<td>Throttle remote clients by their host name or by their public IP address.</td>
</tr>
</tbody>
</table>
IronWebScraper - The C# Web Scraping Library
CommonUserAgents Class

Static helper class which lists common web-browser user-agent strings.

Inheritance Hierarchy

- System
  - Object
    - IronWebScraper
      - CommonUserAgents

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public static class CommonUserAgents
```

The `CommonUserAgents` type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="Image" alt="s" /> All</td>
<td>Common desktop, tablet and mobile user-agent strings.</td>
</tr>
<tr>
<td><img src="Image" alt="s" /> DesktopUserAgents</td>
<td>Common desktop web-browser user-agent strings. Chrome, IE11 and Safari.</td>
</tr>
<tr>
<td><img src="Image" alt="s" /> MobileUserAgents</td>
<td>Common mobile web-browser user-agent strings. iPad, iPhone and Windows</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ChromeDesktopUserAgents</td>
<td>Common desktop Chrome web-browser user-agent strings, most popular first.</td>
</tr>
<tr>
<td>FireFoxDesktopUserAgents</td>
<td>Common desktop FireFox web-browser user-agent strings, most popular first.</td>
</tr>
<tr>
<td>IE11DesktopUserAgents</td>
<td>Common IE11 desktop web-browser user-agent strings, most popular first. Only IE11 is listed for IE to give the best chance of consistent HTML being served.</td>
</tr>
<tr>
<td>IPadUserAgents</td>
<td>Common Apple iApd user agent strings.</td>
</tr>
<tr>
<td>IPhoneUserAgents</td>
<td>Common Apple iPhone user agent strings.</td>
</tr>
<tr>
<td></td>
<td>SafariDesktopUserAgents</td>
</tr>
<tr>
<td>---</td>
<td>-------------------------</td>
</tr>
<tr>
<td></td>
<td>WindowsTabletUserAgents</td>
</tr>
</tbody>
</table>

**See Also**

Reference
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
CommonUserAgents Properties

The `CommonUserAgents` type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>All</code></td>
<td>Common desktop, tablet and mobile user-agent strings.</td>
</tr>
<tr>
<td><code>MobileUserAgents</code></td>
<td>Common mobile web-browser user-agent strings. iPad, iPhone and Windows Tablet.</td>
</tr>
</tbody>
</table>

See Also

Reference

`CommonUserAgents Class`

`IronWebScraper Namespace`
IronWebScraper - The C# Web Scraping Library
CommonUserAgentsAll Property

Common desktop, tablet and mobile user-agent strings.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

C#  
```csharp
public static string[] All { get; }
```

VB

```vbnet
Public Shared Function All As String() As String
```

### Property Value
Type: String

### See Also

Reference  
- [CommonUserAgents Class](#)  
- [IronWebScraper Namespace](#)
CommonUserAgentsDesktopUserAgents Property

Common desktop web-browser user-agent strings. Chrome, IE11 and Safari.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>public static string[] DesktopUserAgents { get; }</td>
</tr>
</tbody>
</table>

Property Value
Type: String

See Also

Reference
CommonUserAgents Class
IronWebScraper Namespace
CommonUserAgentsMobileUserAgent Property

Common mobile web-browser user-agent strings. iPad, iPhone and Windows Tablet.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public static string[] MobileUserAgents { get; }
```

### Property Value

Type: String

### See Also

Reference  
CommonUserAgents Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
CommonUserAgents Fields

The CommonUserAgents type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChromeDesktopUserAgents</td>
<td>Common desktop Chrome web-browser user-agent strings, most popular first.</td>
</tr>
<tr>
<td>FireFoxDesktopUserAgents</td>
<td>Common desktop FireFox web-browser user-agent strings, most popular first.</td>
</tr>
<tr>
<td>IE11DesktopUserAgents</td>
<td>Common IE11 desktop web-browser user-agent strings, most popular first. Only IE11 is listed for IE to give the best chance of consistent HTML being served.</td>
</tr>
<tr>
<td>IPadUserAgents</td>
<td>Common Apple iApd user agent strings.</td>
</tr>
<tr>
<td>IPhoneUserAgents</td>
<td>Common Apple</td>
</tr>
</tbody>
</table>
iPhone user agent strings.

<table>
<thead>
<tr>
<th>s</th>
<th>SafariDesktopUserAgents</th>
<th>Common OS X Safari web-browser user-agent strings, most popular first.</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>WindowsTabletUserAgents</td>
<td>Common Windows Tablet user-agent strings.</td>
</tr>
</tbody>
</table>

## See Also

Reference
- CommonUserAgents Class
- IronWebScraper Namespace
CommonUserAgentsChromeDesktop

Field

Common desktop Chrome web-browser user-agent strings, most popular first.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

▶ Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static string[] ChromeDesktopUserAgents</td>
<td></td>
</tr>
</tbody>
</table>

Field Value
Type: String

▶ See Also

Reference
CommonUserAgents Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
CommonUserAgents.FireFoxDesktopUserAgents

Field

Common desktop FireFox web-browser user-agent strings, most popular first.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>public static string[] FireFoxDesktopUserAgents</td>
</tr>
</tbody>
</table>

Field Value

Type: String

### See Also

Reference

CommonUserAgents Class  
IronWebScraper Namespace
CommonUserAgents\IE11DesktopUserAgents Field

Common IE11 desktop web-browser user-agent strings, most popular first. Only IE11 is listed for IE to give the best chance of consistent HTML being served.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

C# VB

```csharp
public static string[] IE11DesktopUserAgents
```

Field Value
Type: String

See Also

Reference
CommonUserAgents Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
CommonUserAgents.IPadUserAgent Field

Common Apple iApd user agent strings.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public static string[] IPadUserAgents
```

Field Value
Type: String

See Also

Reference
CommonUserAgents Class
IronWebScraper Namespace
CommonUserAgents

Field

Common Apple iPhone user agent strings.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public static string[] IPhoneUserAgents</code></td>
<td></td>
</tr>
</tbody>
</table>

Field Value
Type: String

See Also

Reference
CommonUserAgents Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
Common OS X Safari web-browser user-agent strings, most popular first.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

C#     VB
---   -----  
```csharp
public static string[] SafariDesktopUserAgents
```  

Field Value  
**Type:** String

### See Also

**Reference**  
CommonUserAgents Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
CommonUserAgentsWindowsTabletField

Common Windows Tablet user-agent strings.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

**C#**

```csharp
public static string[] WindowsTabletUserAgents
```

**VB**

```vb
Public Shared Function WindowsTabletUserAgents() As String
```

### Field Value

Type: **String**

### See Also

**Reference**

[CommonUserAgents Class]
[IronWebScraper Namespace]
IronWebScraper - The C# Web Scraping Library
HtmlNode Class

The HtmlNode class represents a single DOM element in a HTML or XML document.

Inheritance Hierarchy

- System
- Object
- IronWebScraper
- HtmlNode

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public class HtmlNode
```

The HtmlNode type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the HTML node (s)</td>
</tr>
<tr>
<td>ChildNodes</td>
<td>An array of nodes nested within the Htn</td>
</tr>
<tr>
<td>InnerHtml</td>
<td>Gets the HtmlNode's inner Html as a string own html markup, only that nested inside</td>
</tr>
<tr>
<td>InnerText</td>
<td>Gets the text content of the HtmlNode a with all groups of whitespace turned into</td>
</tr>
<tr>
<td>InnerTextClean</td>
<td>Gets the text content of the HtmlNode a</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>NodeName</strong></td>
<td>Name of the HtmlNode such as &quot;div&quot; &quot;p&quot;</td>
</tr>
</tbody>
</table>
| **NodeType** | Type of the HtmlNode
Possible values are: "ELEMENT_NODE", "TEXT_NODE", "CDATA_SECTION_NODE", "COMMENT_NODE", "DOCUMENT_NODE", "DOCUMENT_FRAGMENT_NODE" |
| **OuterHtml**| Gets the HtmlNode's outer Html as a string. OuterHTML will include the tag's own markup. |
| **ParentNode**| Gets the Parent (enclosing) tag for this HtmlNode.                           |
| **TextContent** | Gets the text content of the HtmlNode and all of its descendants.            |
| **TextContentClean** | Gets the text content of the HtmlNode, trimmed, with all groups of whitespace turned into a single space character. TextContent will include the content of non-printable elements just as style and script nodes. |

**Top**

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Css</strong></td>
<td>Uses CSS selectors to find all child nodes matching selector. This works in the same way as $(<code>.ClassName</code>) in jQuery or querySelectorAll() in JavaScript.</td>
</tr>
<tr>
<td><strong>CssExists</strong></td>
<td>Uses CSS selectors to find if there are any matching nodes within the Response Document. This works in the same way as <code> </code></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Equals</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetAttribute</td>
<td>Gets a single attribute value from the HtmlNode. Returns null if that attribute is not present.</td>
</tr>
<tr>
<td>GetElementById</td>
<td>Synonym of JavaScript's getElementById function. Searches inside the current HtmlNode.</td>
</tr>
<tr>
<td>GetElementsByTagName</td>
<td>Synonym of JavaScript's getElementsByTagName function. Searches inside the current HtmlNode.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>HasAttribute</td>
<td>Tests if the HtmlNode has a given attribute such as &quot;class&quot; or &quot;href&quot;</td>
</tr>
<tr>
<td>QuerySelector</td>
<td>Synonym of JavaScript's querySelector function. Searches inside the current HtmlNode.</td>
</tr>
<tr>
<td>QuerySelectorAll</td>
<td>Synonym of JavaScript's querySelectorAll function. Searches</td>
</tr>
</tbody>
</table>
inside the current HtmlNode.

<table>
<thead>
<tr>
<th></th>
<th>ToString</th>
<th>(Inherited from Object.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>XPath</td>
<td>Queries the HtmlNode to return all descendant nodes matching an XPath expression.</td>
</tr>
</tbody>
</table>

See Also

Reference
IronWebScraper Namespace
# HtmListNode Properties

The `HtmListNode` type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>Gets the attributes of the HTML node (s)</td>
</tr>
<tr>
<td>ChildNodes</td>
<td>An array of nodes nested within the Htm</td>
</tr>
<tr>
<td>InnerHtml</td>
<td>Gets the HtmNode's inner Html as a str own html markup, only that nested insid</td>
</tr>
<tr>
<td>InnerText</td>
<td>Gets the text content of the HtmNode a with all groups of whitespace turned into</td>
</tr>
<tr>
<td>InnerTextClean</td>
<td>Gets the text content of the HtmNode a</td>
</tr>
<tr>
<td>NodeName</td>
<td>Name of the HtmNode such as &quot;div&quot; &quot;p'</td>
</tr>
<tr>
<td>NodeType</td>
<td>Type of the HtmNode Possible values are: &quot;ELEMENT_NODE&quot;,&quot;TEXT_NODE&quot;</td>
</tr>
<tr>
<td>OuterHtml</td>
<td>Gets the HtmNode's outer Html as a stri markup.</td>
</tr>
<tr>
<td>ParentNode</td>
<td>Gets the Parent (enclosing) tag for this</td>
</tr>
<tr>
<td>TextContent</td>
<td>Gets the text content of the HtmNode a</td>
</tr>
<tr>
<td>TextContentClean</td>
<td>Gets the text content of the HtmNode, a single space character. TextContent w</td>
</tr>
</tbody>
</table>
just as style and script nodes.

See Also

Reference

HtmlNode Class
IronWebScraper Namespace
HtmlNodeAttributes Property

Gets the attributes of the HTML node (such as href, class, style etc).

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public Dictionary&lt;string, string&gt; Attributes { get; }</code></td>
<td></td>
</tr>
</tbody>
</table>

### Property Value

Type: `Dictionary<String, String>`  
The attributes as a `Dictionary<string, string>`

### See Also

Reference  
HtmlNode Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
HtmlNode ChildNodes Property

An array of nodes nested within the HtmlNode.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public HtmlNode[] ChildNodes { get; }</code></td>
<td></td>
</tr>
</tbody>
</table>

**Property Value**

Type: `HtmlNode`

### See Also

Reference

- `HtmlNode Class`
- `IronWebScraper Namespace`
IronWebScraper - The C# Web Scraping Library
HtmlNodeInnerHtml Property

Gets the HtmlNode's inner Html as a string. OuterHTML will not include the current tag's own html markup, only that nested inside it.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public string InnerHtml { get; }</td>
<td></td>
</tr>
</tbody>
</table>

**Property Value**  
Type: **String**

### See Also

Reference  
**HtmlNode Class**  
**IronWebScraper Namespace**
IronWebScraper - The C# Web Scraping Library
**HtmlNodeInnerText Property**

Gets the text content of the HtmlNode as might be shown to a user. Text will be trimmed, with all groups of whitespace turned into a single space character.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public string InnerText { get; }
```

### Property Value

Type: **String**

### See Also

Reference  
HtmlNode Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
**HtmlNodeInnerTextClean Property**

Gets the text content of the HtmlNode as might be shown to a user.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public string InnerTextClean { get; }</code></td>
<td></td>
</tr>
</tbody>
</table>

**Property Value**  
**Type:** String

### See Also

- Reference  
  - HtmlNode Class  
  - IronWebScraper Namespace
HtmlNode.Name Property

Name of the HtmlNode such as "div" "p" or "#text"

Namespace: IronWebScraper  
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

### C#

```csharp
public string NodeName { get; }
```

### VB

```vbnet
Public Property NodeName() As String
```

Property Value

Type: String

See Also

Reference

HtmlNode Class  
IronWebScraper Namespace
HtmlNodeNodeType Property

Type of the HtmlNode

Possible values are: "ELEMENT_NODE", "TEXT_NODE", "CDATA_SECTION_NODE", "COMMENT_NODE", "DOCUMENT_NODE", "DOCUMENT_TYPE_NODE", "DOCUMENT_FRAGMENT_NODE"

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public string NodeType { get; }</td>
<td></td>
</tr>
</tbody>
</table>

Property Value
Type: String

See Also

Reference
HtmlNode Class
IronWebScraper Namespace
HtmlNodeOuterHtml Property

Gets the HtmlNode's outer Html as a string. OuterHTML will include the tag's own html markup.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public string OuterHtml { get; }</td>
<td></td>
</tr>
</tbody>
</table>

Property Value

Type: String

See Also

Reference

HtmlNode Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
HtmlNodeParentNode Property

Gets the Parent (enclosing) tag for this HtmlNode.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public HtmlNode ParentNode { get; }
```

### Property Value

Type: **HtmlNode**  
The parent node as an HtmlNode.

### See Also

**Reference**  
HtmlNode Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
**HtmlNodeTextContent Property**

Gets the text content of the HtmlNode and all of its descendants.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public string TextContent { get; }
```

Property Value  
Type: **String**

### See Also

Reference  
**HtmlNode Class**  
**IronWebScraper Namespace**
HtmlNode TextContentClean Property

Gets the text content of the HtmlNode, Trimmed, with all groups of whitespace turned into a single space character. TextContent will include the content of non-printable elements just as style and script nodes.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public string TextContentClean { get; }</td>
<td></td>
</tr>
</tbody>
</table>

Property Value  
**Type:** String

### See Also

Reference  
HtmlNode Class  
IronWebScraper Namespace
# HtmLNode Methods

The **HtmLNode** type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Css</strong></td>
<td>Uses CSS selectors to find all child nodes matching selector. This works in the same way as <code>$('.ClassName')</code> in jQuery or <code>querySelectorAll()</code> in JavaScript.</td>
</tr>
<tr>
<td><strong>CssExists</strong></td>
<td>Uses CSS selectors to find if there are any matching nodes within the Response Document. This works in the same way as <code>$('.ClassName').length &gt; 0</code> in jQuery or <code>querySelectorAll().length &gt;0</code> in JavaScript.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>(Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td><strong>GetAttribute</strong></td>
<td>Gets a single attribute value from the HtmlNode. Returns null if that attribute is not present.</td>
</tr>
<tr>
<td><strong>GetElementByld</strong></td>
<td>Synonym of JavaScript's <code>getElementByld</code></td>
</tr>
</tbody>
</table>
function. Searches inside the current HtmlNode.

- **GetElementsById**
  - Synonym of JavaScript's getElementsByClassName function. Searches inside the current HtmlNode.

- **GetHashCode**
  - (Inherited from Object.)

- **GetType**
  - (Inherited from Object.)

- **HasAttribute**
  - Tests if the HtmlNode has a given attribute such as "class" or "href"

- **QuerySelector**
  - Synonym of JavaScript's querySelector function. Searches inside the current HtmlNode.

- **QuerySelectorAll**
  - Synonym of JavaScript's querySelectorAll function. Searches inside the current HtmlNode.

- **ToString**
  - (Inherited from Object.)

- **XPath**
  - Queries the HtmlNode to return all descendant nodes matching an XPath expression.

**See Also**
Reference

HtmlNode Class
IronWebScraper Namespace
HtmlNode Css Method

Uses CSS selectors to find all child nodes matching selector. This works in the same way as `$('.ClassName')` in jQuery or `querySelectorAll()` in JavaScript.

Namespace: IronWebScraper  
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public HtmlNode[] Css(
    string selector
)
```

Parameters

`selector`  
Type: System.String

Return Value

Type: HtmlNode

See Also

Reference
HtmlNode Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
HtmlNode.CssExists Method

Uses CSS selectors to find if there are any matching nodes within the Response Document. This works in the same way as 
\$('ClassName').length > 0 in jQuery or querySelectorAll().length >0 in 
JavaScript.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public bool CssExists(string selector)</code></td>
<td></td>
</tr>
</tbody>
</table>

**Parameters**

- **selector**  
  Type: System.String

**Return Value**

Type: Boolean

### See Also

- Reference  
  HtmlNode Class  
  IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
HtmlNode.GetAttribute Method

Gets a single attribute value from the HtmlNode. Returns null if that attribute is not present.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

**C#**

```csharp
public string GetAttribute(
    string Attribute
)
```

**VB**

```vbnet
Public Function GetAttribute(ByVal Attribute As String) As String
End Function
```

### Parameters

*Attribute*  
Type: System.String

### Return Value

Type: String

### See Also

Reference  
HtmlNode Class  
IronWebScraper Namespace
HtmlNodeGetElementById Method

Synonym of JavaScript's getElementById function. Searches inside the current HtmlNode.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public HtmlNode GetElementById(
    string id
)
```

Parameters

`id`
Type: System.String

Return Value
Type: HtmlNode

See Also

Reference
(HtmlNode Class
IronWebScraper Namespace)
IronWebScraper - The C# Web Scraping Library
**HtmlNodeGetElementsByTagName Method**

Synonym of JavaScript's `getElementsByTagName` function. Searches inside the current `HtmlNode`.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

**C#**

```csharp
public HtmlNode[] GetElementsByTagName(  
    string tagName

)
```

**VB**

```vbnet
Public Function GetElementsByTagName(  
    ByVal tagName As String

) As HtmlNode
```

### Parameters

* **tagName**
  Type: System.String

### Return Value

Type: HtmlNode

### See Also

Reference  
HtmlNode Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
HtmlNodeHasAttribute Method

Tests if the HtmlNode has a given attribute such as "class" or "href"

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

C# VB

```csharp
public bool HasAttribute(
    string Attribute
)
```

Parameters

- **Attribute**
  - Type: System.String

Return Value

Type: Boolean

See Also

Reference

- HtmlNode Class
- IronWebScraper Namespace
HtmlNodeQuerySelector Method

Synonym of JavaScript's querySelector function. Searches inside the current HtmlNode.

**Namespace:**  IronWebScraper  
**Assembly:**  IronWebScraper (in IronWebScraper.dll)  
**Version:**  4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
</table>
| `public HtmlNode QuerySelector(  
   string selector  
)  | |

### Parameters

**selector**
Type: `System.String`

### Return Value
Type: `HtmlNode`

### See Also

**Reference**
- `HtmlNode Class`
- `IronWebScraper Namespace`
IronWebScraper - The C# Web Scraping Library
HtmlNodeQuerySelectorAll Method

Synonym of JavaScript's querySelectorAll function. Searches inside the current HtmlNode.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public HtmlNode[] QuerySelectorAll(
    string selector
)
```

Parameters

selector
    Type: System.String

Return Value
    Type: HtmlNode

See Also

Reference
    HtmlNode Class
    IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
**(HtmlNode)XPath Method**

Queries the HtmlNode to return all descendant nodes matching an XPath expression.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

**Syntax**

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>Copy</th>
</tr>
</thead>
</table>
| ```
public HtmlNode[] XPath(
        string expression
    )
``` |

**Parameters**

- **expression**  
  Type: SystemString

**Return Value**  
Type: HtmlNode

**See Also**

**Reference**  
HtmlNode Class  
IronWebScraper Namespace
HtmlNodeExtensions Class

Extension methods for finding elements within with IEnumerable<HtmlNode>

Inheritance Hierarchy

- System
  - Object
    - IronWebScraper
      - HtmlNodeExtensions

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

C# VB

```
public static class HtmlNodeExtensions
```

The HtmlNodeExtensions type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Star] ContainingText</td>
<td>Returns all HtmlNodes in the collection that contain the search text.</td>
</tr>
<tr>
<td>![Star] ContainingTextCaseInsensitive</td>
<td>Returns all HtmlNodes in the collection that contain the search text. Case</td>
</tr>
</tbody>
</table>
Insensitive.

Search multiple HtmlNodes for the given CSS Selector, and returns all distinct matches.

See Also

Reference
IronWebScraper Namespace
# HtmNodeExtensions Methods

The [HtmNodeExtensions](#) type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://example.com/s" alt="s" /> ContainingText</td>
<td>Returns all HtmNodes in the collection that contain the search text.</td>
</tr>
<tr>
<td><img src="https://example.com/s" alt="s" /> ContainingTextCaseInsensitive</td>
<td>Returns all HtmNodes in the collection that contain the search text. Case Insensitive.</td>
</tr>
<tr>
<td><img src="https://example.com/s" alt="s" /> CSS</td>
<td>Searched multiple HtmNodes for the given CSS Selector, and returns all distinct matches.</td>
</tr>
</tbody>
</table>

## See Also

Reference

[HtmNodeExtensions Class](#)
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
HtmlNodeExtensionsContainingText Method

Returns all HtmlNodes in the collection that contain the search text.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public static IEnumerable<HtmlNode> ContainingText(this IEnumerable<HtmlNode> nodes, string search)
```

### Parameters

- **nodes**  
  Type: `System.Collections.Generic.IEnumerable<HtmlNode>`  
  The nodes.

- **search**  
  Type: `System.String`  
  The search string.

### Return Value

Type: `IEnumerable<HtmlNode>`

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type `IEnumerable<HtmlNode>`. When you use instance method syntax to call this method, omit the first parameter. For more information, see [Extension Methods (Visual Basic)](https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/language-features/extension-methods) or
See Also

Reference

(HtmlNodeExtensions Class
IronWebScraper Namespace)
**HtmlNodeExtensionsContainingText**

Method

Returns all HtmlNodes in the collection that contain the search text. Case Insensitive.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

**Syntax**

```csharp
public static IEnumerable<HtmlNode> ContainingText
    (this IEnumerable<HtmlNode> nodes,
     string search)
```

**Parameters**

- `nodes`  
  Type: `System.Collections.Generic(IEnumerable<HtmlNode>)`  
  The nodes.

- `search`  
  Type: `System.String`  
  The search string.

**Return Value**  
Type: `IEnumerable<HtmlNode>`

**Usage Note**  
In Visual Basic and C#, you can call this method as an instance method on any object of type `IEnumerable<HtmlNode>`. When you use instance method syntax to call this method, omit the first parameter.
For more information, see Extension Methods (Visual Basic) or Extension Methods (C# Programming Guide).

See Also

Reference

(HtmlNodeExtensions Class)
(IronWebScraper Namespace)
IronWebScraper - The C# Web Scraping Library
HtmlNodeExtensionsCSS Method

Searched multiple HtmlNodes for the given CSS Selector, and returns all distinct matches.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public static HtmlNode[] CSS(
    this IEnumerable<HtmlNode> nodes,
    string selector
)
```

### Parameters

- **nodes**  
  Type: `System.Collections.Generic.IEnumerable<HtmlNode>`  
  HtmlNodes to search

- **selector**  
  Type: `System.String`  
  A CSS query selector

### Return Value

Type: `HtmlNode`

### Usage Note

In Visual Basic and C#, you can call this method as an instance method on any object of type `IEnumerable<HtmlNode>`. When you use instance method syntax to call this method, omit the first parameter.

For more information, see [Extension Methods (Visual Basic)](https://docs.microsoft.com/en-us/dotnet/visual-basic/tutorials/extension-methods) or
Extension Methods (C# Programming Guide).

See Also

Reference
HtmlNodeExtensions Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
HttpIdentity Class

A class defining the browsing 'identity' to be used to fetch a given Url. Contains Proxy, UserAgent and Http Header information.

Inheritance Hierarchy

- `System.Object`
- `IronWebScraper.HttpIdentity`

Namespace: `IronWebScraper`
Assembly: `IronWebScraper (in IronWebScraper.dll)` Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public class HttpIdentity
```

The `HttpIdentity` type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>HttpIdentity</code></td>
<td>Initializes a new instance of the <code>HttpIdentity</code> class</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Cookies</code></td>
<td>A &quot;Cookie Jar&quot; which persists http cookies for this browser</td>
</tr>
</tbody>
</table>
NetworkLoginCredential  Returns the System.Net.NetworkCredential which will be used on for networks and http authentication (Supports: Windows / NTLM / Kerberos / Linux / BSD / Mac)

ProxyFailureCount  Gets the number of times this proxy has failed.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DefaultHttpRequestHeaders</td>
<td>Default HTTP request headers to be assigned to new identity.</td>
</tr>
<tr>
<td>HttpRequestHeaders</td>
<td>The default headers which will be sent with all http Requests for this identity.</td>
</tr>
<tr>
<td><strong>NetworkDomain</strong></td>
<td>The network domain to be used for user authentication. Supports Windows, NTLM, Kerberos, Linux, Mac OS X networks. See also NetworkUsername, NetworkPassword, Proxy, UseCookies, UserAgent.</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>NetworkPassword</strong></td>
<td>The network/http password to be used for user authentication. Supports Http, Kerberos, Linux, Mac OS X networks. See also NetworkUsername, NetworkPassword, Proxy, UseCookies, UserAgent.</td>
</tr>
<tr>
<td><strong>NetworkUsername</strong></td>
<td>The network/http username to be used for user authentication. Supports Http, Kerberos, Linux, Mac OS X networks. See also NetworkPassword, Proxy, UseCookies, UserAgent.</td>
</tr>
<tr>
<td><strong>Proxy</strong></td>
<td>A proxy string such as &quot;102.192.92.192:8088&quot; or &quot;username:password@102.192.92.192&quot;</td>
</tr>
<tr>
<td><strong>UseCookies</strong></td>
<td>Stores a &quot;Cookie Jar&quot; which persists http cookies for this browser identity.</td>
</tr>
<tr>
<td><strong>UserAgent</strong></td>
<td>A default User-Agent header (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36. Find examples online at: <a href="http://www.useragentstring.com/pages/useragentstring.php">http://www.useragentstring.com/pages/useragentstring.php</a></td>
</tr>
</tbody>
</table>

**See Also**

Reference
IronWebScraper Namespace
HttpIdentity Constructor

Initializes a new instance of the HttpIdentity class

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```c#
public HttpIdentity()
```

See Also

Reference
HttpIdentity Class
IronWebScraper Namespace
# HttpIdentity Properties

The **HttpIdentity** type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cookies</td>
<td>A &quot;Cookie Jar&quot; which persists http cookies for this browser identity.</td>
</tr>
<tr>
<td>NetworkLoginCredential</td>
<td>Returns the System.Net.NetworkCredential which will be used on for networks and http authentication (Supports: Windows / NTLM / Kerberos / Linux / BSD / Mac)</td>
</tr>
<tr>
<td>ProxyFailureCount</td>
<td>Gets the number of times this proxy has failed.</td>
</tr>
</tbody>
</table>

**Top**

## See Also

**Reference**

- [HttpIdentity Class](#)
- [IronWebScraper Namespace](#)
IronWebScraper - The C# Web Scraping Library
HttpIdentityCookies Property

A "Cookie Jar" which persists http cookies for this browser identity.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Copy</td>
</tr>
</tbody>
</table>

```csharp
public CookieContainer Cookies { get; }
```

### Property Value

Type: CookieContainer

### See Also

Reference
- HttpIdentity Class
- IronWebScraper Namespace
HttpIdentityNetworkLoginCredential Property

Returns the System.Net.NetworkCredential which will be used on for networks and http authentication (Supports: Windows / NTLM / Kerberos / Linux / BSD / Mac)

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public NetworkCredential NetworkLoginCredential {</code></td>
<td></td>
</tr>
</tbody>
</table>

**Property Value**

Type: NetworkCredential

### See Also

Reference

HttpIdentity Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
HttpIdentityProxyFailureCount Property

Gets the number of times this proxy has failed.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

**C#**

```csharp
public int ProxyFailureCount { get; }
```

### Property Value

*Type:* `Int32`  
The proxy failure count.

### See Also

**Reference**  
HttpIdentity Class  
IronWebScraper Namespace
HttpIdentity Methods

The `HttpIdentity` type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

See Also

Reference

`HttpIdentity Class`

`IronWebScraper Namespace`
# HttpIdentity Fields

The **HttpIdentity** type exposes the following members.

## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>s</strong> DefaultHttpRequestHeaders</td>
<td>Default HTTP request headers to be assigned to new identity.</td>
</tr>
<tr>
<td><strong>s</strong> HttpRequestHeaders</td>
<td>The default headers which will be sent with all HTTP requests for this identity.</td>
</tr>
<tr>
<td><strong>s</strong> NetworkDomain</td>
<td>The network domain to be used for user authentication. Supports Windows, NTLM, Kerberos, Linux networks, Mac OS X networks. See also NetworkUsername, NetworkPassword.</td>
</tr>
<tr>
<td><strong>s</strong> NetworkPassword</td>
<td>The network/http password to be used for user authentication. Supports Http, Kerberos, Linux networks, BSD networks and Mac OS X networks. See also NetworkUsername, NetworkDomain.</td>
</tr>
<tr>
<td><strong>s</strong> NetworkUsername</td>
<td>The network/http username to be used for user authentication. Supports Http, Kerberos, Linux networks, BSD networks and Mac OS X networks. See also NetworkPassword, NetworkDomain.</td>
</tr>
<tr>
<td><strong>s</strong> Proxy</td>
<td>A proxy string such as &quot;102.1 username:password@102.1&quot;.</td>
</tr>
<tr>
<td><strong>s</strong> UseCookies</td>
<td>Stores a &quot;Cookie Jar&quot; which persists cookies for this browser identity.</td>
</tr>
</tbody>
</table>
UserAgent

A default User-Agent header (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36.

Find examples online at: http://www.useragentstring.com/pages/useragentstring.php

See Also

Reference
HttpIdentity Class
IronWebScraper Namespace
HttpIdentity.DefaultHttpRequestHeaders Field

Default HTTP request headers to be assigned to every new identity.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public static Dictionary&lt;string, string&gt; DefaultHttpRequestHeaders;</code></td>
<td></td>
</tr>
</tbody>
</table>

**Field Value**

Type: `Dictionary<string, string>`

### See Also

**Reference**

- HttpIdentity Class
- IronWebScraper Namespace
HttpIdentity.HttpRequestHeaders Field

The default headers which will be sent with all http Requests for this identity.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public Dictionary<string, string> HttpRequestHeaders
```

### Field Value

Type: `Dictionary<string, string>`

### See Also

Reference  
HttpIdentity Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
HttpIdentityNetworkDomain Field

The network domain to be used for user authentication. Supports Windows, NTLM, Kerberos, Linux, BSD and Mac OS X networks. See also NetworkUsername, NetworkPassword

**Namespace:** IronWebScraper
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

**C#**

```
public string NetworkDomain
```

**VB**

```
Public String NetworkDomain
```

**Field Value**

Type: String

### See Also

**Reference**

HttpIdentity Class
IronWebScraper Namespace
HttpIdentityNetworkPassword Field

The network/http password to be used for user authentication. Supports Http, Windows networks, NTLM, Kerberos, Linux networks, BSD networks and Mac OS X networks. See also NetworkUsername, NetworkDomain

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

## Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```csharp
public string NetworkPassword
```

Field Value
Type: String

## See Also

Reference
HttpIdentity Class
IronWebScraper Namespace
HttpIdentityNetworkUsername Field

The network/http username to be used for user authentication. Supports Http, Windows networks, NTLM, Kerberos, Linux networks, BSD networks and Mac OS X networks. See also NetworkPassword, NetworkDomain

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>public string NetworkUsername</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field Value
Type: String

See Also

Reference
HttpIdentity Class
IronWebScraper Namespace
HttpIdentityProxy Field

A proxy string such as "102.192.92.192:8088" or "username:password@102.192.92.192"

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public string Proxy</td>
<td></td>
</tr>
</tbody>
</table>

Field Value
Type: String

See Also

Reference
HttpIdentity Class
IronWebScraper Namespace
HttpIdentity.UseCookies Field

Stores a "Cookie Jar" which persists http cookies for this browser identity.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public bool UseCookies</td>
<td></td>
</tr>
</tbody>
</table>

Field Value
Type: Boolean

See Also

Reference
HttpIdentity Class
IronWebScraper Namespace
HttpIdentityUserAgent Field

A default User-Agent header string. E.g: "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36".
Find examples online at:
http://www.useragentstring.com/pages/useragentstring.php

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public string UserAgent</td>
<td></td>
</tr>
</tbody>
</table>

Field Value
Type: String

See Also

Reference
HttpIdentity Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
License Class

A public static class used to apply license keys to IronWebScraper.

Inheritance Hierarchy

System \rightarrow Object \rightarrow IronWebScraper \rightarrow License

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public static class License
```

The License type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsValidLicense</td>
<td>Determines whether a license key is valid.</td>
</tr>
</tbody>
</table>

Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
A License key can also be added to Web.Config or App.Config as IronWebScraper.LicenseKey

See Also

Reference
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
License Methods

The License type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsValidLicense</td>
<td>Determines whether a license key is valid.</td>
</tr>
</tbody>
</table>

See Also

Reference
License Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
LicenseIsValidLicense Method

Determines whether a license key is valid.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
Version: 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public static bool IsValidLicense(
    string LicenseKey
)
```

### Parameters

- **LicenseKey**
  - Type: `System.String`
  - IronWebScraper license key as a string

### Return Value

- Type: `Boolean`

### See Also

- Reference
  - License Class
  - IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
License Fields

The License type exposes the following members.

## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LicenseKey</td>
<td>Unlocks IronWebScraper for full professional usage. Get Licensed at</td>
</tr>
<tr>
<td></td>
<td><a href="http://ironsoftware.com/csharp/webscraper/licensing/">http://ironsoftware.com/csharp/webscraper/licensing/</a></td>
</tr>
<tr>
<td></td>
<td>A License key can also be added to Web.Config as IronWebScraper.LicenseKey</td>
</tr>
</tbody>
</table>

### See Also

**Reference**
- License Class
- IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
LicenseLicenseKey Field

Unlocks IronWebScraper for full professional usage. Get Licensed at http://ironsoftware.com/csharp/webscraper/licensing/ A License key can also be added to Web.Config or App.Config as IronWebScraper.LicenseKey

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static string LicenseKey</td>
<td></td>
</tr>
</tbody>
</table>

Field Value  
**Type:** String

### See Also

Reference  
License Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
**MetaData Class**

A flexible dictionary of object values which can be used to attach your own additional data or objects to any Request. Meta can contain objects of any Type including instances of classes, List and Dictionaries. This meta-data can then be accessed while Paring the Response and even passed forwards to the next Request.

Metadata send might include pagination page numbers, referrer Urls, User Ids etc.

E.g:

▲ **Examples**

```csharp
Request["page-number"] = 2;

int pageNumber = Response.Request.Meta.Get<
```

▲ **Inheritance Hierarchy**

```
SystemObject  System.Collections.GenericDictionaryString, Object
IronWebScraperMetaData
```

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

▲ **Syntax**

```csharp
public class MetaData : Dictionary<string, Object
```

The MetaData type exposes the following members.
## Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MetaData</strong></td>
<td>Initializes a new instance of the <strong>MetaData</strong> class</td>
</tr>
</tbody>
</table>

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Comparer</strong></td>
<td>(Inherited from <strong>Dictionary</strong>&lt;<strong>String</strong>, <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Count</strong></td>
<td>(Inherited from <strong>Dictionary</strong>&lt;<strong>String</strong>, <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Item</strong></td>
<td>(Inherited from <strong>Dictionary</strong>&lt;<strong>String</strong>, <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Keys</strong></td>
<td>(Inherited from <strong>Dictionary</strong>&lt;<strong>String</strong>, <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Values</strong></td>
<td>(Inherited from <strong>Dictionary</strong>&lt;<strong>String</strong>, <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add</strong></td>
<td>(Inherited from <strong>Dictionary</strong>&lt;<strong>String</strong>, <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>Clear</strong></td>
<td>(Inherited from <strong>Dictionary</strong>&lt;<strong>String</strong>, <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>
### Dictionary\<\String, \Object\>.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContainsKey</td>
<td>(Inherited from Dictionary&lt;\String, \Object&gt;)</td>
</tr>
<tr>
<td>ContainsValue</td>
<td>(Inherited from Dictionary&lt;\String, \Object&gt;)</td>
</tr>
<tr>
<td>Equals</td>
<td>(Inherited from \Object)</td>
</tr>
<tr>
<td>Get&lt;\Object&gt;</td>
<td>Returns the specified meta object cast as the appropriate type.</td>
</tr>
</tbody>
</table>

#### Examples

E.g:

```csharp
Get\<\int\>("index")
```

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetEnumerator</td>
<td>(Inherited from Dictionary&lt;\String, \Object&gt;)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from \Object)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>(Inherited from Dictionary&lt;\String, \Object&gt;)</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from \Object)</td>
</tr>
<tr>
<td>OnDeserialization</td>
<td>(Inherited from Dictionary&lt;\String, \Object&gt;)</td>
</tr>
<tr>
<td>Remove</td>
<td>(Inherited from Dictionary&lt;\String, \Object&gt;)</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from \Object)</td>
</tr>
<tr>
<td>TryGetValue</td>
<td>(Inherited from \Object)</td>
</tr>
</tbody>
</table>
DictionaryString, Object.

Top

See Also

Reference
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
MetaData Constructor

Initializes a new instance of the MetaData class

**Namespace:**  IronWebScraper  
**Assembly:**  IronWebScraper (in IronWebScraper.dll)  
**Version:**  4.0.4.25470 (4.0.4.3)

### Syntax

**C#**

```csharp
public MetaData()
```

### See Also

**Reference**

- MetaData Class
- IronWebScraper Namespace
# MetaData Properties

The `MetaData` type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparer</td>
<td>(Inherited from <code>Dictionary&lt;String, Object.&gt;</code>)</td>
</tr>
<tr>
<td>Count</td>
<td>(Inherited from <code>Dictionary&lt;String, Object.&gt;</code>)</td>
</tr>
<tr>
<td>Item</td>
<td>(Inherited from <code>Dictionary&lt;String, Object.&gt;</code>)</td>
</tr>
<tr>
<td>Keys</td>
<td>(Inherited from <code>Dictionary&lt;String, Object.&gt;</code>)</td>
</tr>
<tr>
<td>Values</td>
<td>(Inherited from <code>Dictionary&lt;String, Object.&gt;</code>)</td>
</tr>
</tbody>
</table>

## See Also

Reference

- [MetaData Class](#)
- [IronWebScraper Namespace](#)
IronWebScraper - The C# Web Scraping Library
The `MetaData` type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>(Inherited from <code>DictionaryString</code>, <code>Object</code>.)</td>
</tr>
<tr>
<td>Clear</td>
<td>(Inherited from <code>DictionaryString</code>, <code>Object</code>.)</td>
</tr>
<tr>
<td>ContainsKey</td>
<td>(Inherited from <code>DictionaryString</code>, <code>Object</code>.)</td>
</tr>
<tr>
<td>ContainsValue</td>
<td>(Inherited from <code>DictionaryString</code>, <code>Object</code>.)</td>
</tr>
<tr>
<td>Equals</td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetT</td>
<td>Returns the specified meta object cast as the appropriate type.</td>
</tr>
</tbody>
</table>

### Examples

E.g:

```
Get<int>("index")
```
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetHashCode</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>(Inherited from DictionaryString, Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>OnDeserialization</td>
<td>(Inherited from DictionaryString, Object.)</td>
</tr>
<tr>
<td>Remove</td>
<td>(Inherited from DictionaryString, Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>TryGetValue</td>
<td>(Inherited from DictionaryString, Object.)</td>
</tr>
</tbody>
</table>

**See Also**

Reference
MetaData Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
**MetaDataGetT Method**

Returns the specified meta object cast as the appropriate type.

### Examples

E.g:

Get<int>("index")

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public T Get<T>(
    string key
)
```

**Parameters**

- `key`  
  Type: System.String

**Type Parameters**

- `T`  
  Type

**Return Value**

Type: `T`
Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyNotFoundException</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference
- MetaData Class
- IronWebScraper Namespace
Request Class

Represents a http request to be made by IronWebScraper

▲ Inheritance Hierarchy

```
System
Object
IronWebScraperRequest
```

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

▲ Syntax

C#
```
public class Request
```

The Request type exposes the following members.

▲ Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ]</td>
<td>Request</td>
</tr>
</tbody>
</table>

▲ Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ]</td>
<td>Method</td>
</tr>
<tr>
<td>![ ]</td>
<td>Retries</td>
</tr>
</tbody>
</table>
made to fetch content from this Request so far.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Url</td>
<td>Absolute URL to be scraped.</td>
</tr>
</tbody>
</table>

**Methods**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Hash</td>
<td>Hashing function for uniquely identifying web requests.</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from Object.)</td>
</tr>
</tbody>
</table>

**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>The Action&lt;Response&gt; which will be used to parse the response. See IronWebScraper.Parse.</td>
</tr>
<tr>
<td>Identity</td>
<td>An HttpIdentity comprising Proxy, Headers, UserAgent etc.. which will be used to fetch the Request.</td>
</tr>
<tr>
<td>MetaData</td>
<td>Allows additional meta-data of any Type to be attached to a request, and retrieved when the subsequent</td>
</tr>
</tbody>
</table>
Response is parsed.

See Also

Reference
IronWebScraper Namespace
Request Constructor

Initializes a new instance of the `Request` class

**Namespace:**  IronWebScraper  
**Assembly:**  IronWebScraper (in IronWebScraper.dll)  
**Version:**  4.0.4.25470 (4.0.4.3)

### Syntax

**C#**

```csharp
public Request()
```

**VB**

```vbnet
Public Function Request()
```

### See Also

Reference

- Request Class
- IronWebScraper Namespace
Request Properties

The Request type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>Http Request method.</td>
</tr>
<tr>
<td>Retries</td>
<td>Number of attempts which have been made to fetch content from this Request so far.</td>
</tr>
<tr>
<td>Url</td>
<td>Absolute URL to be scraped.</td>
</tr>
</tbody>
</table>

See Also

Reference
- Request Class
- IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
RequestMethod Property

Http Request method.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>public string Method { get; }</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Property Value**

Type: String  
"GET" or "POST"

### See Also

**Reference**

Request Class  
IronWebScraper Namespace
RequestRetries Property

Number of attempts which have been made to fetch content from this Request so far.

**Namespace:**  IronWebScraper  
**Assembly:**  IronWebScraper (in IronWebScraper.dll)  
**Version:**  4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public int Retries { get; }</code></td>
<td></td>
</tr>
</tbody>
</table>

**Property Value**

Type: `Int32`

### See Also

- Reference
  - Request Class
  - IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
RequestUrl Property

Absolute URL to be scraped.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

⚠️ Syntax

C#  
```csharp
public string Url { get; set; }
```

VB

```vbnet
Public String Url Get; Set;
```

Property Value  
Type: String

⚠️ See Also

Reference  
Request Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
# Request Methods

The `Request` type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>Hash</strong></td>
<td>Hashing function for uniquely identifying web requests.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>

See Also

Reference

- Request Class
- IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
RequestHash Method

Hashing function for uniquely identifying web requests.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

## Syntax

```csharp
public static string Hash(  
    Request Request
)
```

## Parameters

- **Request**  
  Type: IronWebScraperRequest  
  A Request object

## Return Value

Type: String

## See Also

- Reference  
  Request Class  
  IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
# Request Fields

The Request type exposes the following members.

## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>The Action&lt;Response&gt; which will be used to parse the response. See IronWebScraper.Parse.</td>
</tr>
<tr>
<td>Identity</td>
<td>An HttpIdentity comprising Proxy, Headers, UserAgent etc.. which will be used to fetch the Request.</td>
</tr>
<tr>
<td>MetaData</td>
<td>Allows additional meta-data of any Type to be attached to a request, and retrieved when the subsequent Response is parsed.</td>
</tr>
</tbody>
</table>

## See Also

Reference
Request Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
RequestAction Field

The Action<Response> which will be used to parse the response. See IronWebScraper.Parse.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public Action&lt;Response&gt; Action</td>
<td></td>
</tr>
</tbody>
</table>

Field Value
Type: ActionResponse

See Also

Reference
Request Class
IronWebScraper Namespace
RequestIdentity Field

An HttpIdentity comprising Proxy, Headers, UserAgent etc.. which will be used to fetch the Request.

**Namespace:**  IronWebScraper  
**Assembly:**  IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public</td>
<td>HttpIdentity Identity</td>
</tr>
</tbody>
</table>

Field Value  
Type: HttpIdentity

### See Also

- Reference  
  - Request Class  
  - IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
RequestMetaData Field

Allows additional meta-data of any Type to be attached to a request, and retrieved when the subsequent Response is parsed.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
</table>

```csharp
public MetaData MetaData
```

Field Value
Type: MetaData

See Also

Reference
Request Class
IronWebScraper Namespace
Response Class

Represents a http response made by IronWebScraper

▲ Inheritance Hierarchy

System Object IronWebScraperResponse

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

▲ Syntax

C# VB

```
public class Response
```

The Response type exposes the following members.

▲ Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>Initializes a new instance of the Response class</td>
</tr>
</tbody>
</table>

▲ Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BinaryContent</td>
<td>The content downloaded from the Url as raw binary data</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Document</td>
<td>Returns an HtmlNode for the entire Response Document. (HTML/XML)</td>
</tr>
<tr>
<td>Html</td>
<td>Synonym of content which checks for an Html content</td>
</tr>
<tr>
<td>MetaData</td>
<td>Returns additional meta-data which was attached to the Request.</td>
</tr>
<tr>
<td>RequestUrl</td>
<td>The Url from which the Response was requested. This may differ from FinalUrl if these was an http redirect.</td>
</tr>
<tr>
<td>TextContent</td>
<td>The content downloaded from the Url encoded as a String.</td>
</tr>
<tr>
<td>WasSuccessful</td>
<td>Gets a value indicating whether this HTTP request yielded a successful response code.</td>
</tr>
</tbody>
</table>

Top

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Css</td>
<td>Uses CSS selectors to find all matching nodes within the Response Document. This works the same way as $('ClassName') in jQuery or querySelectorAll() in JavaScript.</td>
</tr>
<tr>
<td><strong>CssExists</strong></td>
<td>Uses CSS selectors to find if there are any matching nodes within the Response Document. This works the same way as <code>$('ClassName').length &gt; 0</code> in jQuery or <code>querySelectorAll().length &gt; 0</code> in JavaScript.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetElementById</strong></td>
<td>Synonym of JavaScript <code>getElementById</code> function. Searches inside the response as an HTML or XML Document by ID attribute.</td>
</tr>
<tr>
<td><strong>GetElementsByTagName</strong></td>
<td>Synonym of JavaScript <code>getElementById</code> function. Searches inside the response as an HTML or XML Document by tag-name such as &quot;a&quot; or &quot;img&quot;.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>QuerySelector</strong></td>
<td>Synonym of JavaScript <code>querySelector</code> function. Searches inside the Response document as an HTML or XML Document using CSS</td>
</tr>
</tbody>
</table>
selectors.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QuerySelectorAll</td>
<td>Synonym of JavaScript's <code>querySelectorAll</code> function. Searches inside the response document as an HTML or XML Document using CSS selectors.</td>
</tr>
<tr>
<td>ToAbsoluteUrl(IEnumerable&lt;String&gt;)</td>
<td>Makes any relative url strings absolute relative to this Response Document.</td>
</tr>
<tr>
<td>ToAbsoluteUrl(String)</td>
<td>Makes a relative url strings absolute relative to this Response Document.</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>XPath</td>
<td>Searches inside the response as an HTML / XML Document using an XPath expression.</td>
</tr>
</tbody>
</table>

**Fields**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CharSet</td>
<td>Name of the charset used to encode the web resource. When reading TextContent, the result has already been decoded to C# default encoding (utf-8) on your behalf.</td>
</tr>
<tr>
<td><strong>CreatedFromWebCache</strong></td>
<td>True if the Response was created from a cached version of the Url rather than from live data. See WebScraper.EnableWebCache() method, which is can be called from within the WebScraper.Init() method.</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>FinalUrl</strong></td>
<td>The Url from which the Response was returned. This may differ from RequestUrl if these was an http redirect.</td>
</tr>
<tr>
<td><strong>MimeType</strong></td>
<td>The MIME type of the Response content. E.g: &quot;text/html&quot;</td>
</tr>
<tr>
<td><strong>Request</strong></td>
<td>The Request from which the Response was created.</td>
</tr>
<tr>
<td><strong>StatusCode</strong></td>
<td>The http status code given by the url which as 200 or 404. 0 is returned in case the server is uncontactable.</td>
</tr>
</tbody>
</table>

**See Also**

Reference
IronWebScraper Namespace
Response Constructor

Initializes a new instance of the Response class

**Namespace:**  IronWebScraper

**Assembly:**  IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

▶ Syntax

```csharp
public Response()
```

▶ See Also

Reference

- Response Class
- IronWebScraper Namespace
Response Properties

The **Response** type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BinaryContent</strong></td>
<td>The content downloaded from the Url as raw binary data</td>
</tr>
<tr>
<td><strong>Document</strong></td>
<td>Returns an HtmlNode for the entire Response Document. (HTML/XML)</td>
</tr>
<tr>
<td><strong>Html</strong></td>
<td>Synonym of content which checks for an Html content</td>
</tr>
<tr>
<td><strong>MetaData</strong></td>
<td>Returns additional meta-data which was attached to the Request.</td>
</tr>
<tr>
<td><strong>RequestUrl</strong></td>
<td>The Url from which the Response was requested. This may differ from FinalUrl if these was an http redirect.</td>
</tr>
<tr>
<td><strong>TextContent</strong></td>
<td>The content downloaded from the Url encoded as a String.</td>
</tr>
<tr>
<td><strong>WasSuccessful</strong></td>
<td>Gets a value indicating whether this HTTP request yielded a successful response code.</td>
</tr>
</tbody>
</table>
See Also

Reference
Response Class
IronWebScraper Namespace
ResponseBinaryContent Property

The content downloaded from the URL as raw binary data

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public byte[] BinaryContent { get; }</code></td>
<td></td>
</tr>
</tbody>
</table>

**Property Value**

Type: **Byte**

### See Also

Reference  
**Response Class**  
**IronWebScraper Namespace**
IronWebScraper - The C# Web Scraping Library
ResponseDocument Property

Returns an HtmlNode for the entire Response Document. (HTML/XML)

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

**Syntax**

```csharp
public HtmlNode Document { get; }
```

**Property Value**

Type: HtmlNode

**See Also**

Reference
- Response Class
- IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
ResponseHtml Property

Synonym of content which checks for an Html content

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

▷ Syntax

```csharp
public string Html { get; }
```

Property Value
Type: String
The HTML.

▷ Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormatException</td>
<td>Document does not have an HTML mime type or contain a &lt;body&gt; or &lt;html&gt; tag</td>
</tr>
</tbody>
</table>

▷ See Also

Reference
Response Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
ResponseMetaData Property

Returns additional meta-data which was attached to the Request.

**Namespace:**  IronWebScraper  
**Assembly:**  IronWebScraper (in IronWebScraper.dll)  
**Version:**  4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public MetaData MetaData { get; }</code></td>
<td></td>
</tr>
</tbody>
</table>

Property Value  
**Type:**  `MetaData`

### See Also

Reference  
**Response Class**  
**IronWebScraper Namespace**
ResponseRequestUrl Property

The URL from which the Response was requested. This may differ from FinalUrl if there was an HTTP redirect.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public string RequestUrl { get; }</td>
<td></td>
</tr>
</tbody>
</table>

Property Value
Type: String

See Also

Reference
- Response Class
- IronWebScraper Namespace
ResponseTextContent Property

The content downloaded from the URL encoded as a String.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) **Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public string TextContent { get; }</code></td>
<td></td>
</tr>
</tbody>
</table>

Property Value

Type: **String**

### See Also

Reference

**Response Class**  
**IronWebScraper Namespace**
ResponseWasSuccessful Property

Gets a value indicating whether this HTTP request yielded a successful response code.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

C#     VB

```csharp
public bool WasSuccessful { get; }
```

Property Value

Type: Boolean
ture if the response was successful; otherwise, false.

See Also

Reference

Response Class
IronWebScraper Namespace
# Response Methods

The **Response** type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>** Css</td>
<td>Uses CSS selectors to find all matching nodes within the Response Document. This works the same way as <code>$('.ClassName')</code> in jQuery or <code>querySelectorAll()</code> in JavaScript.</td>
</tr>
<tr>
<td>** CssExists</td>
<td>Uses CSS selectors to find if there are any matching nodes within the Response Document. This works the same way as <code>$('.ClassName').length &gt; 0</code> in jQuery or <code>querySelectorAll().length &gt; 0</code> in JavaScript.</td>
</tr>
<tr>
<td>** Equals</td>
<td>(Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td>** GetElementById</td>
<td>Synonym of JavaScript <code>getElementById</code> function. Searches inside the response as an HTML or XML</td>
</tr>
<tr>
<td>Method/Property</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>GetElementsByTagName</strong></td>
<td>Synonym of JavaScript <code>getElementById</code> function. Searches inside the response as an HTML or XML Document by tag-name such as &quot;a&quot; or &quot;img&quot;.</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>(Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>(Inherited from <strong>Object</strong>)</td>
</tr>
<tr>
<td><strong>QuerySelector</strong></td>
<td>Synonym of JavaScript <code>querySelector</code> function. Searches inside the Response document as an HTML or XML Document using CSS selectors.</td>
</tr>
<tr>
<td><strong>QuerySelectorAll</strong></td>
<td>Synonym of JavaScript <code>querySelectorAll</code> function. Searches inside the response document as an HTML or XML Document using CSS selectors.</td>
</tr>
<tr>
<td><strong>ToAbsoluteUrl(IEnumerable&lt;String&gt;)</strong></td>
<td>Makes any relative url strings absolute relative to this Response Document.</td>
</tr>
<tr>
<td><strong>ToAbsoluteUrl(String)</strong></td>
<td>Makes a relative url strings absolute relative to this Response Document.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>ToString</code></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><code>XPath</code></td>
<td>Searches inside the response as an HTML/XML Document using a XPath expression.</td>
</tr>
</tbody>
</table>

**See Also**

- Reference
- `Response Class`
- `IronWebScraper Namespace`
ResponseCss Method

Uses CSS selectors to find all matching nodes within the Response Document. This works in the same way as $('.ClassName') in jQuery or querySelectorAll() in JavaScript.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) **Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public HtmlNode[] Css(  
    string selector  
)
```

#### Parameters

**selector**  
Type: System.String

#### Return Value

Type: HtmlNode

### See Also

Reference  
Response Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
ResponseCssExists Method

Uses CSS selectors to find if there are any matching nodes within the Response Document. This works in the same way as 
$('.'ClassName').length > 0 in jQuery or querySelectorAll().length >0 in JavaScript.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

**C#**

```csharp
public bool CssExists(
    string selector
)
```

**VB**

```vbnet
Public Function CssExists(ByVal selector As String) As Boolean
```

### Parameters

**selector**  
Type: **System.String**

### Return Value

Type: **Boolean**

### See Also

**Reference**

- Response Class
- IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
ResponseGetElementById Method

Synonym of JavaScript's getElementById function. Searches inside the response as an HTML or XML Document by ID attribute.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public HtmlNode GetElementById(
    string id
)
```

**Parameters**

- `id`  
  Type: `System.String`

**Return Value**

Type: `HtmlNode`

### See Also

- Reference
- **Response Class**
- **IronWebScraper Namespace**
IronWebScraper - The C# Web Scraping Library
ResponseGetElementsByTagName Method

Synonym of JavaScript's getElementById function. Searches inside the response as an HTML or XML Document by tag-name such as "a" or "img".

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public HtmlNode[] GetElementsByTagName(
    string tagName
)
```

Parameters

tagName
    Type: System.String

Return Value
    Type: HtmlNode

See Also

Reference
Response Class
IronWebScraper Namespace
ResponseQuerySelector Method


**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public HtmlNode QuerySelector(</code></td>
<td><code>Public Function QuerySelector(</code></td>
</tr>
<tr>
<td><code>string selector)</code></td>
<td><code>String selector)</code></td>
</tr>
</tbody>
</table>

**Parameters**

- `selector`
  - Type: `System.String`

**Return Value**

- Type: `HtmlNode`

### See Also

- **Reference**
  - Response Class
  - IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
ResponseQuerySelectorAll Method

Synonym of JavaScript's querySelectorAll function. Searches inside the response document as an HTML or XML Document using CSS selectors.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public HtmlNode[] QuerySelectorAll(
    string selector
)
```

**Parameters**

- **selector**
  - Type: System.String

**Return Value**

- Type: HtmlNode

### See Also

- Reference
- Response Class
- IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
ResponseToAbsoluteUrl Method

- **Overload List**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ToAbsoluteUrl(IEnumerable&lt;String&gt;)</strong></td>
<td>Makes any relative url strings absolute relative to this Response Document.</td>
</tr>
<tr>
<td><strong>ToAbsoluteUrl(String)</strong></td>
<td>Makes a relative url strings absolute relative to this Response Document.</td>
</tr>
</tbody>
</table>

**See Also**

Reference
- Response Class
- IronWebScraper Namespace
ResponseToAbsoluteUrl Method (IEnumerable<String>)

Makes any relative url strings absolute relative to this Response Document.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```csharp
public IEnumerable<string> ToAbsoluteUrl(
    IEnumerable<string> urls
)
```

**Parameters**

- **urls**
  - Type: `System.Collections.Generic.IEnumerable<String>

**Return Value**

- Type: `IEnumerable<String>`

### See Also

- **Reference**
  - Response Class
  - ToAbsoluteUrl Overload
  - IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
ResponseToAbsoluteUrl Method (String)

Makes a relative url strings absolute relative to this Response Document.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

**C#**

```csharp
public string ToAbsoluteUrl(
    string url
)
```

**VB**

```vbnet
Public Function ToAbsoluteUrl(
    url As String
) As String
```

### Parameters

- **url**  
  Type: `System.String`

### Return Value

Type: `String`

### See Also

Reference  
- Response Class  
- ToAbsoluteUrl Overload  
- IronWebScraper Namespace
ResponseXPath Method

Searches inside the response as an HTML / XML Document using an XPath expression.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

**Syntax**

C#       VB

```csharp
public HtmlNode[] XPath(
    string expression
)
```

**Parameters**

- `expression`  
  Type: SystemString

**Return Value**  
Type: HtmlNode

**See Also**

Reference  
Response Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
Response Fields

The **Response** type exposes the following members.

### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CharSet</td>
<td>Name of the charset used to encode the web resource. When reading TextContent, the result has already been decoded to C# default encoding (utf-8) on your behalf.</td>
</tr>
<tr>
<td>CreatedFromWebCache</td>
<td>True if the Response was created from a cached version of the Url rather than from live data. See WebScraper.EnableWebCache() method, which is can be called from within the WebScraper.Init() method.</td>
</tr>
<tr>
<td>FinalUrl</td>
<td>The Url from which the Response was returned. This may differ from Request.Url if these was an http redirect.</td>
</tr>
<tr>
<td>MimeType</td>
<td>The MIME type of the Response content. E.g: &quot;text/html&quot;</td>
</tr>
<tr>
<td>Request</td>
<td>The Request from which the Response was created.</td>
</tr>
<tr>
<td>StatusCode</td>
<td>The http status code given by</td>
</tr>
</tbody>
</table>
the url which as 200 or 404. 0 is returned in case the server is uncontactable.

See Also

Reference
Response Class
IronWebScraper Namespace
ResponseCharSet Field

Name of the charset used to encode the web resource. When reading TextContent, the result has already been decoded to C# default encoding (utf-8) on your behalf.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public string CharSet</td>
<td></td>
</tr>
</tbody>
</table>

Field Value
Type: String

See Also

Reference
Response Class
IronWebScraper Namespace
ResponseCretaedFromWebCache Field

True if the Response was created from a cached version of the Url rather than from live data. See WebScraper.EnableWebCache() method, which is can be called from within the WebScraper.Init() method.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public bool CretaedFromWebCache
```

Field Value  
Type: Boolean

### See Also

Reference  
Response Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
ResponseFinalUrl Field

The Url from which the Response was returned. This may differ from RequestlUrl if these was an http redirect.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public string FinalUrl</code></td>
<td></td>
</tr>
</tbody>
</table>

### Field Value

Type: **String**

### See Also

**Reference**  
Response Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
ResponseMimeType Field

The MIME type of the Response content. E.g: "text/html"

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public string MimeType
```

Field Value

Type: String

See Also

Reference
Response Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
ResponseRequest Field

The Request from which the Response was created.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

▲ **Syntax**

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public Request Request</code></td>
<td></td>
</tr>
</tbody>
</table>

**Field Value**  
Type: `Request`

▲ **See Also**

**Reference**  
Response Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
ResponseStatusCode Field

The http status code given by the url which as 200 or 404. 0 is returned in case the server is uncontactable.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
<th>Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>public</td>
<td>int StatusCode</td>
<td></td>
</tr>
</tbody>
</table>

Field Value
Type: Int32

See Also

Reference
- Response Class
- IronWebScraper Namespace
ScrapedData Class

A flexible dictionary of object values used to conveniently store scraped data of any Type in a key-value dictionary which can be saved as JSON using the Yield method. ScrapedData can hold data objects of any Type, including Classes.

E.g:

```csharp
var Data = new ScrapedData();
Data['title'] = "Page Title";
Data['date'] = DateTime.Now;
```

Inheritance Hierarchy

SystemObject System.Collections.GenericDictionaryString, Object
IronWebScraper ScrapedData

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public class ScrapedData : Dictionary<string, Object> {
```

The ScrapedData type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**ScrapedData**  Initializes a new instance of the *ScrapedData* class

## Top

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Comparer</code></td>
<td>(Inherited from <em>DictionaryString</em>, <em>Object.</em>)</td>
</tr>
<tr>
<td><code>Count</code></td>
<td>(Inherited from <em>DictionaryString</em>, <em>Object.</em>)</td>
</tr>
<tr>
<td><code>Item</code></td>
<td>(Inherited from <em>DictionaryString</em>, <em>Object.</em>)</td>
</tr>
<tr>
<td><code>Keys</code></td>
<td>(Inherited from <em>DictionaryString</em>, <em>Object.</em>)</td>
</tr>
<tr>
<td><code>Values</code></td>
<td>(Inherited from <em>DictionaryString</em>, <em>Object.</em>)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Add</code></td>
<td>(Inherited from <em>DictionaryString</em>, <em>Object.</em>)</td>
</tr>
<tr>
<td><code>Clear</code></td>
<td>(Inherited from <em>DictionaryString</em>, <em>Object.</em>)</td>
</tr>
<tr>
<td><code>ContainsKey</code></td>
<td>(Inherited from <em>DictionaryString</em>, <em>Object.</em>)</td>
</tr>
</tbody>
</table>
ContainsValue  (Inherited from DictionaryString, Object.)

Equals  (Inherited from Object.)

ToJson  Deserializes a ScrapedData object from JSON created using the toJson() method.

GetT  Returns the specified ScrapedData item cast as the appropriate type.

Examples

E.g:

Get<string>("url");

GetEnumerator  (Inherited from DictionaryString, Object.)

GetHashCode  (Inherited from Object.)

GetObjectData  (Inherited from DictionaryString, Object.)

GetType  (Inherited from Object.)

OnDeserialization  (Inherited from DictionaryString, Object.)

Remove  (Inherited from DictionaryString, Object.)

ToJson  Converts the ScrapedData to a JSON string.

ToString  (Inherited from Object.)
TryGetValue (Inherited from Dictionary<String, Object>.)

See Also

Reference
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
ScrapedData Constructor

Initializes a new instance of the ScrapedData class

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public ScrapedData()
```

See Also

Reference
ScrapedData Class
IronWebScraper Namespace
# ScrapedData Properties

The **ScrapedData** type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Comparer</code></td>
<td>(Inherited from <code>DictionaryString, Object</code>)</td>
</tr>
<tr>
<td><code>Count</code></td>
<td>(Inherited from <code>DictionaryString, Object</code>)</td>
</tr>
<tr>
<td><code>Item</code></td>
<td>(Inherited from <code>DictionaryString, Object</code>)</td>
</tr>
<tr>
<td><code>Keys</code></td>
<td>(Inherited from <code>DictionaryString, Object</code>)</td>
</tr>
<tr>
<td><code>Values</code></td>
<td>(Inherited from <code>DictionaryString, Object</code>)</td>
</tr>
</tbody>
</table>

## See Also

- **Reference**
  - `ScrapedData Class`
  - `IronWebScraper Namespace`
IronWebScraper - The C# Web Scraping Library
## ScrappedData Methods

The *ScrappedData* type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![add_icon]</td>
<td>Add (Inherited from <em>Dictionary</em>&lt;String, Object&gt;.)</td>
</tr>
<tr>
<td>![clear_icon]</td>
<td>Clear (Inherited from <em>Dictionary</em>&lt;String, Object&gt;.)</td>
</tr>
<tr>
<td>![containskey_icon]</td>
<td>ContainsKey (Inherited from <em>Dictionary</em>&lt;String, Object&gt;.)</td>
</tr>
<tr>
<td>![containsvalue_icon]</td>
<td>ContainsValue (Inherited from <em>Dictionary</em>&lt;String, Object&gt;.)</td>
</tr>
<tr>
<td>![equals_icon]</td>
<td>Equals (Inherited from <em>Object</em>.)</td>
</tr>
<tr>
<td>![fromjson_icon]</td>
<td>FromJson Deserializes a ScrappedData object from JSON created using the toJson() method.</td>
</tr>
<tr>
<td>![get_icon]</td>
<td>GetT Returns the specified ScrappedData item cast as the appropriate type.</td>
</tr>
</tbody>
</table>

### Examples

E.g:

```csharp
Get<string>("url");
```
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Inherited From</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetEnumerator</td>
<td>(Inherited from Dictionary[String, Object].)</td>
<td></td>
</tr>
<tr>
<td>GetHashCode</td>
<td>(Inherited from Object.)</td>
<td></td>
</tr>
<tr>
<td>GetObjectData</td>
<td>(Inherited from Dictionary[String, Object].)</td>
<td></td>
</tr>
<tr>
<td>GetType</td>
<td>(Inherited from Object.)</td>
<td></td>
</tr>
<tr>
<td>OnDeserialization</td>
<td>(Inherited from Dictionary[String, Object].)</td>
<td></td>
</tr>
<tr>
<td>Remove</td>
<td>(Inherited from Dictionary[String, Object].)</td>
<td></td>
</tr>
<tr>
<td>ToJson</td>
<td>Converts the ScrapedData to a JSON string.</td>
<td></td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from Object.)</td>
<td></td>
</tr>
<tr>
<td>TryGetValue</td>
<td>(Inherited from Dictionary[String, Object].)</td>
<td></td>
</tr>
</tbody>
</table>

**See Also**

- Reference
- ScrapedData Class
- IronWebScraper Namespace
ScrapedDataFromJson Method

Deserializes a ScrapedData object from JSON created using the toJson() method.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

C#    VB

```csharp
public static ScrapedData FromJson(
    string json
)
```

Parameters

`json`
Type: System.String

Return Value
Type: ScrapedData

See Also

Reference
ScrapedData Class
IronWebScraper Namespace
ScrapedDataGetT Method

Returns the specified ScrapedData item cast as the appropriate type.

Examples

E.g:

```csharp
Get<string>("url");
```

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public T Get<T>(
    string key
)
```

Parameters

**key**
- Type: `System.String`
- Key

Type Parameters

**T**
- Type

Return Value
- Type: `T`
Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyNotFoundException</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Reference
ScrapedData Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
ScrapedDataToJson Method

Converts the ScrapedData to a JSON string.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

**C#**

```csharp
public stringToJson()
```

**VB**

```vbnet
Public Function ToJson() As String
```

### Return Value

Type: **String**

### See Also

Reference

- ScrapedData Class
- IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraper Class

A base class which developers can extend to build custom web-scraping applications.
An easy to use base class which developers can extend to rapidly build custom web-scraping applications.

Inheritance Hierarchy

- System
  - Object
  - IronWebScraper
  - WebScraper

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

C# | VB
---|---
```
public abstract class WebScraper
```

The `WebScraper` type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FailedUrls</td>
<td>Gets the number of failed http requests which have exceeded their total maximum number of retries.</td>
</tr>
<tr>
<td>HttpRetryAttempts</td>
<td>The number of</td>
</tr>
</tbody>
</table>
times WebScraper will retry a failed URL (normally with a new identity) before considering it non-scrapable.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HttpTimeOut</strong></td>
<td>Gets or the time after-which a HTTP request will be considered failed or lost. (non-contactable or Dns unavailable)</td>
</tr>
<tr>
<td><strong>MaxHttpConnectionLimit</strong></td>
<td>Gets or sets the total number of allowed open HTTP requests (threads)</td>
</tr>
<tr>
<td><strong>OpenConnectionLimitPerHost</strong></td>
<td>Gets or sets the allowed number of concurrent HTTP requests (threads) per hostname or IP address. This helps protect hosts against too many requests.</td>
</tr>
<tr>
<td><strong>RateLimitPerHost</strong></td>
<td>Gets or sets minimum polite delay (pause) between request</td>
</tr>
</tbody>
</table>
SuccessfulFileDownloadCount

Gets the number of successful http downloads using the DownloadFile and DownloadImage methods.

SuccessfulRequestCount

Gets the number of successful http requests.

ThrottleMode

Makes the WebScraper intelligently throttle requests not only by hostname, but also by host servers' IP addresses. This is polite in-case multiple scraped domains are hosted on the same machine.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AcceptUrl</td>
<td>Decides if the WebScraper will accept a given url. My be overridden to apply custom middleware logic.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>ChooseIdentityForRequest</strong></td>
<td>Picks a random identity from WebScraper.Identities for each request. Identities with proxy IP addresses, user agents, headers, cookies, username and password in your Init Method are added to the WebScraper.Identities List. Override this method to create your own logic for non-random selection of an HttpIdentity for each request.</td>
</tr>
<tr>
<td><strong>DownloadFile(String, String, Boolean, HttpIdentity)</strong></td>
<td>Requests a file to be downloaded from the given URL to the local file-system for scraping documents, assets and images. Normally called with an IronWebScraper.WebScraper Parse Method.</td>
</tr>
<tr>
<td><strong>DownloadFile(Uri, String, Boolean, HttpIdentity)</strong></td>
<td>Requests a file to be downloaded from the given URL to the local file-system for scraping documents, assets and images. Normally called with an IronWebScraper.WebScraper Parse Method.</td>
</tr>
<tr>
<td><strong>DownloadFileUnique</strong></td>
<td>Much like DownloadFile but if the file has already been downloaded or exists locally, it will not be re-downloaded. Requests a file to be downloaded from the given URL to the local file-system for scraping documents, assets and images. Normally called with an IronWebScraper.WebScraper Parse Method.</td>
</tr>
<tr>
<td><strong>DownloadImage(String, String, Int32, Int32, Boolean, HttpIdentity)</strong></td>
<td>Requests a file to be downloaded from the given URL to the local file-system for scraping documents, assets and images. Normally called with an IronWebScraper.WebScraper Parse Method.</td>
</tr>
<tr>
<td><strong>DownloadImage(Uri, String, Int32, Int32, Boolean, HttpIdentity)</strong></td>
<td>Requests a file to be downloaded from the given URL to the local file-system for scraping documents, assets and images. Normally called with an IronWebScraper.WebScraper Parse Method.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>EnableWebCache</strong></td>
<td>Caches web http responses for reuse. This allows WebScraper classes to be modified and restarted without re-downloading previously scraped urls.</td>
</tr>
<tr>
<td><strong>EnableWebCache(TimeSpan)</strong></td>
<td>Caches web http responses for reuse. This allows WebScraper classes to be modified and restarted without re-downloading previously scraped urls.</td>
</tr>
<tr>
<td><strong>Equals</strong></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>FetchUrlContents</strong></td>
<td>A handy shortcut method that fetches the text content from any Url (synchronously).</td>
</tr>
<tr>
<td><strong>FetchUrlContentsBinary</strong></td>
<td>A handy shortcut method that fetches the text content from any Url (synchronously) as a binary data in a byte array (byte[])</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Init</strong></td>
<td>Override this method in your web-scraper. Important tasks will be to set start url... and set allowed/banned domain or url patterns.</td>
</tr>
<tr>
<td><strong>Log</strong></td>
<td>Logs the specified message. Logs can be enabled using the EnableLogging. This function has been overridable to allow for easy Email and Slack notification integration.</td>
</tr>
<tr>
<td><strong>ObeyRobotsDotTxtForHost</strong></td>
<td>Causes the WebScraper to always obey /robots.txt directives including path restrictions and crawl rates on a domain basis.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Parse</td>
<td>Override this method to create the default Response handler for your web scraper. May be overridden for advanced control.</td>
</tr>
<tr>
<td>ParseWebscraperDownload</td>
<td>Internal method to parse binary files downloaded by a webScraper.</td>
</tr>
<tr>
<td>ParseWebscraperDownloadImage</td>
<td>Internal method to parse images downloaded by a webScraper.</td>
</tr>
<tr>
<td>PostRequest(String, ActionResponse, Dictionary&lt;String, String&gt;)</td>
<td>Request adds a new request to the scrape-job queue using the POST method.</td>
</tr>
<tr>
<td>PostRequest(Uri, ActionResponse, Dictionary&lt;String, String&gt;)</td>
<td>Request adds a new request to the scrape-job queue using the POST method.</td>
</tr>
<tr>
<td>PostRequest(String, ActionResponse, Dictionary&lt;String, String, MetaData&gt;)</td>
<td>Request adds a new request to the scrape-job queue using the POST method.</td>
</tr>
<tr>
<td>PostRequest(Uri, ActionResponse, Dictionary&lt;String, String, MetaData&gt;)</td>
<td>Request adds a new request to the scrape-job queue using the POST method.</td>
</tr>
<tr>
<td>PostRequest(String, ActionResponse, Dictionary&lt;String, String, HttpIdentity, MetaData&gt;)</td>
<td>Request adds a new request to the scrape-job queue using the POST method.</td>
</tr>
<tr>
<td>PostRequest(Uri, ActionResponse, Dictionary&lt;String, String, HttpIdentity, MetaData&gt;)</td>
<td>Request adds a new request to the scrape-job queue using the POST method.</td>
</tr>
<tr>
<td>Request(IEnumerable&lt;String&gt;, ActionResponse)</td>
<td>A key method called from Init and Parse Methods. Request adds new requests to the scrape-job queue, and</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Request(String, ActionResponse)</td>
<td>A key method called from Init and Parse Methods. Request adds to scrape-job queue, and (e.g. Parse) will be used object.</td>
</tr>
<tr>
<td>Request(Uri, ActionResponse)</td>
<td>A key method called from Init and Parse Methods. Request adds to scrape-job queue, and (e.g. Parse) will be used object.</td>
</tr>
<tr>
<td>Request(String, ActionResponse, MetaData)</td>
<td>A key method called from Init and Parse Methods. Request adds to scrape-job queue, and (e.g. Parse) will be used object.</td>
</tr>
<tr>
<td>Request(Uri, ActionResponse, MetaData)</td>
<td>A key method called from Init and Parse Methods. Request adds to scrape-job queue, and (e.g. Parse) will be used object.</td>
</tr>
<tr>
<td>Request(String, ActionResponse, HttpIdentity, MetaData)</td>
<td>A key method called from Init and Parse Methods. Request adds to scrape-job queue, and (e.g. Parse) will be used object.</td>
</tr>
<tr>
<td>Request(Uri, ActionResponse, HttpIdentity, MetaData)</td>
<td>A key method called from Init and Parse Methods. Request adds to scrape-job queue, and (e.g. Parse) will be used object.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Retry</td>
<td>Retries a Response. Usually called in a Parse method if a Captcha or error encountered during HTML parsing.</td>
</tr>
<tr>
<td>Scrape</td>
<td>Appends any scraped data to a file in the JsonLines format. (1 json object per line). Typically used with IronWebScraper.ScrapedData or developer defined classes for scraped data items. The default filename will follow the pattern &quot;NameSpace.TypeName.jsonl&quot;. Example: Scrape/IronWebScraper.ScrapedData.jsonl.</td>
</tr>
<tr>
<td>ScrapeUnique</td>
<td>Appends scraped data to a file in the JsonLines format. (1 json object per line). Ignores duplicates. Will save any .Net object of any kind. This method is typically used with IronWebScraper.ScrapedData or developer defined classes for scraped data items. The default filename will follow the pattern &quot;WorkingDirectory/NameSpace.TypeName.jsonl&quot;. Example: E.g: Scrape/IronWebScraper.ScrapedData.jsonl.</td>
</tr>
<tr>
<td>SetSiteSpecificCrawlRateLimit</td>
<td>Set a throttle limit for a specific domain.</td>
</tr>
<tr>
<td>Start</td>
<td>Starts the WebScraper. Set CrawlId to make this crawl resumable. Will also resume a previous crawl if it exists. Giving a CrawlId also causes the WebScraper to auto-save its state even in case of a crash, system failure or power outage. This feature is particularly useful for long-running web-scraping tasks, allowing hours, days or even weeks of work to be recovered effortlessly.</td>
</tr>
<tr>
<td>StartAsync</td>
<td>Starts the WebScraper Asynchronously. Set CrawlId to make this crawl resumable. Will resume a previous crawl if it exists.</td>
</tr>
</tbody>
</table>
if it exists.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop</td>
<td>Stops this WebScraper instance gracefully. WebScraper may be re-used after</td>
</tr>
<tr>
<td></td>
<td>of data by calling Start() or StartAsync().</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>UnScrape(Boolean)</td>
<td>Retrieves IronWebScraper.ScrapedData objects which were saved using the</td>
</tr>
<tr>
<td></td>
<td>method.</td>
</tr>
<tr>
<td>UnScrape(String, Boolean)</td>
<td>Retrieves IronWebScraper.ScrapedData objects which were saved using the</td>
</tr>
<tr>
<td></td>
<td>method.</td>
</tr>
<tr>
<td>UnScrapeT(Boolean)</td>
<td>Retrieves native C# objects which were saved using the WebScraper.Scrape</td>
</tr>
<tr>
<td></td>
<td>in the JsonLines format.</td>
</tr>
<tr>
<td>UnScrapeT(String, Boolean)</td>
<td>Retrieves native C# objects which were saved using the WebScraper.Scrape</td>
</tr>
<tr>
<td></td>
<td>in the JsonLines format.</td>
</tr>
</tbody>
</table>

### Top

#### Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllowedDomains</td>
<td>If not empty, all requested URLs' hostname must match at least one of the</td>
</tr>
<tr>
<td></td>
<td>AllowedDomains patterns. Patterns may be added using glob wildcard strings</td>
</tr>
<tr>
<td></td>
<td>or Regex.</td>
</tr>
<tr>
<td><strong>AllowedUrls</strong></td>
<td>If not empty, all requested URLs must match at least one of the AllowedUrls patterns. Patterns may be added using glob wildcard strings or Regex.</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>BannedDomains</strong></td>
<td>If not empty, no requested URLs' hostname may match any of the BannedDomains patterns. Patterns may be added using glob wildcard strings or Regex.</td>
</tr>
<tr>
<td><strong>BannedUrls</strong></td>
<td>If not empty, no requested URLs may match any of the BannedUrls patterns. Patterns may be added using glob wildcard strings or Regex.</td>
</tr>
<tr>
<td><strong>CrawlId</strong></td>
<td>A unique string used to identify a crawl job.</td>
</tr>
<tr>
<td><strong>FilesDownloaded</strong></td>
<td>The total number of files downloaded successfully with the DownloadImage and DownloadFile methods.</td>
</tr>
<tr>
<td><strong>Identities</strong></td>
<td>A list of http identities to be used to fetch web resources. Each Identity may have a different proxy IP addresses, userAgent, http headers, persistent cookies, username and password. Best practice is to create Identities in your WebScraper.Init Method and Add them to this WebScraper.Identities List.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LoggingLevel</td>
<td>The level of logging made by the WebScraper engine to the Console. LogLevel.Critical is normally the most useful setting, allowing the developer to write their own, meaningful and application relevant messages inside of Parse methods. LogLevel.ScrapedData is useful when coding and testing a new WebScraper.</td>
</tr>
<tr>
<td>ObeyRobotsDotTxt</td>
<td>Causes the WebScraper to always obey /robots.txt directives including url and path restrictions and crawl rates.</td>
</tr>
<tr>
<td>WorkingDirectory</td>
<td>Path to a local directory where scraped data and state information will be saved.</td>
</tr>
</tbody>
</table>
The **WebScraper** type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FailedUrls</td>
<td>Gets the number of failed http requests which have exceeded their total maximum number of retries.</td>
</tr>
<tr>
<td>HttpRetryAttempts</td>
<td>The number of times WebScraper will retry a failed URL (normally with a new identity) before considering it non-scrapable.</td>
</tr>
<tr>
<td>HttpTimeOut</td>
<td>Gets or the time after-which a HTTP request will be considered failed or lost. (non-contactable or Dns unavailable)</td>
</tr>
<tr>
<td>MaxHttpConnectionLimit</td>
<td>Gets or sets the</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>total number of allowed open</td>
<td>total number of allowed open HTTP requests (threads).</td>
</tr>
<tr>
<td>HTTP requests (threads)</td>
<td></td>
</tr>
<tr>
<td><strong>OpenConnectionLimitPerHost</strong></td>
<td>Gets or sets the allowed number of concurrent HTTP requests (threads) per</td>
</tr>
<tr>
<td></td>
<td>hostname or IP address. This helps protect hosts against too many requests.</td>
</tr>
<tr>
<td><strong>RateLimitPerHost</strong></td>
<td>Gets or sets minimum polite delay (pause) between request to a given domain</td>
</tr>
<tr>
<td></td>
<td>or IP address.</td>
</tr>
<tr>
<td><strong>SuccessfulFileDownloadCount</strong></td>
<td>Gets the number of successful http downloads using the DownloadFile and</td>
</tr>
<tr>
<td></td>
<td>DownloadImage methods.</td>
</tr>
<tr>
<td><strong>SuccessfulRequestCount</strong></td>
<td>Gets the number of successful http requests.</td>
</tr>
<tr>
<td><strong>ThrottleMode</strong></td>
<td>Makes the WebScraper intelligently</td>
</tr>
</tbody>
</table>
throttle requests not only by hostname, but also by host servers' IP addresses. This is polite in-case multiple scraped domains are hosted on the same machine.

See Also

Reference
WebScraper Class
IronWebScraper Namespace
**WebScraperFailedUrls Property**

Gets the number of failed http requests which have exceeded their total maximum number of retries.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public int FailedUrls { get; }</td>
<td>Copy</td>
</tr>
</tbody>
</table>

**Property Value**  
Type: Int32

### See Also

- Reference  
  - WebScraper Class  
  - IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraper.HttpRetryAttempts Property

The number of times WebScraper will retry a failed URL (normally with a new identity) before considering it non-scrappable.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public int HttpRetryAttempts</code> { get; set; }</td>
<td><code>Public Function HttpRetryAttempts As Integer</code></td>
</tr>
</tbody>
</table>

Property Value  
Type: `Int32`

### See Also

Reference  
WebScraper Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperHttpTimeOut Property

Gets or the time after-which a HTTP request will be considered failed or lost. (non-contactable or Dns unavailable)

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public TimeSpan HttpTimeOut { get; set; }</code></td>
<td></td>
</tr>
</tbody>
</table>

**Property Value**  
Type: TimeSpan

### See Also

**Reference**  
WebScraper Class  
IronWebScraper Namespace
WebScraperMaxHttpConnectionLimit Property

Gets or sets the total number of allowed open HTTP requests (threads)

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

**C#**

```csharp
public int MaxHttpConnectionLimit { get; set; }
```

**VB**

```vbnet
Public Property MaxHttpConnectionLimit As Integer
End Property
```

**Property Value**

Type: Int32

### See Also

**Reference**

WebScraper Class  
IronWebScraper Namespace
WebScraperOpenConnectionLimitPerHost Property

Gets or sets the allowed number of concurrent HTTP requests (threads) per hostname or IP address. This helps protect hosts against too many requests.

Namespace:  IronWebScraper  
Assembly:  IronWebScraper (in IronWebScraper.dll)  
Version:  4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public int OpenConnectionLimitPerHost { get; set; }
```

Property Value
Type:  Int32

See Also

Reference
WebScraper Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
## WebScraperRateLimitPerHost Property

Gets or sets minimum polite delay (pause) between request to a given domain or IP address.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public TimeSpan RateLimitPerHost { get; set; }</code></td>
<td></td>
</tr>
</tbody>
</table>

**Property Value**  
Type: `TimeSpan`

### See Also

- Reference  
  - WebScraper Class  
  - IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperSuccessfulFileDownloadCount Property

Gets the number of successful http downloads using the DownloadFile and DownloadImage methods..

**Namespace:**  IronWebScraper  
**Assembly:**  IronWebScraper (in IronWebScraper.dll)  
**Version:**  4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public int SuccessfulFileDownloadCount { get; }
```

**Property Value**  
Type:  **Int32**

### See Also

**Reference**  
WebScraper Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperSuccessfulRequestCount Property

Gets the number of successful http requests.

Namespace: IronWebScraper  
Assembly: IronWebScraper (in IronWebScraper.dll)  
Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public int SuccessfulRequestCount { get; }
```

Property Value

Type: Int32

See Also

Reference
WebScraper Class  
IronWebScraper Namespace
WebScraperThrottleMode Property

Makes the WebScraper intelligently throttle requests not only by hostname, but also by host servers’ IP addresses. This is polite in-case multiple scraped domains are hosted on the same machine.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public WebScraperThrottle ThrottleMode { get; set; }
```

### Property Value

Type: `WebScraperThrottle`  
`true` if we wish to look up hosts' IP addresses for throttling; otherwise, `false`.

### See Also

Reference  
WebScraper Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraper Methods

The *WebScraper* type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![shield] AcceptUrl</td>
<td>Decides if the WebScraper will accept a given url. My be overridden to apply custom middleware logic.</td>
</tr>
<tr>
<td>![shield] ChooseIdentityForRequest</td>
<td>Picks a random identity from WebScraper.Identities for each request. Identities with proxy IP addresses, user agents, headers, cookies, username and password can be added to the WebScraper.Identities list; override this method to create your own logic for non-random selection of a HttpIdentity for each request.</td>
</tr>
<tr>
<td>![shield] DownloadFile(String, String, Boolean, HttpIdentity)</td>
<td>Requests a file to be downloaded from the given url to the local file-system. Normally called with an Parse Method of IronWebScraper.WebScraper.</td>
</tr>
<tr>
<td>![shield] DownloadFileUnique</td>
<td>Much like DownloadFile, but if the file has already been downloaded or exists locally, it will not be re-downloaded.</td>
</tr>
</tbody>
</table>
Requests a file to be downloaded from the given URL to the local file-system for scraping documents, assets, and images. Normally called with an `IronWebScraper.WebScraper` instance.

**DownloadImage(String, String, Int32, Int32, Boolean, HttpIdentity)**

Requests a file to be downloaded from the given URL to the local file-system for scraping documents, assets, and images. Normally called with an `IronWebScraper.WebScraper` instance.

**DownloadImage(Uri, String, Int32, Int32, Boolean, HttpIdentity)**

Requests a file to be downloaded from the given URL to the local file-system for scraping documents, assets, and images. Normally called with an `IronWebScraper.WebScraper` instance.

**EnableWebCache**

Caches web http responses for reuse, allowing `WebScraper` classes to be modified and restarted without re-downloading previously scraped URLs.

**EnableWebCache(TimeSpan)**

Caches web http responses for reuse, allowing `WebScraper` classes to be modified and restarted without re-downloading previously scraped URLs.

**Equals**

(Inherited from `Object`.)

**GetHashCode**

(Inherited from `Object`.)

**GetType**

(Inherited from `Object`.)

**FetchUrlContents**

A handy shortcut method to fetch the text content from any URL (synchronously).

**FetchUrlContentsBinary**

A handy shortcut method to fetch the binary data from any URL (synchronously) as a byte array (byte[ ]).
<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init</td>
<td>Override this method in your custom web-scraper. Important tasks will be to request at least one start url... and set allowed/banned domain patterns.</td>
</tr>
<tr>
<td>Log</td>
<td>Logs the specified message to the console. Logs can be enabled using the EnableLogging function. This function has been exposed and is overridable to allow for easy notification integration.</td>
</tr>
<tr>
<td>ObeyRobotsDotTxtForHost</td>
<td>Causes the WebScraper to always obey robots.txt directives including path restrictions and crawl rates on a domain-by-domain basis. May be overridden for more advanced control.</td>
</tr>
<tr>
<td>Parse</td>
<td>Override this method to create the default Response handler for your web scraper. If you have multiple page types, you can add additional similar methods.</td>
</tr>
<tr>
<td>ParseWebscraperDownload</td>
<td>Internal method to parse binary files downloaded by the webScraper.</td>
</tr>
<tr>
<td>ParseWebscraperDownloadImage</td>
<td>Internal method to parse images downloaded by the webScraper.</td>
</tr>
<tr>
<td>PostRequest(String,</td>
<td>Request adds a new request to the scrape-job queue using the POST http method.</td>
</tr>
<tr>
<td>ActionResponse, DictionaryString, String)</td>
<td></td>
</tr>
<tr>
<td>PostRequest(Uri,</td>
<td>Request adds a new request to the scrape-job queue using the POST http method.</td>
</tr>
<tr>
<td>ActionResponse, DictionaryString, String)</td>
<td></td>
</tr>
<tr>
<td>PostRequest(String,</td>
<td>Request adds a new request to the scrape-job queue using the POST http method.</td>
</tr>
<tr>
<td>ActionResponse, DictionaryString, String, MetaData)</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>PostRequest(Uri, ActionResponse, Dictionary&lt;String, String, MetaData&gt;)</code></td>
<td>Request adds a new request to the scrape-job queue using the POST method.</td>
</tr>
<tr>
<td><code>PostRequest(String, ActionResponse, Dictionary&lt;String, String, HttpIdentity, MetaData&gt;)</code></td>
<td>Request adds a new request to the scrape-job queue using the POST method.</td>
</tr>
<tr>
<td><code>Request(IEnumerable&lt;String&gt;, ActionResponse)</code></td>
<td>A key method called from with the Init and Parse Methods. Request adds new requests to the scrape-job queue, and decides which method (e.g., Parse) will be used to parse the Response object.</td>
</tr>
<tr>
<td><code>Request(String, ActionResponse)</code></td>
<td>A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g., Parse) will be used to parse the Response object.</td>
</tr>
<tr>
<td><code>Request(Uri, ActionResponse)</code></td>
<td>A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g., Parse) will be used to parse the Response object.</td>
</tr>
<tr>
<td><code>Request(String, ActionResponse, MetaData)</code></td>
<td>A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g., Parse) will be used to parse the Response object.</td>
</tr>
<tr>
<td><code>Request(Uri, ActionResponse, MetaData)</code></td>
<td>A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g., Parse) will be used to parse the Response object.</td>
</tr>
</tbody>
</table>
### Request(String, ActionResponse, HttpIdentity, MetaData)
A key method called from Init and Parse Methods. Request adds a new request to the scrape-job queue, and (e.g. Parse) will be used to parse the Response object.

### Request(Uri, ActionResponse, HttpIdentity, MetaData)
A key method called from Init and Parse Methods. Request adds a new request to the scrape-job queue, and (e.g. Parse) will be used to parse the Response object.

### Retry
Retries a Response. Usually called in a Parse method, this method is useful if a Captcha or error screen was encountered during Html parsing.

### Scrape
Appends any scraped data to a file in the JsonLines format. (1 json object per line). Typically used with IronWebScraper.ScrapedData or developer defined classes for scraped data items. The default filename will follow the pattern "NameSpace.TypeName.jsonl". E.g: Scrape/IronWebScraper.ScrapedData.jsonl

### ScrapeUnique
Appends scraped data to a file in the JsonLines format. (1 json object per line). Ignores duplicates. Will save any .Net object of any kind. This method is typically used with IronWebScraper.ScrapedData or developer defined classes for scraped data items. The default filename will follow the pattern "WorkingDirectory/NameSpace.TypeName.jsonl". E.g: Scrape/IronWebScraper.ScrapedData.jsonl
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SetSiteSpecificCrawlRateLimit</strong></td>
<td>Set a throttle limit for a specific domain.</td>
</tr>
<tr>
<td><strong>Start</strong></td>
<td>Starts the WebScraper. Set CrawlId to make this crawl resumable. Also resume a previous crawl with the same CrawlId if it exists. Giving a CrawlId also causes auto-save of its state even in case of crash, system failure or power outage. This feature is particularly useful for long running web-scraping tasks, allowing hours, days, or even weeks of work to be recovered effortlessly.</td>
</tr>
<tr>
<td><strong>StartAsync</strong></td>
<td>Starts the WebScraper asynchronously. Set CrawlId to make this crawl resumable. Resume a previous crawl with the same CrawlId if it exists.</td>
</tr>
<tr>
<td><strong>Stop</strong></td>
<td>Stops this WebScraper instance gracefully. The WebScraper may be re-started later with no loss of data by calling Start(CrawlId) or StartAsync(CrawlId).</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>(Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>UnScrape(Boolean)</strong></td>
<td>Retrieves IronWebScraper.ScrapedData objects which were saved using the WebScraper.Scrape method.</td>
</tr>
<tr>
<td><strong>UnScrape(String, Boolean)</strong></td>
<td>Retrieves IronWebScraper.ScrapedData objects which were saved using the WebScraper.Scrape method.</td>
</tr>
<tr>
<td><strong>UnScrapeT(Boolean)</strong></td>
<td>Retrieves native C# objects which were saved using the WebScraper.Scrape method in the JsonLines format.</td>
</tr>
<tr>
<td><strong>UnScrapeT(String, Boolean)</strong></td>
<td>Retrieves native C# objects which were saved using the WebScraper.Scrape method in the JsonLines format.</td>
</tr>
</tbody>
</table>
See Also

Reference
WebScraper Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperAcceptUrl Method

Decides if the WebScraper will accept a given url. My be overridden to apply custom middleware logic.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public virtual bool AcceptUrl(
    string url
)
```

Parameters

url
Type: System.String

Return Value
Type: Boolean

See Also

Reference
WebScraper Class
IronWebScraper Namespace
WebScraperChooseIdentityForRequest Method

Picks a random identity from WebScraper.Identities for each request. Add Identities with proxy IP addresses, userAgent, headers, cookies, username and password in your Init Method and add them to the WebScraper.Identities List; Override this method to create your own logic for non-random selection of an HttpIdentity for each request.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public virtual HttpIdentity ChooseIdentityForRequest(Request request)
```

**Parameters**

- `request`  
  - Type: `IronWebScraperRequest`  
  - The http Request

**Return Value**

- Type: `HttpIdentity`  
  - An HttpIdentity

### See Also
Reference
WebScraper Class
IronWebScraper Namespace
# WebScraperDownloadFile Method

## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![DownloadFile(String, String, Boolean, HttpIdentity)]</td>
<td>Requests a file to be downloaded from the given URL to the local file-system. Often used for scraping documents, assets and images. Normally called with an Parse Method of IronWebScraper.WebScraper</td>
</tr>
<tr>
<td>![DownloadFile(Uri, String, Boolean, HttpIdentity)]</td>
<td>Requests a file to be downloaded from the given URL to the local file-system. Often used for scraping documents, assets and images. Normally called with an Parse Method of IronWebScraper.WebScraper</td>
</tr>
</tbody>
</table>

## See Also

Reference
- [WebScraper Class](#)
- [IronWebScraper Namespace](#)
IronWebScraper - The C# Web Scraping Library
WebScraperDownloadFile Method (String, String, Boolean, HttpIdentity)

Requests a file to be downloaded from the given URL to the local file-system. Often used for scraping documents, assets and images. Normally called with a Parse Method of IronWebScraper.WebScraper.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

**Syntax**

```csharp
public virtual string DownloadFile(
    string url,
    string path,
    bool overwrite = false,
    HttpIdentity identity = null
)
```

**Parameters**

**url**
Type: System.String  
The absolute URL of the resource to be downloaded.

**path**
Type: System.String  
The path to which the downloaded file should be saved. You may give a directory name or a file name. Relative paths will be resolved relative to WorkingDirectory.
**overWrite (Optional)**
Type: SystemBoolean
If set to **true** any existing file at the given path will be overwritten. If set to **false** a unique name such as "file(1).html" will be created in the case of a naming conflict.

**identity (Optional)**
Type: IronWebScraperHttpIdentity
An HttpIdentity to send the Request. If null, the ChooseIdentityForRequest method will be used to find a suitable identity.

**Return Value**
Type: String
The file path (relative to WorkingDirectory) which the file will be saved to.

**See Also**

Reference
WebScraper Class
DownloadFile Overload
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperDownloadFile Method (Uri, String, Boolean, HttpIdentity)

Requests a file to be downloaded from the given Url to the local file-system. Often used for scraping documents, assets and images. Normally called with a Parse Method of IronWebScraper.WebScraper.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

**Syntax**

```csharp
public virtual string DownloadFile(
    Uri uri,
    string path,
    bool overwrite = false,
    HttpIdentity identity = null)
```

**Parameters**

- **uri**  
  Type: System.Uri  
  The absolute uri of the resource to be downloaded.

- **path**  
  Type: System.String  
  The path to which the downloaded file should be saved. You may give a directory name or a file name. Relative paths will be resolved relative to WorkingDirectory.
**overwrite** *(Optional)*
Type: SystemBoolean
If set to `true` any existing file at the given path will be overwritten. If set to `false` a unique name such as "file(1).html" will be created in the case of a naming conflict.

**identity** *(Optional)*
Type: IronWebScraperHttpIdentity
An HttpIdentity to send the Request. If null, the ChooseIdentityForRequest method will be used to find a suitable identity.

**Return Value**
Type: String
The file path (relative to WorkingDirectory) which the file will be saved to.

**See Also**

**Reference**
WebScraper Class
DownloadFile Overload
IronWebScraper Namespace
WebScraperDownloadFileUnique Method

Much like DownloadFile except if the file has already been downloaded or exists locally, it will not be re-downloaded. Requests a file to be downloaded from the given URL to the local file-system. Often used for scraping documents, assets and images.

Normally called with an Parse Method of IronWebScraper.WebScraper

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

**Syntax**

```csharp
public virtual string DownloadFileUnique(
    string url,
    string path,
    HttpIdentity identity = null
)
```

**Parameters**

- **url**  
  Type: System.String  
  The URL.

- **path**  
  Type: System.String  
  The path.

- **identity (Optional)**  
  Type: IronWebScraper.HttpIdentity  
  The identity.
Return Value
Type: String

See Also

Reference
WebScraper Class
IronWebScraper Namespace
# WebScraperDownloadImage Method

## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DownloadImage(String, String, Int32, Int32, Boolean, HttpIdentity)</td>
<td>Requests a file to be downloaded from the given URL to the local file-system. Often used for scraping documents, assets and images. Normally called with an Parse Method of IronWebScraper.WebScraper</td>
</tr>
<tr>
<td>DownloadImage(Uri, String, Int32, Int32, Boolean, HttpIdentity)</td>
<td>Requests a file to be downloaded from the given URL to the local file-system. Often used for scraping documents, assets and images. Normally called with an Parse Method of IronWebScraper.WebScraper</td>
</tr>
</tbody>
</table>

## See Also

Reference  
WebScraper Class  
IronWebScraper Namespace
WebScraperDownloadImage Method (String, String, Int32, Int32, Boolean, HttpIdentity)

Requests a file to be downloaded from the given Uri to the local file-system. Often used for scraping documents, assets and images. Normally called with a Parse Method of IronWebScraper.WebScraper.

**Namespace:** IronWebScraper
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public virtual string DownloadImage(
    string url,
    string path,
    int maxWidth = 0,
    int maxHeight = 0,
    bool overWrite = false,
    HttpIdentity identity = null
)
```

### Parameters

**url**
Type: **SystemString**
The absolute Uri of the resource to be downloaded.

**path**
Type: **SystemString**
The path to which the downloaded file should be saved. You may give a directory name or a file name.
Relative paths will be resolved relative to WorkingDirectory.

*maxWidth (Optional)*
Type: SystemInt32
The Downloaded image will be scaled proportionally to this maximum width. Zero means no constraint.

*maxHeight (Optional)*
Type: SystemInt32
The Downloaded image will be scaled proportionally to this maximum height. Zero means no constraint.

*overWrite (Optional)*
Type: SystemBoolean
If set to `true` any existing file at the given path will be overwritten. If set to `false` a unique name such as "file(1).html" will be created in the case of a naming conflict.

*identity (Optional)*
Type: IronWebScraperHttpIdentity
An HttpIdentity to send the Request. If null, the ChooseIdentityForRequest method will be used to find a suitable identity.

**Return Value**
Type: String
The file path (relative to WorkingDirectory) which the image will be saved to.

**See Also**

Reference
WebScraper Class
DownloadImage Overload
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScrapperDownloadImage Method (Uri, String, Int32, Int32, Boolean, HttpIdentity)

Requests a file to be downloaded from the given Url to the local filesystem. Often used for scraping documents, assets and images. Normally called with an Parse Method of IronWebScraper.WebScraper

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

C#  VB

```csharp
public virtual string DownloadImage(
    Uri uri,
    string path,
    int maxWidth = 0,
    int maxHeight = 0,
    bool overWrite = false,
    HttpIdentity identity = null
)
```

Parameters

`uri`
Type: System.Uri
The absolute uri of the resource to be downloaded.

`path`
Type: System.String
The path to which the downloaded file should be saved. You may give a directory name or a file name.
Relative paths will be resolved relative to WorkingDirectory.

$maxWidth (Optional)$
Type: SystemInt32
The Downloaded image will be scaled proportionally to this maximum width. Zero means no constraint.

$maxHeight (Optional)$
Type: SystemInt32
The Downloaded image will be scaled proportionally to this maximum height. Zero means no constraint.

$overWrite (Optional)$
Type: SystemBoolean
If set to true any existing file at the given path will be overwritten. If set to false a unique name such as "file(1).html" will be created in the case of a naming conflict.

$identity (Optional)$
Type: IronWebScraperHttpIdentity
An HttpIdentity to send the Request. If null, the ChooseIdentityForRequest method will be used to find a suitable identity.

Return Value
Type: String
The file path (relative to WorkingDirectory) which the image will be saved to.

See Also
Reference
WebScraper Class
DownloadImage Overload
IronWebScraper Namespace
# WebScraperEnableWebCache Method

## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="" /> EnableWebCache</td>
<td>Caches web http responses for reuse. This allows WebScraper classes to be modified and restarted without re-downloading previously scraped urls.</td>
</tr>
<tr>
<td><img src="image" alt="" /> EnableWebCache(TimeSpan)</td>
<td>Caches web http responses for reuse. This allows WebScraper classes to be modified and restarted without re-downloading previously scrape urls.</td>
</tr>
</tbody>
</table>

---

See Also

Reference
WebScraper Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperEnableWebCache Method

Caches web http responses for reuse. This allows WebScraper classes to be modified and restarted without re-downloading previously scraped urls.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

**Syntax**

```csharp
public void EnableWebCache()
```

**See Also**

Reference  
WebScraper Class  
EnableWebCache Overload  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperEnableWebCache Method (TimeSpan)

Caches web http responses for reuse. This allows WebScraper classes to be modified and restarted without re-downloading previously scrape urls.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

For C#:

```csharp
public void EnableWebCache(
    TimeSpan cacheDuration
)
```

For VB:

```vbnet
Public Sub EnableWebCache(cacheDuration As TimeSpan)
```

### Parameters

* `cacheDuration`  
  Type: `System.TimeSpan`  
  Duration that responses will be cached for.

### See Also

Reference:
- WebScraper Class
- EnableWebCache Overload
- IronWebScraper Namespace
WebScraperFetchUrlContents Method

A handy shortcut method that fetches the text content from any URL (synchronously).

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Copy</td>
</tr>
</tbody>
</table>

```csharp
public static string FetchUrlContents(
    string url,
    HttpIdentity identity = null
)
```

Parameters

**url**
Type: System.String
The absolute URL.

**identity (Optional)**
Type: IronWebScraper.HttpIdentity
Optional HTTP identity to choose a proxy, user agent, headers, username and password for the request.

Return Value
Type: String

See Also

Reference
WebScraper Class
IronWebScraper Namespace
WebScraperFetchUrlContentsBinary

A handy shortcut method that fetches the text content from any Url (synchronously) as a binary data in a byye array (byte[]) 

**Namespace:** WebScraper

**Assembly:** WebScraper (in WebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

**Syntax**

```csharp
public byte[] FetchUrlContentsBinary(
    string url, 
    HttpIdentity identity = null
)
```

**Parameters**

`url`
Type: System.String
The absolute URL.

`identity (Optional)`
Type: IronWebScraper.HttpIdentity
Optional HTTP identity to choose a proxy, user agent, headers, username and password for the request.

**Return Value**
Type: Byte

**See Also**

Reference
IronWebScraper - The C# Web Scraping Library
WebScraperInit Method

Override this method initialize your web-scraper. Important tasks will be to Request at least one start url... and set allowed/banned domain or url patterns.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) **Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public abstract void Init()
```

```vbnet

```

### See Also

Reference  
WebScraper Class  
IronWebScraper Namespace
WebScraperLog Method

Logs the specified message to the console. Logs can be Enabled using the EnableLogging. This function has been exposed and is overridable to allow for easy Email and Slack notification integration.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

C#  VB

```csharp
public virtual void Log(
    string Message,
    WebScraperLogLevel Type
)
```

Parameters

*Message*

Type: `System.String`
The string message.

*Type*

Type: `IronWebScraperWebScraperLogLevel`
The LogLevel.

See Also

Reference

WebScraper Class
IronWebScraper Namespace
WebScraperObeyRobotsDotTxtForHost Method

Causes the WebScraper to always obey /robots.txt directives including path restrictions and crawl rates on a domain by domain basis. May be overridden for advanced control.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

**C#**

```csharp
public virtual bool ObeyRobotsDotTxtForHost(string Host)
```

**VB**

```vbnet
Public Function ObeyRobotsDotTxtForHost(Host As String) As Boolean
```

### Parameters

*Host*  
Type: `System.String`

### Return Value

Type: `Boolean`

### See Also

Reference  
WebScraper Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperParse Method

Override this method to create the default Response handler for your web scraper. If you have multiple page types, you can add additional similar methods.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public abstract void Parse(IronWebScraperResponse response)
```

### Parameters

- **response**  
  Type: `IronWebScraperResponse`  
  The http Response object to parse

### See Also

- Reference  
  WebScraper Class  
  IronWebScraper Namespace
WebScraper.ParseWebscraperDownload Method

Internal method to parse binary files downloaded by a webScraper.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

#### Syntax

```csharp
public virtual void ParseWebscraperDownload(
    Response response
)
```

#### Parameters

- **response**  
  Type: IronWebScraperResponse  
  A Response object

#### See Also

Reference  
- WebScraper Class  
- IronWebScraper Namespace
WebScraperParseWebscraperDownloadImage Method

Internal method to parse images downloaded by a webScraper.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public virtual void ParseWebscraperDownloadImage(
    Response response
)
```

C# VB

Parameters

`response`
Type: IronWebScraperResponse
A Response object

See Also

Reference
WebScraper Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
## WebScraperPostRequest Method

### Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostRequest(String, ActionResponse, DictionaryString, String)</td>
<td>Request adds a new request to the scrape-job queue using the POST http method.</td>
</tr>
<tr>
<td>PostRequest(Uri, ActionResponse, DictionaryString, String)</td>
<td>Request adds a new request to the scrape-job queue using the POST http method.</td>
</tr>
<tr>
<td>PostRequest(String, ActionResponse, DictionaryString, String, MetaData)</td>
<td>Request adds a new request to the scrape-job queue using the POST http method.</td>
</tr>
<tr>
<td>PostRequest(Uri, ActionResponse, DictionaryString, String, MetaData)</td>
<td>Request adds a new request to the scrape-job queue using the POST http method.</td>
</tr>
<tr>
<td>PostRequest(String, ActionResponse, DictionaryString, String, HttpIdentity, MetaData)</td>
<td>Request adds a new request to the scrape-job queue using the POST http method.</td>
</tr>
<tr>
<td>PostRequest(Uri, ActionResponse, DictionaryString, String, MetaData)</td>
<td>Request adds a new request to the scrape-job queue using the POST http method.</td>
</tr>
</tbody>
</table>
HttpIdentity, MetaData) POST http method.

See Also

Reference
WebScraper Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperPostRequest Method (String, ActionResponse, Dictionary<String, String>)

Request adds a new request to the scrape-job queue using the POST http method.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public virtual void PostRequest(  
    string url,  
    Action<Response> parse,  
    Dictionary<string, string> postVariables
)
```

### Parameters

- **url**
  - Type: `System.String`  
  - The absolute url to be fetched. Developers may use Response.ToAbsoluteUrl to resolve all relative links to Absolute Url strings.

- **parse**
  - Type: `System.Action<Response>`  
  - The method to be used to parse the Response (often this is WebScraper.Parse)

- **postVariables**
  - String
The POST variables as a dictionary of key-value pairs.

See Also

Reference
WebScraper Class
PostRequest Overload
IronWebScraper Namespace
WebScraperPostRequest Method (Uri, ActionResponse, Dictionary<String, String>)

Request adds a new request to the scrape-job queue using the POST http method.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

```
public virtual void PostRequest(
    Uri url,
    Action<Response> parse,
    Dictionary<string, string> postVariables
)
```

### Parameters

**url**
- Type: System.Uri  
  The absolute url to be fetched. Developers may use Response.ToAbsoluteUrl to resolve all relative links to Absolute Url strings.

**parse**
- Type: System.Action<Response>  
  The method to be used to parse the Response (often this is WebScraper.Parse)

**postVariables**
The POST variables as a dictionary of key-value pairs.

See Also

Reference
WebScraper Class
PostRequest Overload
IronWebScraper Namespace
WebScraperPostRequest Method (String, ActionResponse, Dictionary<String, String>, String, MetaData)

Request adds a new request to the scrape-job queue using the POST http method.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

```
public virtual void PostRequest(
    string url,  
    Action<Response> parse,
    Dictionary<string, string> postVariables,
    MetaData metaData
)
```

### Parameters

- **url**
  - Type: `System.String`
  - The absolute url to be fetched. Developers may use `Response.ToAbsoluteUrl` to resolve all relative links to Absolute Url strings.

- **parse**
  - Type: `System.Action<Response>`
The method to be used to parse the Response (often this is WebScraper.Parse)

`postVariables`
Type: `System.Collections.Generic.Dictionary<String, String>`
The POST variables as a dictionary of key-value pairs.

`metaData`
Type: `IronWebScraperMetaData`
Additional information of any Type can be sent with the request and then re-read when the response is parsed.

**See Also**

**Reference**
WebScraper Class
PostRequest Overload
IronWebScraper Namespace
WebScraperPostRequest Method (Uri, ActionResponse, Dictionary<String, String>, String, MetaData)

Request adds a new request to the scrape-job queue using the POST http method.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public virtual void PostRequest(
    Uri url,
    Action<Response> parse,
    Dictionary<string, string> postVariables,
    MetaData metaData
)
```

### Parameters

**url**
- Type: `System.Uri`  
The absolute url to be fetched. Developers may use `Response.ToAbsoluteUrl` to resolve all relative links to Absolute Url strings.

**parse**
- Type: `System.Action`  

```vbnet
public virtual void PostRequest(
    Uri url,
    Action<Response> parse,
    Dictionary<string, string> postVariables,
    MetaData metaData
)
```
The method to be used to parse the Response (often this is WebScraper.Parse)

**postVariables**
Type: `System.Collections.Generic.Dictionary<string, string>`
The POST variables as a dictionary of key-value pairs.

**metaData**
Type: `IronWebScraper.MetaData`
Additional information of any Type can be sent with the request and then re-read when the response is parsed.

### See Also

**Reference**
- WebScraper Class
- PostRequest Overload
- IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperPostRequest Method (String, ActionResponse, Dictionary<String, String>, String, HttpIdentity, MetaData)

Request adds a new request to the scrape-job queue using the POST http method.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

**Syntax**

```csharp
public virtual void PostRequest(
    string url,
    Action<Response> parse,
    Dictionary<string, string> postVariables,
    HttpIdentity identity = null,
    MetaData metaData = null
)
```

**Parameters**

- **url**  
  Type: `System.String`  
  The absolute url to be fetched. Developers may use Response.ToAbsoluteUrl to resolve all relative links to Absolute Url strings.

- **parse**
Type: **SystemActionResponse**
The method to be used to parse the Response (often this is WebScraper.Parse)

**postVariables**
Type: **System.Collections.GenericDictionary[String, String]**
The POST variables as a dictionary of key-value pairs.

**identity (Optional)**
Type: **IronWebScraperHttpIdentity**
An optional HttpIdentity to send the Request. If null, the ChooseIdentityForRequest method will be used to find a suitable identity.

**metaData (Optional)**
Type: **IronWebScraperMetaData**
Additional information of any Type can be sent with the request and then re-read when the response is parsed.

**See Also**

Reference
WebScraper Class
PostRequest Overload
IronWebScraper Namespace
WebScraperPostRequest Method (Uri, ActionResponse, Dictionary<String, String>, HttpIdentity, MetaData)

Request adds a new request to the scrape-job queue using the POST http method.

Namespace: IronWebScraper  
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public virtual void PostRequest(
    Uri url,
    Action<Response> parse,
    Dictionary<string, string> postVariables,
    HttpIdentity identity = null,
    MetaData metaData = null
)
```

Parameters

url  
Type: System.Uri  
The absolute url to be fetched. Developers may use Response.ToAbsoluteUrl to resolve all relative links to Absolute Url strings.

parse
Type: SystemActionResponse
The method to be used to parse the Response (often this is WebScraper.Parse)

**postVariables**
Type: System.Collections.Generic.Dictionary<String, String>
The POST variables as a dictionary of key-value pairs.

**identity (Optional)**
Type: IronWebScraperHttpIdentity
An optional HttpIdentity to send the Request. If null, the
ChooselIdentityForRequest method will be used to find a
suitable identity.

**metaData (Optional)**
Type: IronWebScraperMetaData
Additional information of any Type can be sent with the request
and then re-read when the response is parsed.

**See Also**

Reference
WebScraper Class
PostRequest Overload
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
# WebScraperRequest Method

## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Request(IEnumerable&lt;String, ActionResponse&gt;)</code></td>
<td>A key method called from with the Init and Parse Methods. Request adds new requests to the scrape-job queue, and decides which method (e.g. Parse) will be used to parse the Response object.</td>
</tr>
<tr>
<td><code>Request(String, ActionResponse)</code></td>
<td>A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g. Parse) will be used to parse the Response object.</td>
</tr>
<tr>
<td><code>Request(Uri, ActionResponse)</code></td>
<td>A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g. Parse) will be used to parse the Response object.</td>
</tr>
</tbody>
</table>
queue, and decides which method (e.g. Parse) will be used to parse the Response object.

<table>
<thead>
<tr>
<th>Request(String, ActionResponse, MetaData)</th>
<th>A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g. Parse) will be used to parse the Response object.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request(Uri, ActionResponse, MetaData)</td>
<td>A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g. Parse) will be used to parse the Response object.</td>
</tr>
<tr>
<td>Request(String, ActionResponse, HttpIdentity, MetaData)</td>
<td>A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g. Parse) will be used</td>
</tr>
</tbody>
</table>
Request(Uri, ActionResponse, HttpIdentity, MetaData)  

A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g. Parse) will be used to parse the Response object.

See Also

Reference
WebScraper Class
IronWebScraper Namespace
WebScraperRequest Method (IEnumerable<String, Action<Response>)

A key method called from with the Init and Parse Methods. Request adds new requests to the scrape-job queue, and decides which method (e.g. Parse) will be used to parse the Response object.

Namespace: IronWebScraper  
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
</table>
| public virtual void Request(  
  IEnumerable<string> urls,  
  Action<Response> parse  
) |

Parameters

urls
Type: System.Collections.Generic(IEnumerable<String>
The Absolute url or urls to be fetched. Developers may use Response.ToAbsoluteUrl to resolve all relative links to Absolute Url strings.

parse
Type: SystemActionResponse  
The method to be used to parse the Response (often this is WebScraper.Parse)
See Also

Reference
WebScraper Class
Request Overload
IronWebScraper Namespace
WebScraperRequest Method
(String, Action<WebResponse>)

A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g. Parse) will be used to parse the Response object.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

#### C#
```csharp
public virtual void Request(
    string url,
    Action<WebResponse> parse
)
```

#### VB
```vbnet
Public Overridable Sub Request(ByVal url As String, ByVal parse As Action(Of WebResponse))
```

### Parameters

**url**
- **Type:** System.String  
  The absolute url to be fetched. Developers may use Response.ToAbsoluteUrl to resolve all relative links to Absolute Url strings.

**parse**
- **Type:** System.Action<WebResponse>  
  The method to be used to parse the Response (often this is WebScraper.Parse)

### See Also

Reference
WebScraper Class
Request Overload
IronWebScraper Namespace
WebScraperRequest Method (Uri, ActionResponse)

A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g., Parse) will be used to parse the Response object.

Namespace: IronWebScraper  
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public virtual void Request(
    Uri url,
    Action<Response> parse
)
```

Parameters

- **url**
  - Type: System.Uri
  - The absolute url to be fetched. Developers may use Response.ToAbsoluteUrl to resolve all relative links to Absolute Url strings.

- **parse**
  - Type: System.Action<Response>
  - The method to be used to parse the Response (often this is WebScraper.Parse)

See Also

Reference
WebScraper Class
Request Overload
IronWebScraper Namespace
WebScraperRequest Method (String, Action<Response, MetaData>)

A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g., Parse) will be used to parse the Response object.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
Version: 4.0.4.25470 (4.0.4.3)

**Syntax**

```csharp
public virtual void Request(
    string url,
    Action<Response> parse,
    MetaData metaData
)
```

**Parameters**

*url*  
Type: System.String  
The absolute url to be fetched. Developers may use Response.ToAbsoluteUrl to resolve all relative links to Absolute Uri strings.

*parse*  
Type: System.Action<Response>  
The method to be used to parse the Response (often this is WebScraper.Parse)

*metaData*
Type: IronWebScraperMetaData
Additional information of any Type can be sent with the request and then re-read when the response is parsed.

See Also

Reference
WebScraper Class
Request Overload
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperRequest Method
(Uri, ActionResponse, MetaData)

A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g. Parse) will be used to parse the Response object.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public virtual void Request(
    Uri url,
    Action<Response> parse,
    MetaData metaData
)
```

Parameters

url
Type: System.Uri
The absolute url to be fetched. Developers may use Response.ToAbsoluteUrl to resolve all relative links to Absolute Url strings.

parse
Type: System.Action<Response>
The method to be used to parse the Response (often this is WebScraper.Parse)

metaData
Type: IronWebScraper.MetaData
Additional information of any Type can be sent with the request
and then re-read when the response is parsed.

See Also

Reference
WebScraper Class
Request Overload
IronWebScraper Namespace
WebScraperRequest Method
(String, ActionResponse, HttpIdentity, MetaData)

A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g. Parse) will be used to parse the Response object.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

**Syntax**

```csharp
public virtual void Request(
    string url,
    Action<Response> parse,
    HttpIdentity identity = null,
    MetaData metaData = null
)
```

**Parameters**

- **url**
  - Type: `System.String`  
  - The absolute url to be fetched. Developers may use Response.ToAbsoluteUrl to resolve all relative links to Absolute Url strings.

- **parse**
  - Type: `System.Action<Response>`  
  - The method to be used to parse the Response (often this is WebScraper.Parse)
**identity (Optional)**
Type: `IronWebScraperHttpIdentity`
An HttpIdentity to send the Request. If null, the ChooseIdentityForRequest method will be used to find a suitable identity.

**metaData (Optional)**
Type: `IronWebScraperMetaData`
Additional information of any Type can be sent with the request and then re-read when the response is parsed.

### See Also

Reference
- WebScraper Class
- Request Overload
- IronWebScraper Namespace
WebScraperRequest Method
(Uri, ActionResponse, HttpIdentity, MetaData)

A key method called from with the Init and Parse Methods. Request adds a new request to the scrape-job queue, and decides which method (e.g. Parse) will be used to parse the Response object.

Namespace:  IronWebScraper
Assembly:  IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public virtual void Request(
    Uri url,
    Action<Response> parse,
    HttpIdentity identity = null,
    MetaData metaData = null
)
```

Parameters

`url`
Type:  System.Uri
The absolute url to be fetched. Developers may use Response.ToAbsoluteUrl to resolve all relative links to Absolute Url strings.

`parse`
Type:  System.Action<Response>
The method to be used to parse the Response (often this is WebScraper.Parse)
**identity (Optional)**
Type: IronWebScraperHttpIdentity
An HttpIdentity to send the Request. If null, the
ChooseIdentityForRequest method will be used to find a
suitable identity.

**metaData (Optional)**
Type: IronWebScraperMetaData
Additional information of any Type can be sent with the request
and then re-read when the response is parsed.

See Also

Reference
WebScraper Class
Request Overload
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperRetry Method

Retries a Response.
Usually called in a Parse method, this method is useful if a Captcha or error screen was encountered during Html parsing.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

**C#**

```csharp
public void Retry(
    Response Response
)
```

**VB**

```vbnet
Public Sub Retry(ByVal Response As Response)
```

### Parameters

- **Response**
  Type: IronWebScraperResponse

### See Also

- **Reference**
  - WebScraper Class
  - IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperScrape Method

Appends any scraped data to a file in the JsonLines format. (1 json object per line). Will save any .Net object of any kind. This method is typically used with IronWebScraper.ScrapedData or developer defined classes for scraped data items. The default filename will follow the pattern "NameSpace.TypeName.jsonl". E.g: IronWebScraper.ScrapedData.jsonl

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

C#  
```csharp
public void Scrape(
    Object Item,
    string fileName = null
)
```

### Parameters

**Item**  
Type: `SystemObject`

**fileName** *(Optional)*  
Type: `System.String`

### See Also

Reference  
WebScraper Class  
IronWebScraper Namespace
WebScraperScrapeUnique Method

Appends scraped data to a file in the JsonLines format. (1 json object per line). Automatically ignores duplicates. Will save any .Net object of any kind. This method is typically used with IronWebScraper.ScrapedData or developer defined classes for scraped data items. The default filename will follow the pattern "WorkingDirectory/NameSpace.TypeName.jsonl". E.g: Scrape/IronWebScraper.ScrapedData.jsonl

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

C#   VB

```csharp
public void ScrapeUnique(
    Object Item,
    string fileName = null
)
```

Parameters

- **Item**
  Type: SystemObject

- **fileName (Optional)**
  Type: SystemString

See Also

Reference
WebScraper Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperSetSiteSpecificCrawlRateLimit Method

Set a throttle limit for a specific domain

Namespace: IronWebScraper  
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```csharp
public void SetSiteSpecificCrawlRateLimit(
    string hostName,
    TimeSpan crawlRate
)
```

Parameters

- **hostName**
  - Type: System.String  
  - The http host (domain name).

- **crawlRate**
  - Type: System.TimeSpan  
  - The maximum frequency of http requests for the given hostName.

See Also

Reference
- WebScraper Class
- IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperStart Method

Starts the WebScraper. Set CrawlId to make this crawl resumable. Will also resume a previous scrawl with the same CrawlId if it exists.

Giving a CrawlId also causes the WebScraper to auto-save its state every 5 minutes in case of a crash, system failure or power outage. This feature is particularly useful for long running web-scraping tasks, allowing hours, days or even weeks of work to be recovered effortlessly.

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

C#   VB

```csharp
public void Start(
    string CrawlId = null
)
```

Parameters

*CrawlId (Optional)*
Type: System.String

See Also

Reference
WebScraper Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperStartAsync Method

Starts the WebScraper Asynchronously. Set CrawlId to make this crawl resumable. Will resume a previous scrawl with the same CrawlId if it exists.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

```csharp
public Task StartAsync(
    string CrawlId = null
)
```

### Parameters

**CrawlId (Optional)**  
Type: `System.String`

### Return Value

Type: `Task`

### See Also

Reference  
**WebScraper Class**  
**IronWebScraper Namespace**
WebScrapperStop Method

Stops this WebScrapper instance graceful. The WebScrapper may be restated later with no loss of data by calling Start(CrawlId) or StartAsync(CrawlId)

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) **Version:** 4.0.4.25470 (4.0.4.3)

▶ **Syntax**

```csharp
public void Stop()
```

▶ **See Also**

Reference  
[WebScrapper Class](#)  
[IronWebScraper Namespace](#)
WebScraperUnScrape Method

## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>UnScrape(Boolean)</code></td>
<td>Retrieves native C# objects which were saved using the WebScraper.Scrape method in the JsonLines format.</td>
</tr>
<tr>
<td><code>UnScrape(Boolean)</code></td>
<td>Retrieves IronWebScraper.ScrapedData objects which were saved using the WebScraper.Scrape method.</td>
</tr>
<tr>
<td><code>UnScrape(String, Boolean)</code></td>
<td>Retrieves IronWebScraper.ScrapedData objects which were saved using the WebScraper.Scrape method.</td>
</tr>
<tr>
<td><code>UnScrapeT(Boolean)</code></td>
<td>Retrieves native C# objects which were saved using the WebScraper.Scrape method in the JsonLines format.</td>
</tr>
<tr>
<td><code>UnScrapeT(Boolean)</code></td>
<td>Retrieves IronWebScraper.ScrapedData objects which were saved using the WebScraper.Scrape method.</td>
</tr>
</tbody>
</table>

## See Also

Reference

- WebScraper Class
- IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperUnScrape<T> Method (Boolean)

Retrieves native C# objects which were saved using the WebScraper.Scrape method in the JsonLines format.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
</table>
| public IEnumerable<T> UnScrape<T>(  
|   bool IgnoreErrors   |
| )                      |                         |

### Parameters

**IgnoreErrors**  
Type: System.Boolean  
if set to true any objects that cant be cast to the specified Type T will be ignored.

### Type Parameters

**T**  
The Type of object to be returned. Giving no value will return an IEnumerable of IronWebScraper.ScrapedData

### Return Value  
Type: IEnumerable<T>

### See Also
Reference
WebScraper Class
UnScrape Overload
IronWebScraper Namespace
WebScraperUnScrape Method (Boolean)

Retrieves IronWebScraper.ScrapedData objects which were saved using the WebScraper.Scrape method.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
</table>
| ```csharp
public IEnumerable<ScrapedData> UnScrape(
        bool IgnoreErrors
    )
``` | |

**Parameters**

*IgnoreErrors*
Type: **SystemBoolean**  
if set to **true** any objects that cant be cast to the specified Type **T** will be ignored.

**Return Value**
Type: **IEnumerableScrapedData**

### See Also

**Reference**  
WebScraper Class  
UnScrape Overload  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperUnScrape Method (String, Boolean)

Retrieves IronWebScraper.ScrapedData objects which were saved using the WebScraper.Scrape method.

**Namespace:**  IronWebScraper  
**Assembly:**  IronWebScraper (in IronWebScraper.dll)  
Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>public</strong></td>
<td><code>IList&lt;ScrapedData&gt;</code></td>
</tr>
<tr>
<td><strong>UnScrape</strong></td>
<td>(<code>fileName</code> = <code>null</code>,</td>
</tr>
<tr>
<td></td>
<td><code>bool IgnoredErrors = false</code>)</td>
</tr>
</tbody>
</table>

### Parameters

**fileName (Optional)**  
Type: `System.String`  
Path of the saved data file.

**IgnoreErrors (Optional)**  
Type: `System.Boolean`  
If set to `true` any objects that can't be cast to the specified type `T` will be ignored.

### Return Value

Type: `IEnumerable<ScrapedData>`

### See Also

Reference
WebScraper Class
UnScrape Overload
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperUnScrape<T> Method (String, Boolean)

Retrieves native C# objects which were saved using the WebScraper.Scrape method in the JsonLines format.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
</table>
| public IEnumerable<T> UnScrape<T>(
  string fileName = null,
  bool IgnoreErrors = false
) |

### Parameters

**fileName (Optional)**  
Type: System.String  
Path of the saved data file.

**IgnoreErrors (Optional)**  
Type: System.Boolean  
if set to true any objects that can't be cast to the specified Type T will be ignored.

### Type Parameters

**T**  
The Type of object to be returned. Giving no value will return an IEnumerable of IronWebScraper.ScrapedData
Return Value
Type: IEnumerable<T

See Also

Reference
WebScraper Class
UnScrape Overload
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
The **WebScraper** type exposes the following members.

## Fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AllowedDomains</strong></td>
<td>If not empty, all requested URLs' hostname must match at least one of the AllowedDomains patterns. Patterns may be added using glob wildcard strings or Regex</td>
</tr>
<tr>
<td><strong>AllowedUrls</strong></td>
<td>If not empty, all requested URLs must match at least one of the AllowedUrls patterns. Patterns may be added using glob wildcard strings or Regex</td>
</tr>
<tr>
<td><strong>BannedDomains</strong></td>
<td>If not empty, no requested URLs' hostname may match any of the BannedDomains patterns. Patterns may be added using glob wildcard strings or Regex</td>
</tr>
<tr>
<td><strong>BannedUrls</strong></td>
<td>If not empty, no requested URLs may match any of the BannedUrls patterns. Patterns may be added using glob wildcard strings or Regex</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>CrawlId</strong></td>
<td>A unique string used to identify a crawl job.</td>
</tr>
<tr>
<td><strong>FilesDownloaded</strong></td>
<td>The total number of files downloaded successfully with the DownloadImage and DownloadFile methods.</td>
</tr>
<tr>
<td><strong>Identities</strong></td>
<td>A list of http identities to be used to fetch web resources. Each Identity may have a different proxy IP addresses, userAgent, http headers, persistent cookies, username and password. Best practice is to create Identities in your WebScraper.Init Method and Add them to this WebScraper.Identities List.</td>
</tr>
<tr>
<td><strong>LoggingLevel</strong></td>
<td>The level of logging made by the WebScraper engine to the Console. LogLevel.Critical is normally the most useful setting, allowing the developer to write their own, meaningful and application relevant messages inside of Parse methods. LogLevel.ScrapedData is useful when coding and testing a new WebScraper.</td>
</tr>
<tr>
<td><strong>ObeyRobotsDotTxt</strong></td>
<td>Causes the WebScraper to always obey /robots.txt directives including url and path restrictions and crawl rates.</td>
</tr>
</tbody>
</table>
WorkingDirectory	Path to a local directory where scraped data and state information will be saved.

See Also

Reference
WebScraper Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperAllowedDomains Field

If not empty, all requested Urls' hostname must match at least one of the AllowedDomains patterns. Patterns may be added using glob wildcard strings or Regex

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public UrlMatchPatternCollection AllowedDomains</td>
<td></td>
</tr>
</tbody>
</table>

Field Value
Type: UrlMatchPatternCollection

See Also

Reference
WebScraper Class
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperAllowedUrls Field

If not empty, all requested URLs must match at least one of the AllowedUrls patterns. Patterns may be added using glob wildcard strings or Regex.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

**Syntax**

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public</code></td>
<td><code>UrlMatchPatternCollection</code> AllowedUrls</td>
</tr>
</tbody>
</table>

**Field Value**

Type: `UrlMatchPatternCollection`

**See Also**

- Reference
  - `WebScraper Class`
  - `IronWebScraper Namespace`
WebScraperBannedDomains Field

If not empty, no requested URLs' hostname may match any of the BannedDomains patterns. Patterns may be added using glob wildcard strings or Regex

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public UrlMatchPatternCollection BannedDomains</td>
<td></td>
</tr>
</tbody>
</table>

Field Value  
Type: UrlMatchPatternCollection

### See Also

Reference  
[WebScraper Class](#)  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperBannedUrls Field

If not empty, no requested URLs may match any of the BannedUrls patterns. Patterns may be added using glob wildcard strings or Regex

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>public UrlMatchPatternCollection BannedUrls</td>
</tr>
</tbody>
</table>

### Field Value

Type: **UrlMatchPatternCollection**

### See Also

Reference

WebScraper Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperCrawlId Field

A unique string used to identify a crawl job.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public string CrawlId</code></td>
<td></td>
</tr>
</tbody>
</table>

Field Value

Type: **String**

### See Also

Reference
- WebScraper Class
- IronWebScraper Namespace
WebScraperFilesDownloaded Field

The total number of files downloaded successfully with the DownloadImage and DownloadFile methods.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public int FilesDownloaded</code></td>
<td></td>
</tr>
</tbody>
</table>

**Field Value**  
**Type:** Int32

### See Also

**Reference**  
- WebScraper Class  
- IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraper.Identities Field

A list of http identities to be used to fetch web resources. Each Identity may have a different proxy IP addresses, userAgent, http headers, persistent cookies, username and password.

Best practice is to create Identities in your WebScraper.Init Method and Add them to this WebScraper.Identities List.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public List&lt;HttpIdentity&gt; Identities</td>
<td></td>
</tr>
</tbody>
</table>

Field Value
Type: List&lt;HttpIdentity&gt;

### See Also

**Reference**
WebScraper Class  
IronWebScraper Namespace
WebScraperLoggingLevel Field

The level of logging made by the WebScraper engine to the Console. LogLevel.Critical is normally the most useful setting, allowing the developer to write their own, meaningful and application relevant messages inside of Parse methods.

LogLevel.ScrapedData is useful when coding and testing a new WebScraper.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll)  
**Version:** 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public WebScraperLogLevel Logginglevel</code></td>
<td><code>Public Variable Logginglevel As WebScraperLogLevel</code></td>
</tr>
</tbody>
</table>

**Field Value**  
**Type:** WebScraperLogLevel

### See Also

**Reference**  
WebScraper Class  
IronWebScraper Namespace
WebScraperObeyRobotsDotTxt Field

Causes the WebScraper to always obey /robots.txt directives including url and path restrictions and crawl rates.

**Namespace:** IronWebScraper  
**Assembly:** IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

**Syntax**

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public bool ObeyRobotsDotTxt</code></td>
<td></td>
</tr>
</tbody>
</table>

**Field Value**  
Type: `Boolean`

**See Also**

Reference  
WebScraper Class  
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
WebScraperWorkingDirectory Field

Path to a local directory where scraped data and state information will be saved.

**Namespace:**  IronWebScraper  
**Assembly:**  IronWebScraper (in IronWebScraper.dll)  
**Version:**  4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public string WorkingDirectory</td>
<td></td>
</tr>
</tbody>
</table>

Field Value

Type:  **String**

### See Also

Reference

- **WebScraper Class**
- **IronWebScraper Namespace**
WebScraperLogLevel Enumeration

Level of WebScraper logging to the Console. Because this Enum is a Flag type options can be combined using a pipe: e.g. LogLevel.Critical | LogLevel.ScrapedData

Namespace: IronWebScraper
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

C# VB

```csharp
[FlagsAttribute]
public enum LogLevel
```

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>No Logs</td>
</tr>
<tr>
<td>Critical</td>
<td>1</td>
<td>Logs critical events such as permanently irretrievable page failures and system information.</td>
</tr>
<tr>
<td>ScrapedData</td>
<td>2</td>
<td>Logs data which has been extracted, in a JSON format. If using YieldUnique, only new records will be logged.</td>
</tr>
<tr>
<td>Level</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Http</td>
<td>4</td>
<td>Logs HTTP request success and failures. Verbose. Useful for advanced debugging.</td>
</tr>
<tr>
<td>Decision</td>
<td>8</td>
<td>Logs about decisions made by the crawler. Verbose. Useful for advanced debugging.</td>
</tr>
<tr>
<td>All</td>
<td>15</td>
<td>All events are logged to the console. Extremely verbose;</td>
</tr>
</tbody>
</table>

See Also

Reference
IronWebScraper Namespace
WebScraperThrottle Enumeration

Throttle remote clients by their host name or by their public IP address.

Namespace: IronWebScraper  
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

C#  VB
public enum Throttle

Members

<table>
<thead>
<tr>
<th>Member name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ByIpAddress</td>
<td>0</td>
<td>Identify unique hosts by resolving the IP address for each host in-case</td>
</tr>
<tr>
<td></td>
<td></td>
<td>multiple host names share an IP address.</td>
</tr>
<tr>
<td>ByDomainHostName</td>
<td>1</td>
<td>Identify unique hosts by their host name</td>
</tr>
</tbody>
</table>

See Also
Reference
IronWebScraper Namespace
IronWebScraper - The C# Web Scraping Library
## IronWebScraper.UrIs

### Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="UrlMatchPatternCollection" alt="Icon" /></td>
<td>A class used for control URL, Domain and File type access using positive and negative wild-card(,*?) and Regex rules.</td>
</tr>
</tbody>
</table>
UrlMatchPatternCollection Class

A class used for control URL, Domain and File type access using positive and negative wild-card(?) and Regex rules.

Inheritance Hierarchy

```
SystemObject  System.Collections.GenericList<Regex>
  IronWebScraper.UrlsUrlMatchPatternCollection
```

Namespace:  IronWebScraper.Urls
Assembly:  IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```
C#   VB

public class UrlMatchPatternCollection : List<Regex>
```

The UrlMatchPatternCollection type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UrlMatchPatternCollection</td>
<td>Initializes a new instance of the UrlMatchPatternCollection class</td>
</tr>
</tbody>
</table>

Properties
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Count</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Item</td>
<td>(Inherited from ListRegex.)</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Add(T)]</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>[Add(String)]</td>
<td>Adds a match pattern using strings with wild-card.</td>
</tr>
<tr>
<td></td>
<td>? matches exactly 1 character</td>
</tr>
<tr>
<td></td>
<td>* matches exactly zero to any number of characters</td>
</tr>
<tr>
<td>[Add(String)]</td>
<td>Adds multiple match patterns using strings with wild-card.</td>
</tr>
<tr>
<td></td>
<td>? matches exactly 1 character</td>
</tr>
<tr>
<td></td>
<td>* matches exactly zero to any number of characters</td>
</tr>
<tr>
<td>AddRange</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>BinarySearch(T)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>BinarySearch(T, IComparer&lt;T&gt;)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>BinarySearch(Int32, Int32, T, IComparer&lt;T&gt;)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Clear</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Contains</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>ConvertAllToOutput</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>CopyTo(T)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>CopyTo(T, Int32)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>CopyTo(Int32, T, Int32, Int32)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Equals</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Exists</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Find</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>FindAll</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>FindIndex(PredicateT)</td>
<td>(Inherited from</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td><code>FindIndex(Int32, PredicateT)</code></td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td><code>FindIndex(Int32, Int32, PredicateT)</code></td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td><code>FindLast</code></td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td><code>FindLastIndex(PredicateT)</code></td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td><code>FindLastIndex(Int32, PredicateT)</code></td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td><code>FindLastIndex(Int32, Int32, PredicateT)</code></td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td><code>ForEach</code></td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td><code>GetEnumerator</code></td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td><code>GetHashCode</code></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><code>GetRange</code></td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td><code>GetType</code></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td><code>IndexOf(T)</code></td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td><code>IndexOf(T, Int32)</code></td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>IndexOf(T)</code></td>
<td>(Inherited from <code>List&lt;Regex&gt;</code>)</td>
</tr>
<tr>
<td><code>Insert</code></td>
<td>(Inherited from <code>List&lt;Regex&gt;</code>)</td>
</tr>
<tr>
<td><code>InsertRange</code></td>
<td>(Inherited from <code>List&lt;Regex&gt;</code>)</td>
</tr>
<tr>
<td><code>LastIndexOf(T)</code></td>
<td>(Inherited from <code>List&lt;Regex&gt;</code>)</td>
</tr>
<tr>
<td><code>LastIndexOf(T, Int32)</code></td>
<td>(Inherited from <code>List&lt;Regex&gt;</code>)</td>
</tr>
<tr>
<td><code>LastIndexOf(T, Int32, Int32)</code></td>
<td>(Inherited from <code>List&lt;Regex&gt;</code>)</td>
</tr>
<tr>
<td><code>Remove</code></td>
<td>(Inherited from <code>List&lt;Regex&gt;</code>)</td>
</tr>
<tr>
<td><code>RemoveAll</code></td>
<td>(Inherited from <code>List&lt;Regex&gt;</code>)</td>
</tr>
<tr>
<td><code>RemoveAt</code></td>
<td>(Inherited from <code>List&lt;Regex&gt;</code>)</td>
</tr>
<tr>
<td><code>RemoveRange</code></td>
<td>(Inherited from <code>List&lt;Regex&gt;</code>)</td>
</tr>
<tr>
<td><code>Reverse</code></td>
<td>(Inherited from <code>List&lt;Regex&gt;</code>)</td>
</tr>
<tr>
<td><code>Reverse(Int32, Int32)</code></td>
<td>(Inherited from <code>List&lt;Regex&gt;</code>)</td>
</tr>
<tr>
<td><code>Sort</code></td>
<td>(Inherited from <code>List&lt;Regex&gt;</code>)</td>
</tr>
<tr>
<td>Method</td>
<td>Type</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Sort(IComparer&lt;T&gt;)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Sort(Comparison&lt;T&gt;)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Sort(Int32, Int32, IComparer&lt;T&gt;)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>ToArray</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>TrimExcess</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>TrueForAll</td>
<td>(Inherited from ListRegex.)</td>
</tr>
</tbody>
</table>

**Top**

**See Also**

Reference
- *IronWebScraper.Urls Namespace*
UrlMatchPatternCollection Constructor

Initializes a new instance of the UrlMatchPatternCollection class

Namespace: IronWebScraper.Urls
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

```
public UrlMatchPatternCollection()
```

See Also

Reference
UrlMatchPatternCollection Class
IronWebScraper.Urls Namespace
IronWebScraper - The C# Web Scraping Library
UrlMatchPatternCollection Properties

The UrlMatchPatternCollection type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Count</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Item</td>
<td>(Inherited from ListRegex.)</td>
</tr>
</tbody>
</table>

See Also

Reference

UrlMatchPatternCollection Class
IronWebScraper.Urls Namespace
UrlMatchPatternCollection

Methods

The `UrlMatchPatternCollection` type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(T)</td>
<td>(Inherited from <code>ListRegex</code>.)</td>
</tr>
<tr>
<td>Add(String)</td>
<td>Adds a match pattern using strings with wild-card. ? matches exactly 1 character * matches exactly zero to any number of characters</td>
</tr>
<tr>
<td>Add(String)</td>
<td>Adds multiple match patterns using strings with wild-card. ? matches exactly 1 character * matches exactly zero to any number of characters</td>
</tr>
<tr>
<td>AddRange</td>
<td>(Inherited from <code>ListRegex</code>.)</td>
</tr>
<tr>
<td>AsReadOnly</td>
<td>(Inherited from <code>ListRegex</code>.)</td>
</tr>
<tr>
<td>Method Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>BinarySearch(T)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>BinarySearch(T, IComparerT)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>BinarySearch(Int32, Int32, T, IComparerT)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Clear</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Contains</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>ConvertAllTOOutput</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>CopyTo(T)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>CopyTo(T, Int32)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>CopyTo(Int32, T, Int32, Int32)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Equals</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Exists</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Find</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>FindAll</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>FindIndex(PredicateT)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Method</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>FindIndex(Int32, PredicateT)</td>
<td></td>
</tr>
<tr>
<td>FindIndex(Int32, Int32, PredicateT)</td>
<td></td>
</tr>
<tr>
<td>FindLast</td>
<td></td>
</tr>
<tr>
<td>FindLastIndex(PredicateT)</td>
<td></td>
</tr>
<tr>
<td>FindLastIndex(Int32, PredicateT)</td>
<td></td>
</tr>
<tr>
<td>FindLastIndex(Int32, Int32, PredicateT)</td>
<td></td>
</tr>
<tr>
<td>ForEach</td>
<td></td>
</tr>
<tr>
<td>GetEnumerator</td>
<td></td>
</tr>
<tr>
<td>GetHashCode</td>
<td></td>
</tr>
<tr>
<td>GetRange</td>
<td></td>
</tr>
<tr>
<td>GetType</td>
<td></td>
</tr>
<tr>
<td>IndexOf(T)</td>
<td></td>
</tr>
<tr>
<td>IndexOf(T, Int32)</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Inherited from ListRegex.</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td><code>IndexOf(T, Int32, Int32)</code></td>
<td></td>
</tr>
<tr>
<td><code>Insert</code></td>
<td></td>
</tr>
<tr>
<td><code>InsertRange</code></td>
<td></td>
</tr>
<tr>
<td><code>LastIndexOf(T)</code></td>
<td></td>
</tr>
<tr>
<td><code>LastIndexOf(T, Int32)</code></td>
<td></td>
</tr>
<tr>
<td><code>LastIndexOf(T, Int32, Int32)</code></td>
<td></td>
</tr>
<tr>
<td><code>Remove</code></td>
<td></td>
</tr>
<tr>
<td><code>RemoveAll</code></td>
<td></td>
</tr>
<tr>
<td><code>RemoveAt</code></td>
<td></td>
</tr>
<tr>
<td><code>RemoveRange</code></td>
<td></td>
</tr>
<tr>
<td><code>Reverse</code></td>
<td></td>
</tr>
<tr>
<td><code>Reverse(Int32, Int32)</code></td>
<td></td>
</tr>
<tr>
<td><code>Sort</code></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Sort(TComparer)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Sort(ComparisonT)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>Sort(Int32, Int32, IComparerT)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>ToArray</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>ToString</td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>TrimExcess</td>
<td>(Inherited from ListRegex.)</td>
</tr>
<tr>
<td>TrueForAll</td>
<td>(Inherited from ListRegex.)</td>
</tr>
</tbody>
</table>

**See Also**

Reference

UrlMatchPatternCollection Class
IronWebScraper.Urls Namespace
IronWebScraper - The C# Web Scraping Library
# UrlMatchPatternCollectionAdd Method

## Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(T)</td>
<td>(Inherited from ListRegex.)</td>
</tr>
</tbody>
</table>
| Add(String) | Adds a match pattern using strings with wild-card.  
  ? matches exactly 1 character  
  * matches exactly zero to any number of characters |
| Add(String) | Adds multiple match patterns using strings with wild-card.  
  ? matches exactly 1 character  
  * matches exactly zero to any number of characters |

## See Also

Reference

UrlMatchPatternCollection Class
IronWebScraper.Urls Namespace
IronWebScraper - The C# Web Scraping Library
UrlMatchPatternCollectionAdd Method (String)

Adds a match pattern using strings with wild-card.
? matches exactly 1 character
* matches exactly zero to any number of characters

Namespace: IronWebScraper.Urls
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

Syntax

C# VB
Copy

```csharp
public void Add(
    string WildCardPattern
)
```

Parameters

*WildCardPattern*
Type: System.String
The wild-card pattern.

See Also

Reference
UrlMatchPatternCollection Class
Add Overload
IronWebScraper.Urls Namespace
UrlMatchPatternCollection.Add Method (String)

Adds multiple match patterns using strings with wild-card.
? matches exactly 1 character
* matches exactly zero to any number of characters

Namespace: IronWebScraper.Urls
Assembly: IronWebScraper (in IronWebScraper.dll) Version: 4.0.4.25470 (4.0.4.3)

### Syntax

<table>
<thead>
<tr>
<th>C#</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td>public void Add( params string[] WildCardPatterns )</td>
<td></td>
</tr>
</tbody>
</table>

### Parameters

* **WildCardPatterns**
  - Type: System.String
  - The wild-card patterns.

### See Also

Reference
UrlMatchPatternCollection Class
Add Overload
IronWebScraper.Urls Namespace