

1 System.Net.HttpWebRequest Class

```
2 [ILAsm]  
3 .class public serializable HttpWebRequest extends System.Net.WebRequest  
4 [C#]  
5 public class HttpWebRequest: WebRequest
```

6 Assembly Info:

- 7 • *Name:* System
- 8 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- 9 • *Version:* 2.0.x.x
- 10 • *Attributes:*
 - 11 ○ CLSCompliantAttribute(true)

12 Summary

13 Provides an HTTP-specific implementation of the `System.Net.WebRequest` class.

14 Inherits From: System.Net.WebRequest

15

16 **Library:** Networking

17

18 **Thread Safety:** All public static members of this type are safe for multithreaded operations.
19 No instance members are guaranteed to be thread safe.

20

21 Description

22 This class implements properties and methods defined in `System.Net.WebRequest` and
23 provides additional properties and methods that enable the user to interact directly with
24 servers using the Hypertext Transfer Protocol (HTTP).

25

26 [*Note:* Instances of this class are automatically created by the `System.Net.WebRequest`
27 class. For example, an instance of `System.Net.HttpWebRequest` is created when the
28 `System.Net.WebRequest.Create` method is called and a Uniform Resource Identifier
29 (URI) beginning with `http://` is specified. It is expected that an instance of this class
30 will be constructed for every request made to the server. For example, after a call to
31 `System.Net.HttpWebRequest.Abort` cancels an asynchronous operation, a call to
32 `System.Net.HttpWebRequest.GetRequestStream` causes a `System.Net.WebException`
33 to be thrown.

34

35 Requests can be sent synchronously or asynchronously. The
36 `System.Net.HttpWebRequest.GetResponse` method sends a request to a server
37 synchronously and returns a `System.Net.HttpWebResponse` instance containing the
38 response. An asynchronous request for a resource is sent using the
39 `System.Net.HttpWebRequest.BeginGetResponse` and
40 `System.Net.HttpWebRequest.EndGetResponse` methods.

41

42 Request data is sent using a request stream. The

1 System.Net.HttpWebRequest.GetRequestStream,
 2 System.Net.HttpWebRequest.BeginGetRequestStream, and
 3 System.Net.HttpWebRequest.EndGetRequestStream methods return a
 4 System.IO.Stream instance used to send data.

5
 6 When errors occur while accessing an Internet resource, the
 7 System.Net.HttpWebRequest class throws a System.Net.WebException, and the
 8 System.Net.WebException.Status property that indicates the source of the error.
 9 When System.Net.WebException.Status is
 10 System.Net.WebExceptionStatus.ProtocolError, the
 11 System.Net.WebException.Response property contains the
 12 System.Net.HttpWebResponse received from the Internet resource.

13
 14 Certain HTTP headers are protected; the user cannot set them directly in the header
 15 collection obtained from the System.Net.HttpWebRequest.Headers property. Instead,
 16 these headers are set using the associated properties of a System.Net.HttpWebRequest
 17 instance, or are set by the system. The following table describes how each protected
 18 header is set.

| Header | Set by |
|-------------------|---|
| Accept | System.Net.HttpWebRequest.Accept |
| Connection | System.Net.HttpWebRequest.Connection System.Net.HttpWebRequest.KeepAlive |
| Content-Length | System.Net.HttpWebRequest.ContentLength |
| Content-Type | System.Net.HttpWebRequest.ContentType |
| Expect | System.Net.HttpWebRequest.Expect |
| Date | Set to current date by the system. |
| Host | Set to current host by the system. |
| if-Modified-since | System.Net.HttpWebRequest.IfModifiedSince |
| Range | System.Net.HttpWebRequest.AddRange |
| Referer | System.Net.HttpWebRequest.Referer |
| Transfer-Encoding | System.Net.HttpWebRequest.TransferEncoding System.Net.HttpWebRequest.SendChunked |

| | |
|------------|-------------------------------------|
| User-Agent | System.Net.HttpWebRequest.UserAgent |
|------------|-------------------------------------|

1

2]

3

1 `HttpRequest.Abort()` Method

```
2 [ILAsm]  
3 .method public hidebysig virtual void Abort()  
4 [C#]  
5 public override void Abort()
```

6 **Summary**

7 Cancels an asynchronous operation.

8 **Description**

9 `System.Net.HttpRequest.Abort` cancels any pending asynchronous operation. After
10 this method is called, calling `System.Net.HttpRequest.GetResponse`,
11 `System.Net.HttpRequest.BeginGetResponse`,
12 `System.Net.HttpRequest.EndGetResponse`,
13 `System.Net.HttpRequest.GetRequestStream`,
14 `System.Net.HttpRequest.BeginGetRequestStream`, or
15 `System.Net.HttpRequest.EndGetRequestStream` will throw a
16 `System.Net.WebException` with `System.Net.WebException.Status` set to
17 `System.Net.WebExceptionStatus.RequestCanceled`.

18
19 *[Note: If no pending request exists, calling this method does not cause an exception to*
20 *be thrown.*

21
22 This method overrides `System.Net.WebRequest.Abort`.

23
24]

25

1 `HttpRequest.AddRange(System.String, System.Int32)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance void AddRange(string rangeSpecifier,  
5 int32 range)  
  
6 [C#]  
7 public void AddRange(string rangeSpecifier, int range)
```

8 Summary

9 Adds a HTTP Range header to the current request for a specific range from the
10 beginning or end of the requested data.

11 Parameters

| Parameter | Description |
|-----------------------|---|
| <i>rangeSpecifier</i> | A <code>System.String</code> that contains the description of the range. |
| <i>range</i> | A <code>System.Int32</code> that designates the starting or ending point of the range. If this value is positive, the range is from the beginning of the data to <i>range</i> . If this value is negative, the range is from <i>range</i> to the end of the data. |

12

13 Description

14 [Note: The HTTP Range header specifies either a single range of bytes or a set of byte
15 ranges in an entity-body to be returned. If the server accessed by the current instance
16 supports the use of this header, this allows for the partial retrieval of the entity due to,
17 for example, the entity being particularly large or there having been a failed transfer of
18 data.

19

20 For more information on the HTTP Range header, see Section 14.35 of RFC 2616.

21

22]

23 Exceptions

| Exception | Condition |
|---|-----------------------------------|
| <code>System.ArgumentNullException</code> | <i>rangeSpecifier</i> is null. |
| <code>System.ArgumentException</code> | <i>rangeSpecifier</i> is invalid. |

System.InvalidOperationException

The range header could not be added.

1

2

1 `HttpRequest.AddRange(System.String,` 2 `System.Int32, System.Int32)` Method

```
3 [ILAsm]  
4 .method public hidebysig instance void AddRange(string rangeSpecifier,  
5 int32 from, int32 to)  
  
6 [C#]  
7 public void AddRange(string rangeSpecifier, int from, int to)
```

8 Summary

9 Adds a HTTP Range header to the current instance for a specified range.

10 Parameters

| Parameter | Description |
|-----------------------|--|
| <i>rangeSpecifier</i> | A <code>System.String</code> that contains the description of the range. |
| <i>from</i> | A <code>System.Int32</code> designating the position at which to start sending data. |
| <i>to</i> | A <code>System.Int32</code> designating the position at which to stop sending data. |

11

12 Description

13 [Note: The HTTP Range header specifies either a single range of bytes or a set of byte
14 ranges in an entity-body to be returned. If the server accessed by the current instance
15 supports the use of this header, this allows for the partial retrieval of the entity due to,
16 for example, the entity being particularly large or there having been a failed transfer of
17 data.

18

19 For more information on the HTTP Range header, see Section 14.35 of IETF RFC 2616.

20

21]

22 Exceptions

| Exception | Condition |
|---|--------------------------------|
| System.ArgumentNullException | <i>rangeSpecifier</i> is null. |
| System.ArgumentOutOfRangeException | <i>from</i> < 0. |
| | -or- |

| | |
|---|---|
| | <i>to</i> < 0. -or- <i>from</i> > <i>to</i> . |
| System.ArgumentException | <i>rangeSpecifier</i> is invalid. |
| System.InvalidOperationException | The range header could not be added. |

1

2

1 `HttpRequest.AddRange(System.Int32)`

2 Method

```
3 [ILAsm]  
4 .method public hidebysig instance void AddRange(int32 range)  
  
5 [C#]  
6 public void AddRange(int range)
```

7 Summary

8 Adds a HTTP Range header to the current instance for a specific range from the
9 beginning or end of the requested data.

10 Parameters

| Parameter | Description |
|--------------|--|
| <i>range</i> | A <code>System.Int32</code> that specifies the starting or ending point of the range. If this value is positive, the range is from the beginning of the data to <i>range</i> . If this value is negative, the range is from <i>range</i> to the end of the data. |

11 Description

13 This method is equivalent to `System.Net.HttpWebRequest.AddRange("bytes", range)`.

14
15 [Note: The HTTP Range header specifies either a single range of bytes or a set of byte
16 ranges in an entity-body to be returned. If the server accessed by the current instance
17 supports the use of this header, this allows for the partial retrieval of the entity due to,
18 for example, the entity being particularly large or there having been a failed transfer of
19 data.

20
21 For more information on the HTTP Range header, see Section 14.35 of RFC 2616.

22]
23

24 Exceptions

| Exception | Condition |
|---|--------------------------------------|
| <code>System.InvalidOperationException</code> | The range header could not be added. |

25

26

1 `HttpRequest.AddRange(System.Int32,` 2 `System.Int32) Method`

```
3 [ILAsm]  
4 .method public hidebysig instance void AddRange(int32 from, int32 to)  
5 [C#]  
6 public void AddRange(int from, int to)
```

7 Summary

8 Adds a HTTP Range header to the current instance for a specified range.

9 Parameters

| Parameter | Description |
|-------------|---|
| <i>from</i> | A <code>System.Int32</code> indicating the starting byte position of the entity-body data to be returned. |
| <i>to</i> | A <code>System.Int32</code> indicating the last byte. |

10

11 Description

12 This method is equivalent to `System.Net.HttpWebRequest.AddRange("bytes", from, to)`.

13

14 [Note: The HTTP Range header specifies either a single range of bytes or a set of byte
15 ranges in an entity-body to be returned. If the server accessed by the current instance
16 supports the use of this header, this allows for the partial retrieval of the entity due to,
17 for example, the entity being particularly large or there having been a failed transfer of
18 data.

19

20 For more information on the HTTP Range header, see Section 14.35 of RFC 2616.

21

22]

23 Exceptions

| Exception | Condition |
|---|--|
| System.ArgumentOutOfRangeException | <i>from</i> < 0. -or- <i>to</i> < 0. |

| | |
|---|--------------------------------------|
| | -or- <i>from > to.</i> |
| System.InvalidOperationException | The range header could not be added. |

1

2

1 2 **HttpRequest.BeginGetRequestStream(System.AsyncCallback, System.Object) Method** 3

```
4 [ILAsm]  
5 .method public hidebysig virtual class System.IAsyncResult  
6 BeginGetRequestStream(class System.AsyncCallback callback, object state)  
7  
8 [C#]  
9 public override IAsyncResult BeginGetRequestStream(AsyncCallback callback,  
10 object state)
```

10 **Summary**

11 Begins an asynchronous request for a stream in which to write data to be sent in the
12 current instance.

13 **Parameters**

| Parameter | Description |
|-----------------|---|
| <i>callback</i> | A System.AsyncCallback delegate to be called when the stream is available. Can be null. |
| <i>state</i> | A System.Object containing state information for the asynchronous request. Can be null. |

14 15 **Return Value**

16 A System.IAsyncResult that contains information about the asynchronous operation.

17 **Description**

18 This method starts an asynchronous operation. To get the request stream, call the
19 System.Net.HttpWebRequest.EndGetRequestStream method and specify the
20 System.IAsyncResult object returned by this method. [Note: The
21 System.Net.HttpWebRequest.EndGetRequestStream method should be called exactly
22 once for each call to System.Net.HttpWebRequest.BeginGetRequestStream.]
23
24
25

26 If the *callback* parameter is not null, the method(s) referenced by *callback* are invoked
27 when the asynchronous operation completes. The System.IAsyncResult object
28 returned by this method is passed as the argument to the method(s) referenced by
29 *callback*.
30

31 The *state* parameter can be any object that the caller wishes to have available for the

1 duration of the asynchronous operation. This object is available via the
 2 `System.IAsyncResult.AsyncState` property of the object returned by this method.
 3
 4 The value of the `System.Net.HttpWebRequest.ContentLength` property of the current
 5 instance is required to be set prior to calling this method.
 6
 7 *[Note:* The method(s) invoked by the callback delegate can call the
 8 `System.Net.HttpWebRequest.EndGetRequestStream` method to retrieve the stream.
 9
 10 This method is the asynchronous version of the
 11 `System.Net.HttpWebRequest.GetRequestStream` method.
 12
 13 This method overrides `System.Net.WebRequest.BeginGetRequestStream`.
 14
 15]

16 **Exceptions**

| Exception | Condition |
|--|---|
| System.InvalidOperationException | The stream is being used by a previous call to <code>System.Net.HttpWebRequest.BeginGetRequestStream</code> . -or- No writeable stream is available. |
| System.Net.ProtocolViolationException | The <code>System.Net.HttpWebRequest.ContentLength</code> property of the current instance is not set. -or- The <code>System.Net.HttpWebRequest.Method</code> property of the current instance is "GET" or "HEAD". |

17

18

1 2 **HttpRequest.BeginGetResponse(System. 3 AsyncCallback, System.Object) Method**

```
4 [ILAsm]  
5 .method public hidebysig virtual class System.IAsyncResult  
6 BeginGetResponse(class System.AsyncCallback callback, object state)  
7  
8 [C#]  
9 public override IAsyncResult BeginGetResponse(AsyncCallback callback,  
10 object state)
```

10 **Summary**

11 Begins sending the current HTTP request asynchronously.

12 **Parameters**

| Parameter | Description |
|-----------------|---|
| <i>callback</i> | A System.AsyncCallback delegate to be called when the stream is available. Can be null. |
| <i>state</i> | A System.Object containing state information for the asynchronous request. Can be null. |

13 14 **Return Value**

15 A System.IAsyncResult that contains information about the asynchronous operation.

16 **Description**

17 This method starts an asynchronous operation. To get the response, call the
18 System.Net.HttpWebRequest.EndGetResponse method and specify the
19 System.IAsyncResult object returned by this method. [Note: The
20 System.Net.HttpWebRequest.EndGetResponse method should be called exactly once
21 for each call to System.Net.HttpWebRequest.BeginGetResponse.]
22
23

24
25 If the *callback* parameter is not null, the method referenced by *callback* is invoked
26 when the asynchronous operation completes. The System.IAsyncResult object
27 returned by this method is passed as the argument to the method referenced by
28 *callback*.
29

30 The *state* parameter can be any object that the caller wishes to have available for the
31 duration of the asynchronous operation. This object is available via the

1 `System.IAsyncResult.AsyncState` property of the object returned by this method.
2
3 [*Note:* The method(s) invoked by the callback delegate can call the
4 `System.Net.HttpWebRequest.EndGetResponse` method to retrieve the response.
5
6 This method is the asynchronous version of the
7 `System.Net.HttpWebRequest.GetResponse` method.
8
9 This method overrides `System.Net.WebRequest.BeginGetResponse`.
10
11]

12 Exceptions

| Exception | Condition |
|--|---|
| System.InvalidOperationException | <code>System.Net.HttpWebRequest.BeginGetResponse</code> or <code>System.Net.HttpWebRequest.GetResponse</code> was previously called on this instance. |
| System.Net.ProtocolViolationException | The <code>System.Net.HttpWebRequest.ContentLength</code> property of the current instance has not been set. |
| System.Net.WebException | <code>System.Net.HttpWebRequest.Abort</code> was previously called. |

13

14

1
2 **HttpRequest.EndGetRequestStream(System.IAsyncResult) Method**
3

```
4 [ILAsm]  
5 .method public hidebysig virtual class System.IO.Stream  
6 EndGetRequestStream(class System.IAsyncResult asyncResult)  
7  
8 [C#]  
9 public override Stream EndGetRequestStream(IAsyncResult asyncResult)
```

9 **Summary**

10 Completes an asynchronous request for a stream that was started by the
11 System.Net.HttpWebRequest.BeginGetRequestStream method.

12 **Parameters**

| Parameter | Description |
|--------------------|---|
| <i>asyncResult</i> | The System.IAsyncResult object that holds the state information for the asynchronous operation. |

13
14 **Return Value**

15 A System.IO.Stream to write request data to.

16 **Description**

17 [Note: The caller is responsible for calling the System.IO.Stream.Close method to close
18 the stream.

19
20 This method overrides System.Net.WebRequest.EndGetRequestStream.

21]
22

23 **Exceptions**

| Exception | Condition |
|-------------------------------------|---|
| System.ArgumentNullException | <i>asyncResult</i> is null. |
| System.ArgumentException | <i>asyncResult</i> was not returned by the current instance from a call to System.Net.WebRequest.BeginGetRequestStream. |

| | |
|---|---|
| System.InvalidOperationException | This method was called previously using <i>asyncResult</i> . -or- No stream is available. |
| System.Net.WebException | System.Net.HttpWebRequest.Abort was previously called. -or- An error occurred while processing the request. |

1

2

1 2 `HttpRequest.EndGetResponse(System.IA` 3 `syncResult)` Method

```
4 [IAsm]  
5 .method public hidebysig virtual class System.Net.WebResponse  
6 EndGetResponse(class System.IAsyncResult asyncResult)  
7 [C#]  
8 public override WebResponse EndGetResponse(IAsyncResult asyncResult)
```

9 Summary

10 Returns a `System.Net.WebResponse` that contains a response to the specified pending
11 Internet request.

12 Parameters

| Parameter | Description |
|--------------------|---|
| <i>asyncResult</i> | The <code>System.IAsyncResult</code> object that hold the state information for the asynchronous operation. |

13 14 Return Value

15 A `System.Net.WebResponse` that contains a response to the Internet request referenced
16 by *asyncResult*.

17 Description

18 [Note: This method completes an asynchronous request for an Internet resource that
19 was started by calling `System.Net.HttpWebRequest.BeginGetResponse`.

20 This method overrides `System.Net.WebRequest.EndGetResponse`.

21]
22
23

24 Exceptions

| Exception | Condition |
|---|--|
| <code>System.ArgumentNullException</code> | <i>asyncResult</i> is null. |
| <code>System.ArgumentException</code> | <i>asyncResult</i> was not returned by the current instance from a call to |

| | |
|---|---|
| | System.Net.WebRequest.BeginGetResponse. |
| System.InvalidOperationException | <p>This method was called previously using <i>asyncResult</i>.</p> <p>-or-</p> <p>The System.Net.HttpWebRequest.ContentLength property of the current instance is greater than 0 but the data has not been written to the request stream.</p> |
| System.Net.WebException | <p>System.Net.HttpWebRequest.Abort was previously called.</p> <p>-or-</p> <p>An error occurred while processing the request.</p> |

1

2

1 `HttpRequest.GetHashCode()` Method

```
2 [ILAsm]  
3 .method public hidebysig virtual int32 GetHashCode()  
4 [C#]  
5 public override int GetHashCode()
```

6 **Summary**

7 Generates a hash code for the current instance.

8 **Return Value**

9 A `System.Int32` containing the hash code for the current instance.

10 **Description**

11 The algorithm used to generate the hash code is unspecified.

12

13 [*Note:* This method overrides `System.Object.GetHashCode()`.]

14

15

16

1 `HttpRequest.GetResponseStream()`

2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual class System.IO.Stream GetRequestStream()  
5 [C#]  
6 public override Stream GetRequestStream()
```

7 Summary

8 Returns a `System.IO.Stream` for writing data to the Internet resource requested by the
9 current instance.

10 Return Value

11 A `System.IO.Stream` for writing data to an Internet resource requested by the current
12 instance.

13 Description

14 The value of the `System.Net.HttpWebRequest.ContentLength` property is required to
15 be set before writing data to the stream.

16
17 [*Note:* This method returns a stream to use to send data for the
18 `System.Net.HttpWebRequest`. Once the `System.IO.Stream` instance has been returned,
19 data can be sent with the `System.Net.HttpWebRequest` by using the
20 `System.IO.Stream.Write` method.

21
22 Call the `System.IO.Stream.Close` method to close the stream and release the
23 connection for reuse. Failure to close the stream might cause the application to run out
24 of connections.

25
26 This method overrides `System.Net.WebRequest.GetResponseStream`.

27
28]

29 Exceptions

| Exception | Condition |
|--|---|
| System.Net.ProtocolViolationException | The <code>System.Net.HttpWebRequest.Method</code> property of the current instance is "GET" or "HEAD". -or- The |

| | |
|--|--|
| | <p><code>System.Net.HttpWebRequest.ContentLength</code> property of the current instance is not set.</p> |
| <p>System.InvalidOperationException</p> | <p>The <code>System.Net.HttpWebRequest.GetRequestStream</code> method was called more than once.</p> <p>-or-</p> <p>No writeable stream is available.</p> |
| <p>System.Net.WebException</p> | <p><code>System.Net.HttpWebRequest.Abort</code> was previously called.</p> <p>-or-</p> <p>The timeout period for the request expired.</p> <p>-or-</p> <p>An error occurred while processing the request.</p> |

1

2

1 `HttpRequest.GetResponse()` Method

```
2 [ILAsm]  
3 .method public hidebysig virtual class System.Net.WebResponse  
4 GetResponse()  
  
5 [C#]  
6 public override WebResponse GetResponse()
```

7 **Summary**

8 Returns a response to an Internet request.

9 **Return Value**

10 A `System.Net.WebResponse` containing the response from the Internet resource
11 requested by the current instance.

12 **Description**

13 [*Note:* This method returns a `System.Net.WebResponse` instance containing the
14 response from the Internet resource requested by the current instance. The actual
15 instance returned is an instance of `System.Net.HttpWebResponse`, and can be typecast
16 to that class to access HTTP-specific properties.

17 This method overrides `System.Net.WebRequest.GetResponse`.

18]
19
20

21 **Exceptions**

| Exception | Condition |
|--|---|
| System.Net.ProtocolViolationException | The <code>System.Net.HttpWebRequest.ContentLength</code> property of the current instance is not set. |
| System.Net.WebException | <code>System.Net.HttpWebRequest.Abort</code> was previously called. |
| | -or- |
| | The timeout period for the request expired. |
| | -or- |
| | An error occurred while processing the request. |

1

2

1 **HttpRequest.Accept Property**

```
2 [ILAsm]  
3 .property string Accept { public hidebysig specialname instance string  
4 get_Accept() public hidebysig specialname instance void set_Accept(string  
5 value) }  
6 [C#]  
7 public string Accept { get; set; }
```

8 **Summary**

9 Gets or sets a `System.String` containing the value of the HTTP Accept header.

10 **Property Value**

11 A `System.String` containing the value of the HTTP Accept header. The default value of
12 this property is `null`.

13 **Description**

14 [*Note:* For additional information see section 14.1 of IETF RFC 2616 - HTTP/1.1.]
15
16
17

1 HttpWebRequest.Address Property

```
2 [ILAsm]  
3 .property class System.Uri Address { public hidebysig specialname instance  
4 class System.Uri get_Address() }  
  
5 [C#]  
6 public Uri Address { get; }
```

7 Summary

8 Gets the URI that responds to the current request.

9 Property Value

10 A `System.Uri` identifying the Internet resource that responds to the current request.
11 The default is the URI used by the `System.Net.WebRequest.Create` method to initialize
12 the current instance.

13 Description

14 This property is read-only.

15
16 The value of this property is set to the URI that is the source of the response after all
17 redirections are complete.

18
19 [*Note:* The URI of the original request is kept in the
20 `System.Net.HttpWebRequest.RequestUri` property.]
21
22

23

1 **HttpRequest.AllowAutoRedirect Property**

```
2 [ILAsm]  
3 .property bool AllowAutoRedirect { public hidebysig specialname instance  
4 bool get_AllowAutoRedirect() public hidebysig specialname instance void  
5 set_AllowAutoRedirect(bool value) }  
  
6 [C#]  
7 public bool AllowAutoRedirect { get; set; }
```

8 **Summary**

9 Gets or sets a `System.Boolean` value that indicates whether the current request will
10 follow redirection responses.

11 **Property Value**

12 `true` if the current request will automatically follow redirection responses from the
13 Internet resource; otherwise `false`. The default value is `true`.

14 **Description**

15 [*Note:* Set `System.Net.HttpWebRequest.AllowAutoRedirect` to `true` to allow the
16 current request to automatically follow HTTP redirection headers to the new location of a
17 resource.

18
19 The maximum number of redirections to follow is set by the
20 `System.Net.HttpWebRequest.MaximumAutomaticRedirections` property.

21]
22]

23

1 HttpWebRequest.AllowWriteStreamBuffering 2 Property

```
3 [ILAsm]  
4 .property bool AllowWriteStreamBuffering { public hidebysig specialname  
5 instance bool get_AllowWriteStreamBuffering() public hidebysig specialname  
6 instance void set_AllowWriteStreamBuffering(bool value) }  
  
7 [C#]  
8 public bool AllowWriteStreamBuffering { get; set; }
```

9 Summary

10 Gets or sets a `System.Boolean` value that indicates whether to buffer the data sent to
11 the Internet resource requested by the current instance.

12 Property Value

13 `true` to enable buffering of the data sent to the Internet resource requested by the
14 current instance; `false` to disable buffering. The default is `true`.

15 Description

16 [*Note:* When `System.Net.HttpWebRequest.AllowWriteStreamBuffering` is `true`, the
17 data is buffered in memory so it is ready to be resent in the event of redirections or
18 authentication requests.

19
20 Depending on available memory, setting
21 `System.Net.HttpWebRequest.AllowWriteStreamBuffering` as `true` might impact
22 system performance when uploading large amounts of data.

23
24]

25

1 HttpWebRequest.Connection Property

```
2 [ILAsm]
3 .property string Connection { public hidebysig specialname instance string
4 get_Connection() public hidebysig specialname instance void
5 set_Connection(string value) }
6
7 [C#]
8 public string Connection { get; set; }
```

8 Summary

9 Gets or sets the value of the Connection HTTP header.

10 Property Value

11 A `System.String` containing the value of the Connection HTTP header. The default value
12 is `null`.

13 Description

14 The current request sends the `System.Net.HttpWebRequest.Connection` property to
15 the Internet resource as the Connection HTTP header.

16
17 *[Note: If `System.Net.HttpWebRequest.KeepAlive` is true, the value "Keep-alive" is
18 appended to the end of the Connection header.*

19
20 For additional information see section 14.10 of IETF RFC 2616 - HTTP/1.1.

21
22]

23 Exceptions

| Exception | Condition |
|---------------------------------|---|
| System.ArgumentException | The value of <code>System.Net.HttpWebRequest.Connection</code> is set to "Keep-alive" or "Close". This value is case insensitive. |

24

25

1 HttpWebRequest.ConnectionGroupName 2 Property

```
3 [ILAsm]  
4 .property string ConnectionGroupName { public hidebysig virtual  
5 specialname string get_ConnectionGroupName() public hidebysig virtual  
6 specialname void set_ConnectionGroupName(string value) }  
7 [C#]  
8 public override string ConnectionGroupName { get; set; }
```

9 Summary

10 Gets or sets the name of the connection group for the current instance.

11 Property Value

12 A `System.String` that contains the name of the connection group for the current
13 instance. The default value is `null`.

14 Description

15 [*Note:* The `System.Net.HttpWebRequest.ConnectionGroupName` property enables a
16 request to be associated with a connection group. This is useful when an application
17 makes requests to one server for different users, such as a Web site that retrieves
18 customer information from a database server.

19
20 Each connection group creates additional connections for a server. This might result in
21 exceeding `System.Net.ServicePoint.ConnectionLimit` for that server.

22
23 This property overrides `System.Net.WebRequest.ConnectionGroupName`.

24
25]

26

1 HttpWebRequest.ContentLength Property

```
2 [ILAsm]  
3 .property int64 ContentLength { public hidebysig virtual specialname int64  
4 get_ContentLength() public hidebysig virtual specialname void  
5 set_ContentLength(int64 value) }  
  
6 [C#]  
7 public override long ContentLength { get; set; }
```

8 Summary

9 Gets or sets the Content-length HTTP header.

10 Property Value

11 A `System.Int64` value that specifies the number of bytes of data to send to the Internet
12 resource. The default is -1, which indicates that this value has not been set.

13 Description

14 The `System.Net.HttpWebRequest.ContentLength` property contains the value to send
15 as the Content-length HTTP header of the request.

16
17 Any value other than -1 in the `System.Net.HttpWebRequest.ContentLength` property
18 indicates that the request will upload data; only methods that upload data are allowed in
19 the `System.Net.HttpWebRequest.Method` property.

20
21 This property is required to be set prior to writing data to the request data stream. Once
22 the `System.Net.HttpWebRequest.ContentLength` property is set to a value, that
23 number of bytes is required to be written to the request data stream. [*Note:* Get the
24 request data stream by calling `System.Net.HttpWebRequest.GetRequestStream`, or
25 `System.Net.HttpWebRequest.BeginGetRequestStream` and
26 `System.Net.HttpWebRequest.EndGetRequestStream`.]
27

28
29
30 [*Note:* For additional information see section 14.13 of IETF RFC 2616 - HTTP/1.1.

31
32 This property overrides `System.Net.WebRequest.ContentLength`.

33]
34

35 Exceptions

| Exception | Condition |
|---|--|
| <code>System.InvalidOperationException</code> | Data has already been written to the request stream. |

| | |
|---|--|
| System.ArgumentOutOfRangeException | A value less than zero is specified for a set operation. |
|---|--|

1

2

1 `HttpRequest.ContentType` Property

```
2 [ILAsm]  
3 .property string ContentType { public hidebysig virtual specialname string  
4 get_ContentType() public hidebysig virtual specialname void  
5 set_ContentType(string value) }  
6 [C#]  
7 public override string ContentType { get; set; }
```

8 **Summary**

9 Gets or sets the value of the Content-type HTTP header of the current instance.

10 **Property Value**

11 The value of the Content-type HTTP header of the current instance. The default value is
12 null.

13 **Description**

14 The `System.Net.HttpWebRequest.ContentType` property contains the media type of the
15 current instance. Values assigned to the `System.Net.HttpWebRequest.ContentType`
16 property replace any existing contents when the request sends the Content-type HTTP
17 header.

18
19 *[Note:* To clear the Content-type HTTP header, set the
20 `System.Net.HttpWebRequest.ContentType` property to null.

21
22 For additional information see section 14.17 of IETF RFC 2616 - HTTP/1.1.

23
24 This property overrides `System.Net.WebRequest.ContentType`.

25
26]

27

1 HttpWebRequest.ContinueDelegate Property

```
2 [ILAsm]  
3 .property class System.Net.HttpContinueDelegate ContinueDelegate { public  
4 hidebysig specialname instance class System.Net.HttpContinueDelegate  
5 get_ContinueDelegate() public hidebysig specialname instance void  
6 set_ContinueDelegate(class System.Net.HttpContinueDelegate value) }  
7  
8 [C#]  
9 public HttpContinueDelegate ContinueDelegate { get; set; }
```

9 Summary

10 Gets or sets the delegate method whose methods are invoked when an HTTP 100-
11 continue response is received by the current instance.

12 Property Value

13 A `System.Net.HttpContinueDelegate` that references the methods that are invoked
14 when an HTTP Continue response is received. The default value is `null`.

15 Description

16 [*Note:* This delegate is useful to display the status of responses received by the current
17 instance.
18
19]

20

1 HttpWebRequest.Credentials Property

```
2 [ILAsm]
3 .property class System.Net.ICredentials Credentials { public hidebysig
4 virtual specialname class System.Net.ICredentials get_Credentials() public
5 hidebysig virtual specialname void set_Credentials(class
6 System.Net.ICredentials value) }
7
8 [C#]
9 public override ICredentials Credentials { get; set; }
```

9 Summary

10 Gets or sets the credentials used for authenticating the current request.

11 Property Value

12 A `System.Net.ICredentials` object containing the authentication credentials associated
13 with the current instance. The default is `null`.

14 Description

15 [*Note:* The `System.Net.HttpWebRequest.Credentials` property contains authentication
16 information to identify the client making the request. The
17 `System.Net.HttpWebRequest.Credentials` property can be either an instance of
18 `System.Net.NetworkCredential`, in which case the user, password, and domain
19 information contained in the `System.Net.NetworkCredential` instance is used to
20 authenticate the request, or it can be an instance of `System.Net.CredentialCache`, in
21 which case the uniform resource identifier (URI) of the request is used to determine the
22 user, password, and domain information to use to authenticate the request.

23
24 This property overrides `System.Net.WebRequest.Credentials`.

25
26]

27

1 HttpWebRequest.Expect Property

```
2 [ILAsm]  
3 .property string Expect { public hidebysig specialname instance string  
4 get_Expect() public hidebysig specialname instance void set_Expect(string  
5 value) }  
  
6 [C#]  
7 public string Expect { get; set; }
```

8 Summary

9 Gets or sets the value of the HTTP Expect header.

10 Property Value

11 A System.String that contains the contents of the HTTP Expect header. The default
12 value is null.

13 Description

14 [*Note:* By default, System.Net.HttpWebRequest.Expect is null. Other values can be
15 added to the list that System.Net.HttpWebRequest.Expect maintains, or all values
16 except "100-continue" can be deleted from the list by setting
17 System.Net.HttpWebRequest.Expect to null.

18 For additional information see section 14.20 of IETF RFC 2616 - HTTP/1.1.

20
21]

22 Exceptions

| Exception | Condition |
|---------------------------------|--|
| System.ArgumentException | The value specified for a set operation is "100-continue". This value is case insensitive. |

23

24

1 HttpWebRequest.HaveResponse Property

```
2 [ILAsm]  
3 .property bool HaveResponse { public hidebysig specialname instance bool  
4 get_HaveResponse() }  
5 [C#]  
6 public bool HaveResponse { get; }
```

7 Summary

8 Gets a `System.Boolean` value indicating whether a response has been received for the
9 current instance.

10 Property Value

11 `true` if a response has been received; otherwise `false`.

12 Description

13 This property is read-only.

14

1 HttpWebRequest.Headers Property

```
2 [ILAsm]
3 .property class System.Net.WebHeaderCollection Headers { public hidebysig
4 virtual specialname class System.Net.WebHeaderCollection get_Headers()
5 public hidebysig virtual specialname void set_Headers(class
6 System.Net.WebHeaderCollection value) }
7
8 [C#]
9 public override WebHeaderCollection Headers { get; set; }
```

9 Summary

10 Gets or sets the collection of HTTP header name/value pairs associated with the current
11 instance.

12 Property Value

13 A `System.Net.WebHeaderCollection` containing the name/value pairs of the headers
14 for the current instance.

15 Description

16 The following table lists the HTTP headers that cannot be set using the collection
17 returned by this property.

| Header | Set by |
|----------------|---|
| Accept | <code>System.Net.HttpWebRequest.Accept.</code> |
| Connection | <code>System.Net.HttpWebRequest.Connection.</code> <code>System.Net.HttpWebRequest.KeepAlive.</code> |
| Content-Length | <code>System.Net.HttpWebRequest.ContentLength.</code> |
| Content-Type | <code>System.Net.HttpWebRequest.ContentType.</code> |
| Expect | <code>System.Net.HttpWebRequest.Expect.</code> |
| Date | Set by the system to the current date. |
| Host | Set by the system to the current host information. |
| Range | <code>System.Net.HttpWebRequest.AddRange.</code> |

| | |
|-------------------|---|
| Referer | <code>System.Net.HttpWebRequest.Referer.</code> |
| Transfer-Encoding | <code>System.Net.HttpWebRequest.TransferEncoding.</code> <code>System.Net.HttpWebRequest.SendChunked.</code> |
| User-Agent | <code>System.Net.HttpWebRequest.UserAgent.</code> |

1
2
3
4

[*Note:* This property overrides `System.Net.WebRequest.Headers.`]

5 **Exceptions**

| Exception | Condition |
|---|---|
| System.InvalidOperationException | A set operation was requested but data has already been written to the request data stream. |

6
7

1 HttpWebRequest.IfModifiedSince Property

```
2 [ILAsm]
3 .property valuetype System.DateTime IfModifiedSince { public hidebysig
4 specialname instance valuetype System.DateTime get_IfModifiedSince()
5 public hidebysig specialname instance void set_IfModifiedSince(valuetype
6 System.DateTime value) }
7 [C#]
8 public DateTime IfModifiedSince { get; set; }
```

9 Summary

10 Gets or sets the value of the HTTP If-Modified-Since header.

11 Property Value

12 A `System.DateTime` that contains the contents of the HTTP If-Modified-Since header.
13 The default value is the current date and time of the system.

14 Description

15 [*Note:* For additional information see section 14.25 of IETF RFC 2616 - HTTP/1.1.]
16
17

18

1 HttpWebRequest.KeepAlive Property

```
2 [ILAsm]  
3 .property bool KeepAlive { public hidebysig specialname instance bool  
4 get_KeepAlive() public hidebysig specialname instance void  
5 set_KeepAlive(bool value) }  
  
6 [C#]  
7 public bool KeepAlive { get; set; }
```

8 Summary

9 Gets or sets a `System.Boolean` value indicating whether to make a persistent
10 connection to the server hosting the Internet resource requested by the current
11 instance.

12 Property Value

13 `true` indicates that the current request will contain an HTTP Connection header with the
14 value "Keep-alive"; otherwise, `false`. The default value is `true`.

15 Description

16 [*Note:* An application uses `System.Net.HttpWebRequest.KeepAlive` to indicate a
17 preference for persistent connections. When this property is `true`, the application makes
18 persistent connections to the servers that support them.]
19
20

21

1 2 **HttpRequest.MaximumAutomaticRedirect** 3 **ions Property**

```
4 [ILAsm]  
5 .property int32 MaximumAutomaticRedirections { public hidebysig  
6 specialname instance int32 get_MaximumAutomaticRedirections() public  
7 hidebysig specialname instance void set_MaximumAutomaticRedirections(int32  
8 value) }  
  
9 [C#]  
10 public int MaximumAutomaticRedirections { get; set; }
```

11 **Summary**

12 Gets or sets the maximum number of redirects that the current instance will follow.

13 **Property Value**

14 A `System.Int32` value that indicates the maximum number of redirection responses that
15 the current instance will follow. The default value is implementation-specific.

16 **Description**

17 [*Note:* This property sets the maximum number of redirections for the request to follow
18 if the `System.Net.HttpWebRequest.AllowAutoRedirect` property is true.]
19
20

21 **Exceptions**

| Exception | Condition |
|---------------------------------|--|
| System.ArgumentException | The value specified for a set operation is less than or equal to zero. |

22
23

1 **HttpRequest.MediaType Property**

```
2 [ILAsm]  
3 .property string MediaType { public hidebysig specialname instance string  
4 get_MediaType() public hidebysig specialname instance void  
5 set_MediaType(string value) }  
  
6 [C#]  
7 public string MediaType { get; set; }
```

8 **Summary**

9 Gets or sets the media type of the current request.

10 **Property Value**

11 A `System.String` that identifies the media type of the current request. The default value
12 is `null`.

13 **Description**

14 [*Note:* The value of this property affects the
15 `System.Net.HttpWebResponse.CharacterSet` property. When this property is set in the
16 current instance, the corresponding media type is chosen from the list of character sets
17 returned in the response HTTP Content-type header.]
18
19

20

1 `HttpRequest.Method` Property

```
2 [ILAsm]  
3 .property string Method { public hidebysig virtual specialname string  
4 get_Method() public hidebysig virtual specialname void set_Method(string  
5 value) }  
  
6 [C#]  
7 public override string Method { get; set; }
```

8 **Summary**

9 Gets or sets the HTTP protocol request method used by the current instance.

10 **Property Value**

11 A `System.String` containing an HTTP method. The default value is "GET".

12 **Description**

13 If the `System.Net.HttpWebRequest.ContentLength` property is set to any value other
14 than -1, the `System.Net.HttpWebRequest.Method` property is required to be set to a
15 protocol method that sends request data.

16
17 The `System.Net.HttpWebRequest.Method` property can be set to any of the following
18 HTTP 1.1 protocol methods:

| HTTP Method | Description |
|-------------|--|
| GET | Retrieves in entity-body form the information identified by the <code>System.Net.HttpWebRequest.RequestUri</code> property of the current instance. |
| HEAD | Identical to GET except that the message-body is not returned in the response. |
| POST | Requests that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified the Request-URI in the Request-Line. |
| PUT | Requests that the enclosed entity be stored under the supplied Request-URI. |
| DELETE | Requests that the origin server delete the resource identified by the Request-URI. |
| TRACE | Invokes a remote, application-layer loopback of the request message. |
| OPTIONS | Requests information about the communication options available on the request/response chain identified by the Request-URI. [<i>Note:</i> This allows the |

| | |
|--|--|
| | client to determine the options and/or requirements associated with a resource, or the capabilities of a server, without implying a resource action or initiating a resource retrieval.] |
|--|--|

1
2 [Note: For detailed information regarding these methods, see sections 9.2 to 9.8 of RFC
3 2616.

4
5 This property overrides `System.Net.WebRequest.Method`.

6
7]

8 Exceptions

| Exception | Condition |
|---------------------------------|---|
| System.ArgumentException | null, <code>System.String.Empty</code> , or an invalid value was specified for a set operation. |

9

10

1 **HttpRequest.Pipelined Property**

```
2 [ILAsm]  
3 .property bool Pipelined { public hidebysig specialname instance bool  
4 get_Pipelined() public hidebysig specialname instance void  
5 set_Pipelined(bool value) }  
6 [C#]  
7 public bool Pipelined { get; set; }
```

8 **Summary**

9 Gets or sets a `System.Boolean` value indicating whether to pipeline the current request
10 to the Internet resource.

11 **Property Value**

12 `true` if the current request can be pipelined; otherwise, `false`. The default is `true`.

13 **Description**

14 An application uses this property to indicate a preference for pipelined connections. If
15 `System.Net.HttpWebRequest.Pipelined` is `true`, an application makes pipelined
16 connections to servers that support them.

17
18 [*Note:* Pipelined connections are made only when the
19 `System.Net.HttpWebRequest.KeepAlive` property is `true`.]
20
21

22

1 HttpWebRequest.PreAuthenticate Property

```
2 [ILAsm]  
3 .property bool PreAuthenticate { public hidebysig virtual specialname bool  
4 get_PreAuthenticate() public hidebysig virtual specialname void  
5 set_PreAuthenticate(bool value) }  
  
6 [C#]  
7 public override bool PreAuthenticate { get; set; }
```

8 Summary

9 Gets or sets a Boolean value that indicates whether to send HTTP preauthentication
10 header information with current instance without waiting for an authentication challenge
11 from the requested resource.

12 Property Value

13 true to send a HTTP WWW-authenticate header with the current instance without
14 waiting for an authentication challenge from the requested resource; otherwise, false.
15 The default is false.

16 Description

17 When `System.Net.HttpWebRequest.PreAuthenticate` is true and credentials are
18 supplied, the HTTP WWW-authenticate header is sent with the current instance without
19 waiting for an authentication challenge from the requested resource; otherwise the
20 request uses standard authentication procedures.

21
22 [*Note:* Set this property to true to allow clients to improve server efficiency by avoiding
23 extra round trips caused by authentication challenges.

24
25 This property overrides `System.Net.WebRequest.PreAuthenticate`.

26
27]

28

1 HttpWebRequest.ProtocolVersion Property

```
2 [ILAsm]  
3 .property class System.Version ProtocolVersion { public hidebyref  
4 specialname instance class System.Version get_ProtocolVersion() public  
5 hidebyref specialname instance void set_ProtocolVersion(class  
6 System.Version value) }  
7 [C#]  
8 public Version ProtocolVersion { get; set; }
```

9 Summary

10 Gets or sets the version of the HTTP protocol to use for the current request.

11 Property Value

12 A `System.Version` that represents the HTTP version to use for the request. The default
13 is `System.Net.HttpVersion.Version11`.

14 Description

15 The `System.Net.HttpWebRequest` class supports only versions 1.0 and 1.1 of HTTP.
16 Setting `System.Net.HttpWebRequest.ProtocolVersion` to a different version causes a
17 `System.ArgumentException` exception to be thrown.

18
19 [*Note:* To set the `System.Net.HttpWebRequest.ProtocolVersion` property of the
20 current instance, specify one of the members of the use the `System.Net.HttpVersion`
21 class.]
22
23

24 Exceptions

| Exception | Condition |
|---------------------------------------|---|
| <code>System.ArgumentException</code> | The HTTP version is set to a value other than 1.0 or 1.1. |

25

26

1 HttpWebRequest.Proxy Property

```
2 [ILAsm]  
3 .property class System.Net.IWebProxy Proxy { public hidebysig virtual  
4 specialname class System.Net.IWebProxy get_Proxy() public hidebysig  
5 virtual specialname void set_Proxy(class System.Net.IWebProxy value) }  
6 [C#]  
7 public override IWebProxy Proxy { get; set; }
```

8 Summary

9 Gets or sets network proxy information for the current instance.

10 Property Value

11 The `System.Net.WebProxy` instance to use as a proxy for the current instance. The
12 default value is set by calling `System.Net.GlobalProxySelection.Select`.

13 Description

14 The `System.Net.HttpWebRequest.Proxy` property identifies the `System.Net.WebProxy`
15 instance to use to communicate with the destination server.

16
17 *[Note:* To specify that no proxy should be used, set the
18 `System.Net.HttpWebRequest.Proxy` property to the proxy instance returned by the
19 `System.Net.GlobalProxySelection.GetEmptyWebProxy` method.

20
21 This property overrides `System.Net.WebRequest.Proxy`.

22
23]

24 Exceptions

| Exception | Condition |
|--|---|
| System.ArgumentNullException | A set operation was requested and the specified value was <code>null</code> . |
| System.InvalidOperationException | A set operation was requested but data has already been sent to the request stream. |
| System.Security.SecurityException | The caller does not have permission for the requested operation. |

25 26 Permissions

| Permission | Description |
|--|---|
| System.Security.Permissions.WebPermission | Requires unrestricted <code>System.Net.WebPermission</code> . See <code>System.Security.Permissions.PermissionState.Unrestricted</code> . |

1

2

1 HttpWebRequest.Referer Property

```
2 [ILAsm]  
3 .property string Referer { public hidebysig specialname instance string  
4 get_Referer() public hidebysig specialname instance void  
5 set_Referer(string value) }  
6 [C#]  
7 public string Referer { get; set; }
```

8 Summary

9 Gets or sets the value of the HTTP Referer header.

10 Property Value

11 A System.String containing the value of the HTTP Referer header. The default value is
12 null.

13 Description

14 [*Note:* For additional information see section 14.36 of IETF RFC 2616 - HTTP/1.1.]
15
16
17

1 `HttpRequest.RequestUri` Property

```
2 [ILAsm]  
3 .property class System.Uri RequestUri { public hidebysig virtual  
4 specialname class System.Uri get_RequestUri() }  
  
5 [C#]  
6 public override Uri RequestUri { get; }
```

7 **Summary**

8 Gets the `System.Uri` of the resource that receives requests sent by the current
9 instance.

10 **Property Value**

11 The `System.Uri` of the resource that receives requests sent by the current instance.

12 **Description**

13 This property is read-only.

14

15 This property is the `System.Uri` instance passed to the current instance via the
16 `System.Net.WebRequest.Create` method.

17

18 [*Note:* Following a redirection header does not change the
19 `System.Net.HttpWebRequest.RequestUri` property. The URI of the resource that
20 actually responded to the current instance is contained by
21 `System.Net.HttpWebRequest.Address` property of the current instance.

22

23 This property overrides `System.Net.WebRequest.RequestUri`.

24

25]

26

1 `HttpRequest.SendChunked` Property

```
2 [ILAsm]  
3 .property bool SendChunked { public hidebysig specialname instance bool  
4 get_SendChunked() public hidebysig specialname instance void  
5 set_SendChunked(bool value) }  
  
6 [C#]  
7 public bool SendChunked { get; set; }
```

8 **Summary**

9 Gets or sets a value indicating whether to send data in segments.

10 **Property Value**

11 `true` to send data in segments; otherwise, `false`. The default value is `false`.

12 **Description**

13 When `System.Net.HttpWebRequest.SendChunked` is `true`, the request sends data to the
14 destination in segments. The destination server is required to support receiving chunked
15 data.

16
17 [*Note:* Set this property to `true` only if the server specified by the
18 `System.Net.HttpWebRequest.Address` property of the current instance accepts
19 chunked data (i.e. is HTTP/1.1 or greater in compliance). If the server does not accept
20 chunked data, buffer all data to be written and send a HTTP Content-Length header with
21 the buffered data.]
22
23

24 **Exceptions**

| Exception | Condition |
|---|---|
| System.InvalidOperationException | A set operation was requested but data has already been written to the request data stream. |

25

26

1 HttpWebRequest.ServicePoint Property

```
2 [ILAsm]  
3 .property class System.Net.ServicePoint ServicePoint { public hidebysig  
4 specialname instance class System.Net.ServicePoint get_ServicePoint() }  
5 [C#]  
6 public ServicePoint ServicePoint { get; }
```

7 Summary

8 Gets the service point to use for the current instance.

9 Property Value

10 A `System.Net.ServicePoint` that represents the network connection to the destination.
11 The value of this property can be, but is not required to be, `null` until the
12 `System.Net.HttpWebRequest.GetResponse` method is called.

13 Description

14 This property is read-only.

15

1 HttpWebRequest.Timeout Property

```
2 [ILAsm]  
3 .property int32 Timeout { public hidebysig virtual specialname int32  
4 get_Timeout() public hidebysig virtual specialname void set_Timeout(int32  
5 value) }  
6 [C#]  
7 public override int Timeout { get; set; }
```

8 Summary

9 Gets or sets the length of time before the request times out.

10 Property Value

11 A `System.Int32` indicating the number of milliseconds to wait for a response until the
12 request times out, or `System.Threading.Timeout.Infinite` to indicate that the request
13 does not time out.

14 Description

15 `System.Net.HttpWebRequest.Timeout` is the number of milliseconds that a synchronous
16 request made with the `System.Net.HttpWebRequest.GetResponse` method waits for a
17 response. If a resource does not respond within the time-out period, the request throws
18 a `System.Net.WebException` with the `System.Net.WebException.Status` property set
19 to `System.Net.WebExceptionStatus.Timeout`.

20
21 [*Note:* This property overrides `System.Net.WebRequest.Timeout`.]
22
23

24 Exceptions

| Exception | Condition |
|---|--|
| System.ArgumentOutOfRangeException | A value less than zero and not equal to <code>System.Threading.Timeout.Infinite</code> is specified for a set operation. |

25

26

HttpWebRequest.TransferEncoding Property

```
[ILAsm]
.property string TransferEncoding { public hidebysig specialname instance
string get_TransferEncoding() public hidebysig specialname instance void
set_TransferEncoding(string value) }

[C#]
public string TransferEncoding { get; set; }
```

Summary

Gets or sets the value of the HTTP Transfer-encoding header.

Property Value

A `System.String` that contains the value of the HTTP Transfer-encoding header. The default value is `null`.

Description

This property can be set in the current instance only if the `System.Net.HttpWebRequest.SendChunked` property in the current instance is `true`.

[*Note:* Clearing `System.Net.HttpWebRequest.TransferEncoding` by setting it to `null` has no effect on the value of `System.Net.HttpWebRequest.SendChunked`.

Values assigned to the `System.Net.HttpWebRequest.TransferEncoding` property replace any existing contents.

For additional information see section 14.41 of IETF RFC 2616 - HTTP/1.1.

]

Exceptions

| Exception | Condition |
|---|--|
| System.InvalidOperationException | <code>System.Net.HttpWebRequest.TransferEncoding</code> is set when <code>System.Net.HttpWebRequest.SendChunked</code> is <code>false</code> . |
| System.ArgumentException | <code>System.Net.HttpWebRequest.TransferEncoding</code> is set to the value "Chunked". This value is case insensitive. |

1 HttpWebRequest.UserAgent Property

```
2 [ILAsm]  
3 .property string UserAgent { public hidebysig specialname instance string  
4 get_UserAgent() public hidebysig specialname instance void  
5 set_UserAgent(string value) }  
  
6 [C#]  
7 public string UserAgent { get; set; }
```

8 Summary

9 Gets or sets the value of the HTTP User-agent header.

10 Property Value

11 A System.String containing the value of the HTTP User-agent header. The default value
12 is null.

13 Description

14 [*Note:* For additional information see section 14.43 of IETF RFC 2616 - HTTP/1.1.]
15
16
17