

Namespace Sisk

Namespaces

[Sisk.BasicAuth](#)

[Sisk.Cadente](#)

[Sisk.Core](#)

[Sisk.Documenting](#)

[Sisk.Helpers](#)

[Sisk.IniConfiguration](#)

[Sisk.JsonRPC](#)

[Sisk.ModelContextProtocol](#)

[Sisk.Ssl](#)

Namespace Sisk.BasicAuth

Classes

[BasicAuthenticateRequestHandler](#)

Gets a [IRequestHandler](#) that serves as an authenticator for the Basic Authentication scheme, which can validate a user id and password.

[BasicAuthenticationCredentials](#)

Represents basic authentication credentials for an HTTP request.

Class BasicAuthenticateRequestHandler

Namespace:[Sisk.BasicAuth](#)

Assembly:Sisk.BasicAuth.dll

Gets a [IRequestHandler](#) that serves as an authenticator for the Basic Authentication scheme, which can validate a user id and password.

```
public class BasicAuthenticateRequestHandler : IRequestHandler
```

Implements

[IRequestHandler](#)

Constructors

[BasicAuthenticateRequestHandler\(\)](#)

Properties

[ExecutionMode](#)

Gets or sets when this RequestHandler should run.

[Realm](#)

Gets or sets a message to show the client which protection scope it needs to authenticate to.

Methods

[CreateUnauthorizedResponse\(\)](#)

Creates an empty HTTP response with the WWW-Authenticate header and with the realm message defined in this class instance.

[CreateUnauthorizedResponse\(string\)](#)

Creates an empty HTTP response with the WWW-Authenticate header and an custom realm message.

[Execute\(HttpRequest, HttpContext\)](#)

This method is called by the [Router](#) before executing a request when the [Route](#) instantiates an object that implements this interface. If it returns a [HttpResponse](#) object, the route callback is not called and all execution of the route is stopped. If it returns "null", the execution is continued.

[OnValidating\(BasicAuthenticationCredentials, HttpContext\)](#)

Indicates the method that is called to validate a request with client credentials. When returning an [Http Response](#), it will be sent immediately to the client and the rest of the stack will not be executed. If the return is null, it is interpretable that the authentication was successful and the execution should continue.

Constructor BasicAuthenticateRequestHandler

Namespace:[Sisk.BasicAuth](#)

Assembly:Sisk.BasicAuth.dll

BasicAuthenticateRequestHandler()

```
public BasicAuthenticateRequestHandler()
```

Property ExecutionMode

Namespace:[Sisk.BasicAuth](#)

Assembly:Sisk.BasicAuth.dll

ExecutionMode

Gets or sets when this RequestHandler should run.

```
public RequestHandlerExecutionMode ExecutionMode { get; init; }
```

Property Value

[RequestHandlerExecutionMode](#)

Property Realm

Namespace:[Sisk.BasicAuth](#)

Assembly:Sisk.BasicAuth.dll

Realm

Gets or sets a message to show the client which protection scope it needs to authenticate to.

```
public string Realm { get; set; }
```

Property Value

[string](#)

Method CreateUnauthorizedResponse

Namespace:[Sisk.BasicAuth](#)

Assembly:Sisk.BasicAuth.dll

CreateUnauthorizedResponse(string)

Creates an empty HTTP response with the WWW-Authenticate header and an custom realm message.

```
public HttpResponse CreateUnauthorizedResponse(string realm)
```

Parameters

realm string ↗

Defines the realm message to send back to the client.

Returns

[HttpResponse](#)

CreateUnauthorizedResponse()

Creates an empty HTTP response with the WWW-Authenticate header and with the realm message defined in this class instance.

```
public HttpResponse CreateUnauthorizedResponse()
```

Returns

[HttpResponse](#)

Method Execute

Namespace:[Sisk.BasicAuth](#)

Assembly:Sisk.BasicAuth.dll

Execute(HttpRequest, HttpContext)

This method is called by the [Router](#) before executing a request when the [Route](#) instantiates an object that implements this interface. If it returns a [HttpResponse](#) object, the route callback is not called and all execution of the route is stopped. If it returns "null", the execution is continued.

```
public HttpResponse? Execute(HttpContext context)
```

Parameters

request [HttpRequest](#)

context [HttpContext](#)

Returns

[HttpResponse](#)

Method OnValidating

Namespace:[Sisk.BasicAuth](#)

Assembly:Sisk.BasicAuth.dll

OnValidating(BasicAuthenticationCredentials, HttpContext)

Indicates the method that is called to validate a request with client credentials. When returning an [Http Response](#), it will be sent immediately to the client and the rest of the stack will not be executed. If the return is null, it is interpretable that the authentication was successful and the execution should continue.

```
public virtual HttpResponse? OnValidating(BasicAuthenticationCredentials credentials,  
HttpContext context)
```

Parameters

credentials [BasicAuthenticationCredentials](#)

Represents the credentials sent by the client, already decoded and ready for use.

context [HttpContext](#)

Represents the Http context.

Returns

[HttpResponse](#)

Class BasicAuthenticationCredentials

Namespace:[Sisk.BasicAuth](#)

Assembly:Sisk.BasicAuth.dll

Represents basic authentication credentials for an HTTP request.

```
public class BasicAuthenticationCredentials
```

Properties

[Password](#)

Gets the plain password component from this credentials.

[UserId](#)

Gets the user id component from this credentials.

Property Password

Namespace:[Sisk.BasicAuth](#)

Assembly:Sisk.BasicAuth.dll

Password

Gets the plain password component from this credentials.

```
public string Password { get; }
```

Property Value

[string](#)

Property UserId

Namespace:[Sisk.BasicAuth](#)

Assembly:Sisk.BasicAuth.dll

UserId

Gets the user id component from this credentials.

```
public string UserId { get; }
```

Property Value

[string](#)

Namespace Sisk.Cadente

Namespaces

[Sisk.Cadente.CoreEngine](#)

Classes

[HttpEventStreamWriter](#)

Provides methods to write data and events to an underlying stream in the Server-Sent Events (SSE) format.

[HttpHeaderExtensions](#)

Provides extension methods for collections of [HttpHeader](#).

[HttpHost](#)

Represents an HTTP host that listens for incoming TCP connections and handles HTTP requests.

[HttpHostClient](#)

Represents an HTTP host client with its endpoint and certificate information.

[HttpHostContext](#)

Represents an HTTP session that manages the request and response for a single connection.

[HttpHostContext.HttpRequest](#)

Represents an HTTP request.

[HttpHostContext.HttpResponse](#)

Represents an HTTP response.

[HttpHostHandler](#)

Provides a base class for handling HTTP host events.

[HttpHostTimeoutManager](#)

Manages timeouts for HTTP hosts.

[HttpsOptions](#)

Represents the options for configuring an HTTPS server.

Structs

[HttpHeader](#)

Represents an HTTP header, consisting of a name and a value.

Enums

[TransferEncoding](#)

Represents an HTTP transfer-encoding algorithm.

Namespace Sisk.Cadente.CoreEngine

Classes

[CadenteHttpServerEngine](#)

Represents an HTTP server engine based on the Cadente host. This class implements [HttpServerEngine](#) and [IDisposable](#) to manage the lifecycle of the HTTP server.

[CadenteHttpServerEngineContext](#)

Represents the context for an HTTP request and response within the Cadente engine.

[CadenteHttpServerEngineRequest](#)

Represents an HTTP request within the Cadente engine context.

[CadenteHttpServerEngineResponse](#)

Represents an HTTP response within the Cadente engine context.

Class CadenteHttpServerEngine

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:[Sisk.Cadente.CoreEngine.dll](#)

Represents an HTTP server engine based on the Cadente host. This class implements [HttpServerEngine](#) and [IDisposable](#) to manage the lifecycle of the HTTP server.

```
public sealed class CadenteHttpServerEngine : HttpServerEngine, IDisposable
```

Implements

[IDisposable](#)

Constructors

[CadenteHttpServerEngine\(\)](#)

Initializes a new instance of the [CadenteHttpServerEngine](#) class.

[CadenteHttpServerEngine\(Action<HttpHost>\)](#)

Initializes a new instance of the [CadenteHttpServerEngine](#) class with a specified action to set up each HTTP host.

Properties

[IdleConnectionTimeout](#)

Gets or sets the timeout for idle connections.

Methods

[AddListeningPrefix\(string\)](#)

Adds a listening prefix to the server.

[BeginGetContext\(AsyncCallback?, object?\)](#)

Begins an asynchronous operation to get an HTTP context.

[ClearPrefixes\(\)](#)

Clears all listening prefixes from the server.

[Dispose\(\)](#)

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

[EndGetContext\(IAsyncResult\)](#)

Ends an asynchronous operation to get an HTTP context.

[StartServer\(\)](#)

Starts the HTTP server.

[StopServer\(\)](#)

Stops the HTTP server.

Constructor CadenteHttpServerEngine

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

CadenteHttpServerEngine()

Initializes a new instance of the [CadenteHttpServerEngine](#) class.

```
public CadenteHttpServerEngine()
```

CadenteHttpServerEngine(Action<HttpHost>)

Initializes a new instance of the [CadenteHttpServerEngine](#) class with a specified action to set up each HTTP host.

```
public CadenteHttpServerEngine(Action<HttpHost> hostSetupAction)
```

Parameters

hostSetupAction [Action](#)<HttpHost>

An action that is executed for each [HttpHost](#) to configure it.

Property IdleConnectionTimeout

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

IdleConnectionTimeout

Gets or sets the timeout for idle connections.

```
public override TimeSpan IdleConnectionTimeout { get; set; }
```

Property Value

[TimeSpan](#)

The [TimeSpan](#) representing the idle connection timeout.

Method AddListeningPrefix

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

AddListeningPrefix(string)

Adds a listening prefix to the server.

```
public override void AddListeningPrefix(string prefix)
```

Parameters

prefix string ↗

The prefix to add.

Method BeginGetContext

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

BeginGetContext(AsyncCallback?, object?)

Begins an asynchronous operation to get an HTTP context.

```
public override IAsyncResult BeginGetContext(AsyncCallback? callback, object? state)
```

Parameters

callback [AsyncCallback](#)

The [AsyncCallback](#) delegate.

state [object](#)

An object that provides state information for the asynchronous operation.

Returns

[IAsyncResult](#)

An [IAsyncResult](#) that references the asynchronous operation.

Method ClearPrefixes

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

ClearPrefixes()

Clears all listening prefixes from the server.

```
public override void ClearPrefixes()
```

Method Dispose

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

```
public override void Dispose()
```

Method EndGetContext

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

EndGetContext(IAsyncResult)

Ends an asynchronous operation to get an HTTP context.

```
public override HttpServerEngineContext EndGetContext(IAsyncResult asyncResult)
```

Parameters

asyncResult [IAsyncResult](#)

The [IAsyncResult](#) that references the pending asynchronous operation.

Returns

[HttpServerEngineContext](#)

An [HttpServerEngineContext](#) representing the HTTP context.

Method StartServer

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

StartServer()

Starts the HTTP server.

```
public override void StartServer()
```

Method StopServer

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

StopServer()

Stops the HTTP server.

```
public override void StopServer()
```

Class CadenteHttpServerEngineContext

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

Represents the context for an HTTP request and response within the Cadente engine.

```
public sealed class CadenteHttpServerEngineContext : HttpServerEngineContext
```

Constructors

[CadenteHttpServerEngineContext\(CadenteHttpServerEngineRequest, CadenteHttpServerEngineResponse\)](#)

Properties

[ContextAbortedToken](#)

Gets a value that indicates whether the HTTP connection has been aborted.

[ProcessingTask](#)

Gets a task that represents the completion of the processing for this context.

[Request](#)

Gets the HTTP request associated with the context.

[Response](#)

Gets the HTTP response associated with the context.

Methods

[AcceptWebSocketAsync\(string?\)](#)

Accepts a WebSocket connection asynchronously.

Constructor CadenteHttpServerEngineContext

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

CadenteHttpServerEngineContext(CadenteHttpServerEngineRequest, CadenteHttpServerEngineResponse)

```
public CadenteHttpServerEngineContext(CadenteHttpServerEngineRequest request,  
CadenteHttpServerEngineResponse response)
```

Parameters

request [CadenteHttpServerEngineRequest](#)

response [CadenteHttpServerEngineResponse](#)

Property ContextAbortedToken

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

ContextAbortedToken

Gets a value that indicates whether the HTTP connection has been aborted.

```
public override CancellationToken ContextAbortedToken { get; }
```

Property Value

[CancellationToken](#)

A [CancellationToken](#) that can be used to signal that the HTTP connection has been aborted.

Property ProcessingTask

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

ProcessingTask

Gets a task that represents the completion of the processing for this context.

```
public Task ProcessingTask { get; }
```

Property Value

[Task](#)

Property Request

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

Request

Gets the HTTP request associated with the context.

```
public override HttpServerEngineContextRequest Request { get; }
```

Property Value

[HttpServerEngineContextRequest](#)

The [HttpServerEngineContextRequest](#) representing the HTTP request.

Property Response

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

Response

Gets the HTTP response associated with the context.

```
public override HttpServerEngineContextResponse Response { get; }
```

Property Value

[HttpServerEngineContextResponse](#)

The [HttpServerEngineContextResponse](#) representing the HTTP response.

Method AcceptWebSocketAsync

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

AcceptWebSocketAsync(string?)

Accepts a WebSocket connection asynchronously.

```
public override Task<HttpServerEngineWebSocket> AcceptWebSocketAsync(string? subProtocol)
```

Parameters

subProtocol [string](#)

The subprotocol to use for the WebSocket connection.

Returns

[Task](#)<[HttpServerEngineWebSocket](#)>

A [Task](#)<[TResult](#)> representing the asynchronous operation. The result contains the [HttpServerEngineWeb](#) Socket.

Class CadenteHttpServerEngineRequest

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:[Sisk.Cadente.CoreEngine.dll](#)

Represents an HTTP request within the Cadente engine context.

```
public sealed class CadenteHttpServerEngineRequest : HttpServerEngineContextRequest
```

Constructors

[CadenteHttpServerEngineRequest\(HttpContext\)](#)

Initializes a new instance of the [CadenteHttpServerEngineRequest](#) class.

Properties

[ContentEncoding](#)

Gets the content encoding of the request.

[ContentLength64](#)

Gets the content length of the request.

[Headers](#)

Gets the HTTP headers.

[HttpMethod](#)

Gets the HTTP method of the request.

[InputStream](#)

Gets the input stream of the request.

[IsLocal](#)

Gets a value indicating whether the request is from the local machine.

[IsSecureConnection](#)

Gets a value indicating whether the connection is secure.

[LocalEndPoint](#)

Gets the local endpoint of the request.

[ProtocolVersion](#)

Gets the HTTP protocol version.

[QueryString](#)

Gets the query string collection.

[RawUrl](#)

Gets the raw URL of the request.

[RemoteEndPoint](#)

Gets the remote endpoint of the request.

[RequestTraceIdentifier](#)

Gets the request trace identifier.

[Url](#)

Gets the URL of the request.

[UserHostName](#)

Gets the host name of the user.

Constructor CadenteHttpServerEngineRequest

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:[Sisk.Cadente.CoreEngine.dll](#)

CadenteHttpServerEngineRequest(HttpRequest, HttpContext)

Initializes a new instance of the [CadenteHttpServerEngineRequest](#) class.

```
public CadenteHttpServerEngineRequest(HttpContext.HttpRequest request,  
HttpContext context)
```

Parameters

request [HttpContext.HttpRequest](#)

The underlying HTTP request object.

context [HttpContext](#)

The HTTP host context.

Property ContentEncoding

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

ContentEncoding

Gets the content encoding of the request.

```
public override Encoding ContentEncoding { get; }
```

Property Value

[Encoding](#)

The [Encoding](#) representing the content encoding.

Property ContentLength64

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

ContentLength64

Gets the content length of the request.

```
public override long ContentLength64 { get; }
```

Property Value

[long](#)

The content length of the request.

Property Headers

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

Headers

Gets the HTTP headers.

```
public override NameValueCollection Headers { get; }
```

Property Value

[NameValueCollection](#)

The [WebHeaderCollection](#) containing the HTTP headers.

Property HttpMethod

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

HttpMethod

Gets the HTTP method of the request.

```
public override string HttpMethod { get; }
```

Property Value

[string](#)

The HTTP method of the request.

Property InputStream

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

InputStream

Gets the input stream of the request.

```
public override Stream InputStream { get; }
```

Property Value

[Stream](#)

The [Stream](#) representing the input stream.

Property IsLocal

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

IsLocal

Gets a value indicating whether the request is from the local machine.

```
public override bool IsLocal { get; }
```

Property Value

[bool](#)

[true](#) if the request is from the local machine; otherwise, [false](#).

Property IsSecureConnection

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

IsSecureConnection

Gets a value indicating whether the connection is secure.

```
public override bool IsSecureConnection { get; }
```

Property Value

`bool`

`true` if the connection is secure; otherwise, `false`.

Property LocalEndPoint

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

LocalEndPoint

Gets the local endpoint of the request.

```
public override IPPEndPoint LocalEndPoint { get; }
```

Property Value

[IPPEndPoint](#)

The [IPPEndPoint](#) representing the local endpoint.

Property ProtocolVersion

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

ProtocolVersion

Gets the HTTP protocol version.

```
public override Version ProtocolVersion { get; }
```

Property Value

[Version](#)

The [Version](#) representing the HTTP protocol version.

Property QueryString

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

QueryString

Gets the query string collection.

```
public override NameValueCollection QueryString { get; }
```

Property Value

[NameValueCollection](#)

The [NameValueCollection](#) containing the query string parameters.

Property RawUrl

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

RawUrl

Gets the raw URL of the request.

```
public override string? RawUrl { get; }
```

Property Value

[string](#)

The raw URL of the request.

Property RemoteEndPoint

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

RemoteEndPoint

Gets the remote endpoint of the request.

```
public override IPPEndPoint RemoteEndPoint { get; }
```

Property Value

[IPEndPoint](#)

The [IPEndPoint](#) representing the remote endpoint.

Property RequestTraceIdentifier

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

RequestTraceIdentifier

Gets the request trace identifier.

```
public override Guid RequestTraceIdentifier { get; }
```

Property Value

[Guid](#)

The [Guid](#) representing the request trace identifier.

Property Url

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

Url

Gets the URL of the request.

```
public override Uri? Url { get; }
```

Property Value

Uri

The System.Uri representing the URL of the request.

Property UserHostName

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

UserHostName

Gets the host name of the user.

```
public override string UserHostName { get; }
```

Property Value

[string](#)

The host name of the user.

Class CadenteHttpServerEngineResponse

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:[Sisk.Cadente.CoreEngine.dll](#)

Represents an HTTP response within the Cadente engine context.

```
public sealed class CadenteHttpServerEngineResponse : HttpServerEngineContextResponse,  
IDisposable
```

Implements

[IDisposable](#) ↗

Constructors

[CadenteHttpServerEngineResponse\(HttpResponse, HttpContext\)](#)

Initializes a new instance of the [CadenteHttpServerEngineResponse](#) class.

Properties

[ContentLength64](#)

Gets or sets the content length of the response.

[ContentType](#)

Gets or sets the content type of the response.

[Headers](#)

[KeepAlive](#)

Gets or sets a value indicating whether the connection should be kept alive.

[OutputStream](#)

Gets the output stream of the response.

[SendChunked](#)

Gets or sets a value indicating whether chunked transfer encoding is used.

[StatusCode](#)

Gets or sets the HTTP status code.

[StatusDescription](#)

Gets or sets the status description.

Methods

[Abort\(\)](#)

Aborts the response.

[AppendHeader\(string, string\)](#)

Appends a header to the response.

[Close\(\)](#)

Closes the response.

[Dispose\(\)](#)

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Constructor CadenteHttpServerEngineResponse

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:[Sisk.Cadente.CoreEngine.dll](#)

CadenteHttpServerEngineResponse(HttpResponse, HttpContext)

Initializes a new instance of the [CadenteHttpServerEngineResponse](#) class.

```
public CadenteHttpServerEngineResponse(HttpContext.HttpResponse response,  
HttpContext httpHostContext)
```

Parameters

response [HttpContext.HttpResponse](#)

The underlying HTTP response object.

httpHostContext [HttpContext](#)

The HTTP host context.

Property ContentLength64

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

ContentLength64

Gets or sets the content length of the response.

```
public override long ContentLength64 { get; set; }
```

Property Value

[long](#)

The content length of the response.

Property ContentType

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

ContentType

Gets or sets the content type of the response.

```
public override string? ContentType { get; set; }
```

Property Value

[string](#)

The content type of the response.

Property Headers

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

Headers

```
public override IHttpEngineHeaderList Headers { get; }
```

Property Value

IHttpEngineHeaderList

Property KeepAlive

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

KeepAlive

Gets or sets a value indicating whether the connection should be kept alive.

```
public override bool KeepAlive { get; set; }
```

Property Value

[bool](#)

[true](#) if the connection should be kept alive; otherwise, [false](#).

Property OutputStream

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

OutputStream

Gets the output stream of the response.

```
public override Stream OutputStream { get; }
```

Property Value

[Stream](#)

The [Stream](#) representing the output stream.

Property SendChunked

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

SendChunked

Gets or sets a value indicating whether chunked transfer encoding is used.

```
public override bool SendChunked { get; set; }
```

Property Value

[bool](#)

[true](#) if chunked transfer encoding is used; otherwise, [false](#).

Property StatusCode

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

StatusCode

Gets or sets the HTTP status code.

```
public override int StatusCode { get; set; }
```

Property Value

[int](#)

The HTTP status code.

Property StatusDescription

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

StatusDescription

Gets or sets the status description.

```
public override string StatusDescription { get; set; }
```

Property Value

[string](#)

The status description.

Method Abort

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

Abort()

Aborts the response.

```
public override void Abort()
```

Method AppendHeader

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

AppendHeader(string, string)

Appends a header to the response.

```
public override void AppendHeader(string name, string value)
```

Parameters

name [string](#)

The name of the header.

value [string](#)

The value of the header.

Method Close

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

Close()

Closes the response.

```
public override void Close()
```

Method Dispose

Namespace:[Sisk.Cadente.CoreEngine](#)

Assembly:Sisk.Cadente.CoreEngine.dll

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

```
public override void Dispose()
```

Class HttpEventStreamWriter

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Provides methods to write data and events to an underlying stream in the Server-Sent Events (SSE) format.

```
public sealed class HttpEventStreamWriter
```

Methods

[WriteDataAsync\(string\)](#)

Asynchronously writes a data message to the underlying stream.

[WriteEventAsync\(string\)](#)

Asynchronously writes an event message to the underlying stream.

Method WriteDataAsync

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

WriteDataAsync(string)

Asynchronously writes a data message to the underlying stream.

```
public Task WriteDataAsync(string data)
```

Parameters

data [string](#)

The data to write.

Returns

[Task](#)

A task that represents the asynchronous write operation.

Method WriteEventAsync

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

WriteEventAsync(string)

Asynchronously writes an event message to the underlying stream.

```
public Task WriteEventAsync(string eventName)
```

Parameters

eventName [string](#)

The name of the event to write.

Returns

[Task](#)

A task that represents the asynchronous write operation.

Struct `HttpHeader`

Namespace:[Sisk.Cadente](#)

Assembly:[Sisk.Cadente.dll](#)

Represents an HTTP header, consisting of a name and a value.

```
public readonly struct HttpHeader : IEquatable<HttpHeader>
```

Implements

[IEquatable](#)<HttpHeader>

Constructors

[HttpHeader\(in ReadOnlyMemory<byte>, in ReadOnlyMemory<byte>\)](#)

Initializes a new instance of the `HttpHeader` struct with the specified name and value as byte arrays.

[HttpHeader\(string, string\)](#)

Initializes a new instance of the `HttpHeader` struct with the specified name and value as strings.

Properties

[IsEmpty](#)

Gets a value indicating whether this `HttpHeader` has any empty value or name.

[Name](#)

Gets the name of the header as a string.

[Value](#)

Gets the value of the header as a string.

Methods

[Equals\(HttpHeader\)](#)

[Equals\(object?\)](#)

[GetHashCode\(\)](#)

[ToString\(\)](#)

Gets the string representation of this [HttpHeader](#).

Constructor HttpHeaders

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

HttpHeader(in ReadOnlyMemory<byte>, in ReadOnlyMemory<byte>)

Initializes a new instance of the [HttpHeader](#) struct with the specified name and value as byte arrays.

```
public HttpHeaders(in ReadOnlyMemory<byte> nameBytes, in ReadOnlyMemory<byte> valueBytes)
```

Parameters

nameBytes [ReadOnlyMemory<byte>](#)

The byte array representing the name of the header.

valueBytes [ReadOnlyMemory<byte>](#)

The byte array representing the value of the header.

HttpHeader(string, string)

Initializes a new instance of the [HttpHeader](#) struct with the specified name and value as strings.

```
public HttpHeaders(string name, string value)
```

Parameters

name [string](#)

The name of the header.

value [string](#)

The value of the header.

Property IsEmpty

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

IsEmpty

Gets a value indicating whether this [HttpHeader](#) has any empty value or name.

```
public bool IsEmpty { get; }
```

Property Value

[bool](#)

Property Name

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Name

Gets the name of the header as a string.

```
public string Name { get; }
```

Property Value

[string](#)

Property Value

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Value

Gets the value of the header as a string.

```
public string Value { get; }
```

Property Value

[string](#)

Method Equals

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Equals(object?)

```
public override bool Equals(object? obj)
```

Parameters

obj object

Returns

bool

Equals(Header)

```
public bool Equals(Header other)
```

Parameters

other Header

Returns

bool

Method GetHashCode

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

GetHashCode()

```
public override int GetHashCode()
```

Returns

[int](#)

Method ToString

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

ToString()

Gets the string representation of this [HttpHeader](#).

```
public override string ToString()
```

Returns

[string](#)

Class `HttpHeaderExtensions`

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Provides extension methods for collections of [HttpHeader](#).

```
public static class HttpHeaderExtensions
```

Methods

[Set\(List<HttpHeader>, in HttpHeaders\)](#)

Sets an [HttpHeader](#) in the list. If a header with the same name already exists, it is removed before the new header is added. This operation is thread-safe.

Method Set

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Set(List<HttpHeader>, in HttpHeaders)

Sets an [HttpHeader](#) in the list. If a header with the same name already exists, it is removed before the new header is added. This operation is thread-safe.

```
public static void Set(this List<HttpHeader> headers, in HttpHeaders header)
```

Parameters

headers [List](#)<HttpHeader>

The list of [HttpHeader](#) to modify.

header [HttpHeader](#)

The [HttpHeader](#) to set.

Class HttpHost

Namespace:[Sisk.Cadente](#)

Assembly:[Sisk.Cadente.dll](#)

Represents an HTTP host that listens for incoming TCP connections and handles HTTP requests.

```
public sealed class HttpHost : IDisposable
```

Implements

[IDisposable](#)

Constructors

[HttpHost\(int\)](#)

Initializes a new instance of the [HttpHost](#) class using the specified port on the loopback address.

[HttpHost\(IPEndPoint\)](#)

Initializes a new instance of the [HttpHost](#) class using the specified [IPEndPoint](#).

Properties

[Endpoint](#)

Gets the endpoint of the [HttpHost](#).

[Handler](#)

Gets or sets an [HttpHostHandler](#) instance for this [HttpHost](#).

[HttpsOptions](#)

Gets or sets the HTTPS options for secure connections. Setting an [HttpsOptions](#) object in this property, the [HttpHost](#) will use HTTPS instead of HTTP.

[IsDisposed](#)

Gets a value indicating whether this [HttpHost](#) has been disposed.

[QueueSize](#)

Gets or sets the client queue size of all [HttpHost](#) instances. This value indicates how many connections the server can maintain simultaneously before queueing other connection attempts.

[ServerNameHeader](#)

Gets or sets the name of the server in the header name.

[TimeoutManager](#)

Gets the [HttpHostTimeoutManager](#) for this [HttpHost](#).

Methods

[Dispose\(\)](#)

[~HttpHost\(\)](#)

[Start\(\)](#)

Starts the HTTP host and begins listening for incoming connections.

[Stop\(\)](#)

Stops the HTTP host from listening for incoming HTTP requests.

Constructor HttpHost

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

HttpHost(IPEndPoint)

Initializes a new instance of the [HttpHost](#) class using the specified [IPEndPoint](#).

```
public HttpHost(IPEndPoint endpoint)
```

Parameters

endpoint [IPEndPoint](#)

The [IPEndPoint](#) to listen on.

HttpHost(int)

Initializes a new instance of the [HttpHost](#) class using the specified port on the loopback address.

```
public HttpHost(int port)
```

Parameters

port [int](#)

The port number to listen on.

Property Endpoint

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Endpoint

Gets the endpoint of the [HttpHost](#).

```
public IPEndPoint Endpoint { get; }
```

Property Value

[IPEndPoint](#)

Property Handler

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Handler

Gets or sets an [HttpHostHandler](#) instance for this [HttpHost](#).

```
public HttpHostHandler? Handler { get; set; }
```

Property Value

[HttpHostHandler](#)

Property `HttpsOptions`

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

`HttpsOptions`

Gets or sets the HTTPS options for secure connections. Setting an [HttpsOptions](#) object in this property, the [Http Host](#) will use HTTPS instead of HTTP.

```
public HttpsOptions? HttpsOptions { get; set; }
```

Property Value

[HttpsOptions](#)

Property IsDisposed

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

IsDisposed

Gets a value indicating whether this `HttpHost` has been disposed.

```
public bool IsDisposed { get; }
```

Property Value

`bool`[↗](#)

Property QueueSize

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

QueueSize

Gets or sets the client queue size of all [HttpHost](#) instances. This value indicates how many connections the server can maintain simultaneously before queueing other connection attempts.

```
public static int QueueSize { get; set; }
```

Property Value

[int](#)

Property ServerNameHeader

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

ServerNameHeader

Gets or sets the name of the server in the header name.

```
public static string ServerNameHeader { get; set; }
```

Property Value

[string](#)

Property TimeoutManager

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

TimeoutManager

Gets the [HttpHostTimeoutManager](#) for this [HttpHost](#).

```
public HttpHostTimeoutManager TimeoutManager { get; }
```

Property Value

[HttpHostTimeoutManager](#)

Method Dispose

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Dispose()

```
public void Dispose()
```

Method Start

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Start()

Starts the HTTP host and begins listening for incoming connections.

```
public void Start()
```

Method Stop

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Stop()

Stops the HTTP host from listening for incoming HTTP requests.

```
public void Stop()
```

Method ~HttpHost

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

~HttpHost()

```
protected ~HttpHost()
```

Class `HttpHostClient`

Namespace:[Sisk.Cadente](#)

Assembly:[Sisk.Cadente.dll](#)

Represents an HTTP host client with its endpoint and certificate information.

```
public sealed class HttpHostClient
```

Properties

[ClientCertificate](#)

Gets the client certificate, if any.

[ClientEndpoint](#)

Gets the endpoint of the client.

[DisconnectToken](#)

Gets a token that raises whether the HTTP client was disconnected.

[State](#)

Gets or sets an optional state object associated with the client.

Property ClientCertificate

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

ClientCertificate

Gets the client certificate, if any.

```
public X509Certificate? ClientCertificate { get; }
```

Property Value

[X509Certificate](#)

Property ClientEndpoint

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

ClientEndpoint

Gets the endpoint of the client.

```
public IPPEndPoint ClientEndpoint { get; }
```

Property Value

[IPPEndPoint](#)

Property DisconnectToken

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

DisconnectToken

Gets a token that raises whether the HTTP client was disconnected.

```
public CancellationToken DisconnectToken { get; }
```

Property Value

[CancellationToken](#)

Property State

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

State

Gets or sets an optional state object associated with the client.

```
public object? State { get; set; }
```

Property Value

[object](#)

Class HttpHostContext

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Represents an HTTP session that manages the request and response for a single connection.

```
public sealed class HttpHostContext
```

Properties

[Client](#)

Gets the associated [HttpHostClient](#) with this HTTP context.

[Host](#)

Gets the associated [HttpHost](#) which created this HTTP context.

[KeepAlive](#)

Gets or sets a value indicating whether the connection should be kept alive.

[Request](#)

Gets the HTTP request associated with this session.

[Response](#)

Gets the HTTP response associated with this session.

Property Client

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Client

Gets the associated [HttpClient](#) with this HTTP context.

```
public HttpClient Client { get; }
```

Property Value

[HttpClient](#)

Property Host

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Host

Gets the associated [HttpHost](#) which created this HTTP context.

```
public HttpHost Host { get; }
```

Property Value

[HttpHost](#)

Property KeepAlive

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

KeepAlive

Gets or sets a value indicating whether the connection should be kept alive.

```
public bool KeepAlive { get; set; }
```

Property Value

[bool](#)

Property Request

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Request

Gets the HTTP request associated with this session.

```
public HttpContext.HttpRequest Request { get; }
```

Property Value

[HttpContext.HttpRequest](#)

Property Response

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Response

Gets the HTTP response associated with this session.

```
public HttpContext.HttpResponse Response { get; }
```

Property Value

[HttpContext.HttpResponse](#)

Class HttpContext.HttpRequest

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Represents an HTTP request.

```
public sealed class HttpContext.HttpRequest
```

Properties

[ContentLength](#)

Gets the content length of the request.

[Headers](#)

Gets the headers associated with the request.

[Method](#)

Gets the HTTP method (e.g., GET, POST) of the request.

[Path](#)

Gets the path of the requested resource.

Methods

[GetRequestStream\(\)](#)

Gets the stream containing the content of the request.

Property ContentLength

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

ContentLength

Gets the content length of the request.

```
public long ContentLength { get; }
```

Property Value

[long](#)

Property Headers

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Headers

Gets the headers associated with the request.

```
public HttpHeaders Headers { get; }
```

Property Value

[HttpHeaders\[\]](#)

Property Method

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Method

Gets the HTTP method (e.g., GET, POST) of the request.

```
public string Method { get; }
```

Property Value

[string](#)

Property Path

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Path

Gets the path of the requested resource.

```
public string Path { get; }
```

Property Value

[string](#)

Method GetRequestStream

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

GetRequestStream()

Gets the stream containing the content of the request.

```
public Stream GetRequestStream()
```

Returns

[Stream](#)

Class `HttpContext.HttpResponse`

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Represents an HTTP response.

```
public sealed class HttpContext.HttpResponse
```

Properties

[Headers](#)

Gets or sets the list of headers associated with the response.

[ResponseStream](#)

Gets or sets the stream for the response content.

[SendChunked](#)

Gets or sets an boolean indicating if this [HttpContext.HttpResponse](#) should be send in chunks or not.

[StatusCode](#)

Gets or sets the HTTP status code of the response.

[StatusDescription](#)

Gets or sets the status description of the response.

Methods

[GetEventStream\(\)](#)

Asynchronously gets an event stream writer with UTF-8 encoding.

[GetEventStream\(Encoding\)](#)

Asynchronously gets an event stream writer with the specified encoding.

`GetResponseStream()`

Asynchronously gets the content stream for the response.

Property Headers

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Headers

Gets or sets the list of headers associated with the response.

```
public List<HttpHeader> Headers { get; set; }
```

Property Value

[List](#)<HttpHeader>

Property ResponseStream

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

ResponseStream

Gets or sets the stream for the response content.

```
public Stream? ResponseStream { get; set; }
```

Property Value

[Stream](#)

Property SendChunked

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

SendChunked

Gets or sets an boolean indicating if this [HttpHostContext.HttpResponse](#) should be send in chunks or not.

```
public bool SendChunked { get; set; }
```

Property Value

bool 

Property StatusCode

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

StatusCode

Gets or sets the HTTP status code of the response.

```
public int StatusCode { get; set; }
```

Property Value

[int](#)

Property StatusDescription

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

StatusDescription

Gets or sets the status description of the response.

```
public string StatusDescription { get; set; }
```

Property Value

[string](#)

Method GetEventStream

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

GetEventStream()

Asynchronously gets an event stream writer with UTF-8 encoding.

```
public HttpEventStreamWriter GetEventStream()
```

Returns

[HttpEventStreamWriter](#)

A task that represents the asynchronous operation, with a [HttpEventStreamWriter](#) as the result.

GetEventStream(Encoding)

Asynchronously gets an event stream writer with the specified encoding.

```
public HttpEventStreamWriter GetEventStream(Encoding encoding)
```

Parameters

encoding [Encoding](#) ↗

The encoding to use for the event stream.

Returns

[HttpEventStreamWriter](#)

A task that represents the asynchronous operation, with a [HttpEventStreamWriter](#) as the result.

Exceptions

[InvalidOperationException](#) ↗

Thrown when unable to obtain an output stream for the response.

Method GetResponseStream

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

GetResponseStream()

Asynchronously gets the content stream for the response.

```
public Stream GetResponseStream()
```

Returns

[Stream](#)

A task that represents the asynchronous operation, with the response content stream as the result.

Exceptions

[InvalidOperationException](#)

Thrown when unable to obtain an output stream for the response.

Class HttpHostHandler

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Provides a base class for handling HTTP host events.

```
public abstract class HttpHostHandler
```

Constructors

[HttpHostHandler\(\)](#)

Methods

[OnClientConnectedAsync\(HttpHost, HttpHostClient\)](#)

Called when a new client connects to the specified HTTP host.

[OnClientDisconnectedAsync\(HttpHost, HttpHostClient\)](#)

Called when a client disconnects from the specified HTTP host.

[OnContextCreatedAsync\(HttpHost, HttpHostContext\)](#)

Called when a new context is created for the specified HTTP host.

Constructor HttpHostHandler

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

HttpHostHandler()

```
protected HttpHostHandler()
```

Method OnClientConnectedAsync

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

OnClientConnectedAsync(HttpHost, HttpHostClient)

Called when a new client connects to the specified HTTP host.

```
public virtual Task OnClientConnectedAsync(HttpHost host, HttpHostClient client)
```

Parameters

host [HttpHost](#)

The HTTP host that the client connected to.

client [HttpHostClient](#)

The client that connected to the host.

Returns

[Task](#) ↗

A task that represents the asynchronous operation.

Method OnClientDisconnectedAsync

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

OnClientDisconnectedAsync(HttpHost, HttpHostClient)

Called when a client disconnects from the specified HTTP host.

```
public virtual Task OnClientDisconnectedAsync(HttpHost host, HttpHostClient client)
```

Parameters

host [HttpHost](#)

The HTTP host that the client disconnected from.

client [HttpHostClient](#)

The client that disconnected from the host.

Returns

[Task](#) ↗

A task that represents the asynchronous operation.

Method OnContextCreatedAsync

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

OnContextCreatedAsync(HttpHost, HttpHostContext)

Called when a new context is created for the specified HTTP host.

```
public virtual Task OnContextCreatedAsync(HttpHost host, HttpHostContext context)
```

Parameters

host [HttpHost](#)

The HTTP host that created the context.

context [HttpHostContext](#)

The newly created context.

Returns

[Task](#)

A task that represents the asynchronous operation.

Class HttpHostTimeoutManager

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Manages timeouts for HTTP hosts.

```
public sealed class HttpHostTimeoutManager
```

Properties

[ClientReadTimeout](#)

Gets or sets the timeout for client read operations.

[ClientWriteTimeout](#)

Gets or sets the timeout for client write operations.

Property ClientReadTimeout

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

ClientReadTimeout

Gets or sets the timeout for client read operations.

```
public TimeSpan ClientReadTimeout { get; set; }
```

Property Value

[TimeSpan](#) ↗

Property ClientWriteTimeout

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

ClientWriteTimeout

Gets or sets the timeout for client write operations.

```
public TimeSpan ClientWriteTimeout { get; set; }
```

Property Value

[TimeSpan](#) ↗

Class `HttpsOptions`

Namespace:[Sisk.Cadente](#)

Assembly:[Sisk.Cadente.dll](#)

Represents the options for configuring an HTTPS server.

```
public sealed class HttpsOptions
```

Constructors

[`HttpsOptions\(X509Certificate\)`](#)

Initializes a new instance of the [HttpsOptions](#) class.

Properties

[`AllowedProtocols`](#)

Gets or sets the SSL/HTTPS protocols allowed for connections.

[`CheckCertificateRevocation`](#)

Gets or sets a value indicating whether to check for certificate revocation.

[`ClientCertificateRequired`](#)

Gets or sets a value indicating whether client certificates are required for authentication.

[`ServerCertificate`](#)

Gets the SSL certificate used by the proxy server.

Constructor HttpsOptions

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

HttpsOptions(X509Certificate)

Initializes a new instance of the [HttpsOptions](#) class.

```
public HttpsOptions(X509Certificate certificate)
```

Parameters

certificate [X509Certificate](#)

The [X509Certificate](#) used to encrypt data between the client and the server.

Property AllowedProtocols

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

AllowedProtocols

Gets or sets the SSL/HTTPS protocols allowed for connections.

```
public SslProtocols AllowedProtocols { get; set; }
```

Property Value

[SslProtocols](#)

Property CheckCertificateRevocation

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

CheckCertificateRevocation

Gets or sets a value indicating whether to check for certificate revocation.

```
public bool CheckCertificateRevocation { get; set; }
```

Property Value

bool

Property ClientCertificateRequired

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

ClientCertificateRequired

Gets or sets a value indicating whether client certificates are required for authentication.

```
public bool ClientCertificateRequired { get; set; }
```

Property Value

bool

Property ServerCertificate

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

ServerCertificate

Gets the SSL certificate used by the proxy server.

```
public X509Certificate ServerCertificate { get; }
```

Property Value

[X509Certificate](#)

Enum TransferEncoding

Namespace:[Sisk.Cadente](#)

Assembly:Sisk.Cadente.dll

Represents an HTTP transfer-encoding algorithm.

[Flags]

```
public enum TransferEncoding
```

Fields

Chunked = 2

Indicates that the response is sent in a series of chunks.

Deflate = 8

Indicates that the response is compressed using Deflate encoding.

GZip = 4

Indicates that the response is compressed using GZip encoding.

Namespace Sisk.Core

Namespaces

[Sisk.Core.Entity](#)

[Sisk.Core.Helpers](#)

[Sisk.Core.Http](#)

[Sisk.Core.Routing](#)

Namespace Sisk.Core.Entity

Classes

[CrossOriginResourceSharingHeaders](#)

Provides a class to provide Cross Origin response headers for when communicating with a browser.

[HttpHeaderCollection](#)

Represents an collection of HTTP headers with their name and values.

[MultipartFormCollection](#)

Represents an class which hosts an multipart form data contents.

[MultipartObject](#)

Represents an multipart/form-data object.

[StringKeyStoreCollection](#)

Represents a collection of string keys associated with multiple string values.

[StringValueCollection](#)

Represents an collection of [StringValue](#).

[TypedValueDictionary](#)

Represents the base class for storing and retrieving data by their type.

Structs

[StringValue](#)

Represents an option/monad item that wraps an string value and allows conversion to most common types.

Enums

[MultipartObjectCommonFormat](#)

Represents an image format for Multipart objects.

Class CrossOriginResourceSharingHeaders

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Provides a class to provide Cross Origin response headers for when communicating with a browser.

```
public sealed class CrossOriginResourceSharingHeaders
```

Constructors

[CrossOriginResourceSharingHeaders\(\)](#)

Creates an empty [CrossOriginResourceSharingHeaders](#) instance with no predefined CORS headers.

[CrossOriginResourceSharingHeaders\(string?, string\[\]?, string\[\]?, string\[\]?, string\[\]?, TimeSpan?, bool\)](#)

Initializes a new instance of the [CrossOriginResourceSharingHeaders](#) class with the specified CORS headers.

Fields

[AutoAllowOrigin](#)

When applied to the [AllowOrigin](#) property, the HTTP server automatically applies the incoming request Origin header value to the Access-Control-Allow-Origin header.

[AutoFromRequestHeaders](#)

When applied to the [AllowHeaders](#) property, the HTTP server automatically applies the incoming request headers to the Access-Control-Allow-Origin header.

[AutoFromRequestMethod](#)

When applied to the [AllowMethods](#) property, the HTTP server automatically applies the incoming request method to the Access-Control-Allow-Origin header.

Properties

AllowCredentials

Gets or sets the Access-Control-Allow-Credentials header indicates whether or not the response to the request can be exposed when the credentials flag is true. When used as part of a response to a preflight request, this indicates whether or not the actual request can be made using credentials.

AllowHeaders

Gets or sets the Access-Control-Allow-Headers header is used in response to a preflight request to indicate which HTTP headers can be used when making the actual request.

AllowMethods

Gets or sets the Access-Control-Allow-Methods header specifies the method or methods allowed when accessing the resource.

AllowOrigin

From MDN: Access-Control-Allow-Origin specifies either a single origin which tells browsers to allow that origin to access the resource; or else — for requests without credentials — the "*" wildcard tells browsers to allow any origin to access the resource.

AllowOrigins

Gets or sets domains which will define the source header according to one of the domains present below.

Empty

Gets an instance of an empty CrossOriginResourceSharingHeaders.

ExposeHeaders

Gets or sets the Access-Control-Expose-Headers header adds the specified headers to the allowlist that JavaScript in browsers is allowed to access.

MaxAge

Gets or sets the Access-Control-Max-Age header indicates how long the results of a preflight request can be cached.

Methods

CreatePublicContext()

Create an instance of Cross-Origin Resource Sharing that allows any origin, any method and any header in the request.

Constructor CrossOriginResourceSharingHeaders

Namespace:[Sisk.Core.Entity](#)

Assembly:[Sisk.Core.dll](#)

CrossOriginResourceSharingHeaders()

Creates an empty [CrossOriginResourceSharingHeaders](#) instance with no predefined CORS headers.

```
public CrossOriginResourceSharingHeaders()
```

CrossOriginResourceSharingHeaders(string?, string[]?, string[]?, string[]?, string[]?, TimeSpan?, bool)

Initializes a new instance of the [CrossOriginResourceSharingHeaders](#) class with the specified CORS headers.

```
public CrossOriginResourceSharingHeaders(string? allowOrigin = null, string[]? allowOrigins = null,  
string[]? allowMethods = null, string[]? allowHeaders = null, string[]? exposeHeaders = null,  
TimeSpan? maxAge = null, bool allowCredentials = false)
```

Parameters

allowOrigin [string](#)

The value of the Access-Control-Allow-Origin header.

allowOrigins [string\[\]](#)

The values of the Access-Control-Allow-Origin header.

allowMethods [string\[\]](#)

The values of the Access-Control-Allow-Methods header.

allowHeaders [string\[\]](#)

The values of the Access-Control-Allow-Headers header.

exposeHeaders [string](#)[]

The values of the Access-Control-Expose-Headers header.

maxAge [TimeSpan](#)?

The value of the Access-Control-Max-Age header.

allowCredentials [bool](#)

The value of the Access-Control-Allow-Credentials header.

Field AutoAllowOrigin

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

When applied to the [AllowOrigin](#) property, the HTTP server automatically applies the incoming request Origin header value to the Access-Control-Allow-Origin header.

```
public const string AutoAllowOrigin = "<SISK_AUTO_ALLOW_ORIGIN_NAME>"
```

Returns

[string](#)

When applied to the property, the HTTP server automatically applies the incoming request Origin header value to the Access-Control-Allow-Origin header.

Field AutoFromRequestHeaders

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

When applied to the [AllowHeaders](#) property, the HTTP server automatically applies the incoming request headers to the Access-Control-Allow-Origin header.

```
public const string AutoFromRequestHeaders = "<SISK_AUTO_FROM_REQUEST_HEADERS>"
```

Returns

[string](#)

When applied to the property, the HTTP server automatically applies the incoming request headers to the Access-Control-Allow-Origin header.

Field AutoFromRequestMethod

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

When applied to the [AllowMethods](#) property, the HTTP server automatically applies the incoming request method to the Access-Control-Allow-Origin header.

```
public const string AutoFromRequestMethod = "<SISK_AUTO_FROM_REQUEST_METHOD>"
```

Returns

[string](#)

When applied to the property, the HTTP server automatically applies the incoming request method to the Access-Control-Allow-Origin header.

Property AllowCredentials

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AllowCredentials

Gets or sets the Access-Control-Allow-Credentials header indicates whether or not the response to the request can be exposed when the credentials flag is true. When used as part of a response to a preflight request, this indicates whether or not the actual request can be made using credentials.

```
public bool? AllowCredentials { get; set; }
```

Property Value

[bool](#)?

Property AllowHeaders

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AllowHeaders

Gets or sets the Access-Control-Allow-Headers header is used in response to a preflight request to indicate which HTTP headers can be used when making the actual request.

```
public string[] AllowHeaders { get; set; }
```

Property Value

[string](#) 

Property AllowMethods

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AllowMethods

Gets or sets the Access-Control-Allow-Methods header specifies the method or methods allowed when accessing the resource.

```
public string[] AllowMethods { get; set; }
```

Property Value

[string](#) ↗

Property AllowOrigin

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AllowOrigin

From MDN: Access-Control-Allow-Origin specifies either a single origin which tells browsers to allow that origin to access the resource; or else — for requests without credentials — the "*" wildcard tells browsers to allow any origin to access the resource.

```
public string? AllowOrigin { get; set; }
```

Property Value

[string](#)

Property AllowOrigins

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AllowOrigins

Gets or sets domains which will define the source header according to one of the domains present below.

```
public string[] AllowOrigins { get; set; }
```

Property Value

[string](#) 

Remarks

This property makes the server compare the origin of the request and associate the domain that corresponds to it.

Property Empty

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Empty

Gets an instance of an empty CrossOriginResourceSharingHeaders.

```
public static CrossOriginResourceSharingHeaders Empty { get; }
```

Property Value

[CrossOriginResourceSharingHeaders](#)

Property ExposeHeaders

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ExposeHeaders

Gets or sets the Access-Control-Expose-Headers header adds the specified headers to the allowlist that JavaScript in browsers is allowed to access.

```
public string[] ExposeHeaders { get; set; }
```

Property Value

[string](#) ↗

Property MaxAge

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

MaxAge

Gets or sets the Access-Control-Max-Age header indicates how long the results of a preflight request can be cached.

```
public TimeSpan MaxAge { get; set; }
```

Property Value

[TimeSpan](#)

Method CreatePublicContext

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

CreatePublicContext()

Create an instance of Cross-Origin Resource Sharing that allows any origin, any method and any header in the request.

```
public static CrossOriginResourceSharingHeaders CreatePublicContext()
```

Returns

[CrossOriginResourceSharingHeaders](#)

Class HttpHeadersCollection

Namespace:[Sisk.Core.Entity](#)

Assembly:[Sisk.Core.dll](#)

Represents an collection of HTTP headers with their name and values.

```
public sealed class HttpHeadersCollection : StringKeyStoreCollection, IDictionary<string, string[]>, ICollection<KeyValuePair<string, string[]>>, IEnumerable<KeyValuePair<string, string[]>>, IEnumerable
```

Implements

[IDictionary<string, string\[\]>](#), [ICollection<KeyValuePair<string, string\[\]>>](#),
[IEnumerable<KeyValuePair<string, string\[\]>>](#), [IEnumerable](#)

Constructors

[HttpHeadersCollection\(\)](#)

Create an new instance of the [HttpHeadersCollection](#) class.

[HttpHeadersCollection\(IDictionary<string, string\[\]>\)](#)

Create an new instance of the [HttpHeadersCollection](#) class with values from another collection.

[HttpHeadersCollection\(IDictionary<string, string?>\)](#)

Create an new instance of the [HttpHeadersCollection](#) class with values from another collection.

[HttpHeadersCollection\(WebHeaderCollection\)](#)

Create an new instance of the [HttpHeadersCollection](#) class with values from another collection.

Properties

[Accept](#)

Gets the value of the HTTP Accept header.

Specifies the media types that are acceptable for the response, allowing the client to indicate its preferences.

[AcceptCH](#)

Gets or sets the value of the HTTP Accept-CH header.

Specifies the client hints that the server supports, allowing clients to provide additional information about their capabilities.

[AcceptCharset](#)

Gets the value of the HTTP Accept-Charset header.

Indicates the character sets that are acceptable for the response, allowing the client to specify its preferred encoding.

[AcceptEncoding](#)

Gets the value of the HTTP Accept-Encoding header.

Specifies the content encodings that are acceptable for the response, allowing the client to indicate its preferences for compression.

[AcceptLanguage](#)

Gets the value of the HTTP Accept-Language header.

Indicates the natural languages that are preferred for the response, allowing the client to specify its language preferences.

[AcceptPatch](#)

Gets the value of the HTTP Accept-Patch header.

Indicates the patch document formats that are acceptable for the response, allowing the client to specify its preferences for patching resources.

[AcceptPost](#)

Gets or sets the value of the HTTP Accept-Post header.

Specifies the accepted formats for posting data to the server.

[AcceptRanges](#)

Gets or sets the value of the HTTP Accept-Ranges header.

Indicates that the server supports range requests for the resource, allowing clients to request specific byte ranges.

[AccessControlAllowCredentials](#)

Gets or sets the value of the HTTP Access-Control-Allow-Credentials header.

Indicates whether the response to the request can expose credentials, allowing cross-origin requests to include credentials.

[AccessControlAllowHeaders](#)

Gets or sets the value of the HTTP Access-Control-Allow-Headers header.

Specifies which headers can be used when making the actual request in a cross-origin resource sharing (CORS) context.

[AccessControlAllowMethods](#)

Gets or sets the value of the HTTP Access-Control-Allow-Methods header.

Specifies the methods that are allowed when accessing the resource in a CORS context.

[AccessControlAllowOrigin](#)

Gets or sets the value of the HTTP Access-Control-Allow-Origin header.

Specifies which origins are allowed to access the resource in a CORS context, helping to control cross-origin requests.

[AccessControlExposeHeaders](#)

Gets or sets the value of the HTTP Access-Control-Expose-Headers header.

Indicates which headers can be exposed as part of the response to a cross-origin request.

[AccessControlMaxAge](#)

Gets or sets the value of the HTTP Access-Control-Max-Age header.

Specifies how long the results of a preflight request can be cached, reducing the number of preflight requests made.

[Age](#)

Gets or sets the value of the HTTP Age header.

Indicates the age of the object in a cache, helping clients understand how fresh the cached response is.

[Allow](#)

Gets or sets the value of the HTTP Allow header.

Lists the HTTP methods that are supported by the resource, informing clients about the available actions.

[Authorization](#)

Gets or sets the value of the HTTP Authorization header.

Contains credentials for authenticating the client with the server, often used for basic or bearer token authentication.

[CacheControl](#)

Gets or sets the value of the HTTP Cache-Control header.

Directs caching mechanisms on how to cache the response, including directives for expiration and revalidation.

[Connection](#)

Gets the value of the HTTP Connection header. To set this header in a HTTP response, use the [KeepAlive](#) property.

[ContentDisposition](#)

Gets or sets the value of the HTTP Content-Disposition header.

Indicates if the content should be displayed inline in the browser or treated as an attachment to be downloaded.

[ContentEncoding](#)

Gets or sets the value of the HTTP Content-Encoding header.

Specifies the encoding transformations that have been applied to the response body, such as gzip or deflate. This header should not be interpreted as the content text charset.

[ContentLanguage](#)

Gets or sets the value of the HTTP Content-Language header.

Indicates the natural language(s) of the intended audience for the response, helping clients understand the content's language.

[ContentLocation](#)

Gets or sets the value of the HTTP Content-Location header.

Indicates an alternate location for the returned data, often used for redirecting clients to a different resource.

[ContentMD5](#)

Gets or sets the value of the HTTP Content-MD5 header.

Contains the MD5 hash of the response body in an base-64 format, allowing clients to verify the integrity of the received data.

[ContentRange](#)

Gets or sets the value of the HTTP Content-Range header.

Indicates the size of the response body in bytes, allowing the client to know how much data to expect.

[ContentSecurityPolicy](#)

Gets or sets the value of the HTTP Content-Security-Policy header.

Defines security policies for the content, helping to prevent cross-site scripting (XSS) and other code injection attacks.

[ContentType](#)

Gets or sets the value of the HTTP Content-Type header.

Indicates the media type of the resource, allowing the client to understand how to process the response body.

[Cookie](#)

Gets the value of the HTTP Cookie header.

Contains stored HTTP cookies previously sent by the server, allowing the server to identify the client on subsequent requests.

[Dnt](#)

Gets the value of the HTTP DNT header.

The HTTP DNT (Do Not Track) request header indicates the user's tracking preference.

[ETag](#)

Gets or sets the value of the HTTP ETag header.

Provides a unique identifier for a specific version of a resource, allowing clients to cache and validate resources efficiently.

[Expect](#)

Gets the value of the HTTP Expect header.

Indicates that the client expects certain behaviors from the server, such as support for specific features or conditions.

[Expires](#)

Gets or sets the value of the HTTP Expires header.

Indicates the date and time after which the response is considered stale, helping clients manage caching.

[Host](#)

Gets the value of the HTTP Host header.

Specifies the domain name of the server and the TCP port number on which the server is listening, allowing for virtual hosting.

[IfMatch](#)

Gets the value of the HTTP If-Match header.

Used to make a conditional request, allowing the client to specify that the request should only be processed if the resource matches the given ETag.

IfModifiedSince

Gets the value of the HTTP If-Modified-Since header.

Used to make a conditional request, allowing the client to specify that the resource should only be returned if it has been modified since the given date.

IfNoneMatch

Gets the value of the HTTP If-None-Match header.

Used to make a conditional request, allowing the client to specify that the resource should only be returned if it has been modified since the given date.

IfRange

Gets the value of the HTTP If-Range header.

Used to make a conditional range request, allowing the client to specify that the range should only be returned if the resource has not changed.

IfUnmodifiedSince

Gets the value of the HTTP If-Unmodified-Since header.

Used to make a conditional request, allowing the client to specify that the resource should only be returned if it has not been modified since the given date.

Location

Gets or sets the value of the HTTP Location header.

Indicates an alternate location for the returned data, often used for redirecting clients to a different resource.

MaxForwards

Gets or sets the value of the HTTP Max-Forwards header.

Used in OPTIONS requests to limit the number of times the request can be forwarded by proxies.

Origin

Gets the value of the HTTP Origin header.

Indicates the origin of the request, helping servers implement CORS and manage cross-origin requests.

Pragma

Gets or sets the value of the HTTP Pragma header.

Used to include implementation-specific directives that might apply to any recipient along the request/response chain.

ProxyAuthenticate

Gets or sets the value of the HTTP Proxy-Authenticate header.

Used by a proxy server to request authentication from the client, indicating the authentication method required.

ProxyAuthorization

Gets or sets the value of the HTTP Proxy-Authorization header.

Contains credentials for authenticating the client with a proxy server, allowing access to the requested resource.

Range

Gets the value of the HTTP Range header.

Used to request a specific range of bytes from a resource, allowing clients to download large files in parts.

Referer

Gets the value of the HTTP Referer header.

Indicates the URL of the resource from which the request originated, helping servers understand the source of traffic.

RetryAfter

Gets or sets the value of the HTTP Retry-After header.

Indicates how long the client should wait before making a follow-up request, often used in rate limiting scenarios.

SecGPC

Gets the value of the HTTP Sec-GPC header.

The HTTP Sec-GPC request header is part of the Global Privacy Control (GPC) mechanism to indicate whether the user consents to a website or service selling or sharing their personal information with third parties.

SetCookie

Gets or sets the value of the HTTP Set-Cookie header.

Used to send cookies from the server to the client, allowing the server to store state information on the client.

TE

Gets the value of the HTTP TE header.

Indicates the transfer encodings that are acceptable for the response, allowing for content negotiation.

UserAgent

Gets the value of the HTTP User-Agent header.

Contains information about the user agent (browser or application) making the request, including its version and platform.

Vary

Gets or sets the value of the HTTP Vary header.

Indicates that the response varies based on the value of the specified request headers, allowing for content negotiation.

Via

Gets or sets the value of the HTTP Via header.

Used to track message forwards and proxies, indicating the intermediate protocols and recipients involved in the request/response chain.

WWWAuthenticate

Gets or sets the value of the HTTP WWW-Authenticate header.

Used in response to a request for authentication, indicating the authentication method that should be used to access the resource.

XForwardedFor

Gets the value of the HTTP X-Forwarded-For header.

Used to identify the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer.

XForwardedHost

Gets the value of the HTTP X-Forwarded-Host header

Used to identify the original host requested by the client in the Host HTTP request header, often used in proxy setups.

XFrameOptions

Gets or sets the value of the HTTP X-Frame-Options header.

Used to control whether a browser should be allowed to render a page in a iframe, frame, embed or object tag, helping to prevent clickjacking attacks.

XUACOMPATIBLE

Gets or sets the value of the HTTP X-UA-Compatible header.

Used to specify the document mode that Internet Explorer should use to render the page, helping to ensure compatibility with older versions.

Constructor HttpHeadersCollection

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

HttpHeadersCollection()

Create an new instance of the [HttpHeadersCollection](#) class.

```
public HttpHeadersCollection()
```

HttpHeadersCollection(IDictionary<string, string[]>)

Create an new instance of the [HttpHeadersCollection](#) class with values from another collection.

```
public HttpHeadersCollection(IDictionary<string, string[]> items)
```

Parameters

items [IDictionary<string\[\], string\[\]>](#)

The inner collection to add to this collection.

HttpHeadersCollection(IDictionary<string, string?>)

Create an new instance of the [HttpHeadersCollection](#) class with values from another collection.

```
public HttpHeadersCollection(IDictionary<string, string?> items)
```

Parameters

items [IDictionary<string\[\], string?>](#)

The inner collection to add to this collection.

HttpHeaderCollection(WebHeaderCollection)

Create an new instance of the [HttpHeaderCollection](#) class with values from another collection.

```
public HttpHeaderCollection(WebHeaderCollection items)
```

Parameters

items [WebHeaderCollection](#)

The inner collection to add to this collection.

Property Accept

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Accept

Gets the value of the HTTP Accept header.

Specifies the media types that are acceptable for the response, allowing the client to indicate its preferences.

```
public string? Accept { get; }
```

Property Value

[string](#) ↗

Property AcceptCH

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AcceptCH

Gets or sets the value of the HTTP Accept-CH header.

Specifies the client hints that the server supports, allowing clients to provide additional information about their capabilities.

```
public string? AcceptCH { get; set; }
```

Property Value

[string](#)

Property AcceptCharset

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AcceptCharset

Gets the value of the HTTP Accept-Charset header.

Indicates the character sets that are acceptable for the response, allowing the client to specify its preferred encoding.

```
public string? AcceptCharset { get; }
```

Property Value

[string](#) ↗

Property AcceptEncoding

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AcceptEncoding

Gets the value of the HTTP Accept-Encoding header.

Specifies the content encodings that are acceptable for the response, allowing the client to indicate its preferences for compression.

```
public string? AcceptEncoding { get; }
```

Property Value

[string](#) ↗

Property AcceptLanguage

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AcceptLanguage

Gets the value of the HTTP Accept-Language header.

Indicates the natural languages that are preferred for the response, allowing the client to specify its language preferences.

```
public string? AcceptLanguage { get; }
```

Property Value

[string](#) ↗

Property AcceptPatch

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AcceptPatch

Gets the value of the HTTP Accept-Patch header.

Indicates the patch document formats that are acceptable for the response, allowing the client to specify its preferences for patching resources.

```
public string? AcceptPatch { get; }
```

Property Value

[string](#) ↗

Property AcceptPost

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AcceptPost

Gets or sets the value of the HTTP Accept-Post header.

Specifies the accepted formats for posting data to the server.

```
public string? AcceptPost { get; set; }
```

Property Value

[string](#) ↗

Property AcceptRanges

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AcceptRanges

Gets or sets the value of the HTTP Accept-Ranges header.

Indicates that the server supports range requests for the resource, allowing clients to request specific byte ranges.

```
public string? AcceptRanges { get; set; }
```

Property Value

[string](#)

Property AccessControlAllowCredentials

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AccessControlAllowCredentials

Gets or sets the value of the HTTP Access-Control-Allow-Credentials header.

Indicates whether the response to the request can expose credentials, allowing cross-origin requests to include credentials.

```
public string? AccessControlAllowCredentials { get; set; }
```

Property Value

[string](#) ↗

Remarks

Note: this header can be overwritten by the current [CrossOriginResourceSharingHeaders](#) configuration.

Property AccessControlAllowHeaders

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AccessControlAllowHeaders

Gets or sets the value of the HTTP Access-Control-Allow-Headers header.

Specifies which headers can be used when making the actual request in a cross-origin resource sharing (CORS) context.

```
public string? AccessControlAllowHeaders { get; set; }
```

Property Value

[string](#) ↗

Remarks

Note: this header can be overwritten by the current [CrossOriginResourceSharingHeaders](#) configuration.

Property AccessControlAllowMethods

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AccessControlAllowMethods

Gets or sets the value of the HTTP Access-Control-Allow-Methods header.

Specifies the methods that are allowed when accessing the resource in a CORS context.

```
public string? AccessControlAllowMethods { get; set; }
```

Property Value

[string](#)

Property AccessControlAllowOrigin

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AccessControlAllowOrigin

Gets or sets the value of the HTTP Access-Control-Allow-Origin header.

Specifies which origins are allowed to access the resource in a CORS context, helping to control cross-origin requests.

```
public string? AccessControlAllowOrigin { get; set; }
```

Property Value

[string](#) ↗

Remarks

Note: this header can be overwritten by the current [CrossOriginResourceSharingHeaders](#) configuration.

Property AccessControlExposeHeaders

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AccessControlExposeHeaders

Gets or sets the value of the HTTP Access-Control-Expose-Headers header.

Indicates which headers can be exposed as part of the response to a cross-origin request.

```
public string? AccessControlExposeHeaders { get; set; }
```

Property Value

[string](#)

Remarks

Note: this header can be overwritten by the current [CrossOriginResourceSharingHeaders](#) configuration.

Property AccessControlMaxAge

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AccessControlMaxAge

Gets or sets the value of the HTTP Access-Control-Max-Age header.

Specifies how long the results of a preflight request can be cached, reducing the number of preflight requests made.

```
public string? AccessControlMaxAge { get; set; }
```

Property Value

[string](#) ↗

Remarks

Note: this header can be overwritten by the current [CrossOriginResourceSharingHeaders](#) configuration.

Property Age

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Age

Gets or sets the value of the HTTP Age header.

Indicates the age of the object in a cache, helping clients understand how fresh the cached response is.

```
public string? Age { get; set; }
```

Property Value

[string](#) ↗

Property Allow

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Allow

Gets or sets the value of the HTTP Allow header.

Lists the HTTP methods that are supported by the resource, informing clients about the available actions.

```
public string? Allow { get; set; }
```

Property Value

[string](#) ↗

Property Authorization

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Authorization

Gets or sets the value of the HTTP Authorization header.

Contains credentials for authenticating the client with the server, often used for basic or bearer token authentication.

```
public string? Authorization { get; set; }
```

Property Value

[string](#) ↗

Property CacheControl

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

CacheControl

Gets or sets the value of the HTTP Cache-Control header.

Directs caching mechanisms on how to cache the response, including directives for expiration and revalidation.

```
public string? CacheControl { get; set; }
```

Property Value

[string](#) ↗

Property Connection

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Connection

Gets the value of the HTTP Connection header. To set this header in a HTTP response, use the [KeepAlive](#) property.

```
public string? Connection { get; }
```

Property Value

[string](#)

Property ContentDisposition

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ContentDisposition

Gets or sets the value of the HTTP Content-Disposition header.

Indicates if the content should be displayed inline in the browser or treated as an attachment to be downloaded.

```
public string? ContentDisposition { get; set; }
```

Property Value

[string](#) ↗

Property ContentEncoding

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ContentEncoding

Gets or sets the value of the HTTP Content-Encoding header.

Specifies the encoding transformations that have been applied to the response body, such as gzip or deflate.
This header should not be interpreted as the content text charset.

```
public string? ContentEncoding { get; set; }
```

Property Value

[string](#)

Property ContentLanguage

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ContentLanguage

Gets or sets the value of the HTTP Content-Language header.

Indicates the natural language(s) of the intended audience for the response, helping clients understand the content's language.

```
public string? ContentLanguage { get; set; }
```

Property Value

[string](#)

Property ContentLocation

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ContentLocation

Gets or sets the value of the HTTP Content-Location header.

Indicates an alternate location for the returned data, often used for redirecting clients to a different resource.

```
public string? ContentLocation { get; set; }
```

Property Value

[string](#) ↗

Property ContentMD5

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ContentMD5

Gets or sets the value of the HTTP Content-MD5 header.

Contains the MD5 hash of the response body in an base-64 format, allowing clients to verify the integrity of the received data.

```
public string? ContentMD5 { get; set; }
```

Property Value

[string](#) ↗

Property ContentRange

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ContentRange

Gets or sets the value of the HTTP Content-Range header.

Indicates the size of the response body in bytes, allowing the client to know how much data to expect.

```
public string? ContentRange { get; set; }
```

Property Value

[string](#) ↗

Property ContentSecurityPolicy

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ContentSecurityPolicy

Gets or sets the value of the HTTP Content-Security-Policy header.

Defines security policies for the content, helping to prevent cross-site scripting (XSS) and other code injection attacks.

```
public string? ContentSecurityPolicy { get; set; }
```

Property Value

[string](#)

Property ContentType

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ContentType

Gets or sets the value of the HTTP Content-Type header.

Indicates the media type of the resource, allowing the client to understand how to process the response body.

```
public string? ContentType { get; set; }
```

Property Value

[string](#)

Remarks

Note: setting the value of this header, the value present in the response's [HttpContent](#) will be overwritten.

Property Cookie

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Cookie

Gets the value of the HTTP Cookie header.

Contains stored HTTP cookies previously sent by the server, allowing the server to identify the client on subsequent requests.

```
public string? Cookie { get; }
```

Property Value

[string](#) ↗

Remarks

Tip: use [Cookies](#) property to getting cookies values from requests and [SetCookie\(string, string\)](#) on [HttpResponse](#) to set cookies.

Property Dnt

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Dnt

Gets the value of the HTTP DNT header.

The HTTP DNT (Do Not Track) request header indicates the user's tracking preference.

```
public string? Dnt { get; set; }
```

Property Value

[string](#)

Property ETag

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ETag

Gets or sets the value of the HTTP ETag header.

Provides a unique identifier for a specific version of a resource, allowing clients to cache and validate resources efficiently.

```
public string? ETag { get; set; }
```

Property Value

[string](#)

Property Expect

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Expect

Gets the value of the HTTP Expect header.

Indicates that the client expects certain behaviors from the server, such as support for specific features or conditions.

```
public string? Expect { get; }
```

Property Value

[string](#) ↗

Property Expires

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Expires

Gets or sets the value of the HTTP Expires header.

Indicates the date and time after which the response is considered stale, helping clients manage caching.

```
public string? Expires { get; set; }
```

Property Value

[string](#) ↗

Property Host

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Host

Gets the value of the HTTP Host header.

Specifies the domain name of the server and the TCP port number on which the server is listening, allowing for virtual hosting.

```
public string? Host { get; }
```

Property Value

[string](#) ↗

Property IfMatch

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

IfMatch

Gets the value of the HTTP If-Match header.

Used to make a conditional request, allowing the client to specify that the request should only be processed if the resource matches the given ETag.

```
public string? IfMatch { get; }
```

Property Value

[string](#) ↗

Property IfModifiedSince

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

IfModifiedSince

Gets the value of the HTTP If-Modified-Since header.

Used to make a conditional request, allowing the client to specify that the resource should only be returned if it has been modified since the given date.

```
public string? IfModifiedSince { get; }
```

Property Value

[string](#) ↗

Property IfNoneMatch

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

IfNoneMatch

Gets the value of the HTTP If-None-Match header.

Used to make a conditional request, allowing the client to specify that the resource should only be returned if it has been modified since the given date.

```
public string? IfNoneMatch { get; }
```

Property Value

[string](#) ↗

Property IfRange

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

IfRange

Gets the value of the HTTP If-Range header.

Used to make a conditional range request, allowing the client to specify that the range should only be returned if the resource has not changed.

```
public string? IfRange { get; }
```

Property Value

[string](#)

Property IfUnmodifiedSince

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

IfUnmodifiedSince

Gets the value of the HTTP If-Unmodified-Since header.

Used to make a conditional request, allowing the client to specify that the resource should only be returned if it has not been modified since the given date.

```
public string? IfUnmodifiedSince { get; }
```

Property Value

[string](#) ↗

Property Location

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Location

Gets or sets the value of the HTTP Location header.

Indicates an alternate location for the returned data, often used for redirecting clients to a different resource.

```
public string? Location { get; set; }
```

Property Value

[string](#) ↗

Property MaxForwards

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

MaxForwards

Gets or sets the value of the HTTP Max-Forwards header.

Used in OPTIONS requests to limit the number of times the request can be forwarded by proxies.

```
public string? MaxForwards { get; set; }
```

Property Value

[string](#) ↗

Property Origin

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Origin

Gets the value of the HTTP Origin header.

Indicates the origin of the request, helping servers implement CORS and manage cross-origin requests.

```
public string? Origin { get; }
```

Property Value

[string](#) ↗

Property Pragma

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Pragma

Gets or sets the value of the HTTP Pragma header.

Used to include implementation-specific directives that might apply to any recipient along the request/response chain.

```
public string? Pragma { get; set; }
```

Property Value

[string](#) ↗

Property ProxyAuthenticate

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ProxyAuthenticate

Gets or sets the value of the HTTP Proxy-Authenticate header.

Used by a proxy server to request authentication from the client, indicating the authentication method required.

```
public string? ProxyAuthenticate { get; set; }
```

Property Value

[string](#) ↗

Property ProxyAuthorization

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ProxyAuthorization

Gets or sets the value of the HTTP Proxy-Authorization header.

Contains credentials for authenticating the client with a proxy server, allowing access to the requested resource.

```
public string? ProxyAuthorization { get; set; }
```

Property Value

[string](#) ↗

Property Range

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Range

Gets the value of the HTTP Range header.

Used to request a specific range of bytes from a resource, allowing clients to download large files in parts.

```
public string? Range { get; }
```

Property Value

[string](#)

Property Referer

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Referer

Gets the value of the HTTP Referer header.

Indicates the URL of the resource from which the request originated, helping servers understand the source of traffic.

```
public string? Referer { get; }
```

Property Value

[string](#)

Property RetryAfter

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

RetryAfter

Gets or sets the value of the HTTP Retry-After header.

Indicates how long the client should wait before making a follow-up request, often used in rate limiting scenarios.

```
public string? RetryAfter { get; set; }
```

Property Value

[string](#) ↗

Property SecGPC

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

SecGPC

Gets the value of the HTTP Sec-GPC header.

The HTTP Sec-GPC request header is part of the Global Privacy Control (GPC) mechanism to indicate whether the user consents to a website or service selling or sharing their personal information with third parties.

```
public string? SecGPC { get; set; }
```

Property Value

[string](#) ↗

Property SetCookie

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

SetCookie

Gets or sets the value of the HTTP Set-Cookie header.

Used to send cookies from the server to the client, allowing the server to store state information on the client.

```
public string? SetCookie { get; set; }
```

Property Value

[string](#) ↗

Remarks

Note: setting this property, it will override all previous Set-Cookie headers. Use the [Add\(string, string\)](#) method to add more than one Set-Cookie header or use the [SetCookie\(string, string\)](#) method.

Property TE

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

TE

Gets the value of the HTTP TE header.

Indicates the transfer encodings that are acceptable for the response, allowing for content negotiation.

```
public string? TE { get; }
```

Property Value

[string](#) ↗

Property UserAgent

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

UserAgent

Gets the value of the HTTP User-Agent header.

Contains information about the user agent (browser or application) making the request, including its version and platform.

```
public string? UserAgent { get; }
```

Property Value

[string](#)

Property Vary

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Vary

Gets or sets the value of the HTTP Vary header.

Indicates that the response varies based on the value of the specified request headers, allowing for content negotiation.

```
public string? Vary { get; set; }
```

Property Value

[string](#)

Property Via

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Via

Gets or sets the value of the HTTP Via header.

Used to track message forwards and proxies, indicating the intermediate protocols and recipients involved in the request/response chain.

```
public string? Via { get; set; }
```

Property Value

[string](#) ↗

Property WWWAuthenticate

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

WWWAuthenticate

Gets or sets the value of the HTTP WWW-Authenticate header.

Used in response to a request for authentication, indicating the authentication method that should be used to access the resource.

```
public string? WWWAuthenticate { get; set; }
```

Property Value

[string](#)

Property XForwardedFor

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

XForwardedFor

Gets the value of the HTTP X-Forwarded-For header.

Used to identify the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer.

```
public string? XForwardedFor { get; }
```

Property Value

[string](#) ↗

Remarks

Tip: use the [ForwardingResolver](#) property to obtain the user client proxied IP through [RemoteAddress](#).

Property XForwardedHost

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

XForwardedHost

Gets the value of the HTTP X-Forwarded-Host header

Used to identify the original host requested by the client in the Host HTTP request header, often used in proxy setups.

```
public string? XForwardedHost { get; }
```

Property Value

[string](#) ↗

Remarks

Tip: use the [ForwardingResolver](#) property to obtain the client requested host through [Host](#).

Property XFrameOptions

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

XFrameOptions

Gets or sets the value of the HTTP X-Frame-Options header.

Used to control whether a browser should be allowed to render a page in a iframe, frame, embed or object tag, helping to prevent clickjacking attacks.

```
public string? XFrameOptions { get; set; }
```

Property Value

[string](#)

Property XUACOMPATIBLE

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

XUACOMPATIBLE

Gets or sets the value of the HTTP X-UA-Compatible header.

Used to specify the document mode that Internet Explorer should use to render the page, helping to ensure compatibility with older versions.

```
public string? XUACOMPATIBLE { get; set; }
```

Property Value

[string](#)

Class MultipartFormCollection

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Represents an class which hosts an multipart form data contents.

```
public sealed class MultipartFormCollection : IReadOnlyList<MultipartObject>,
IReadOnlyCollection<MultipartObject>, IEnumerable<MultipartObject>, IReadOnlyDictionary<string,
MultipartObject>, IReadOnlyCollection<KeyValuePair<string, MultipartObject>>,
IEnumerable<KeyValuePair<string, MultipartObject>>, IEnumerable
```

Implements

[IReadOnlyList](#)<MultipartObject>, [IReadOnlyCollection](#)<MultipartObject>, [IEnumerable](#)<MultipartObject>,
[IReadOnlyDictionary](#)<string, MultipartObject>,
[IReadOnlyCollection](#)<KeyValuePair<string, MultipartObject>>,
[IEnumerable](#)<KeyValuePair<string, MultipartObject>>, [IEnumerable](#)

Properties

[Count](#)

[Files](#)

Gets a collection of [MultipartObject](#) instances that represent files.

[Keys](#)

[Values](#)

Methods

[ContainsKey\(string\)](#)

[GetEnumerator\(\)](#)

[GetFile\(string\)](#)

Retrieves a [MultipartObject](#) instance by its file name.

[GetItem\(string\)](#)

Gets the last form item by their name. This search is case-insensitive.

[GetItems\(string\)](#)

Gets all form items that shares the specified name. This search is case-insensitive.

[GetStringValue\(string\)](#)

Gets an [StringValue](#) object from the form item content string. This method reads the contents of the last matched last item with the request encoding.

[ToArray\(\)](#)

Creates an array with the [MultipartObject](#) in this collection.

[TryGetValue\(string, out MultipartObject\)](#)

Property Count

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Count

```
public int Count { get; }
```

Property Value

[int](#)

Property Files

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Files

Gets a collection of [MultipartObject](#) instances that represent files.

```
public IEnumerable<MultipartObject> Files { get; }
```

Property Value

[IEnumerable](#)<[MultipartObject](#)>

An [IEnumerable](#)<[T](#)> of [MultipartObject](#) instances.

Property Keys

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Keys

```
public IEnumerable<string> Keys { get; }
```

Property Value

[IEnumerable](#)<[string](#)>

Property Values

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Values

```
public IEnumerable<MultipartObject> Values { get; }
```

Property Value

[IEnumerable](#)<MultipartObject>

Method ContainsKey

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ContainsKey(string)

```
public bool ContainsKey(string key)
```

Parameters

key string ↗

Returns

bool ↗

Method GetEnumerator

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetEnumerator()

```
public IEnumarator<MultipartObject> GetEnumerator()
```

Returns

[IEnumarator](#)<MultipartObject>

Method GetFile

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetFile(string)

Retrieves a [MultipartObject](#) instance by its file name.

```
public MultipartObject? GetFile(string name)
```

Parameters

name [string](#)

The filename of the [MultipartObject](#) to retrieve.

Returns

[MultipartObject](#)

The [MultipartObject](#) instance with the specified filename, or [null](#) if no matching file is found.

Method GetItem

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetItem(string)

Gets the last form item by their name. This search is case-insensitive.

```
public MultipartObject? GetItem(string name)
```

Parameters

name [string](#)

The form item name.

Returns

[MultipartObject](#)

Method GetItems

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetItems(string)

Gets all form items that shares the specified name. This search is case-insensitive.

```
public MultipartObject[] GetItems(string name)
```

Parameters

name [string](#)

The form item name.

Returns

[MultipartObject](#)

An array of [MultipartObject](#) with the specified name.

Method GetStringValue

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetStringValue(string)

Gets an [StringValue](#) object from the form item content string. This method reads the contents of the last matched last item with the request encoding.

```
public StringValue GetStringValue(string name)
```

Parameters

name [string](#)

The form item name.

Returns

[StringValue](#)

Method ToArray

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ToArray()

Creates an array with the [MultipartObject](#) in this collection.

```
public MultipartObject[] ToArray()
```

Returns

[MultipartObject](#)[]

Method TryGetValue

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

TryGetValue(string, out MultipartObject)

```
public bool TryGetValue(string key, out MultipartObject value)
```

Parameters

key [string](#)

value [MultipartObject](#)

Returns

[bool](#)

Class MultipartObject

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Represents an multipart/form-data object.

```
public sealed class MultipartObject : IEquatable<MultipartObject>
```

Implements

[IEquatable](#)<MultipartObject>

Properties

[ContentBytes](#)

Gets this [MultipartObject](#) form data content in bytes.

[ContentLength](#)

Gets this [MultipartObject](#) form data content length in byte count.

[ContentType](#)

Gets the Content-Type header value from this multipart-object.

[Filename](#)

Gets this [MultipartObject](#) provided file name. If this object isn't disposing a file, nothing is returned.

[HasContents](#)

Gets an boolean indicating if this [MultipartObject](#) has contents or not.

[Headers](#)

Gets this [MultipartObject](#) headers.

[IsFile](#)

Gets an boolean indicating if this [MultipartObject](#) is a file or not.

[Name](#)

Gets this [MultipartObject](#) field name.

Methods

[Equals\(MultipartObject?\)](#)

[Equals\(object?\)](#)

[GetCommonFileFormat\(\)](#)

Determines the image format based in the file header for each image content type.

[GetHashCode\(\)](#)

[ReadContentAsString\(\)](#)

Reads the content bytes using the HTTP request content-encoding.

[ReadContentAsString\(Encoding\)](#)

Reads the content bytes with the given encoder.

Property ContentBytes

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ContentBytes

Gets this [MultipartObject](#) form data content in bytes.

```
public byte[] ContentBytes { get; }
```

Property Value

[byte](#) ↗

Property ContentLength

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ContentLength

Gets this [MultipartObject](#) form data content length in byte count.

```
public int ContentLength { get; }
```

Property Value

[int](#)

Property ContentType

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ContentType

Gets the Content-Type header value from this multipart-object.

```
public string? ContentType { get; }
```

Property Value

[string](#)

Property Filename

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Filename

Gets this [MultipartObject](#) provided file name. If this object isn't disposing a file, nothing is returned.

```
public string? Filename { get; }
```

Property Value

[string](#)

Property HasContents

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

HasContents

Gets an boolean indicating if this [MultipartObject](#) has contents or not.

```
public bool HasContents { get; }
```

Property Value

[bool](#)

Property Headers

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Headers

Gets this [MultipartObject](#) headers.

```
public HttpHeaders Headers { get; }
```

Property Value

[HttpHeaderCollection](#)

Property IsFile

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

IsFile

Gets an boolean indicating if this [MultipartObject](#) is a file or not.

```
public bool IsFile { get; }
```

Property Value

bool ↗

Property Name

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Name

Gets this [MultipartObject](#) field name.

```
public string Name { get; }
```

Property Value

[string](#)

Method Equals

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Equals(object?)

```
public override bool Equals(object? obj)
```

Parameters

obj object

Returns

bool

Equals(MultipartObject?)

```
public bool Equals(MultipartObject? other)
```

Parameters

other MultipartObject

Returns

bool

Method GetCommonFileFormat

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetCommonFileFormat()

Determines the image format based in the file header for each image content type.

```
public MultipartObjectCommonFormat GetCommonFileFormat()
```

Returns

[MultipartObjectCommonFormat](#)

Method GetHashCode

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetHashCode()

```
public override int GetHashCode()
```

Returns

[int](#)

Method ReadContentAsString

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ReadContentAsString(Encoding)

Reads the content bytes with the given encoder.

```
public string ReadContentAsString(Encoding encoder)
```

Parameters

encoder [Encoding](#)

Returns

[string](#)

ReadContentAsString()

Reads the content bytes using the HTTP request content-encoding.

```
public string ReadContentAsString()
```

Returns

[string](#)

Enum MultipartObjectCommonFormat

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Represents an image format for Multipart objects.

```
public enum MultipartObjectCommonFormat
```

Fields

BMP = 104

Represents an bitmap image.

GIF = 101

Represents an GIF image.

JPEG = 100

Represents an JPEG/JPG image.

PDF = 200

Represents an PDF file.

PNG = 102

Represents an PNG image.

TIFF = 103

Represents an TIFF image.

Unknown = 0

Represents that the object is not a recognized image.

WEBP = 105

Represents an WebP image.

Class StringKeyStoreCollection

Namespace:[Sisk.Core.Entity](#)

Assembly:[Sisk.Core.dll](#)

Represents a collection of string keys associated with multiple string values.

```
public class StringKeyStoreCollection : IDictionary<string, string[]>,
ICollection<KeyValuePair<string, string[]>>, IEnumerable<KeyValuePair<string,
string[]>>, IEnumerable
```

Implements

[IDictionary<string, string\[\]>](#), [ICollection<KeyValuePair<string, string\[\]>>](#),
[IEnumerable<KeyValuePair<string, string\[\]>>](#), [IEnumerable](#)

Constructors

[StringKeyStoreCollection\(\)](#)

Initializes a new instance of the [StringKeyStoreCollection](#) class,

[StringKeyStoreCollection\(IEqualityComparer<string>\)](#)

Initializes a new instance of the [StringKeyStoreCollection](#) class with a specified comparer.

[StringKeyStoreCollection\(IEqualityComparer<string>, IDictionary<string, string\[\]>?\)](#)

Initializes a new instance of the [StringKeyStoreCollection](#) class,

Properties

[Comparer](#)

Gets the [IEqualityComparer<T>](#) used to compare keys in this [StringKeyStoreCollection](#).

[Count](#)

Gets the number of key-value pairs in the [StringKeyStoreCollection](#).

[IsReadOnly](#)

Gets a value indicating whether the [StringKeyStoreCollection](#) is read-only.

this[string]

Gets or sets the array of values associated with the specified key. Returns [null](#) if the key does not exist in the store.

Keys

Gets the collection of keys in the [StringKeyStoreCollection](#).

Values

Gets the collection of values in the [StringKeyStoreCollection](#) as arrays. Each key may have multiple associated values.

Methods

Add(KeyValuePair<string, string[]>)

Adds a key-value pair to the [StringKeyStoreCollection](#).

Add(string, IEnumerable<string>)

Adds a collection of values associated with the specified key.

Add(string, string)

Adds a single value associated with the specified key.

Add(string, string[])

Adds an array of values associated with the specified key.

AddRange(IEnumerable<KeyValuePair<string, string[]>>)

Adds the elements of the specified collection to the end of this collection.

AddRange(IEnumerable<KeyValuePair<string, string?>>)

Adds the elements of the specified collection to the end of this collection.

AsDictionary()

Copies the contents of this [StringKeyStoreCollection](#) into an [Dictionary< TKey, TValue >](#).

AsNameValueCollection()

Copies the contents of this [StringKeyStoreCollection](#) into an [NameValueCollection](#), with values separated with an comma (,).

AsStringValueCollection()

Copies the contents of this [StringKeyStoreCollection](#) into an [StringValueCollection](#).

[Clear\(\)](#)

Removes all key-value pairs from the [StringKeyStoreCollection](#). Throws an exception if the store is read-only.

[ContainsKey\(string\)](#)

Determines whether the [StringKeyStoreCollection](#) contains a specific key.

[FromCookieString\(string\)](#)

Creates a new instance of the [StringKeyStoreCollection](#) from a cookie string. The query string should be in the format of "key1=value1; key2=value2".

[FromNameValueCollection\(NameValueCollection\)](#)

Creates a new instance of the [StringKeyStoreCollection](#) from a [NameValueCollection](#).

[FromQueryString\(string\)](#)

Creates a new instance of the [StringKeyStoreCollection](#) from a query string. The query string should be in the format of "key1=value1&key2=value2".

[GetEnumerator\(\)](#)

[GetValue\(string\)](#)

Retrieves the last value associated with the specified key. Returns [null](#) if the key does not exist.

[GetValues\(string\)](#)

Retrieves all values associated with the specified key. Returns an empty array if the key does not exist.

[ImportCookieString\(string\)](#)

Imports key-value pairs from a cookie string into the [StringKeyStoreCollection](#). The query string should be in the format of "key1=value1; key2=value2".

[ImportNameValueCollection\(NameValueCollection\)](#)

Imports key-value pairs from a [NameValueCollection](#) into the [StringKeyStoreCollection](#). Each key can have multiple associated values.

[ImportQueryString\(string\)](#)

Imports key-value pairs from a query string into the [StringKeyStoreCollection](#). The query string should be in the format of "key1=value1&key2=value2".

[MakeReadOnly\(\)](#)

Marks the [StringKeyStoreCollection](#) as read-only, preventing further modifications.

[Remove\(string\)](#)

Removes the value associated with the specified key from the [StringKeyStoreCollection](#). Throws an exception if the store is read-only.

[Set\(KeyValuePair<string, string\[\]>\)](#)

Sets the value associated with the specified key, replacing any existing values.

[Set\(string, IEnumerable<string>\)](#)

Sets the collection of values associated with the specified key, replacing any existing values.

[Set\(string, string\)](#)

Sets the value associated with the specified key, replacing any existing values.

[SetRange\(IEnumerable<KeyValuePair<string, string\[\]>>\)](#)

Sets the elements of the specified collection, replacing existing values.

[ToString\(\)](#)

Returns a string that represents the current object.

[ToString\(IFormatProvider?\)](#)

Returns a string that represents the current object, using the specified format provider.

[TryGetValue\(string, out string\[\]\)](#)

Tries to get the array of values associated with the specified key.

Constructor StringKeyStoreCollection

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

StringKeyStoreCollection()

Initializes a new instance of the [StringKeyStoreCollection](#) class,

```
public StringKeyStoreCollection()
```

StringKeyStoreCollection(IEqualityComparer<string>)

Initializes a new instance of the [StringKeyStoreCollection](#) class with a specified comparer.

```
public StringKeyStoreCollection(IEqualityComparer<string> comparer)
```

Parameters

comparer [IEqualityComparer<string>](#)

The comparer used for key equality.

StringKeyStoreCollection(IEqualityComparer<string>, IDictionary<string, string[]>?)

Initializes a new instance of the [StringKeyStoreCollection](#) class,

```
public StringKeyStoreCollection(IEqualityComparer<string> comparer, IDictionary<string, string[]?> items)
```

Parameters

comparer [IEqualityComparer<string>](#)

The comparer used for key equality.

items [IDictionary<string, string\[\]>](#)

The inner collection to add to this instance.

Property Comparer

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Comparer

Gets the [IEqualityComparer<T>](#) used to compare keys in this [StringKeyStoreCollection](#).

```
public IEqualityComparer<string> Comparer { get; }
```

Property Value

[IEqualityComparer<string>](#)

Property Count

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Count

Gets the number of key-value pairs in the [StringKeyStoreCollection](#).

```
public int Count { get; }
```

Property Value

[int](#)

Property IsReadOnly

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

IsReadOnly

Gets a value indicating whether the [StringKeyStoreCollection](#) is read-only.

```
public bool IsReadOnly { get; }
```

Property Value

[bool](#)

Property Keys

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Keys

Gets the collection of keys in the [StringKeyStoreCollection](#).

```
public ICollection<string> Keys { get; }
```

Property Value

[ICollection](#)<[string](#)>

Property Values

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Values

Gets the collection of values in the [StringKeyStoreCollection](#) as arrays. Each key may have multiple associated values.

```
public ICollection<string[]> Values { get; }
```

Property Value

[ICollection](#)<[string](#)[]>

Property this

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

this[string]

Gets or sets the array of values associated with the specified key. Returns [null](#) if the key does not exist in the store.

```
public string? this[string key] { get; set; }
```

Parameters

key [string](#)

Property Value

[string](#)

Method Add

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Add(string, string[])

Adds an array of values associated with the specified key.

```
public void Add(string key, string[] value)
```

Parameters

key [string](#)

The key to which the values will be added.

value [string\[\]](#)

The array of values to associate with the key.

Add(string, IEnumerable<string>)

Adds a collection of values associated with the specified key.

```
public void Add(string key, IEnumerable<string> value)
```

Parameters

key [string](#)

The key to which the values will be added.

value [IEnumerable<string>](#)

The collection of values to associate with the key.

Add(string, string)

Adds a single value associated with the specified key.

```
public void Add(string key, string value)
```

Parameters

key `string`

The key to which the value will be added.

value `string`

The value to associate with the key.

Add(KeyValuePair<string, string[]>)

Adds a key-value pair to the `StringKeyStoreCollection`.

```
public void Add(KeyValuePair<string, string[]> item)
```

Parameters

item `KeyValuePair<string, string[]>`

The key-value pair to add, where the key is associated with an array of values.

Method AddRange

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AddRange(IEnumerable<KeyValuePair<string, string[]>>)

Adds the elements of the specified collection to the end of this collection.

```
public void AddRange(IEnumerable<KeyValuePair<string, string[]>> items)
```

Parameters

items `IEnumerable<KeyValuePair<string[], string[]>>`

The collection whose items should be added to the end of this collection.

AddRange(IEnumerable<KeyValuePair<string, string?>>)

Adds the elements of the specified collection to the end of this collection.

```
public void AddRange(IEnumerable<KeyValuePair<string, string?>> items)
```

Parameters

items `IEnumerable<KeyValuePair<string?, string?>>`

The collection whose items should be added to the end of this collection.

Method AsDictionary

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AsDictionary()

Copies the contents of this [StringKeyStoreCollection](#) into an [Dictionary<TKey, TValue>](#).

```
public IDictionary<string, string[]> AsDictionary()
```

Returns

[IDictionary<string\[\], string\[\]>](#)

Method AsNameValueCollection

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AsNameValueCollection()

Copies the contents of this [StringKeyStoreCollection](#) into an [NameValuePairCollection](#), with values separated with an comma (,).

```
public NameValueCollection AsNameValueCollection()
```

Returns

[NameValuePairCollection](#)

Method AsStringValueCollection

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

AsStringValueCollection()

Copies the contents of this [StringKeyStoreCollection](#) into an [StringValueCollection](#).

```
public StringValueCollection AsStringValueCollection()
```

Returns

[StringValueCollection](#)

Method Clear

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Clear()

Removes all key-value pairs from the [StringKeyStoreCollection](#). Throws an exception if the store is read-only.

```
public void Clear()
```

Method ContainsKey

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ContainsKey(string)

Determines whether the [StringKeyStoreCollection](#) contains a specific key.

```
public bool ContainsKey(string key)
```

Parameters

key [string](#)

The key to locate in the [StringKeyStoreCollection](#).

Returns

[bool](#)

[true](#) if the [StringKeyStoreCollection](#) contains an element with the specified key; otherwise, [false](#).

Method FromCookieString

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

FromCookieString(string)

Creates a new instance of the [StringKeyStoreCollection](#) from a cookie string. The query string should be in the format of "key1=value1; key2=value2".

```
public static StringKeyStoreCollection FromCookieString(string queryString)
```

Parameters

queryString string ↗

The query string containing the key-value pairs to import.

Returns

[StringKeyStoreCollection](#)

A new [StringKeyStoreCollection](#) populated with the key-value pairs from the query string.

Method FromNameValueCollection

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

FromNameValueCollection(NameValueCollection)

Creates a new instance of the [StringKeyStoreCollection](#) from a [NameValueCollection](#).

```
public static StringKeyStoreCollection FromNameValueCollection(NameValueCollection collection)
```

Parameters

collection [NameValuePairCollection](#)

The [NameValuePairCollection](#) containing the key-value pairs to import.

Returns

[StringKeyStoreCollection](#)

A new [StringKeyStoreCollection](#) populated with the key-value pairs from the query string.

Method FromQueryString

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

FromQueryString(string)

Creates a new instance of the [StringKeyStoreCollection](#) from a query string. The query string should be in the format of "key1=value1&key2=value2".

```
public static StringKeyStoreCollection FromQueryString(string queryString)
```

Parameters

queryString string ↗

The query string containing the key-value pairs to import.

Returns

[StringKeyStoreCollection](#)

A new [StringKeyStoreCollection](#) populated with the key-value pairs from the query string.

Method GetEnumerator

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetEnumerator()

```
public IEnumarator<KeyValuePair<string, string[]>> GetEnumerator()
```

Returns

[IEnumarator](#)<KeyValuePair<string[], string[]>>

Method GetValue

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetValue(string)

Retrieves the last value associated with the specified key. Returns [null](#) if the key does not exist.

```
public string? GetValue(string name)
```

Parameters

name [string](#)

The key for which to retrieve the value.

Returns

[string](#)

The last value associated with the specified key, or [null](#) if the key is not found.

Method GetValues

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetValues(string)

Retrieves all values associated with the specified key. Returns an empty array if the key does not exist.

```
public string[] GetValues(string name)
```

Parameters

name [string](#)

The key for which to retrieve the values.

Returns

[string](#)[]

An array of values associated with the specified key, or an empty array if the key is not found.

Method ImportCookieString

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ImportCookieString(string)

Imports key-value pairs from a cookie string into the [StringKeyStoreCollection](#). The query string should be in the format of "key1=value1; key2=value2".

```
public void ImportCookieString(string queryString)
```

Parameters

queryString string ↗

The query string containing the key-value pairs to import.

Method ImportNameValueCollection

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ImportNameValueCollection(NameValueCollection)

Imports key-value pairs from a [NameValuePairCollection](#) into the [StringKeyStoreCollection](#). Each key can have multiple associated values.

```
public void ImportNameValueCollection(NameValueCollection items)
```

Parameters

items [NameValuePairCollection](#)

The [NameValuePairCollection](#) containing the key-value pairs to import.

Method ImportQueryString

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ImportQueryString(string)

Imports key-value pairs from a query string into the [StringKeyStoreCollection](#). The query string should be in the format of "key1=value1&key2=value2".

```
public void ImportQueryString(string queryString)
```

Parameters

queryString string ↗

The query string containing the key-value pairs to import.

Method MakeReadOnly

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

MakeReadOnly()

Marks the [StringKeyStoreCollection](#) as read-only, preventing further modifications.

```
public void MakeReadOnly()
```

Method Remove

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Remove(string)

Removes the value associated with the specified key from the [StringKeyStoreCollection](#). Throws an exception if the store is read-only.

```
public bool Remove(string key)
```

Parameters

key [string](#)

The key of the value to remove.

Returns

[bool](#)

[true](#) if the key was successfully removed; otherwise, [false](#).

Method Set

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Set(KeyValuePair<string, string[]>)

Sets the value associated with the specified key, replacing any existing values.

```
public void Set(KeyValuePair<string, string[]> item)
```

Parameters

item KeyValuePair<string, string[]>

The key-value pair to add, where the key is associated with an array of values.

Set(string, string)

Sets the value associated with the specified key, replacing any existing values.

```
public void Set(string key, string value)
```

Parameters

key string

The key for which to set the value.

value string

The value to associate with the key.

Set(string, IEnumerable<string>)

Sets the collection of values associated with the specified key, replacing any existing values.

```
public void Set(string key, IEnumerable<string> value)
```

Parameters

key `string`

The key for which to set the values.

value `IEnumerable<string>`

The collection of values to associate with the key.

Method SetRange

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

SetRange(IEnumerable<KeyValuePair<string, string[]>>)

Sets the elements of the specified collection, replacing existing values.

```
public void SetRange(IEnumerable<KeyValuePair<string, string[]>> items)
```

Parameters

items `IEnumerable<KeyValuePair<string[], string[]>>`

The collection whose items should be replaced or added to this collection.

Method ToString

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

ToString()

Returns a string that represents the current object.

```
public override string ToString()
```

Returns

[string](#)

A string that represents the current object.

ToString(IFormatProvider?)

Returns a string that represents the current object, using the specified format provider.

```
public string ToString(IFormatProvider? formatProvider = null)
```

Parameters

formatProvider [IFormatProvider](#)

The format provider to use when formatting the string. If null, the current culture is used.

Returns

[string](#)

A string that represents the current object, formatted using the specified format provider.

Method TryGetValue

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

TryGetValue(string, out string[])

Tries to get the array of values associated with the specified key.

```
public bool TryGetValue(string key, out string[] value)
```

Parameters

key [string](#)

The key for which to retrieve the values.

value [string\[\]](#)

When this method returns, contains the array of values associated with the specified key, or an empty array if the key is not found.

Returns

[bool](#)

[true](#) if the key was found and the values were retrieved; otherwise, [false](#).

Struct StringValue

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Represents an option/monad item that wraps an string value and allows conversion to most common types.

```
public readonly struct StringValue : ICloneable, IEquatable<StringValue>,  
IComparable<StringValue>
```

Implements

[ICloneable](#), [IEquatable](#)<[StringValue](#)>, [IComparable](#)<[StringValue](#)>

Constructors

[StringValue\(string\)](#)

Creates an new empty value of the [StringValue](#) with no predefined value.

[StringValue\(string, string?\)](#)

Creates an new value of the [StringValue](#).

Properties

[IsNull](#)

Gets an boolean indicating if this object value is null.

[IsNullOrEmpty](#)

Gets an boolean indicating if this object value is null or an empty string.

[Name](#)

Gets the name of the property that hosts this [StringValue](#).

[Value](#)

Gets the value of the current [StringValue](#) string if it has been assigned a valid underlying value.

Methods

[Clone\(\)](#)

[CompareTo\(StringValue, in StringComparison\)](#)

Compares the current object with another object of the same type, using the specified string comparison.

[Create\(string?\)](#)

Creates an new [StringValue](#) from the specified string.

[GetBoolean\(\)](#)

Parses the value contained in this [StringValue](#) as a [bool](#). Throws an exception if the value couldn't be parsed to the target type.

[GetByte\(IFormatProvider?\)](#)

Parses the value contained in this [StringValue](#) as a [byte](#). Throws an exception if the value couldn't be parsed to the target type.

[GetChar\(\)](#)

Parses the value contained in this [StringValue](#) as a [char](#). Throws an exception if the value couldn't be parsed to the target type.

[GetDateTime\(IFormatProvider?\)](#)

Parses the value contained in this [StringValue](#) as a [DateTime](#). Throws an exception if the value couldn't be parsed to the target type.

[GetDecimal\(IFormatProvider?\)](#)

Parses the value contained in this [StringValue](#) as a [decimal](#). Throws an exception if the value couldn't be parsed to the target type.

[GetDouble\(IFormatProvider?\)](#)

Parses the value contained in this [StringValue](#) as a [double](#). Throws an exception if the value couldn't be parsed to the target type.

[GetEnum<TEnum>\(\)](#)

Gets an [Enum](#) object representation from this [StringValue](#), parsing the current string expression into an value of [TEnum](#). This method will throw an [NullReferenceException](#) if the value stored in this instance is null.

[GetGuid\(IFormatProvider?\)](#)

Parses the value contained in this [StringValue](#) as a [Guid](#). Throws an exception if the value couldn't be parsed to the target type.

[GetInteger\(IFormatProvider?\)](#)

Parses the value contained in this [StringValue](#) as an [int](#). Throws an exception if the value couldn't be parsed to the target type.

[GetLong\(IFormatProvider?\)](#)

Parses the value contained in this [StringValue](#) as a [long](#). Throws an exception if the value couldn't be parsed to the target type.

[GetShort\(IFormatProvider?\)](#)

Parses the value contained in this [StringValue](#) as a [short](#). Throws an exception if the value couldn't be parsed to the target type.

[GetSingle\(IFormatProvider?\)](#)

Parses the value contained in this [StringValue](#) as a [float](#). Throws an exception if the value couldn't be parsed to the target type.

[GetString\(\)](#)

Gets a non-null string from this [StringValue](#). This method will throw an [NullReferenceException](#) if the value stored in this instance is null.

[Get<T>\(IFormatProvider?\)](#)

Parses the value contained in this [StringValue](#) as a type `T` that implements [IParsable<TSelf>](#). Throws an exception if the value couldn't be parsed to the target type.

[MaybeNull\(\)](#)

Returns a self-reference to this object when it's value is not null.

[MaybeNullOrEmpty\(\)](#)

Returns a self-reference to this object when it's value is not null or an empty string.

Constructor StringValue

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

StringValue(string)

Creates an new empty value of the [StringValue](#) with no predefined value.

```
public StringValue(string name)
```

Parameters

name string ↗

The [StringValue](#) name.

StringValue(string, string?)

Creates an new value of the [StringValue](#).

```
public StringValue(string name, string? value)
```

Parameters

name string ↗

The [StringValue](#) name.

value string ↗

The [StringValue](#) value.

Property IsNull

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

IsNull

Gets an boolean indicating if this object value is null.

```
public bool IsNull { get; }
```

Property Value

bool ↗

Property IsNullOrEmpty

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

IsNullOrEmpty

Gets an boolean indicating if this object value is null or an empty string.

```
public bool IsNullOrEmpty { get; }
```

Property Value

bool ↗

Property Name

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Name

Gets the name of the property that hosts this [StringValue](#).

```
public string Name { get; }
```

Property Value

[string](#)

Property Value

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Value

Gets the value of the current [StringValue](#) string if it has been assigned a valid underlying value.

```
public string? Value { get; }
```

Property Value

[string](#) ↗

Method Clone

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Clone()

```
public object Clone()
```

Returns

[object](#)

Method CompareTo

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

CompareTo(StringValue, in StringComparison)

Compares the current object with another object of the same type, using the specified string comparison.

```
public int CompareTo(StringValue other, in StringComparison stringComparison)
```

Parameters

other [StringValue](#)

The object to compare with the current object.

stringComparison [StringComparison](#)

One of the [StringComparison](#) values that specifies the comparison rules to use.

Returns

[int](#)

A value that indicates the relative order of the objects being compared.

See Also

[StringComparison](#)

Method Create

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Create(string?)

Creates an new [StringValue](#) from the specified string.

```
public static StringValue Create(string? value)
```

Parameters

value [string](#)

The string value.

Returns

[StringValue](#)

Method Get

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Get<T>(IFormatProvider?)

Parses the value contained in this [StringValue](#) as a type `T` that implements [IParsable<TSelf>](#). Throws an exception if the value couldn't be parsed to the target type.

```
public T Get<T>(IFormatProvider? formatProvider = null) where T : IParsable<T>
```

Parameters

formatProvider [IFormatProvider](#)

The [IFormatProvider](#) to use for parsing. Defaults to [null](#).

Returns

`T`

The converted value of type `T`.

Type Parameters

`T`

The type to parse the value to.

Exceptions

[FormatException](#)

Thrown when the value cannot be parsed to type `T`.

Method GetBoolean

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetBoolean()

Parses the value contained in this [StringValue](#) as a [bool](#). Throws an exception if the value couldn't be parsed to the target type.

```
public bool GetBoolean()
```

Returns

[bool](#)

The converted [bool](#).

Exceptions

[FormatException](#)

Thrown when the value cannot be parsed to a [bool](#).

Method GetByte

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetByte(IFormatProvider?)

Parses the value contained in this [StringValue](#) as a [byte](#). Throws an exception if the value couldn't be parsed to the target type.

```
public int GetByte(IFormatProvider? formatProvider = null)
```

Parameters

formatProvider [IFormatProvider](#)

The [IFormatProvider](#) to use for parsing. Defaults to [null](#).

Returns

[int](#)

The converted [byte](#).

Exceptions

[FormatException](#)

Thrown when the value cannot be parsed to a [byte](#).

Method GetChar

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetChar()

Parses the value contained in this [StringValue](#) as a [char](#). Throws an exception if the value couldn't be parsed to the target type.

```
public char GetChar()
```

Returns

[char](#)

The converted [char](#).

Method GetDateTime

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetDateTime(IFormatProvider?)

Parses the value contained in this [StringValue](#) as a [DateTime](#). Throws an exception if the value couldn't be parsed to the target type.

```
public DateTime GetDateTime(IFormatProvider? formatProvider = null)
```

Parameters

formatProvider [IFormatProvider](#)

The [IFormatProvider](#) to use for parsing. Defaults to [null](#).

Returns

[DateTime](#)

The converted [DateTime](#).

Exceptions

[FormatException](#)

Thrown when the value cannot be parsed to a [DateTime](#).

Method GetDecimal

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetDecimal(IFormatProvider?)

Parses the value contained in this [StringValue](#) as a [decimal](#). Throws an exception if the value couldn't be parsed to the target type.

```
public decimal GetDecimal(IFormatProvider? formatProvider = null)
```

Parameters

formatProvider [IFormatProvider](#)

The [IFormatProvider](#) to use for parsing. Defaults to [null](#).

Returns

[decimal](#)

The converted [decimal](#).

Exceptions

[FormatException](#)

Thrown when the value cannot be parsed to a [decimal](#).

Method GetDouble

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetDouble(IFormatProvider?)

Parses the value contained in this [StringValue](#) as a [double](#). Throws an exception if the value couldn't be parsed to the target type.

```
public double GetDouble(IFormatProvider? formatProvider = null)
```

Parameters

formatProvider [IFormatProvider](#)

The [IFormatProvider](#) to use for parsing. Defaults to [null](#).

Returns

[double](#)

The converted [double](#).

Exceptions

[FormatException](#)

Thrown when the value cannot be parsed to a [double](#).

Method GetEnum

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetEnum<TEnum>()

Gets an [Enum](#) object representation from this [StringValue](#), parsing the current string expression into an value of [TEnum](#). This method will throw an [NullReferenceException](#) if the value stored in this instance is null.

```
public TEnum GetEnum<TEnum>() where TEnum : struct, Enum
```

Returns

TEnum

Type Parameters

TEnum

The [Enum](#) type.

Method GetGuid

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetGuid(IFormatProvider?)

Parses the value contained in this [StringValue](#) as a [Guid](#). Throws an exception if the value couldn't be parsed to the target type.

```
public Guid GetGuid(IFormatProvider? formatProvider = null)
```

Parameters

formatProvider [IFormatProvider](#)

The [IFormatProvider](#) to use for parsing. Defaults to [null](#).

Returns

[Guid](#)

The converted [Guid](#).

Exceptions

[FormatException](#)

Thrown when the value cannot be parsed to a [Guid](#).

Method GetInteger

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetInteger(IFormatProvider?)

Parses the value contained in this [StringValue](#) as an [int](#). Throws an exception if the value couldn't be parsed to the target type.

```
public int GetInteger(IFormatProvider? formatProvider = null)
```

Parameters

formatProvider [IFormatProvider](#)

The [IFormatProvider](#) to use for parsing. Defaults to [null](#).

Returns

[int](#)

The converted [int](#).

Exceptions

[FormatException](#)

Thrown when the value cannot be parsed to an [int](#).

Method GetLong

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetLong(IFormatProvider?)

Parses the value contained in this [StringValue](#) as a [long](#). Throws an exception if the value couldn't be parsed to the target type.

```
public long GetLong(IFormatProvider? formatProvider = null)
```

Parameters

formatProvider [IFormatProvider](#)

The [IFormatProvider](#) to use for parsing. Defaults to [null](#).

Returns

[long](#)

The converted [long](#).

Exceptions

[FormatException](#)

Thrown when the value cannot be parsed to a [long](#).

Method GetShort

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetShort(IFormatProvider?)

Parses the value contained in this [StringValue](#) as a [short](#). Throws an exception if the value couldn't be parsed to the target type.

```
public short GetShort(IFormatProvider? formatProvider = null)
```

Parameters

formatProvider [IFormatProvider](#)

The [IFormatProvider](#) to use for parsing. Defaults to [null](#).

Returns

[short](#)

The converted [short](#).

Exceptions

[FormatException](#)

Thrown when the value cannot be parsed to a [short](#).

Method GetSingle

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetSingle(IFormatProvider?)

Parses the value contained in this [StringValue](#) as a [float](#). Throws an exception if the value couldn't be parsed to the target type.

```
public float GetSingle(IFormatProvider? formatProvider = null)
```

Parameters

formatProvider [IFormatProvider](#)

The [IFormatProvider](#) to use for parsing. Defaults to [null](#).

Returns

[float](#)

The converted [float](#).

Exceptions

[FormatException](#)

Thrown when the value cannot be parsed to a [float](#).

Method GetString

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetString()

Gets a non-null string from this [StringValue](#). This method will throw an [NullReferenceException](#) if the value stored in this instance is null.

```
public string GetString()
```

Returns

[string](#)

An non-null string value.

Exceptions

[NullReferenceException](#)

Thrown when the value stored in this instance is null.

Method MaybeNull

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

MaybeNull()

Returns a self-reference to this object when it's value is not null.

```
public StringValue? MaybeNull()
```

Returns

[StringValue?](#)

Method MaybeNullOrEmpty

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

MaybeNullOrEmpty()

Returns a self-reference to this object when it's value is not null or an empty string.

```
public StringValue? MaybeNullOrEmpty()
```

Returns

[StringValue?](#)

Class StringValueCollection

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Represents an collection of [StringValue](#).

```
public sealed class StringValueCollection : StringKeyStoreCollection, IDictionary<string, string[]>, ICollection<KeyValuePair<string, string[]>>, IEnumerable<KeyValuePair<string, string[]>>, I Enumerable
```

Implements

[IDictionary](#)<string, string[]>, [ICollection](#)<KeyValuePair<string, string[]>>, [IEnumerable](#)<KeyValuePair<string, string[]>>, [I Enumerable](#)

Constructors

[StringValueCollection\(\)](#)

Creates an new empty [StringValueCollection](#).

[StringValueCollection\(IDictionary<string, string\[\]>\)](#)

Creates an new [StringValueCollection](#) instance with values from another [IDictionary](#) instance.

[StringValueCollection\(IDictionary<string, string?>\)](#)

Creates an new [StringValueCollection](#) instance with values from another [IDictionary](#) instance.

Properties

[this\[string\]](#)

Gets or sets an [StringValue](#) item by their key name.

Methods

[GetItem\(string\)](#)

Gets an [StringValue](#) from their key name. If the object was not found by their name, an empty non-null [StringValue](#) with no value is returned.

[GetItems\(string\)](#)

Gets an array of [StringValue](#) from their key name. If the object was not found by their name, an empty array of [StringValue](#) is returned.

[TryGetValue\(string, out StringValue\)](#)

Tries to get the last [StringValue](#) associated with the specified key.

Constructor StringValueCollection

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

StringValueCollection(IDictionary<string, string?>)

Creates an new [StringValueCollection](#) instance with values from another [IDictionary](#) instance.

```
public StringValueCollection(IDictionary<string, string?> values)
```

Parameters

values [IDictionary](#)<string, string>

StringValueCollection(IDictionary<string, string[]>)

Creates an new [StringValueCollection](#) instance with values from another [IDictionary](#) instance.

```
public StringValueCollection(IDictionary<string, string[]> values)
```

Parameters

values [IDictionary](#)<string, string[]>

StringValueCollection()

Creates an new empty [StringValueCollection](#).

```
public StringValueCollection()
```

Property this

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

this[string]

Gets or sets an [StringValue](#) item by their key name.

```
public StringValue this[string name] { get; set; }
```

Parameters

name [string](#)

Property Value

[StringValue](#)

Method GetItem

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetItem(string)

Gets an [StringValue](#) from their key name. If the object was not found by their name, an empty non-null [String Value](#) with no value is returned.

```
public StringValue GetItem(string name)
```

Parameters

name [string](#) ↗

Returns

[StringValue](#)

Method GetItems

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetItems(string)

Gets an array of [StringValue](#) from their key name. If the object was not found by their name, an empty array of [StringValue](#) is returned.

```
public StringValue[] GetItems(string name)
```

Parameters

name [string](#)

Returns

[StringValue\[\]](#)

Method TryGetValue

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

TryGetValue(string, out StringValue)

Tries to get the last [StringValue](#) associated with the specified key.

```
public bool TryGetValue(string key, out StringValue value)
```

Parameters

key [string](#)

The key for which to retrieve the values.

value [StringValue](#)

When this method returns, the [StringValue](#) containing the value, or empty [StringValue](#).

Returns

[bool](#)

[true](#) if the key was found; otherwise, [false](#).

Class TypedValueDictionary

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Represents the base class for storing and retrieving data by their type.

```
public class TypedValueDictionary : IDictionary<string, object?>,
ICollection<KeyValuePair<string, object?>>, IEnumerable<KeyValuePair<string, object?>>,
>>, IEnumerable
```

Implements

[IDictionary](#)<string, object>, [ICollection](#)<KeyValuePair<string, object>>,
[IEnumerable](#)<KeyValuePair<string, object>>, [IEnumerable](#)

Constructors

[TypedValueDictionary\(\)](#)

Creates an new [TypedValueDictionary](#) instance with default parameters.

[TypedValueDictionary\(StringComparer\)](#)

Creates an new [TypedValueDictionary](#) instance with default parameters with the specified [StringComparer](#).

Methods

[GetOrAddAsync<T>\(Func<Task<T>>\)](#)

Asynchronously gets a singleton previously defined in this context bag via its type `T`. If it does not exist, it adds the object to the context bag using the provided asynchronous getter function.

[GetOrAdd<T>\(\)](#)

Gets a singleton previously defined in this context bag via its type `T`. If it does not exist, it adds the object to the context bag by creating a new instance of `T`.

[GetOrAdd<T>\(Func<T>\)](#)

Gets a singleton previously defined in this context bag via its type `T`. If it does not exist, it adds the object to the context bag using the provided `getter` function.

[GetOrDefault<T>\(\)](#)

Gets a singleton previously defined in this context bag via its type `T`. Returns the default value if the object is not defined.

[GetTypeKeyName\(Type\)](#)

Gets the Type full qualified key name.

[GetValue\(string\)](#)

Gets the value associated with the specified key.

[GetValue<T>\(string\)](#)

Gets the value associated with the specified key, and converts it to the specified type.

[Get<T>\(\)](#)

Gets a singleton previously defined in this context bag via it's type `T`.

[IsSet<T>\(\)](#)

Determines whether the specified `T` singleton is defined in this context.

[IsSet<T>\(out T\)](#)

Determines whether the specified `T` singleton is defined in this context and tries to output it.

[SetValue\(string, object?\)](#)

Sets the value associated with the specified key.

[Set<T>\(\)](#)

Creates and adds an singleton of `T` in this context bag.

[Set<T>\(T\)](#)

Adds an singleton of `T` in this context bag.

[TryGetValue<TResult>\(string, out TResult?\)](#)

Gets the value associated with the specified key and casts it into `TResult`.

[Unset<T>\(\)](#)

Removes an singleton object from it's type `T`.

Constructor TypedValueDictionary

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

TypedValueDictionary()

Creates an new [TypedValueDictionary](#) instance with default parameters.

```
public TypedValueDictionary()
```

TypedValueDictionary(StringComparer)

Creates an new [TypedValueDictionary](#) instance with default parameters with the specified [StringComparer](#).

```
public TypedValueDictionary(StringComparer keyComparer)
```

Parameters

keyComparer [StringComparer](#)

Method Get

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Get<T>()

Gets a singleton previously defined in this context bag via it's type `T`.

```
public T Get<T>() where T : notnull
```

Returns

`T`

Type Parameters

`T`

The type of the object defined in this context bag.

Method GetOrAdd

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetOrAdd<T>(Func<T>)

Gets a singleton previously defined in this context bag via its type `T`. If it does not exist, it adds the object to the context bag using the provided `getter` function.

```
public T GetOrAdd<T>(Func<T> getter) where T : notnull
```

Parameters

`getter Func<T>`

A function that provides the object to be added if it does not exist.

Returns

`T`

The object of type `T` from the context bag.

Type Parameters

`T`

The type of the object defined in this context bag.

GetOrAdd<T>()

Gets a singleton previously defined in this context bag via its type `T`. If it does not exist, it adds the object to the context bag by creating a new instance of `T`.

```
public T GetOrAdd<T>() where T : notnull, new()
```

Returns

T

The object of type T from the context bag.

Type Parameters

T

The type of the object defined in this context bag. It must have a public parameterless constructor.

Method GetOrAddAsync

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetOrAddAsync<T>(Func<Task<T>>)

Asynchronously gets a singleton previously defined in this context bag via its type `T`. If it does not exist, it adds the object to the context bag using the provided asynchronous getter function.

```
public Task<T> GetOrAddAsync<T>(Func<Task<T>> getter) where T : notnull
```

Parameters

getter `Func<Task<T>>`

An asynchronous function that provides the object to be added if it does not exist.

Returns

`Task<T>`

A task that represents the asynchronous operation. The task result contains the object of type `T` from the context bag.

Type Parameters

`T`

The type of the object defined in this context bag.

Method GetOrDefault

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetOrDefault<T>()

Gets a singleton previously defined in this context bag via its type `T`. Returns the default value if the object is not defined.

```
public T? GetOrDefault<T>() where T : notnull
```

Returns

`T`

The object of type `T` if it exists; otherwise, [null](#).

Type Parameters

`T`

The type of the object defined in this context bag.

Method GetTypeKeyName

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetTypeKeyName(Type)

Gets the Type full qualified key name.

```
protected string GetTypeKeyName(Type t)
```

Parameters

t [Type](#)

The type to get their qualified key name.

Returns

[string](#)

Method GetValue

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

GetValue(string)

Gets the value associated with the specified key.

```
public object? GetValue(string key)
```

Parameters

key [string](#)

The key of the value to get.

Returns

[object](#)

The value associated with the specified key, or [null](#) if the key is not found.

GetValue<T>(string)

Gets the value associated with the specified key, and converts it to the specified type.

```
public T? GetValue<T>(string key)
```

Parameters

key [string](#)

The key of the value to get.

Returns

T

The value associated with the specified key, converted to the specified type, or the default value for the type if the key is not found.

Type Parameters

T

The type to convert the value to.

Method IsSet

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

IsSet<T>()

Determines whether the specified `T` singleton is defined in this context.

```
public bool IsSet<T>() where T : notnull
```

Returns

`bool` ↗

Type Parameters

`T`

The singleton type.

IsSet<T>(out T)

Determines whether the specified `T` singleton is defined in this context and tries to output it.

```
public bool IsSet<T>(out T value) where T : notnull
```

Parameters

`value` `T`

When this method returns, the value associated with the specified key, if the key is found; otherwise, the default value for the type of the value parameter. This parameter is passed uninitialized.

Returns

`bool` ↗

True if the object is find with the specified key; otherwise, false.

Type Parameters

T

The singleton type.

Method Set

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Set<T>()

Creates and adds an singleton of `T` in this context bag.

```
public T Set<T>() where T : notnull, new()
```

Returns

`T`

Type Parameters

`T`

The object that will be defined in this context bag.

Set<T>(T)

Adds an singleton of `T` in this context bag.

```
public T Set<T>(T value) where T : notnull
```

Parameters

value `T`

The instance of `T` which will be defined in this context bag.

Returns

`T`

Type Parameters

T

The object that will be defined in this context bag.

Method SetValue

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

SetValue(string, object?)

Sets the value associated with the specified key.

```
public void SetValue(string key, object? value)
```

Parameters

key [string](#) ↗

The key of the value to set.

value [object](#) ↗

The value to set.

Method TryGetValue

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

TryGetValue<TResult>(string, out TResult?)

Gets the value associated with the specified key and casts it into `TResult`.

```
public bool TryGetValue<TResult>(string key, out TResult? value)
```

Parameters

key `string` ↗

The key whose to get.

value `TResult`

When this method returns, the value associated with the specified key, if the key is found; otherwise, the default value for the type of the value parameter. This parameter is passed uninitialized.

Returns

`bool` ↗

true if the object is find with the specified key; otherwise, false.

Type Parameters

TResult

The type which will be casted into.

Method Unset

Namespace:[Sisk.Core.Entity](#)

Assembly:Sisk.Core.dll

Unset<T>()

Removes an singleton object from it's type `T`.

```
public bool Unset<T>() where T : notnull
```

Returns

`bool`

Type Parameters

`T`

The singleton type.

Namespace Sisk.Core.Helpers

Classes

[CookieHelper](#)

Provides a class that contains useful methods for working with cookies in HTTP responses.

[HeaderHelper](#)

Provides helper methods for working with HTTP headers.

[MimeHelper](#)

Provides useful helper methods for resolving mime-types from common formats.

[PathHelper](#)

Provides useful path-dedicated helper members.

[SizeHelper](#)

Provides useful size-dedicated helper members.

Class CookieHelper

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

Provides a class that contains useful methods for working with cookies in HTTP responses.

```
public static class CookieHelper
```

Methods

[BuildCookieHeaderValue\(Cookie\)](#)

Builds the cookie header value and returns an string from it.

[BuildCookieHeaderValue\(string, string, DateTime?, TimeSpan?, string?, string?, bool?, bool?, string?\)](#)

Builds the cookie header value and returns an string from it.

Method BuildCookieHeaderValue

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

BuildCookieHeaderValue(Cookie)

Builds the cookie header value and returns an string from it.

```
public static string BuildCookieHeaderValue(Cookie cookie)
```

Parameters

cookie [Cookie](#)

The [Cookie](#) instance to build the cookie string.

Returns

[string](#)

BuildCookieHeaderValue(string, string, DateTime?, TimeSpan?, string?, string?, bool?, bool?, string?)

Builds the cookie header value and returns an string from it.

```
public static string BuildCookieHeaderValue(string name, string value, DateTime? expires = null,  
TimeSpan? maxAge = null, string? domain = null, string? path = null, bool? secure = null, bool?  
httpOnly = null, string? sameSite = null)
```

Parameters

name [string](#)

The cookie name.

value [string](#)

The cookie value.

expires [DateTime](#)?

The cookie expiry date.

maxAge [TimeSpan](#)?

The cookie max duration after being set.

domain [string](#)

The domain where the cookie will be valid.

path [string](#)

The path where the cookie will be valid.

secure [bool](#)?

Determines if the cookie will only be stored in an secure context.

httpOnly [bool](#)?

Determines if the cookie will be only available in the HTTP context.

sameSite [string](#)

The cookie SameSite parameter.

Returns

[string](#)

Class HeaderHelper

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

Provides helper methods for working with HTTP headers.

```
public static class HeaderHelper
```

Methods

[CopyHttpHeaders\(HttpContentHeaders, HttpContentHeaders, bool\)](#)

Copies HTTP headers from one [HttpContentHeaders](#) instance to another.

Method CopyHttpHeaders

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

CopyHttpHeaders([HttpContentHeaders](#), [HttpContentHeaders](#), bool)

Copies HTTP headers from one [HttpContentHeaders](#) instance to another.

```
public static void CopyHttpHeaders(HttpContentHeaders from, HttpContentHeaders to, bool safe = true)
```

Parameters

from [HttpContentHeaders](#)

The source [HttpContentHeaders](#) instance.

to [HttpContentHeaders](#)

The target [HttpContentHeaders](#) instance.

safe [bool](#)

If set to [true](#), headers that are added will be validated (an exception can be thrown if an header is invalid). If [false](#), invalid headers could be discarded, but no exception is thrown.

Class MimeHelper

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

Provides useful helper methods for resolving mime-types from common formats.

```
public static class MimeHelper
```

Properties

[DefaultMimeType](#)

Gets or sets the [MimeHelper](#) default fallback mime-type.

Methods

[GetMimeType\(string, string?\)](#)

Gets the content mime-type from the specified file extension.

[IsBrowserKnownInlineMimeType\(string\)](#)

Determines whether the specified mime-type is considered an inline content type that can be displayed directly in most browsers.

[IsPlainTextFile\(string?\)](#)

Gets an boolean indicating if the specified file is an well-known plain text file.

[IsPlainTextMimeType\(string?\)](#)

Gets a value indicating whether the specified MIME type is a plain text MIME type.

Property DefaultMimeType

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

DefaultMimeType

Gets or sets the [MimeHelper](#) default fallback mime-type.

```
public static string DefaultMimeType { get; set; }
```

Property Value

[string](#)

Remarks

This property is not used by the HTTP server itself, only this helper class.

Method GetMimeType

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

GetMimeType(string, string?)

Gets the content mime-type from the specified file extension.

```
public static string GetMimeType(string fileExtension, string? fallback = null)
```

Parameters

fileExtension [string](#)

The file extension, file path or the extension, with or without the dot.

fallback [string](#)

Optional. The default mime-type when the file best mime-type is not found. If this argument is null, [DefaultMimeType](#) is used.

Returns

[string](#)

The best matched mime-type, or the default if no mime-type was matched with the specified extension.

Method IsBrowserKnownInlineMimeType

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

IsBrowserKnownInlineMimeType(string)

Determines whether the specified mime-type is considered an inline content type that can be displayed directly in most browsers.

```
public static bool IsBrowserKnownInlineMimeType(string mimeType)
```

Parameters

mimeType [string](#)

The mime-type to evaluate.

Returns

[bool](#)

[true](#) if the content type is an inline content type; otherwise, [false](#).

Method IsPlainTextFile

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

IsPlainTextFile(string?)

Gets an boolean indicating if the specified file is an well-known plain text file.

```
public static bool IsPlainTextFile(string? fileExtension)
```

Parameters

fileExtension [string](#)

The file extension, with or without the initial dot.

Returns

[bool](#)

Method IsPlainTextMimeType

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

IsPlainTextMimeType(string?)

Gets a value indicating whether the specified MIME type is a plain text MIME type.

```
public static bool IsPlainTextMimeType(string? mimeType)
```

Parameters

mimeType [string](#)

The MIME type to check.

Returns

[bool](#)

[true](#) if the MIME type is a plain text MIME type; otherwise, [false](#).

Class PathHelper

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

Provides useful path-dedicated helper members.

```
public sealed class PathHelper
```

Constructors

[PathHelper\(\)](#)

Methods

[CombinePaths\(params string\[\]\)](#)

Combines the specified URL paths into one.

[FilesystemCombinePaths\(bool, char, ReadOnlySpan<string>\)](#)

Normalizes and combines the specified file-system paths into one.

[FilesystemCombinePaths\(bool, char, params string\[\]\)](#)

Normalizes and combines the specified file-system paths into one.

[FilesystemCombinePaths\(params string\[\]\)](#)

Normalizes and combines the specified file-system paths into one, using the default environment directory separator char.

[NormalizePath\(string, char\)](#)

Normalize the given path to use the specified directory separator, trim the last separator and remove empty entries.

Constructor PathHelper

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

PathHelper()

```
public PathHelper()
```

Method CombinePaths

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

CombinePaths(params string[])

Combines the specified URL paths into one.

```
public static string CombinePaths(params string[] paths)
```

Parameters

paths string[]

The string array which contains parts that will be combined.

Returns

string

Method FilesystemCombinePaths

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

FilesystemCombinePaths(bool, char, params string[])

Normalizes and combines the specified file-system paths into one.

```
public static string FilesystemCombinePaths(bool allowRelativeReturn, char separator, params  
string[] paths)
```

Parameters

allowRelativeReturn [bool](#)

Specifies if relative paths should be merged and ".." returns should be respected.

separator [char](#)

Specifies the path separator character.

paths [string](#)[]

Specifies the array of paths to combine.

Returns

[string](#)

FilesystemCombinePaths(bool, char, [ReadOnlySpan<string>](#))

Normalizes and combines the specified file-system paths into one.

```
public static string FilesystemCombinePaths(bool allowRelativeReturn, char separator,  
ReadOnlySpan<string> paths)
```

Parameters

allowRelativeReturn [bool](#)

Specifies if relative paths should be merged and ".." returns should be respected.

separator [char](#)

Specifies the path separator character.

paths [ReadOnlySpan<string>](#)

Specifies the array of paths to combine.

Returns

[string](#)

FilesystemCombinePaths(params string[])

Normalizes and combines the specified file-system paths into one, using the default environment directory separator char.

```
public static string FilesystemCombinePaths(params string[] paths)
```

Parameters

paths [string\[\]](#)

Specifies the array of paths to combine.

Returns

[string](#)

Method NormalizePath

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

NormalizePath(string, char)

Normalize the given path to use the specified directory separator, trim the last separator and remove empty entries.

```
public static string NormalizePath(string path, char directorySeparator = '/')
```

Parameters

path [string](#)

The path to normalize.

directorySeparator [char](#)

The directory separator.

Returns

[string](#)

Class SizeHelper

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

Provides useful size-dedicated helper members.

```
public sealed class SizeHelper
```

Constructors

[SizeHelper\(\)](#)

Fields

[UnitEb](#)

Represents the number of bytes in one exbibyte (EiB). This is calculated as 1024 pebibytes.

[UnitGb](#)

Represents the number of bytes in one gibibyte (GiB). This is calculated as 1024 mebibytes.

[UnitKb](#)

Represents the number of bytes in one kibibyte (KiB). This is calculated as 1024 bytes.

[UnitMb](#)

Represents the number of bytes in one mebibyte (MiB). This is calculated as 1024 kibibytes.

[UnitPb](#)

Represents the number of bytes in one pebibyte (PiB). This is calculated as 1024 tebibytes.

[UnitTb](#)

Represents the number of bytes in one tebibyte (TiB). This is calculated as 1024 gibibytes.

Methods

[HumanReadableSize\(double\)](#)

Converts a byte count into a human-readable string representation.

[HumanReadableSize\(long\)](#)

Converts a byte count into a human-readable string representation.

[Parse\(string\)](#)

Parses a human-readable size string (e.g., "10 KB", "2.5 MB") into a long representing the number of bytes.

Constructor SizeHelper

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

SizeHelper()

```
public SizeHelper()
```

Field UnitEb

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

Represents the number of bytes in one exibibyte (EiB). This is calculated as 1024 pebibytes.

```
public const long UnitEb = 1152921504606846976
```

Returns

[long](#)

Represents the number of bytes in one exibibyte (EiB). This is calculated as 1024 pebibytes.

Field UnitGb

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

Represents the number of bytes in one gibibyte (GiB). This is calculated as 1024 mebibytes.

```
public const long UnitGb = 1073741824
```

Returns

[long](#) ↗

Represents the number of bytes in one gibibyte (GiB). This is calculated as 1024 mebibytes.

Field UnitKb

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

Represents the number of bytes in one kibibyte (KiB). This is calculated as 1024 bytes.

```
public const long UnitKb = 1024
```

Returns

[long](#)

Represents the number of bytes in one kibibyte (KiB). This is calculated as 1024 bytes.

Field UnitMb

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

Represents the number of bytes in one mebibyte (MiB). This is calculated as 1024 kibibytes.

```
public const long UnitMb = 1048576
```

Returns

[long](#)

Represents the number of bytes in one mebibyte (MiB). This is calculated as 1024 kibibytes.

Field UnitPb

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

Represents the number of bytes in one pebibyte (PiB). This is calculated as 1024 tebibytes.

```
public const long UnitPb = 1125899906842624
```

Returns

[long](#)

Represents the number of bytes in one pebibyte (PiB). This is calculated as 1024 tebibytes.

Field UnitTb

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

Represents the number of bytes in one tebibyte (TiB). This is calculated as 1024 gibibytes.

```
public const long UnitTb = 1099511627776
```

Returns

[long](#) ↗

Represents the number of bytes in one tebibyte (TiB). This is calculated as 1024 gibibytes.

Method HumanReadableSize

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

HumanReadableSize(long)

Converts a byte count into a human-readable string representation.

```
public static string HumanReadableSize(long byteCount)
```

Parameters

byteCount [long](#)

The total number of bytes to convert.

Returns

[string](#)

A string representing the byte count in a human-readable format.

HumanReadableSize(double)

Converts a byte count into a human-readable string representation.

```
public static string HumanReadableSize(double byteCount)
```

Parameters

byteCount [double](#)

The total number of bytes to convert.

Returns

string ↗

A string representing the byte count in a human-readable format.

Method Parse

Namespace:[Sisk.Core.Helpers](#)

Assembly:Sisk.Core.dll

Parse(string)

Parses a human-readable size string (e.g., "10 KB", "2.5 MB") into a long representing the number of bytes.

```
public static long Parse(string humanReadableSize)
```

Parameters

humanReadableSize [string](#)

The human-readable size string to parse.

Returns

[long](#)

The size in bytes, represented as a long.

Exceptions

[ArgumentNullException](#)

Thrown if `humanReadableSize` is null or whitespace.

[ArgumentException](#)

Thrown if `humanReadableSize` is not in a valid format.

Namespace Sisk.Core.Http

Namespaces

[Sisk.Core.Http.Engine](#)

[Sisk.Core.Http.Handlers](#)

[Sisk.Core.Http.Hosting](#)

[Sisk.Core.Http.Streams](#)

Classes

[BrotliContent](#)

Represents an HTTP content that is compressed using the Brotli algorithm.

[CompressedContent](#)

Represents a base class for HTTP contents served over an compressing stream.

[DefaultMessagePage](#)

Provides methods for creating informative static pages used by Sisk.

[DeflateContent](#)

Represents an HTTP content that is compressed using the Deflate algorithm.

[ForwardingResolver](#)

Provides HTTP forwarding resolving methods that can be used to resolving the client remote address, host and protocol of a proxy, load balancer or CDN, through the HTTP request.

[GZipContent](#)

Represents an HTTP content that is compressed using the GZip algorithm.

[HtmlContent](#)

Provides HTTP content based on HTML contents.

[HttpContext](#)

Represents an context that is shared in a entire HTTP session.

[HttpKnownHeaderNames](#)

Provides most of the most commonly known HTTP headers for constants.

[HttpRequest](#)

Represents an HTTP request received by a Sisk server.

[HttpRequestException](#)

Represents an exception that is thrown while a request is being interpreted by the HTTP server.

[HttpResponse](#)

Represents an HTTP Response.

[HttpResponseExtensions](#)

Provides useful extensions for [HttpResponse](#) objects.

[HttpServer](#)

Provides an lightweight HTTP server powered by Sisk.

[HttpServerConfiguration](#)

Provides execution parameters for an [HttpServer](#).

[HttpServerExecutionResult](#)

Represents the results of an request execution on the HTTP server.

[ListeningHost](#)

Provides a structure to contain the fields needed by an http server host.

[ListeningHostRepository](#)

Represents an fluent repository of [ListeningHost](#) that can add, modify, or remove listening hosts while an [Http Server](#) is running.

[LogStream](#)

Provides a managed, asynchronous log writer which supports writing safe data to log files or text streams.

[PrefixedLogStream](#)

Represents a log stream that prefixes log messages with a custom string.

[RotatingLogPolicy](#)

Provides a managed utility for rotating log files by their file size.

Structs

[HttpStatusInformation](#)

Represents a value that holds an HTTP response status information, with it's status code and description.

[ListeningPort](#)

Provides a structure to contain a listener port for an [ListeningHost](#) instance.

Enums

[HttpServerExecutionStatus](#)

Represents the status of an execution of a request on an [HttpServer](#).

[RequestListenAction](#)

Represents the HTTP server action when receiving an request.

Namespace Sisk.Core.Http.Engine

Classes

[HttpEngineException](#)

Represents an exception that occurred during the execution of the HTTP engine.

[HttpListenerAbstractEngine](#)

Provides an implementation of [HttpServerEngine](#) using [HttpListener](#).

[HttpServerEngine](#)

Provides an abstract base class for HTTP server engines.

[HttpServerEngineContext](#)

Provides an abstract base class for HTTP contexts.

[HttpServerEngineContextRequest](#)

Provides an abstract base class for HTTP requests.

[HttpServerEngineContextResponse](#)

Provides an abstract base class for HTTP responses.

[HttpServerEngineWebSocket](#)

Provides an abstract base class for WebSocket contexts.

Class HttpEngineException

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Represents an exception that occurred during the execution of the HTTP engine.

```
public class HttpEngineException : Exception, ISerializable
```

Implements

[ISerializable](#) ↗

Constructors

[HttpEngineException\(Exception\)](#)

Initializes a new instance of the [HttpEngineException](#) class with a specified inner exception.

[HttpEngineException\(string\)](#)

Initializes a new instance of the [HttpEngineException](#) class with a specified error message.

Constructor HttpEngineException

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

HttpEngineException(string)

Initializes a new instance of the [HttpEngineException](#) class with a specified error message.

```
public HttpEngineException(string message)
```

Parameters

message [string](#)

The message that describes the error.

HttpEngineException(Exception)

Initializes a new instance of the [HttpEngineException](#) class with a specified inner exception.

```
public HttpEngineException(Exception inner)
```

Parameters

inner [Exception](#)

The exception that is the cause of the current exception.

Class HttpListenerAbstractEngine

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:[Sisk.Core.dll](#)

Provides an implementation of [HttpServerEngine](#) using [HttpListener](#).

```
public sealed class HttpListenerAbstractEngine : HttpServerEngine, IDisposable
```

Implements

[IDisposable](#)

Constructors

[HttpListenerAbstractEngine\(\)](#)

Initializes a new instance of the [HttpListenerAbstractEngine](#) class.

Properties

[IdleConnectionTimeout](#)

Gets or sets the timeout for idle connections.

[Shared](#)

Gets the shared instance of the [HttpListenerAbstractEngine](#) class.

Methods

[AddListeningPrefix\(string\)](#)

Adds a listening prefix to the server.

[BeginGetContext\(AsyncCallback?, object?\)](#)

Begins an asynchronous operation to get an HTTP context.

[**ClearPrefixes\(\)**](#)

Clears all listening prefixes from the server.

[**Dispose\(\)**](#)

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

[**EndGetContext\(IAsyncResult\)**](#)

Ends an asynchronous operation to get an HTTP context.

[**StartServer\(\)**](#)

Starts the HTTP server.

[**StopServer\(\)**](#)

Stops the HTTP server.

Constructor HttpListenerAbstractEngine

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

HttpListenerAbstractEngine()

Initializes a new instance of the [HttpListenerAbstractEngine](#) class.

```
public HttpListenerAbstractEngine()
```

Property IdleConnectionTimeout

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

IdleConnectionTimeout

Gets or sets the timeout for idle connections.

```
public override TimeSpan IdleConnectionTimeout { get; set; }
```

Property Value

[TimeSpan](#)

The [TimeSpan](#) representing the idle connection timeout.

Property Shared

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Shared

Gets the shared instance of the [HttpListenerAbstractEngine](#) class.

```
public static HttpListenerAbstractEngine Shared { get; }
```

Property Value

[HttpListenerAbstractEngine](#)

Method AddListeningPrefix

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

AddListeningPrefix(string)

Adds a listening prefix to the server.

```
public override void AddListeningPrefix(string prefix)
```

Parameters

prefix string ↗

The prefix to add.

Method BeginGetContext

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

BeginGetContext(AsyncCallback?, object?)

Begins an asynchronous operation to get an HTTP context.

```
public override IAsyncResult BeginGetContext(AsyncCallback? callback, object? state)
```

Parameters

callback [AsyncCallback](#)

The [AsyncCallback](#) delegate.

state [object](#)

An object that provides state information for the asynchronous operation.

Returns

[IAsyncResult](#)

An [IAsyncResult](#) that references the asynchronous operation.

Method ClearPrefixes

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

ClearPrefixes()

Clears all listening prefixes from the server.

```
public override void ClearPrefixes()
```

Method Dispose

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

```
public override void Dispose()
```

Method EndGetContext

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

EndGetContext(IAsyncResult)

Ends an asynchronous operation to get an HTTP context.

```
public override HttpServerEngineContext EndGetContext(IAsyncResult asyncResult)
```

Parameters

asyncResult [IAsyncResult](#)

The [IAsyncResult](#) that references the pending asynchronous operation.

Returns

[HttpServerEngineContext](#)

An [HttpServerEngineContext](#) representing the HTTP context.

Method StartServer

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

StartServer()

Starts the HTTP server.

```
public override void StartServer()
```

Method StopServer

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

StopServer()

Stops the HTTP server.

```
public override void StopServer()
```

Class HttpServerEngine

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Provides an abstract base class for HTTP server engines.

```
public abstract class HttpServerEngine : IDisposable
```

Implements

[IDisposable](#) ↗

Constructors

[HttpServerEngine\(\)](#)

Properties

[IdleConnectionTimeout](#)

Gets or sets the timeout for idle connections.

Methods

[AddListeningPrefix\(string\)](#)

Adds a listening prefix to the server.

[BeginGetContext\(AsyncCallback?, object?\)](#)

Begins an asynchronous operation to get an HTTP context.

[ClearPrefixes\(\)](#)

Clears all listening prefixes from the server.

[Dispose\(\)](#)

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

[EndGetContext\(IAsyncResult\)](#)

Ends an asynchronous operation to get an HTTP context.

[StartServer\(\)](#)

Starts the HTTP server.

[StopServer\(\)](#)

Stops the HTTP server.

Constructor HttpServerEngine

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

HttpServerEngine()

```
protected HttpServerEngine()
```

Property IdleConnectionTimeout

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

IdleConnectionTimeout

Gets or sets the timeout for idle connections.

```
public abstract TimeSpan IdleConnectionTimeout { get; set; }
```

Property Value

[TimeSpan](#)

The [TimeSpan](#) representing the idle connection timeout.

Method AddListeningPrefix

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

AddListeningPrefix(string)

Adds a listening prefix to the server.

```
public abstract void AddListeningPrefix(string prefix)
```

Parameters

prefix string ↗

The prefix to add.

Method BeginGetContext

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

BeginGetContext(AsyncCallback?, object?)

Begins an asynchronous operation to get an HTTP context.

```
public abstract IAsyncResult BeginGetContext(AsyncCallback? callback, object? state)
```

Parameters

callback [AsyncCallback](#)

The [AsyncCallback](#) delegate.

state [object](#)

An object that provides state information for the asynchronous operation.

Returns

[IAsyncResult](#)

An [IAsyncResult](#) that references the asynchronous operation.

Method ClearPrefixes

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

ClearPrefixes()

Clears all listening prefixes from the server.

```
public abstract void ClearPrefixes()
```

Method Dispose

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

```
public abstract void Dispose()
```

Method EndGetContext

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

EndGetContext(IAsyncResult)

Ends an asynchronous operation to get an HTTP context.

```
public abstract HttpServerEngineContext EndGetContext(IAsyncResult asyncResult)
```

Parameters

asyncResult [IAsyncResult](#)

The [IAsyncResult](#) that references the pending asynchronous operation.

Returns

[HttpServerEngineContext](#)

An [HttpServerEngineContext](#) representing the HTTP context.

Method StartServer

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

StartServer()

Starts the HTTP server.

```
public abstract void StartServer()
```

Method StopServer

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

StopServer()

Stops the HTTP server.

```
public abstract void StopServer()
```

Class HttpServerEngineContext

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Provides an abstract base class for HTTP contexts.

```
public abstract class HttpServerEngineContext
```

Constructors

[HttpServerEngineContext\(\)](#)

Properties

[ContextAbortedToken](#)

Gets a value that indicates whether the HTTP connection has been aborted.

[Request](#)

Gets the HTTP request associated with the context.

[Response](#)

Gets the HTTP response associated with the context.

Methods

[AcceptWebSocketAsync\(string?\)](#)

Accepts a WebSocket connection asynchronously.

Constructor HttpServerEngineContext

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

HttpServerEngineContext()

```
protected HttpServerEngineContext()
```

Property ContextAbortedToken

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

ContextAbortedToken

Gets a value that indicates whether the HTTP connection has been aborted.

```
public virtual CancellationToken ContextAbortedToken { get; }
```

Property Value

[CancellationToken](#)

A [CancellationToken](#) that can be used to signal that the HTTP connection has been aborted.

Property Request

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Request

Gets the HTTP request associated with the context.

```
public abstract HttpServerEngineContextRequest Request { get; }
```

Property Value

[HttpServerEngineContextRequest](#)

The [HttpServerEngineContextRequest](#) representing the HTTP request.

Property Response

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Response

Gets the HTTP response associated with the context.

```
public abstract HttpServerEngineContextResponse Response { get; }
```

Property Value

[HttpServerEngineContextResponse](#)

The [HttpServerEngineContextResponse](#) representing the HTTP response.

Method AcceptWebSocketAsync

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

AcceptWebSocketAsync(string?)

Accepts a WebSocket connection asynchronously.

```
public abstract Task<HttpServerEngineWebSocket> AcceptWebSocketAsync(string? subProtocol)
```

Parameters

subProtocol [string](#)

The subprotocol to use for the WebSocket connection.

Returns

[Task](#)<[HttpServerEngineWebSocket](#)>

A [Task](#)<[TResult](#)> representing the asynchronous operation. The result contains the [HttpServerEngineWeb](#) Socket.

Class HttpServerEngineContextRequest

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Provides an abstract base class for HTTP requests.

```
public abstract class HttpServerEngineContextRequest
```

Constructors

[HttpServerEngineContextRequest\(\)](#)

Properties

[ContentEncoding](#)

Gets the content encoding of the request.

[ContentLength64](#)

Gets the content length of the request.

[Headers](#)

Gets the HTTP headers.

[HttpMethod](#)

Gets the HTTP method of the request.

[InputStream](#)

Gets the input stream of the request.

[IsLocal](#)

Gets a value indicating whether the request is from the local machine.

[IsSecureConnection](#)

Gets a value indicating whether the connection is secure.

[LocalEndPoint](#)

Gets the local endpoint of the request.

[ProtocolVersion](#)

Gets the HTTP protocol version.

[QueryString](#)

Gets the query string collection.

[RawUrl](#)

Gets the raw URL of the request.

[RemoteEndPoint](#)

Gets the remote endpoint of the request.

[RequestTraceIdentifier](#)

Gets the request trace identifier.

[Url](#)

Gets the URL of the request.

[UserHostName](#)

Gets the host name of the user.

Constructor HttpServerEngineContextRequest

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

HttpServerEngineContextRequest()

```
protected HttpServerEngineContextRequest()
```

Property ContentEncoding

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

ContentEncoding

Gets the content encoding of the request.

```
public abstract Encoding ContentEncoding { get; }
```

Property Value

[Encoding](#)

The [Encoding](#) representing the content encoding.

Property ContentLength64

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

ContentLength64

Gets the content length of the request.

```
public abstract long ContentLength64 { get; }
```

Property Value

[long](#)

The content length of the request.

Property Headers

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Headers

Gets the HTTP headers.

```
public abstract NameValueCollection Headers { get; }
```

Property Value

[NameValueCollection](#)

The [WebHeaderCollection](#) containing the HTTP headers.

Property HttpMethod

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

HttpMethod

Gets the HTTP method of the request.

```
public abstract string HttpMethod { get; }
```

Property Value

[string](#)

The HTTP method of the request.

Property InputStream

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

InputStream

Gets the input stream of the request.

```
public abstract Stream InputStream { get; }
```

Property Value

[Stream](#)

The [Stream](#) representing the input stream.

Property IsLocal

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

IsLocal

Gets a value indicating whether the request is from the local machine.

```
public abstract bool IsLocal { get; }
```

Property Value

[bool](#)

[true](#) if the request is from the local machine; otherwise, [false](#).

Property IsSecureConnection

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

IsSecureConnection

Gets a value indicating whether the connection is secure.

```
public abstract bool IsSecureConnection { get; }
```

Property Value

`bool`

`true` if the connection is secure; otherwise, `false`.

Property LocalEndPoint

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

LocalEndPoint

Gets the local endpoint of the request.

```
public abstract IPPEndPoint LocalEndPoint { get; }
```

Property Value

[IPPEndPoint](#)

The [IPPEndPoint](#) representing the local endpoint.

Property ProtocolVersion

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

ProtocolVersion

Gets the HTTP protocol version.

```
public abstract Version ProtocolVersion { get; }
```

Property Value

[Version](#)

The [Version](#) representing the HTTP protocol version.

Property QueryString

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

QueryString

Gets the query string collection.

```
public abstract NameValueCollection QueryString { get; }
```

Property Value

[NameValueCollection](#)

The [NameValueCollection](#) containing the query string parameters.

Property RawUrl

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

RawUrl

Gets the raw URL of the request.

```
public abstract string? RawUrl { get; }
```

Property Value

[string](#)

The raw URL of the request.

Property RemoteEndPoint

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

RemoteEndPoint

Gets the remote endpoint of the request.

```
public abstract IPEndPoint RemoteEndPoint { get; }
```

Property Value

[IPEndPoint](#)

The [IPEndPoint](#) representing the remote endpoint.

Property RequestTraceIdentifier

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

RequestTraceIdentifier

Gets the request trace identifier.

```
public abstract Guid RequestTraceIdentifier { get; }
```

Property Value

[Guid](#)

The [Guid](#) representing the request trace identifier.

Property Url

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Url

Gets the URL of the request.

```
public abstract Uri? Url { get; }
```

Property Value

Uri

The System.Uri representing the URL of the request.

Property UserHostName

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

UserHostName

Gets the host name of the user.

```
public abstract string UserHostName { get; }
```

Property Value

[string](#)

The host name of the user.

Class HttpServerEngineContextResponse

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:[Sisk.Core.dll](#)

Provides an abstract base class for HTTP responses.

```
public abstract class HttpServerEngineContextResponse : IDisposable
```

Implements

[IDisposable](#) ↗

Constructors

[HttpServerEngineContextResponse\(\)](#)

Properties

[ContentLength64](#)

Gets or sets the content length of the response.

[ContentType](#)

Gets or sets the content type of the response.

[Headers](#)

Gets or sets the HTTP headers.

[KeepAlive](#)

Gets or sets a value indicating whether the connection should be kept alive.

[OutputStream](#)

Gets the output stream of the response.

[SendChunked](#)

Gets or sets a value indicating whether chunked transfer encoding is used.

[StatusCode](#)

Gets or sets the HTTP status code.

[StatusDescription](#)

Gets or sets the status description.

Methods

[Abort\(\)](#)

Aborts the response.

[AppendHeader\(string, string\)](#)

Appends a header to the response.

[Close\(\)](#)

Closes the response.

[Dispose\(\)](#)

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

Constructor HttpServerEngineContextResponse

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

HttpServerEngineContextResponse()

```
protected HttpServerEngineContextResponse()
```

Property ContentLength64

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

ContentLength64

Gets or sets the content length of the response.

```
public abstract long ContentLength64 { get; set; }
```

Property Value

[long](#)

The content length of the response.

Property ContentType

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

ContentType

Gets or sets the content type of the response.

```
public abstract string? ContentType { get; set; }
```

Property Value

[string](#)

The content type of the response.

Property Headers

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Headers

Gets or sets the HTTP headers.

```
public abstract WebHeaderCollection Headers { get; set; }
```

Property Value

[WebHeaderCollection](#)

The [WebHeaderCollection](#) containing the HTTP headers.

Property KeepAlive

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

KeepAlive

Gets or sets a value indicating whether the connection should be kept alive.

```
public abstract bool KeepAlive { get; set; }
```

Property Value

[bool](#)

[true](#) if the connection should be kept alive; otherwise, [false](#).

Property OutputStream

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

OutputStream

Gets the output stream of the response.

```
public abstract Stream OutputStream { get; }
```

Property Value

[Stream](#)

The [Stream](#) representing the output stream.

Property SendChunked

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

SendChunked

Gets or sets a value indicating whether chunked transfer encoding is used.

```
public abstract bool SendChunked { get; set; }
```

Property Value

[bool](#)

[true](#) if chunked transfer encoding is used; otherwise, [false](#).

Property StatusCode

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

StatusCode

Gets or sets the HTTP status code.

```
public abstract int StatusCode { get; set; }
```

Property Value

[int](#)

The HTTP status code.

Property StatusDescription

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

StatusDescription

Gets or sets the status description.

```
public abstract string StatusDescription { get; set; }
```

Property Value

[string](#)

The status description.

Method Abort

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Abort()

Aborts the response.

```
public abstract void Abort()
```

Method AppendHeader

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

AppendHeader(string, string)

Appends a header to the response.

```
public abstract void AppendHeader(string name, string value)
```

Parameters

name [string](#)

The name of the header.

value [string](#)

The value of the header.

Method Close

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Close()

Closes the response.

```
public abstract void Close()
```

Method Dispose

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

```
public abstract void Dispose()
```

Class HttpServerEngineWebSocket

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:[Sisk.Core.dll](#)

Provides an abstract base class for WebSocket contexts.

```
public abstract class HttpServerEngineWebSocket
```

Constructors

[HttpServerEngineWebSocket\(\)](#)

Properties

State

Gets the state of the WebSocket.

Methods

[CloseAsync\(WebSocketCloseStatus, string?, CancellationToken\)](#)

Closes the WebSocket connection asynchronously.

[CloseOutputAsync\(WebSocketCloseStatus, string?, CancellationToken\)](#)

Closes the output stream of the WebSocket asynchronously.

[ReceiveAsync\(Memory<byte>, CancellationToken\)](#)

Receives data from the WebSocket asynchronously.

[SendAsync\(ReadOnlyMemory<byte>, WebSocketMessageType, bool, CancellationToken\)](#)

Sends data over the WebSocket asynchronously.

Constructor HttpServerEngineWebSocket

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

HttpServerEngineWebSocket()

```
protected HttpServerEngineWebSocket()
```

Property State

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

State

Gets the state of the WebSocket.

```
public abstract WebSocketState State { get; }
```

Property Value

[WebSocketState](#)

The [WebSocketState](#) representing the state of the WebSocket.

Method CloseAsync

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

CloseAsync(WebSocketCloseStatus, string?, CancellationToken)

Closes the WebSocket connection asynchronously.

```
public abstract Task CloseAsync(WebSocketCloseStatus closeStatus, string? reason,  
CancellationToken cancellation)
```

Parameters

closeStatus [WebSocketCloseStatus](#)

The status code for closing the WebSocket.

reason [string](#)

The reason for closing the WebSocket.

cancellation [CancellationToken](#)

The cancellation token.

Returns

[Task](#)

A [Task](#) representing the asynchronous operation.

Method CloseOutputAsync

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

`CloseOutputAsync(WebSocketCloseStatus, string?, CancellationToken)`

Closes the output stream of the WebSocket asynchronously.

```
public abstract Task CloseOutputAsync(WebSocketCloseStatus closeStatus, string? reason,  
CancellationToken cancellation)
```

Parameters

closeStatus [WebSocketCloseStatus](#)

The status code for closing the WebSocket.

reason [string](#)

The reason for closing the WebSocket.

cancellation [CancellationToken](#)

The cancellation token.

Returns

[Task](#)

A [Task](#) representing the asynchronous operation.

Method ReceiveAsync

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

ReceiveAsync(Memory<byte>, CancellationToken)

Receives data from the WebSocket asynchronously.

```
public abstract ValueTask<ValueWebSocketReceiveResult> ReceiveAsync(Memory<byte> buffer,  
CancellationToken cancellationToken)
```

Parameters

buffer [Memory<byte>](#)

The buffer to receive the data into.

cancellationToken [CancellationToken](#)

The cancellation token.

Returns

[ValueTask<ValueWebSocketReceiveResult>](#)

A [ValueTask<TResult>](#) representing the asynchronous operation. The result contains the [ValueWebSocketReceiveResult](#).

Method SendAsync

Namespace:[Sisk.Core.Http.Engine](#)

Assembly:Sisk.Core.dll

SendAsync(ReadOnlyMemory<byte>, WebSocketMessageType, bool, CancellationToken)

Sends data over the WebSocket asynchronously.

```
public abstract ValueTask SendAsync(ReadOnlyMemory<byte> buffer, WebSocketMessageType  
messageType, bool endOfMessage, CancellationToken cancellationToken)
```

Parameters

buffer [ReadOnlyMemory<byte>](#)

The buffer containing the data to send.

messageType [WebSocketMessageType](#)

The type of message to send.

endOfMessage [bool](#)

A value indicating whether this is the end of the message.

cancellationToken [CancellationToken](#)

The cancellation token.

Returns

[ValueTask](#)

A [ValueTask](#) representing the asynchronous operation.

Namespace Sisk.Core.Http.Handlers

Classes

[AsyncHttpServerHandler](#)

Represents an asynchronous event handler for the [HttpServer](#), router, and related events.

[HttpServerHandler](#)

Represents an event handler for the [HttpServer](#), router, and related events.

Class AsyncHttpServerHandler

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

Represents an asynchronous event handler for the [HttpServer](#), router, and related events.

```
public abstract class AsyncHttpServerHandler : HttpServerHandler
```

Constructors

[AsyncHttpServerHandler\(\)](#)

Methods

[OnContextBagCreated\(TypedValueDictionary\)](#)

Event that is called when an HTTP context is created within an [HttpRequest](#) object.

[OnContextBagCreatedAsync\(TypedValueDictionary\)](#)

Method that is called when an HTTP context is created within an [HttpRequest](#) object.

[OnException\(Exception\)](#)

Event that is called when an exception is caught in the HTTP server. This method is called regardless of whether [ThrowExceptions](#) is enabled or not.

[OnExceptionAsync\(Exception\)](#)

Method that is called when an exception is caught in the HTTP server. This method is called regardless of whether [ThrowExceptions](#) is enabled or not.

[OnHttpRequestClose\(HttpServerExecutionResult\)](#)

Event that is called when an [HttpRequest](#) is closed in the HTTP server.

[OnHttpRequestCloseAsync\(HttpServerExecutionResult\)](#)

Method that is called when an [HttpRequest](#) is closed in the HTTP server.

[OnHttpRequestOpen\(HttpRequest\)](#)

Event that is called when an [HttpRequest](#) is received in the HTTP server.

[OnHttpRequestOpenAsync\(HttpRequest\)](#)

Method that is called when an [HttpRequest](#) is received in the HTTP server.

[OnServerStarted\(HttpServer\)](#)

Event that is called immediately after starting the [HttpServer](#), when it's ready and listening.

[OnServerStartedAsync\(HttpServer\)](#)

Method that is called immediately after starting the [HttpServer](#), when it's ready and listening.

[OnServerStarting\(HttpServer\)](#)

Event that is called immediately before starting the [HttpServer](#).

[OnServerStartingAsync\(HttpServer\)](#)

Method that is called immediately before starting the [HttpServer](#).

[OnServerStopped\(HttpServer\)](#)

Event that is called after the [HttpServer](#) is stopped, meaning it has stopped from listening to requests.

[OnServerStoppedAsync\(HttpServer\)](#)

Method that is called after the [HttpServer](#) is stopped, meaning it has stopped from listening to requests.

[OnServerStopping\(HttpServer\)](#)

Event that is called before the [HttpServer](#) stop, when it is stopping from listening requests.

[OnServerStoppingAsync\(HttpServer\)](#)

Method that is called before the [HttpServer](#) stop, when it is stopping from listening requests.

[OnSetupRouter\(Router\)](#)

Event that is called when an [Router](#) is binded to the HTTP server.

[OnSetupRouterAsync\(Router\)](#)

Method that is called when an [Router](#) is binded to the HTTP server.

Constructor AsyncHttpServerHandler

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

AsyncHttpServerHandler()

```
protected AsyncHttpServerHandler()
```

Method OnContextBagCreated

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnContextBagCreated(TypedValueDictionary)

Event that is called when an HTTP context is created within an [HttpRequest](#) object.

```
protected override sealed void OnContextBagCreated(TypedValueDictionary contextBag)
```

Parameters

contextBag [TypedValueDictionary](#)

The creating context bag.

Method OnContextBagCreatedAsync

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnContextBagCreatedAsync(TypedValueDictionary)

Method that is called when an HTTP context is created within an [HttpRequest](#) object.

```
protected virtual Task OnContextBagCreatedAsync(TypedValueDictionary contextBag)
```

Parameters

contextBag [TypedValueDictionary](#)

The creating context bag.

Returns

[Task](#) ↗

Method OnException

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnException(Exception)

Event that is called when an exception is caught in the HTTP server. This method is called regardless of whether [ThrowExceptions](#) is enabled or not.

```
protected override sealed void OnException(Exception exception)
```

Parameters

exception [Exception](#)

The exception object.

Method OnExceptionAsync

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnExceptionAsync(Exception)

Method that is called when an exception is caught in the HTTP server. This method is called regardless of whether [ThrowExceptions](#) is enabled or not.

```
protected virtual Task OnExceptionAsync(Exception exception)
```

Parameters

exception [Exception](#)

The exception object.

Returns

[Task](#)

Method OnHttpRequestClose

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnHttpRequestClose(HttpServerExecutionResult)

Event that is called when an [HttpRequest](#) is closed in the HTTP server.

```
protected override sealed void OnHttpRequestClose(HttpServerExecutionResult result)
```

Parameters

result [HttpServerExecutionResult](#)

The result of the execution of the request.

Method OnHttpRequestCloseAsync

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnHttpRequestCloseAsync(HttpServerExecutionResult)

Method that is called when an [HttpRequest](#) is closed in the HTTP server.

```
protected virtual Task OnHttpRequestCloseAsync(HttpServerExecutionResult result)
```

Parameters

result [HttpServerExecutionResult](#)

The result of the execution of the request.

Returns

[Task](#) ↗

Method OnHttpRequestOpen

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnHttpRequestOpen(HttpRequest)

Event that is called when an [HttpRequest](#) is received in the HTTP server.

```
protected override sealed void OnHttpRequestOpen(HttpRequest request)
```

Parameters

request [HttpRequest](#)

The connecting HTTP request entity.

Method OnHttpRequestOpenAsync

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnHttpRequestOpenAsync(HttpRequest)

Method that is called when an [HttpRequest](#) is received in the HTTP server.

```
protected virtual Task OnHttpRequestOpenAsync(HttpRequest request)
```

Parameters

request [HttpRequest](#)

The connecting HTTP request entity.

Returns

[Task](#) ↗

Method OnServerStarted

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnServerStarted(HttpServer)

Event that is called immediately after starting the [HttpServer](#), when it's ready and listening.

```
protected override sealed void OnServerStarted(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which is ready.

Method OnServerStartedAsync

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnServerStartedAsync(HttpServer)

Method that is called immediately after starting the [HttpServer](#), when it's ready and listening.

```
protected virtual Task OnServerStartedAsync(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which is ready.

Returns

[Task](#) ↗

Method OnServerStarting

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnServerStarting(HttpServer)

Event that is called immediately before starting the [HttpServer](#).

```
protected override sealed void OnServerStarting(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which is starting.

Method OnServerStartingAsync

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnServerStartingAsync(HttpServer)

Method that is called immediately before starting the [HttpServer](#).

```
protected virtual Task OnServerStartingAsync(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which is starting.

Returns

[Task](#) ↗

Method OnServerStopped

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnServerStopped(HttpServer)

Event that is called after the [HttpServer](#) is stopped, meaning it has stopped from listening to requests.

```
protected override sealed void OnServerStopped(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which has stopped.

Method OnServerStoppedAsync

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnServerStoppedAsync(HttpServer)

Method that is called after the [HttpServer](#) is stopped, meaning it has stopped from listening to requests.

```
protected virtual Task OnServerStoppedAsync(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which has stopped.

Returns

[Task](#) ↗

Method OnServerStopping

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnServerStopping(HttpServer)

Event that is called before the [HttpServer](#) stop, when it is stopping from listening requests.

```
protected override sealed void OnServerStopping(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which is stopping.

Method OnServerStoppingAsync

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnServerStoppingAsync(HttpServer)

Method that is called before the [HttpServer](#) stop, when it is stopping from listening requests.

```
protected virtual Task OnServerStoppingAsync(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which is stopping.

Returns

[Task](#) ↗

Method OnSetupRouter

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnSetupRouter(Router)

Event that is called when an [Router](#) is binded to the HTTP server.

```
protected override sealed void OnSetupRouter(Router router)
```

Parameters

router Router

The router entity which is binded.

Method OnSetupRouterAsync

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnSetupRouterAsync(Router)

Method that is called when an [Router](#) is binded to the HTTP server.

```
protected virtual Task OnSetupRouterAsync(Router router)
```

Parameters

router Router

The router entity which is binded.

Returns

[Task](#) ↗

Class HttpServerHandler

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:[Sisk.Core.dll](#)

Represents an event handler for the [HttpServer](#), router, and related events.

```
public abstract class HttpServerHandler
```

Constructors

[HttpServerHandler\(\)](#)

Methods

[OnContextBagCreated\(TypedValueDictionary\)](#)

Event that is called when an HTTP context is created within an [HttpRequest](#) object.

[OnException\(Exception\)](#)

Event that is called when an exception is caught in the HTTP server. This method is called regardless of whether [ThrowExceptions](#) is enabled or not.

[OnHttpRequestClose\(HttpServerExecutionResult\)](#)

Event that is called when an [HttpRequest](#) is closed in the HTTP server.

[OnHttpRequestOpen\(HttpRequest\)](#)

Event that is called when an [HttpRequest](#) is received in the HTTP server.

[OnServerStarted\(HttpServer\)](#)

Event that is called immediately after starting the [HttpServer](#), when it's ready and listening.

[OnServerStarting\(HttpServer\)](#)

Event that is called immediately before starting the [HttpServer](#).

[OnServerStopped\(HttpServer\)](#)

Event that is called after the [HttpServer](#) is stopped, meaning it has stopped from listening to requests.

[OnServerStopping\(HttpServer\)](#)

Event that is called before the [HttpServer](#) stop, when it is stopping from listening requests.

[OnSetupRouter\(Router\)](#)

Event that is called when an [Router](#) is binded to the HTTP server.

Constructor HttpServerHandler

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

HttpServerHandler()

```
protected HttpServerHandler()
```

Method OnContextBagCreated

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnContextBagCreated(TypedValueDictionary)

Event that is called when an HTTP context is created within an [HttpRequest](#) object.

```
protected virtual void OnContextBagCreated(TypedValueDictionary contextBag)
```

Parameters

contextBag [TypedValueDictionary](#)

The creating context bag.

Method OnException

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnException(Exception)

Event that is called when an exception is caught in the HTTP server. This method is called regardless of whether [ThrowExceptions](#) is enabled or not.

```
protected virtual void OnException(Exception exception)
```

Parameters

exception [Exception](#)

The exception object.

Method OnHttpRequestClose

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnHttpRequestClose(HttpServerExecutionResult)

Event that is called when an [HttpRequest](#) is closed in the HTTP server.

```
protected virtual void OnHttpRequestClose(HttpServerExecutionResult result)
```

Parameters

result [HttpServerExecutionResult](#)

The result of the execution of the request.

Method OnHttpRequestOpen

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnHttpRequestOpen(HttpRequest)

Event that is called when an [HttpRequest](#) is received in the HTTP server.

```
protected virtual void OnHttpRequestOpen(HttpRequest request)
```

Parameters

request [HttpRequest](#)

The connecting HTTP request entity.

Method OnServerStarted

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnServerStarted(HttpServer)

Event that is called immediately after starting the [HttpServer](#), when it's ready and listening.

```
protected virtual void OnServerStarted(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which is ready.

Method OnServerStarting

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnServerStarting(HttpServer)

Event that is called immediately before starting the [HttpServer](#).

```
protected virtual void OnServerStarting(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which is starting.

Method OnServerStopped

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnServerStopped(HttpServer)

Event that is called after the [HttpServer](#) is stopped, meaning it has stopped from listening to requests.

```
protected virtual void OnServerStopped(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which has stopped.

Method OnServerStopping

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnServerStopping(HttpServer)

Event that is called before the [HttpServer](#) stop, when it is stopping from listening requests.

```
protected virtual void OnServerStopping(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which is stopping.

Method OnSetupRouter

Namespace:[Sisk.Core.Http.Handlers](#)

Assembly:Sisk.Core.dll

OnSetupRouter(Router)

Event that is called when an [Router](#) is binded to the HTTP server.

```
protected virtual void OnSetupRouter(Router router)
```

Parameters

router Router

The router entity which is binded.

Namespace Sisk.Core.Http.Hosting

Classes

[ConfigurationContext](#)

Represents a reading context for a portable configuration file.

[HttpServerHostContext](#)

Represents the class that hosts most of the components needed to run a Sisk application.

[HttpServerHostContextBuilder](#)

Represents a context constructor for [HttpServerHostContext](#).

[InitializationParameterCollection](#)

Provides a collection of HTTP server initialization variables.

[PortableConfigurationBuilder](#)

Represents the portable configuration builder for [HttpServerHostContextBuilder](#).

Interfaces

[IConfigurationReader](#)

Represents an interface that reads and applies settings from a settings file.

Enums

[ConfigurationFileLookupDirectory](#)

Represents the base directory where the [IConfigurationReader](#) should search for the configuration file.

[HttpServerHostContextBuilderExceptionMode](#)

Represents how the builder event error message should be displayed.

[InitializationParameterCollection.GetValueOption](#)

Represents the option used in the method `GetValueOrThrow(string, GetValueOption)`.

Class ConfigurationContext

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Represents a reading context for a portable configuration file.

```
public sealed class ConfigurationContext
```

Properties

[ConfigurationFile](#)

Gets the absolute path to the configuration file. The file is guaranteed to exist when getting this property value.

[Host](#)

Gets the [HttpServerHostContext](#) which are configuring this context.

[Parameters](#)

Gets the [InitializationParameterCollection](#) collection for defining configuration parameters of the host application.

[TargetListeningHost](#)

Gets the target [ListeningHost](#) which are configuring this context.

Property ConfigurationFile

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

ConfigurationFile

Gets the absolute path to the configuration file. The file is guaranteed to exist when getting this property value.

```
public string ConfigurationFile { get; }
```

Property Value

[string](#)

Property Host

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Host

Gets the [HttpServerHostContext](#) which are configuring this context.

```
public HttpServerHostContext Host { get; }
```

Property Value

[HttpServerHostContext](#)

Property Parameters

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Parameters

Gets the [InitializationParameterCollection](#) collection for defining configuration parameters of the host application.

```
public InitializationParameterCollection Parameters { get; }
```

Property Value

[InitializationParameterCollection](#)

Property TargetListeningHost

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

TargetListeningHost

Gets the target [ListeningHost](#) which are configuring this context.

```
public ListeningHost TargetListeningHost { get; }
```

Property Value

[ListeningHost](#)

Enum ConfigurationFileLookupDirectory

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Represents the base directory where the [IConfigurationReader](#) should search for the configuration file.

```
[Flags]  
public enum ConfigurationFileLookupDirectory
```

Fields

All = CurrentDirectory | AppDirectory

Represents all possible directories.

AppDirectory = 4

The [IConfigurationReader](#) should search in the executable base directory.

CurrentDirectory = 2

The [IConfigurationReader](#) should search in the process current/base directory.

Class HttpServerHostContext

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Represents the class that hosts most of the components needed to run a Sisk application.

```
public sealed class HttpServerHostContext : IDisposable
```

Implements

[IDisposable](#) ↗

Properties

[AccessLogs](#)

Gets the configured access log stream. This property is inherited from [ServerConfiguration](#).

[CrossOriginResourceSharingPolicy](#)

Gets the host [CrossOriginResourceSharingPolicy](#).

[ErrorLogs](#)

Gets the configured error log stream. This property is inherited from [ServerConfiguration](#).

[HttpServer](#)

Gets the host HTTP server.

[Parameters](#)

Gets the initialization parameters from the portable configuration file.

[Router](#)

Gets the host router.

[ServerConfiguration](#)

Gets the host server configuration.

Methods

[Dispose\(\)](#)

Invalidates this class and releases the resources used by it, and permanently closes the HTTP server.

[Start\(bool, bool\)](#)

Starts the HTTP server.

[StartAsync\(bool, bool\)](#)

Asynchronously starts the HTTP server.

Property AccessLogs

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

AccessLogs

Gets the configured access log stream. This property is inherited from [ServerConfiguration](#).

```
public LogStream? AccessLogs { get; }
```

Property Value

[LogStream](#)

Property CrossOriginResourceSharingPolicy

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

CrossOriginResourceSharingPolicy

Gets the host [CrossOriginResourceSharingPolicy](#).

```
public CrossOriginResourceSharingHeaders CrossOriginResourceSharingPolicy { get; set; }
```

Property Value

[CrossOriginResourceSharingHeaders](#)

Property ErrorLogs

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

ErrorLogs

Gets the configured error log stream. This property is inherited from [ServerConfiguration](#).

```
public LogStream? ErrorLogs { get; }
```

Property Value

[LogStream](#)

Property HttpServer

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

HttpServer

Gets the host HTTP server.

```
public HttpServer HttpServer { get; }
```

Property Value

[HttpServer](#)

Property Parameters

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Parameters

Gets the initialization parameters from the portable configuration file.

```
public InitializationParameterCollection Parameters { get; }
```

Property Value

[InitializationParameterCollection](#)

Property Router

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Router

Gets the host router.

```
public Router Router { get; set; }
```

Property Value

[Router](#)

Property ServerConfiguration

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

ServerConfiguration

Gets the host server configuration.

```
public HttpServerConfiguration ServerConfiguration { get; }
```

Property Value

[HttpServerConfiguration](#)

Method Dispose

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Dispose()

Invalidates this class and releases the resources used by it, and permanently closes the HTTP server.

```
public void Dispose()
```

Method Start

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Start(bool, bool)

Starts the HTTP server.

```
public void Start(bool verbose = true, bool preventHault = true)
```

Parameters

verbose [bool](#)

Optional. Specifies if the application should write the listening prefix welcome message.

preventHault [bool](#)

Optional. Specifies if the application should pause the main application loop.

Method StartAsync

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

StartAsync(bool, bool)

Asynchronously starts the HTTP server.

```
public Task StartAsync(bool verbose = true, bool preventHault = true)
```

Parameters

verbose [bool](#)

Optional. Specifies if the application should write the listening prefix welcome message.

preventHault [bool](#)

Optional. Specifies if the application should pause the main application loop.

Returns

[Task](#)

Class HttpServerHostContextBuilder

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Represents a context constructor for [HttpServerHostContext](#).

```
public sealed class HttpServerHostContextBuilder
```

Properties

[CatchConfigurationExceptions](#)

Defines how the constructor should capture errors thrown within

[UsePortableConfiguration\(Action<PortableConfigurationBuilder>\)](#) and display in the Console.

[ServerConfiguration](#)

Gets the Server Configuration object.

Methods

[Build\(\)](#)

Builds an [HttpServerHostContext](#) with the specified parameters.

[UseAutoScan<TModule>\(\)](#)

This method is an shortcut for calling [AutoScanModules<TModule>\(\)](#).

[UseAutoScan<TModule>\(Assembly\)](#)

This method is an shortcut for calling [AutoScanModules<TModule>\(\)](#).

[UseBootstraper\(Action\)](#)

Adds an function that will be executed immediately before starting the HTTP server.

[UseBootstraper\(Func<Task>\)](#)

Adds an asynchronous function that will be executed immediately before starting the HTTP server.

[UseBootstraper\(string, Action\)](#)

Adds an function that will be executed immediately before starting the HTTP server.

[UseBootstraper\(string, Func<Task>\)](#)

Adds an asynchronous function that will be executed immediately before starting the HTTP server.

[UseConfiguration\(Action<HttpServerConfiguration>\)](#)

Calls an action that has the HTTP server configuration as an argument.

[UseCors\(CrossOriginResourceSharingHeaders\)](#)

Sets an [CrossOriginResourceSharingHeaders](#) instance in the current listening host.

[UseCors\(Action<CrossOriginResourceSharingHeaders>\)](#)

Calls an action that has an [CrossOriginResourceSharingHeaders](#) instance from the main listening host as an argument.

[UseEngine\(HttpServerEngine\)](#)

Sets the HTTP server engine.

[UseEngine<TEngine>\(\)](#)

Sets the HTTP server engine using a default constructor.

[UseForwardingResolver\(ForwardingResolver\)](#)

This method is a shortcut for setting [ForwardingResolver](#).

[UseForwardingResolver<TForwardingResolver>\(\)](#)

This method is a shortcut for setting [ForwardingResolver](#).

[UseHandler\(HttpServerHandler\)](#)

This method is an shortcut for calling [RegisterHandler\(HttpServerHandler\)](#).

[UseHandler<THandler>\(\)](#)

This method is an shortcut for calling [RegisterHandler<T>\(\)](#).

[UseHttpServer\(Action<HttpServer>\)](#)

Calls an action that has the HTTP server instance as an argument.

[UseListeningPort\(ListeningPort\)](#)

Sets the main [ListeningPort](#) of this host builder.

[UseListeningPort\(string\)](#)

Sets the main [ListeningPort](#) of this host builder.

[UseListeningPort\(ushort\)](#)

Sets the main [ListeningPort](#) of this host builder.

[UseLocale\(CultureInfo\)](#)

Changes the default thread current culture through [DefaultThreadCurrentCulture](#).

[UsePortableConfiguration\(Action<PortableConfigurationBuilder>\)](#)

Enables the portable configuration for this application, which imports settings, parameters, and other information from a JSON settings file.

[UseRouter\(Router\)](#)

Sets an [Router](#) instance in the current listening host.

[UseRouter\(Action<Router>\)](#)

Calls an action that has an [Router](#) instance from the host HTTP server.

[UseStartupMessage\(Func<string>\)](#)

Adds a function that returns an optional initialization message to the [HttpServerHostContext](#) output verbose.

[UseStartupMessage\(string\)](#)

Add an optional message to the [HttpServerHostContext](#) output verbose.

Property CatchConfigurationExceptions

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

CatchConfigurationExceptions

Defines how the constructor should capture errors thrown within

[UsePortableConfiguration\(Action<PortableConfigurationBuilder>\)](#) and display in the Console.

```
public static HttpServerHostContextBuilderExceptionMode CatchConfigurationExceptions { get;  
    set; }
```

Property Value

[HttpServerHostContextBuilderExceptionMode](#)

Property ServerConfiguration

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

ServerConfiguration

Gets the Server Configuration object.

```
public HttpServerConfiguration ServerConfiguration { get; }
```

Property Value

[HttpServerConfiguration](#)

Method Build

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Build()

Builds an [HttpServerHostContext](#) with the specified parameters.

```
public HttpServerHostContext Build()
```

Returns

[HttpServerHostContext](#)

Method UseAutoScan

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

UseAutoScan<TModule>()

This method is an shortcut for calling [AutoScanModules<TModule>\(\)](#).

```
public HttpServerHostContextBuilder UseAutoScan<TModule>() where TModule : RouterModule
```

Returns

[HttpServerHostContextBuilder](#)

Type Parameters

TModule

An class which implements [RouterModule](#), or the router module itself.

UseAutoScan<TModule>(Assembly)

This method is an shortcut for calling [AutoScanModules<TModule>\(\)](#).

```
public HttpServerHostContextBuilder UseAutoScan<TModule>(Assembly t) where TModule : RouterModule
```

Parameters

t [Assembly](#)

The assembly where the scanning types are.

Returns

[HttpServerHostContextBuilder](#)

Type Parameters

TModule

An class which implements [RouterModule](#), or the router module itself.

Method UseBootstraper

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

UseBootstraper(Action)

Adds an function that will be executed immediately before starting the HTTP server.

```
public HttpServerHostContextBuilder UseBootstraper(Action bootstrapAction)
```

Parameters

bootstrapAction [Action](#)

The action which will be executed before the HTTP server start.

Returns

[HttpServerHostContextBuilder](#)

UseBootstraper(string, Action)

Adds an function that will be executed immediately before starting the HTTP server.

```
public HttpServerHostContextBuilder UseBootstraper(string name, Action bootstrapAction)
```

Parameters

name [string](#)

Defines an custom label for the bootstraping action name.

bootstrapAction [Action](#)

The action which will be executed before the HTTP server start.

Returns

[HttpServerHostContextBuilder](#)

UseBootstraper(Func<Task>)

Adds an asynchronous function that will be executed immediately before starting the HTTP server.

```
public HttpServerHostContextBuilder UseBootstraper(Func<Task> asyncBootstrapAction)
```

Parameters

asyncBootstrapAction [Func<Task>](#)

The asynchronous action which will be executed before the HTTP server start.

Returns

[HttpServerHostContextBuilder](#)

UseBootstraper(string, Func<Task>)

Adds an asynchronous function that will be executed immediately before starting the HTTP server.

```
public HttpServerHostContextBuilder UseBootstraper(string name, Func<Task> asyncBootstrapAction)
```

Parameters

name [string](#)

Defines an custom label for the bootstraping action name.

asyncBootstrapAction [Func<Task>](#)

The asynchronous action which will be executed before the HTTP server start.

Returns

[HttpServerHostContextBuilder](#)

The [HttpServerHostContextBuilder](#) instance.

Method UseConfiguration

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

UseConfiguration(Action<HttpServerConfiguration>)

Calls an action that has the HTTP server configuration as an argument.

```
public HttpServerHostContextBuilder UseConfiguration(Action<HttpServerConfiguration> handler)
```

Parameters

handler [Action](#)<HttpServerConfiguration>

An action where the first argument is an [HttpServerConfiguration](#).

Returns

[HttpServerHostContextBuilder](#)

Method UseCors

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

UseCors(Action<CrossOriginResourceSharingHeaders>)

Calls an action that has an [CrossOriginResourceSharingHeaders](#) instance from the main listening host as an argument.

```
public HttpServerHostContextBuilder UseCors(Action<CrossOriginResourceSharingHeaders> handler)
```

Parameters

handler [Action](#)<CrossOriginResourceSharingHeaders>

An action where the first argument is the main [CrossOriginResourceSharingHeaders](#) object.

Returns

[HttpServerHostContextBuilder](#)

UseCors(CrossOriginResourceSharingHeaders)

Sets an [CrossOriginResourceSharingHeaders](#) instance in the current listening host.

```
public HttpServerHostContextBuilder UseCors(CrossOriginResourceSharingHeaders cors)
```

Parameters

cors [CrossOriginResourceSharingHeaders](#)

The [CrossOriginResourceSharingHeaders](#) to the current host builder.

Returns

[HttpServerHostContextBuilder](#)

Method UseEngine

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

UseEngine(HttpServerEngine)

Sets the HTTP server engine.

```
public HttpServerHostContextBuilder UseEngine(HttpServerEngine engine)
```

Parameters

engine [HttpServerEngine](#)

The [HttpServerEngine](#) to use.

Returns

[HttpServerHostContextBuilder](#)

The current [HttpServerHostContextBuilder](#) instance.

UseEngine<TEngine>()

Sets the HTTP server engine using a default constructor.

```
public HttpServerHostContextBuilder UseEngine<TEngine>() where TEngine : HttpServerEngine, new()
```

Returns

[HttpServerHostContextBuilder](#)

The current [HttpServerHostContextBuilder](#) instance.

Type Parameters

TEngine

The type of the HTTP server engine to use, which must inherit from [HttpServerEngine](#) and have a parameterless constructor.

Method UseForwardingResolver

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

UseForwardingResolver(ForwardingResolver)

This method is a shortcut for setting [ForwardingResolver](#).

```
public HttpServerHostContextBuilder UseForwardingResolver(ForwardingResolver resolver)
```

Parameters

resolver ForwardingResolver

The [ForwardingResolver](#) object.

Returns

[HttpServerHostContextBuilder](#)

UseForwardingResolver<TForwardingResolver>()

This method is a shortcut for setting [ForwardingResolver](#).

```
public HttpServerHostContextBuilder UseForwardingResolver<TForwardingResolver>() where  
TForwardingResolver : ForwardingResolver, new()
```

Returns

[HttpServerHostContextBuilder](#)

Type Parameters

TForwardingResolver

The type which implements [ForwardingResolver](#).

Method UseHandler

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

UseHandler<THandler>()

This method is an shortcut for calling [RegisterHandler<T>\(\)](#).

```
public HttpServerHostContextBuilder UseHandler<THandler>() where THandler :  
HttpServerHandler, new()
```

Returns

[HttpServerHostContextBuilder](#)

Type Parameters

THandler

The handler which implements [HttpServerHandler](#).

UseHandler(HttpServerHandler)

This method is an shortcut for calling [RegisterHandler\(HttpServerHandler\)](#).

```
public HttpServerHostContextBuilder UseHandler(HttpServerHandler handler)
```

Parameters

handler [HttpServerHandler](#)

The instance of the server handler.

Returns

[HttpServerHostContextBuilder](#)

Method UseHttpServer

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

UseHttpServer(Action<HttpServer>)

Calls an action that has the HTTP server instance as an argument.

```
public HttpServerHostContextBuilder UseHttpServer(Action<HttpServer> handler)
```

Parameters

handler [Action](#)<HttpServer>

An action where the first argument is the main [HttpServer](#) object.

Returns

[HttpServerHostContextBuilder](#)

Method UseListeningPort

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

UseListeningPort(ushort)

Sets the main [ListeningPort](#) of this host builder.

```
public HttpServerHostContextBuilder UseListeningPort(ushort port)
```

Parameters

port `ushort`

The port the server will listen on.

Returns

[HttpServerHostContextBuilder](#)

UseListeningPort(string)

Sets the main [ListeningPort](#) of this host builder.

```
public HttpServerHostContextBuilder UseListeningPort(string uri)
```

Parameters

uri `string`

The URI component that will be parsed to the listening port format.

Returns

[HttpServerHostContextBuilder](#)

UseListeningPort(ListeningPort)

Sets the main [ListeningPort](#) of this host builder.

```
public HttpServerHostContextBuilder UseListeningPort(ListeningPort listeningPort)
```

Parameters

listeningPort [ListeningPort](#)

The [ListeningPort](#) object which the HTTP server will listen to.

Returns

[HttpServerHostContextBuilder](#)

Method UseLocale

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

UseLocale(CultureInfo)

Changes the default thread current culture through [DefaultThreadCurrentCulture](#).

```
public HttpServerHostContextBuilder UseLocale(CultureInfo locale)
```

Parameters

locale CultureInfo

The default [CultureInfo](#) object which the HTTP server will apply to the request handlers and callbacks thread.

Returns

[HttpServerHostContextBuilder](#)

Method UsePortableConfiguration

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

UsePortableConfiguration(Action<PortableConfigurationBuilder>)

Enables the portable configuration for this application, which imports settings, parameters, and other information from a JSON settings file.

```
public HttpServerHostContextBuilder  
UsePortableConfiguration(Action<PortableConfigurationBuilder> portableConfigHandler)
```

Parameters

portableConfigHandler [Action](#)<PortableConfigurationBuilder>

The handler of [PortableConfigurationBuilder](#).

Returns

[HttpServerHostContextBuilder](#)

Remarks

This method overrides almost all of your [CreateBuilder\(\)](#) configuration. To avoid this, call this method at the beginning of your builder, as the first immediate method.

Method UseRouter

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

UseRouter(Action<Router>)

Calls an action that has an [Router](#) instance from the host HTTP server.

```
public HttpServerHostContextBuilder UseRouter(Action<Router> handler)
```

Parameters

handler [Action](#)<Router>

An action where the first argument is the main [Router](#) object.

Returns

[HttpServerHostContextBuilder](#)

UseRouter(Router)

Sets an [Router](#) instance in the current listening host.

```
public HttpServerHostContextBuilder UseRouter(Router r)
```

Parameters

r [Router](#)

The [Router](#) to the current host builder.

Returns

[HttpServerHostContextBuilder](#)

Method UseStartupMessage

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

UseStartupMessage(string)

Add an optional message to the [HttpServerHostContext](#) output verbose.

```
public HttpServerHostContextBuilder UseStartupMessage(string startupMessage)
```

Parameters

startupMessage [string](#)

The startup message.

Returns

[HttpServerHostContextBuilder](#)

UseStartupMessage(Func<string>)

Adds a function that returns an optional initialization message to the [HttpServerHostContext](#) output verbose.

```
public HttpServerHostContextBuilder UseStartupMessage(Func<string> startupMessage)
```

Parameters

startupMessage [Func](#)<string>

The startup message function.

Returns

[HttpServerHostContextBuilder](#)

Enum `HttpServerHostContextBuilderExceptionMode`

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:[Sisk.Core.dll](#)

Represents how the builder event error message should be displayed.

```
public enum HttpServerHostContextBuilderExceptionMode
```

Fields

Detailed = 2

Detailed messages, including detailed exception trace and information, should be displayed.

Normal = 1

Normal messages, including their exception type and message, should be displayed.

Silent = 0

No message should be displayed.

Throw = 3

No message should be displayed and exceptions should be thrown instead being caughts.

Interface IConfigurationReader

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Represents an interface that reads and applies settings from a settings file.

```
public interface IConfigurationReader
```

Methods

[ReadConfiguration\(ConfigurationContext\)](#)

Represents the method that reads and applies settings from a settings file.

Method ReadConfiguration

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

ReadConfiguration(ConfigurationContext)

Represents the method that reads and applies settings from a settings file.

```
void ReadConfiguration(ConfigurationContext context)
```

Parameters

context [ConfigurationContext](#)

The configuration context object.

Class InitializationParameterCollection

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Provides a collection of HTTP server initialization variables.

```
public sealed class InitializationParameterCollection : IDictionary<string, string?>, ICollection<KeyValuePair<string, string?>>, IEnumerable<KeyValuePair<string, string?>>, IEnumerable
```

Implements

[IDictionary](#)<string, string>, [ICollection](#)<[KeyValuePair](#)<string, string>>, [IEnumerable](#)<[KeyValuePair](#)<string, string>>, [IEnumerable](#)

Constructors

[InitializationParameterCollection\(\)](#)

Methods

[AsNameValueCollection\(\)](#)

Gets an instance of [NameValueCollection](#) with the values of this class.

[EnsureNotNull\(string\)](#)

Ensures that the parameter defined by name `parameterName` is present in this collection.

[EnsureNotNullOrEmpty\(string\)](#)

Ensures that the parameter defined by name `parameterName` is present and not empty in this collection.

[GetStringValue\(string\)](#)

Retrieves a [StringValue](#) instance representing the specified parameter.

[GetValueOrDefault\(string, GetValueOption\)](#)

Gets the specified value if present in this parameter collection, or throw an exception if the value is not present.

Constructor InitializationParameterCollection

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

InitializationParameterCollection()

```
public InitializationParameterCollection()
```

Method AsNameValueCollection

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

AsNameValueCollection()

Gets an instance of [NameValuePairCollection](#) with the values of this class.

```
public NameValueCollection AsNameValueCollection()
```

Returns

[NameValuePairCollection](#)

Method EnsureNotNull

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

EnsureNotNull(string)

Ensures that the parameter defined by name `parameterName` is present in this collection.

```
public void EnsureNotNull(string parameterName)
```

Parameters

parameterName [string](#)

The parameter name which will be evaluated.

Remarks

If the parameter doesn't meet the above requirements, an [ArgumentNullException](#) exception is thrown.

Method EnsureNotNullOrEmpty

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

EnsureNotNullOrEmpty(string)

Ensures that the parameter defined by name `parameterName` is present and not empty in this collection.

```
public void EnsureNotNullOrEmpty(string parameterName)
```

Parameters

parameterName [string](#)

The parameter name which will be evaluated.

Remarks

If the parameter doesn't meet the above requirements, an [ArgumentNullException](#) exception is thrown.

Method GetStringValue

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

GetStringValue(string)

Retrieves a [StringValue](#) instance representing the specified parameter.

```
public StringValue GetStringValue(string parameterName)
```

Parameters

parameterName [string](#)

The name of the parameter to retrieve.

Returns

[StringValue](#)

A [StringValue](#) instance containing the parameter name and value.

Method GetValueOrThrow

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

GetValueOrThrow(string, GetValueOption)

Gets the specified value if present in this parameter collection, or throw an exception if the value is not present.

```
public string GetValueOrThrow(string parameterName,  
InitializationParameterCollection.GetValueOption option = GetValueOption.NotNullOrEmpty)
```

Parameters

parameterName [string](#)

The parameter name.

option [InitializationParameterCollection.GetValueOption](#)

Specifies the [InitializationParameterCollection.GetValueOption](#) used for getting the value.

Returns

[string](#)

Enum InitializationParameterCollection.GetValueOption

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Represents the option used in the method [GetValueOrThrow\(string, GetValueOption\)](#).

```
public enum InitializationParameterCollection.GetValueOption
```

Fields

NotNull = 0

The method should throw if the value is not present in the collection, but allow empty values.

NotNullOrEmpty = 1

The method should throw if the value is not present in the collection or has an empty value.

NotNullOrWhiteSpace = 2

The method should throw if the value is not present in the collection, has an empty value or consists of whitespaces (spaces, tabs, etc.).

Class PortableConfigurationBuilder

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

Represents the portable configuration builder for [HttpServerHostContextBuilder](#).

```
public sealed class PortableConfigurationBuilder
```

Methods

[WithConfigFile\(string, bool, ConfigurationFileLookupDirectory\)](#)

Specifies the name of the server configuration file.

[WithConfigReader\(IConfigurationReader\)](#)

Defines an custom [IConfigurationReader](#) configuration pipeline to the builder.

[WithConfigReader<TReader>\(\)](#)

Defines an custom [IConfigurationReader](#) configuration pipeline to the builder.

[WithParameters\(Action<InitializationParameterCollection>\)](#)

Invokes a method on the initialization parameter collection.

Method WithConfigFile

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

WithConfigFile(string, bool, ConfigurationFileLookupDirectory)

Specifies the name of the server configuration file.

```
public PortableConfigurationBuilder WithConfigFile(string filename, bool createIfDontExists =  
false, ConfigurationFileLookupDirectory lookupDirectories =  
ConfigurationFileLookupDirectory.CurrentDirectory)
```

Parameters

filename [string](#)

The name of the JSON configuration file.

createIfDontExists [bool](#)

Optional. Determines if the configuration file should be created if it doesn't exist.

lookupDirectories [ConfigurationFileLookupDirectory](#)

Optional. Specifies the directories which the [IConfigurationReader](#) should search for the configuration file.

Returns

[PortableConfigurationBuilder](#)

Method WithConfigReader

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

WithConfigReader(IConfigurationReader)

Defines an custom [IConfigurationReader](#) configuration pipeline to the builder.

```
public PortableConfigurationBuilder WithConfigReader(IConfigurationReader reader)
```

Parameters

reader IConfigurationReader

The [IConfigurationReader](#) object.

Returns

PortableConfigurationBuilder

WithConfigReader<TReader>()

Defines an custom [IConfigurationReader](#) configuration pipeline to the builder.

```
public PortableConfigurationBuilder WithConfigReader<TReader>() where TReader :  
IConfigurationReader, new()
```

Returns

PortableConfigurationBuilder

Type Parameters

TReader

The [IConfigurationReader](#) type.

Method WithParameters

Namespace:[Sisk.Core.Http.Hosting](#)

Assembly:Sisk.Core.dll

WithParameters(Action<InitializationParameterCollection>)

Invokes a method on the initialization parameter collection.

```
public PortableConfigurationBuilder WithParameters(Action<InitializationParameterCollection> handler)
```

Parameters

handler [Action](#)<InitializationParameterCollection>

The handler of [InitializationParameterCollection](#).

Returns

[PortableConfigurationBuilder](#)

Namespace Sisk.Core.Http.Streams

Classes

[HttpEventSourceCollection](#)

Provides a managed object to manage [HttpRequestEventSource](#) connections.

[HttpRequestEventSource](#)

An [HttpRequestEventSource](#) instance opens a persistent connection to the request, which sends events in text/event-stream format.

[HttpResponseStreamManager](#)

Represents a way to manage HTTP requests with their output streams, without relying on synchronous content.

[HttpStreamPingPolicy](#)

Provides an automatic ping sender for HTTP Event Source connections.

[HttpWebSocket](#)

Provides an persistent bi-directional socket between the client and the HTTP server.

[HttpWebSocketConnectionCollection](#)

Provides a managed object to manage [HttpWebSocket](#) connections.

[WebSocketMessage](#)

Represents an websocket request message received by an websocket server.

Delegates

[EventSourceRegistrationHandler](#)

Represents an function that is called when an [HttpEventSourceCollection](#) registers an new event source connection.

[EventSourceUnregistrationHandler](#)

Represents an function that is called when an [HttpEventSourceCollection](#) is removed and had their connection closed.

[WebSocketRegistrationHandler](#)

Represents an function that is called when an [HttpWebSocketConnectionCollection](#) registers an new web socket connection.

WebSocketUnregistrationHandler

Represents a function that is called when an [HttpWebSocketConnectionCollection](#) is removed and had its connection closed.

Delegate EventSourceRegistrationHandler

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:[Sisk.Core.dll](#)

Represents an function that is called when an [HttpEventSourceCollection](#) registers an new event source connection.

```
public delegate void EventSourceRegistrationHandler(object sender,  
HttpRequestEventSource eventSource)
```

Parameters

sender [object](#)

Represents the caller [HttpEventSourceCollection](#) object.

eventSource [HttpRequestEventSource](#)

Represents the registered [HttpRequestEventSource](#) event source connection.

Constructors

[EventSourceRegistrationHandler\(object, nint\)](#)

Methods

[BeginInvoke\(object, HttpRequestEventSource, AsyncCallback, object\)](#)

[EndInvoke\(IAsyncResult\)](#)

[Invoke\(object, HttpRequestEventSource\)](#)

Constructor EventSourceRegistrationHandler

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

EventSourceRegistrationHandler(object, nint)

```
public EventSourceRegistrationHandler(object @object, nint method)
```

Parameters

object object ↗

method nint ↗

Method BeginInvoke

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

BeginInvoke(object, HttpRequestEventSource, AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(object sender, HttpRequestEventSource eventSource,
AsyncCallback callback, object @object)
```

Parameters

sender object ↗

eventSource HttpRequestEventSource

callback AsyncCallback ↗

object object ↗

Returns

IAsyncResult ↗

Method EndInvoke

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

EndInvoke(IAsyncResult)

```
public virtual void EndInvoke(IAsyncResult result)
```

Parameters

result [IAsyncResult](#)

Method Invoke

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Invoke(object, HttpRequestEventArgs)

```
public virtual void Invoke(object sender, HttpRequestEventArgs eventSource)
```

Parameters

sender object ↗

eventSource HttpRequestEventArgs

Delegate EventSourceUnregistrationHandler

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:[Sisk.Core.dll](#)

Represents an function that is called when an [HttpEventSourceCollection](#) is removed and had their connection closed.

```
public delegate void EventSourceUnregistrationHandler(object sender,  
HttpRequestEventSource eventSource)
```

Parameters

sender [object](#)

Represents the caller [HttpEventSourceCollection](#) object.

eventSource [HttpRequestEventSource](#)

Represents the closed [HttpRequestEventSource](#) event source connection.

Constructors

[EventSourceUnregistrationHandler\(object, nint\)](#)

Methods

[BeginInvoke\(object, HttpRequestEventSource, AsyncCallback, object\)](#)

[EndInvoke\(IAsyncResult\)](#)

[Invoke\(object, HttpRequestEventSource\)](#)

Constructor EventSourceUnregistrationHandler

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

EventSourceUnregistrationHandler(object, nint)

```
public EventSourceUnregistrationHandler(object @object, nint method)
```

Parameters

object object

method nint

Method BeginInvoke

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

BeginInvoke(object, HttpRequestEventSource, AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(object sender, HttpRequestEventSource eventSource,
AsyncCallback callback, object @object)
```

Parameters

sender object ↗

eventSource HttpRequestEventSource

callback AsyncCallback ↗

object object ↗

Returns

IAsyncResult ↗

Method EndInvoke

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

EndInvoke(IAsyncResult)

```
public virtual void EndInvoke(IAsyncResult result)
```

Parameters

result [IAsyncResult](#)

Method Invoke

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Invoke(object, HttpRequestEventArgs)

```
public virtual void Invoke(object sender, HttpRequestEventArgs eventSource)
```

Parameters

sender object ↗

eventSource HttpRequestEventArgs

Class HttpEventSourceCollection

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:[Sisk.Core.dll](#)

Provides a managed object to manage [HttpRequestEventSource](#) connections.

```
public sealed class HttpEventSourceCollection : IReadOnlyCollection<HttpRequestEventSource>,  
IEnumerable<HttpRequestEventSource>, IEnumerable
```

Implements

[IReadOnlyCollection](#)<[HttpRequestEventSource](#)>, [IEnumerable](#)<[HttpRequestEventSource](#)>, [IEnumerable](#)

Properties

[ActiveConnections](#)

Gets an number indicating the amount of active event source connections.

[Count](#)

Methods

[All\(\)](#)

Gets all actives [HttpRequestEventSource](#) instances.

[DropAll\(\)](#)

Closes and disposes all registered and active [HttpRequestEventSource](#) in this collections.

[Find\(Func<string, bool>\)](#)

Gets all actives [HttpRequestEventSource](#) instances that matches their identifier predicate.

[GetByIdentifier\(string\)](#)

Gets the event source connection for the specified identifier.

[GetEnumerator\(\)](#)

Events

[OnEventSourceRegistered](#)

Represents an event that is fired when an [HttpRequestEventArgs](#) is registered in this collection.

[OnEventSourceUnregistration](#)

Represents an event that is fired when an [HttpRequestEventArgs](#) is closed and removed from this collection.

Property ActiveConnections

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

ActiveConnections

Gets an number indicating the amount of active event source connections.

```
public int ActiveConnections { get; }
```

Property Value

[int](#)

Property Count

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Count

```
public int Count { get; }
```

Property Value

[int](#)

Method All

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

All()

Gets all actives [HttpRequestEventArgs](#) instances.

```
public HttpRequestEventArgs[] All()
```

Returns

[HttpRequestEventArgs](#)[]

Method DropAll

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

DropAll()

Closes and disposes all registered and active [HttpRequestEventArgs](#) in this collections.

```
public void DropAll()
```

Method Find

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Find(Func<string, bool>)

Gets all actives [HttpRequestEventSource](#) instances that matches their identifier predicate.

```
public HttpRequestEventSource[ ] Find(Func<string, bool> predicate)
```

Parameters

predicate [Func<string, bool>](#)

The expression on the an non-empty event source identifier.

Returns

[HttpRequestEventSource\[\]](#)

Method GetByIdentifier

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

GetByIdentifier(string)

Gets the event source connection for the specified identifier.

```
public HttpRequestEventSource? GetByIdentifier(string identifier)
```

Parameters

identifier string ↗

The event source identifier.

Returns

[HttpRequestEventSource](#)

Method GetEnumerator

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

GetEnumerator()

```
public IEnumarator<HttpRequestEventSource> GetEnumerator()
```

Returns

[IEnumarator](#)<HttpRequestEventSource>

Event OnEventSourceRegistered

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Represents an event that is fired when an [HttpRequestEventArgs](#) is registered in this collection.

```
public event EventSourceRegistrationHandler? OnEventSourceRegistered
```

Returns

[EventSourceRegistrationHandler](#)

Represents an event that is fired when an is registered in this collection.

Event OnEventSourceUnregistration

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Represents an event that is fired when an [HttpRequestEventArgs](#) is closed and removed from this collection.

```
public event EventSourceUnregistrationHandler? OnEventSourceUnregistration
```

Returns

[EventSourceUnregistrationHandler](#)

Represents an event that is fired when an is closed and removed from this collection.

Class HttpRequestEventSource

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

An [HttpRequestEventSource](#) instance opens a persistent connection to the request, which sends events in text/event-stream format.

```
public sealed class HttpRequestEventSource : IDisposable
```

Implements

[IDisposable](#) ↗

Properties

[HttpRequest](#)

Gets the [HttpRequest](#) object which created this Event Source instance.

[Identifier](#)

Gets an unique identifier label to this EventStream connection, useful for finding this connection's reference later.

[IsActive](#)

Gets an boolean indicating if this connection is open and this instance can send messages.

[PingPolicy](#)

Gets the [HttpStreamPingPolicy](#) for this HTTP event source connection.

[SentContentLength](#)

Gets an integer indicating the total bytes sent by this instance to the client.

Methods

[AppendHeader\(string, string\)](#)

Sends an header to the streaming context.

[Cancel\(\)](#)

Cancels the sending queue from sending pending messages and clears the queue.

[Close\(\)](#)

Closes the event listener and it's connection.

[Dispose\(\)](#)

Flushes and releases the used resources of this class instance.

[KeepAlive\(\)](#)

Asynchronously waits for the connection to close before continuing execution. This method is released when either the client or the server reaches an sending failure.

[Send\(string?, string\)](#)

Sends an event to the client over the HTTP connection.

[SendAsync\(string?, string\)](#)

Asynchronously sends an event to the client over the HTTP connection.

[WaitForFail\(TimeSpan\)](#)

Asynchronously waits for the connection to close before continuing execution with an maximum keep alive timeout. This method is released when either the client or the server reaches an sending failure.

[WithPing\(Action<HttpStreamPingPolicy>\)](#)

Configures the ping policy for this instance of HTTP Event Source.

Property HttpRequest

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

HttpRequest

Gets the [HttpRequest](#) object which created this Event Source instance.

```
public HttpRequest HttpRequest { get; }
```

Property Value

[HttpRequest](#)

Property Identifier

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Identifier

Gets an unique identifier label to this EventStream connection, useful for finding this connection's reference later.

```
public string? Identifier { get; }
```

Property Value

[string](#)

Property IsActive

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

IsActive

Gets an boolean indicating if this connection is open and this instance can send messages.

```
public bool IsActive { get; }
```

Property Value

bool

Property PingPolicy

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

PingPolicy

Gets the [HttpStreamPingPolicy](#) for this HTTP event source connection.

```
public HttpStreamPingPolicy PingPolicy { get; }
```

Property Value

[HttpStreamPingPolicy](#)

Property SentContentLength

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

SentContentLength

Gets an integer indicating the total bytes sent by this instance to the client.

```
public int SentContentLength { get; }
```

Property Value

[int](#)

Method AppendHeader

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

AppendHeader(string, string)

Sends an header to the streaming context.

```
public void AppendHeader(string name, string value)
```

Parameters

name [string](#)

The header name.

value [string](#)

The header value.

Method Cancel

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Cancel()

Cancels the sending queue from sending pending messages and clears the queue.

```
public void Cancel()
```

Method Close

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Close()

Closes the event listener and it's connection.

```
public HttpResponse Close()
```

Returns

[HttpResponse](#)

Method Dispose

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Dispose()

Flushes and releases the used resources of this class instance.

```
public void Dispose()
```

Method KeepAlive

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

KeepAlive()

Asynchronously waits for the connection to close before continuing execution. This method is released when either the client or the server reaches an sending failure.

```
public void KeepAlive()
```

Method Send

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Send(string?, string)

Sends an event to the client over the HTTP connection.

```
public bool Send(string? data, string fieldName = "data")
```

Parameters

data [string](#)

The data to be sent as part of the event.

fieldName [string](#)

The field name for the event data. Defaults to "data".

Returns

[bool](#)

True if the event was sent successfully, false otherwise.

Method SendAsync

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

SendAsync(string?, string)

Asynchronously sends an event to the client over the HTTP connection.

```
public ValueTask<bool> SendAsync(string? data, string fieldName = "data")
```

Parameters

data [string](#)

The data to be sent as part of the event.

fieldName [string](#)

The field name for the event data. Defaults to "data".

Returns

[ValueTask](#)<[bool](#)>

A [ValueTask](#)<[TResult](#)> that represents the asynchronous operation. The result is true if the event was sent successfully, false otherwise.

Method WaitForFail

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

WaitForFail(TimeSpan)

Asynchronously waits for the connection to close before continuing execution with an maximum keep alive timeout. This method is released when either the client or the server reaches an sending failure.

```
public void WaitForFail(TimeSpan maximumIdleTolerance)
```

Parameters

maximumIdleTolerance [TimeSpan](#)

The maximum timeout interval for an idle connection to automatically release this method.

Method WithPing

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

WithPing(Action<HttpStreamPingPolicy>)

Configures the ping policy for this instance of HTTP Event Source.

```
public void WithPing(Action<HttpStreamPingPolicy> act)
```

Parameters

act [Action<HttpStreamPingPolicy>](#)

The method that runs on the ping policy for this HTTP Event Source.

Class HttpResponseStreamManager

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:[Sisk.Core.dll](#)

Represents a way to manage HTTP requests with their output streams, without relying on synchronous content.

```
public sealed class HttpResponseStreamManager
```

Properties

[ResponseStream](#)

Gets the [Stream](#) that represents the HTTP response output stream.

[SendChunked](#)

Gets or sets whether this HTTP response stream should use chunked transfer encoding.

Methods

[Close\(\)](#)

Closes this HTTP response stream connection between the server and the client and returns an empty [Http Response](#) to finish the HTTP server context.

[GetStreamWriter\(string?, Encoding?, string?\)](#)

Returns a [StreamWriter](#) for writing to the response stream, setting the content type and encoding.

[SetContentLength\(long\)](#)

Sets the Content-Length header of this response stream. If this response stream is using chunked transfer encoding, this method will do nothing.

[SetCookie\(Cookie\)](#)

Sets a cookie and sends it in the response to be set by the client.

[SetCookie\(string, string\)](#)

Sets a cookie and sends it in the response to be set by the client.

[SetCookie\(string, string, DateTime?, TimeSpan?, string?, string?, bool?, bool?, string?\)](#)

Sets a cookie and sends it in the response to be set by the client.

[SetHeader\(string, object?\)](#)

Sets the specific HTTP header into this response stream.

[SetStatus\(HttpStatusCode\)](#)

Sets the HTTP status code and description for this response stream.

[SetStatus\(int\)](#)

Sets the HTTP status code for this response stream.

[SetStatus\(HttpStatusCode\)](#)

Sets the HTTP status code for this response stream.

[Write\(byte\[\]\)](#)

Writes an sequence of bytes to the HTTP response stream.

[Write\(byte\[\], int, int\)](#)

Writes a range of bytes from a byte array to the HTTP response stream.

[Write\(ReadOnlySpan<byte>\)](#)

Writes an sequence of bytes to the HTTP response stream.

Property ResponseStream

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

ResponseStream

Gets the [Stream](#) that represents the HTTP response output stream.

```
public Stream ResponseStream { get; }
```

Property Value

[Stream](#)

Property SendChunked

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

SendChunked

Gets or sets whether this HTTP response stream should use chunked transfer encoding.

```
public bool SendChunked { get; set; }
```

Property Value

bool ↗

Method Close

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Close()

Closes this HTTP response stream connection between the server and the client and returns an empty [Http Response](#) to finish the HTTP server context.

```
public HttpResponse Close()
```

Returns

[HttpResponse](#)

Method GetStreamWriter

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

GetStreamWriter(string?, Encoding?, string?)

Returns a [StreamWriter](#) for writing to the response stream, setting the content type and encoding.

```
public StreamWriter GetStreamWriter(string? contentType, Encoding? encoding, string? newLine  
= "\r\n")
```

Parameters

contentType [string](#)

The content type of the response, or [null](#) to omit the content type header.

encoding [Encoding](#)

The encoding to use for the response, or [null](#) to use the default encoding.

newLine [string](#)

The new line string literal for the writer.

Returns

[StreamWriter](#)

A [StreamWriter](#) for writing to the response stream.

Method SetContentLength

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

SetContentLength(long)

Sets the Content-Length header of this response stream. If this response stream is using chunked transfer encoding, this method will do nothing.

```
public void SetContentLength(long contentLength)
```

Parameters

contentLength [long](#)

The length in bytes of the content stream.

Method SetCookie

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

SetCookie(Cookie)

Sets a cookie and sends it in the response to be set by the client.

```
public void SetCookie(Cookie cookie)
```

Parameters

cookie [Cookie](#)

The cookie object.

SetCookie(string, string)

Sets a cookie and sends it in the response to be set by the client.

```
public void SetCookie(string name, string value)
```

Parameters

name [string](#)

The cookie name.

value [string](#)

The cookie value.

SetCookie(string, string, DateTime?, TimeSpan?, string?, string?, bool?, bool?, string?)

Sets a cookie and sends it in the response to be set by the client.

```
public void SetCookie(string name, string value, DateTime? expires = null, TimeSpan? maxAge = null, string? domain = null, string? path = null, bool? secure = null, bool? httpOnly = null, string? sameSite = null)
```

Parameters

name `string`

The cookie name.

value `string`

The cookie value.

expires `DateTime?`

The cookie expiry date.

maxAge `TimeSpan?`

The cookie max duration after being set.

domain `string`

The domain where the cookie will be valid.

path `string`

The path where the cookie will be valid.

secure `bool?`

Determines if the cookie will only be stored in a secure context.

httpOnly `bool?`

Determines if the cookie will be only available in the HTTP context.

sameSite `string`

The cookie SameSite parameter.

Method SetHeader

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

SetHeader(string, object?)

Sets the specific HTTP header into this response stream.

```
public void SetHeader(string headerName, object? value)
```

Parameters

headerName [string](#)

The HTTP header name.

value [object](#)

The HTTP header value.

Remarks

Headers are sent immediately, along with the HTTP response code, after starting to send content or closing this stream.

Method SetStatus

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

SetStatus(int)

Sets the HTTP status code for this response stream.

```
public void SetStatus(int httpStatusCode)
```

Parameters

httpStatusCode [int](#)

The HTTP status code.

SetStatus(HttpStatusCode)

Sets the HTTP status code for this response stream.

```
public void SetStatus(HttpStatusCode statusCode)
```

Parameters

statusCode [HttpStatusCode](#)

The HTTP status code.

SetStatus(HttpStatusCode)

Sets the HTTP status code and description for this response stream.

```
public void SetStatus(HttpStatusCode statusCode)
```

Parameters

statusCode [HttpStatusInformation](#)

The custom HTTP status code information.

Method Write

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Write(ReadOnlySpan<byte>)

Writes an sequence of bytes to the HTTP response stream.

```
public void Write(ReadOnlySpan<byte> buffer)
```

Parameters

buffer [ReadOnlySpan<byte>](#)

The read only memory that includes the buffer which will be written to the HTTP response.

Write(byte[])

Writes an sequence of bytes to the HTTP response stream.

```
public void Write(byte[] buffer)
```

Parameters

buffer [byte\[\]](#)

The byte array that includes the buffer which will be written to the HTTP response.

Write(byte[], int, int)

Writes a range of bytes from a byte array to the HTTP response stream.

```
public void Write(byte[] buffer, int offset, int count)
```

Parameters

buffer `byte[]`

The byte array to write data from.

offset `int`

The zero-based byte offset in `buffer` at which to begin writing bytes.

count `int`

The maximum number of bytes to write.

Class HttpStreamPingPolicy

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Provides an automatic ping sender for HTTP Event Source connections.

```
public sealed class HttpStreamPingPolicy : IDisposable
```

Implements

[IDisposable](#) ↗

Properties

[DataMessage](#)

Gets or sets the payload message that is sent to the server as a ping message.

[Interval](#)

Gets or sets the sending interval for each ping message.

Methods

[Dispose\(\)](#)

[~HttpStreamPingPolicy\(\)](#)

[Start\(\)](#)

Starts sending periodic pings to the client.

[Start\(string, TimeSpan\)](#)

Configures and starts sending periodic pings to the client.

Property DataMessage

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

DataMessage

Gets or sets the payload message that is sent to the server as a ping message.

```
public string DataMessage { get; set; }
```

Property Value

[string](#)

Property Interval

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Interval

Gets or sets the sending interval for each ping message.

```
public TimeSpan Interval { get; set; }
```

Property Value

[TimeSpan](#) ↗

Method Dispose

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Dispose()

```
public void Dispose()
```

Method Start

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Start()

Starts sending periodic pings to the client.

```
public void Start()
```

Start(string, TimeSpan)

Configures and starts sending periodic pings to the client.

```
public void Start(string dataMessage, TimeSpan interval)
```

Parameters

dataMessage [string](#)

The payload message that is sent to the server as a ping message.

interval [TimeSpan](#)

The sending interval for each ping message.

Method ~HttpStreamPingPolicy

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

~HttpStreamPingPolicy()

```
protected ~HttpStreamPingPolicy()
```

Class HttpWebSocket

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:[Sisk.Core.dll](#)

Provides an persistent bi-directional socket between the client and the HTTP server.

```
public sealed class HttpWebSocket : IDisposable
```

Implements

[IDisposable](#)

Properties

[HttpRequest](#)

Gets the [HttpRequest](#) object which created this Web Socket instance.

[Identifier](#)

Gets an unique identifier label to this Web Socket connection, useful for finding this connection's reference later.

[IsClosed](#)

Gets an boolean indicating if this Web Socket connection is closed.

[PingPolicy](#)

Gets the [HttpStreamPingPolicy](#) for this HTTP web socket connection.

[State](#)

Gets or sets an object linked with this [WebSocket](#) session.

Methods

[CloseAsync\(CancellationToken\)](#)

Closes the WebSocket connection asynchronously.

[Dispose\(\)](#)

[ReceiveMessageAsync\(\)](#)

Receives a message from the WebSocket endpoint asynchronously with a default timeout of 30 seconds.

[ReceiveMessageAsync\(CancellationToken\)](#)

Receives a message from the WebSocket endpoint asynchronously.

[ReceiveMessageAsync\(TimeSpan\)](#)

Receives a message from the WebSocket endpoint asynchronously with a specified timeout.

[SendAsync\(ReadOnlyMemory<byte>, CancellationToken\)](#)

Sends an asynchronous binary message to the WebSocket endpoint.

[SendAsync\(string, CancellationToken\)](#)

Sends an asynchronous text message to the WebSocket endpoint.

Property HttpRequest

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

HttpRequest

Gets the [HttpRequest](#) object which created this Web Socket instance.

```
public HttpRequest HttpRequest { get; }
```

Property Value

[HttpRequest](#)

Property Identifier

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Identifier

Gets an unique identifier label to this Web Socket connection, useful for finding this connection's reference later.

```
public string? Identifier { get; }
```

Property Value

[string](#)

Property IsClosed

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

IsClosed

Gets an boolean indicating if this Web Socket connection is closed.

```
public bool IsClosed { get; }
```

Property Value

[bool](#)

Property PingPolicy

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

PingPolicy

Gets the [HttpStreamPingPolicy](#) for this HTTP web socket connection.

```
public HttpStreamPingPolicy PingPolicy { get; }
```

Property Value

[HttpStreamPingPolicy](#)

Property State

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

State

Gets or sets an object linked with this [WebSocket](#) session.

```
public object? State { get; set; }
```

Property Value

[object](#)

Method CloseAsync

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

CloseAsync(CancellationToken)

Closes the WebSocket connection asynchronously.

```
public Task<HttpResponse> CloseAsync(CancellationToken cancellation = default)
```

Parameters

cancellation [CancellationToken](#)

The [CancellationToken](#) to use for cancellation.

Returns

[Task](#)<HttpResponse>

A [Task](#)<TResult> that represents the asynchronous close operation, which returns an [HttpResponse](#) indicating the result of the close operation.

Method Dispose

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Dispose()

```
public void Dispose()
```

Method ReceiveMessageAsync

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

ReceiveMessageAsync(CancellationToken)

Receives a message from the WebSocket endpoint asynchronously.

```
public ValueTask<WebSocketMessage?> ReceiveMessageAsync(CancellationToken cancellation  
= default)
```

Parameters

cancellation [CancellationToken](#)

The [CancellationToken](#) to use for cancellation.

Returns

[ValueTask](#)< [WebSocketMessage](#)>

A [ValueTask](#)< [TResult](#)> that represents the asynchronous receive operation, which returns a [WebSocket Message](#) if a message is received; otherwise, `null`.

ReceiveMessageAsync(TimeSpan)

Receives a message from the WebSocket endpoint asynchronously with a specified timeout.

```
public ValueTask<WebSocketMessage?> ReceiveMessageAsync(TimeSpan timeout)
```

Parameters

timeout [TimeSpan](#)

The time to wait for a message before timing out.

Returns

`ValueTask<WebSocketMessage>`

A `ValueTask<TResult>` that represents the asynchronous receive operation, which returns a [WebSocket Message](#) if a message is received; otherwise, `null`.

ReceiveMessageAsync()

Receives a message from the WebSocket endpoint asynchronously with a default timeout of 30 seconds.

```
public ValueTask<WebSocketMessage?> ReceiveMessageAsync()
```

Returns

`ValueTask<WebSocketMessage>`

A `ValueTask<TResult>` that represents the asynchronous receive operation, which returns a [WebSocket Message](#) if a message is received; otherwise, `null`.

Method SendAsync

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

SendAsync(string, CancellationToken)

Sends an asynchronous text message to the WebSocket endpoint.

```
public ValueTask<bool> SendAsync(string message, CancellationToken cancellation = default)
```

Parameters

message [string](#)

The text message to send.

cancellation [CancellationToken](#)

The [CancellationToken](#) to use for cancellation.

Returns

[ValueTask](#)<bool>

A [ValueTask](#)<TResult> that represents the asynchronous send operation, which returns [true](#) if the message was sent successfully; otherwise, [false](#).

SendAsync(ReadOnlyMemory<byte>, CancellationToken)

Sends an asynchronous binary message to the WebSocket endpoint.

```
public ValueTask<bool> SendAsync(ReadOnlyMemory<byte> buffer, CancellationToken cancellation = default)
```

Parameters

buffer `ReadOnlyMemory<byte>`

The binary data to send.

cancellation `CancellationToken`

The `CancellationToken` to use for cancellation.

Returns

`ValueTask<bool>`

A `ValueTask<TResult>` that represents the asynchronous send operation, which returns `true` if the message was sent successfully; otherwise, `false`.

Class HttpWebSocketConnectionCollection

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:[Sisk.Core.dll](#)

Provides a managed object to manage [HttpWebSocket](#) connections.

```
public sealed class HttpWebSocketConnectionCollection : IReadOnlyCollection<HttpWebSocket>,  
IEnumerable<HttpWebSocket>, IEnumerable
```

Implements

[IReadOnlyCollection](#)<HttpWebSocket>, [IEnumerable](#)<HttpWebSocket>, [IEnumerable](#)

Properties

[ActiveConnections](#)

Gets an number indicating the amount of active web socket connections.

[Count](#)

Methods

[All\(\)](#)

Gets all actives [HttpWebSocket](#) instances.

[DropAll\(\)](#)

Closes all registered and active [HttpWebSocket](#) in this collections.

[Find\(Func<string, bool>\)](#)

Gets all actives [HttpWebSocket](#) instances that matches their identifier predicate.

[GetByIdentifier\(string\)](#)

Gets the Web Socket connection for the specified identifier.

[GetEnumerator\(\)](#)

Events

[OnWebSocketRegister](#)

Represents an event that is fired when an [HttpWebSocket](#) is registered in this collection.

[OnWebSocketUnregister](#)

Represents an event that is fired when an [HttpWebSocket](#) is closed and removed from this collection.

Property ActiveConnections

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

ActiveConnections

Gets an number indicating the amount of active web socket connections.

```
public int ActiveConnections { get; }
```

Property Value

[int](#)

Property Count

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Count

```
public int Count { get; }
```

Property Value

[int](#)

Method All

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

All()

Gets all actives [HttpWebSocket](#) instances.

```
public HttpWebSocket[] All()
```

Returns

[HttpWebSocket](#)[]

Method DropAll

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

DropAll()

Closes all registered and active [HttpWebSocket](#) in this collections.

```
public void DropAll()
```

Method Find

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Find(Func<string, bool>)

Gets all actives [HttpWebSocket](#) instances that matches their identifier predicate.

```
public HttpWebSocket[] Find(Func<string, bool> predicate)
```

Parameters

predicate [Func](#)<[string](#), [bool](#)>

The expression on the an non-empty Web Socket identifier.

Returns

[HttpWebSocket](#)[]

Method GetByIdentifier

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

GetByIdentifier(string)

Gets the Web Socket connection for the specified identifier.

```
public HttpWebSocket? GetByIdentifier(string identifier)
```

Parameters

identifier string ↗

The Web Socket identifier.

Returns

[HttpWebSocket](#)

Method GetEnumerator

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

GetEnumerator()

```
public IEnumarator<HttpWebSocket> GetEnumerator()
```

Returns

[IEnumarator](#)<HttpWebSocket>

Event OnWebSocketRegister

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Represents an event that is fired when an [HttpWebSocket](#) is registered in this collection.

```
public event WebSocketRegistrationHandler? OnWebSocketRegister
```

Returns

[WebSocketRegistrationHandler](#)

Represents an event that is fired when an is registered in this collection.

Event OnWebSocketUnregister

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Represents an event that is fired when an [HttpWebSocket](#) is closed and removed from this collection.

```
public event WebSocketRegistrationHandler? OnWebSocketUnregister
```

Returns

[WebSocketRegistrationHandler](#)

Represents an event that is fired when an is closed and removed from this collection.

Class WebSocketMessage

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Represents an websocket request message received by an websocket server.

```
public sealed class WebSocketMessage
```

Properties

[Length](#)

Gets the message length in byte count.

[MessageBytes](#)

Gets an byte array with the message contents.

[Sender](#)

Gets the sender [HttpWebSocket](#) object instance which received this message.

Methods

[GetString\(\)](#)

Reads the message bytes as string using the HTTP request encoding.

[GetString\(Encoding\)](#)

Reads the message bytes as string using the specified encoding.

Property Length

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Length

Gets the message length in byte count.

```
public int Length { get; }
```

Property Value

[int](#)

Property MessageBytes

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

MessageBytes

Gets an byte array with the message contents.

```
public byte[] MessageBytes { get; }
```

Property Value

[byte](#)

Property Sender

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Sender

Gets the sender [HttpWebSocket](#) object instance which received this message.

```
public HttpWebSocket Sender { get; }
```

Property Value

[HttpWebSocket](#)

Method GetString

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

GetString(Encoding)

Reads the message bytes as string using the specified encoding.

```
public string GetString(Encoding encoding)
```

Parameters

encoding [Encoding](#)

The encoding which will be used to decode the message.

Returns

[string](#)

GetString()

Reads the message bytes as string using the HTTP request encoding.

```
public string GetString()
```

Returns

[string](#)

Delegate WebSocketRegistrationHandler

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:[Sisk.Core.dll](#)

Represents an function that is called when an [HttpWebSocketConnectionCollection](#) registers an new web socket connection.

```
public delegate void WebSocketRegistrationHandler(object sender, HttpWebSocket ws)
```

Parameters

sender [object](#)

Represents the caller [HttpWebSocketConnectionCollection](#) object.

ws [HttpWebSocket](#)

Represents the registered [HttpWebSocket](#) web socket connection.

Constructors

[WebSocketRegistrationHandler\(object, nint\)](#)

Methods

[BeginInvoke\(object, HttpWebSocket, AsyncCallback, object\)](#)

[EndInvoke\(IAsyncResult\)](#)

[Invoke\(object, HttpWebSocket\)](#)

Constructor WebSocketRegistrationHandler

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

WebSocketRegistrationHandler(object, nint)

```
public WebSocketRegistrationHandler(object @object, nint method)
```

Parameters

object object ↗

method nint ↗

Method BeginInvoke

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

BeginInvoke(object, HttpWebSocket, AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(object sender, HttpWebSocket ws, AsyncCallback callback,
object @object)
```

Parameters

sender object ↗

ws [HttpWebSocket](#)

callback AsyncCallback ↗

object object ↗

Returns

IAsyncResult ↗

Method EndInvoke

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

EndInvoke(IAsyncResult)

```
public virtual void EndInvoke(IAsyncResult result)
```

Parameters

result [IAsyncResult](#)

Method Invoke

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Invoke(object, HttpWebSocket)

```
public virtual void Invoke(object sender, HttpWebSocket ws)
```

Parameters

sender object ↗

ws [HttpWebSocket](#)

Delegate WebSocketUnregistrationHandler

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:[Sisk.Core.dll](#)

Represents an function that is called when an [HttpWebSocketConnectionCollection](#) is removed and had it's connection closed.

```
public delegate void WebSocketUnregistrationHandler(object sender, HttpWebSocket ws)
```

Parameters

sender [object](#)

Represents the caller [HttpWebSocketConnectionCollection](#) object.

ws [HttpWebSocket](#)

Represents the closed [HttpWebSocket](#) web socket connection.

Constructors

[WebSocketUnregistrationHandler\(object, nint\)](#)

Methods

[BeginInvoke\(object, HttpWebSocket, AsyncCallback, object\)](#)

[EndInvoke\(IAsyncResult\)](#)

[Invoke\(object, HttpWebSocket\)](#)

Constructor WebSocketUnregistrationHandler

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

WebSocketUnregistrationHandler(object, nint)

```
public WebSocketUnregistrationHandler(object @object, nint method)
```

Parameters

object object

method nint

Method BeginInvoke

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

BeginInvoke(object, HttpWebSocket, AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(object sender, HttpWebSocket ws, AsyncCallback callback,
object @object)
```

Parameters

sender object ↗

ws [HttpWebSocket](#)

callback AsyncCallback ↗

object object ↗

Returns

IAsyncResult ↗

Method EndInvoke

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

EndInvoke(IAsyncResult)

```
public virtual void EndInvoke(IAsyncResult result)
```

Parameters

result [IAsyncResult](#)

Method Invoke

Namespace:[Sisk.Core.Http.Streams](#)

Assembly:Sisk.Core.dll

Invoke(object, HttpWebSocket)

```
public virtual void Invoke(object sender, HttpWebSocket ws)
```

Parameters

sender object ↗

ws [HttpWebSocket](#)

Class BrotliContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Represents an HTTP content that is compressed using the Brotli algorithm.

```
public sealed class BrotliContent : CompressedContent, IDisposable
```

Implements

[IDisposable](#) ↗

Constructors

[BrotliContent\(byte\[\]\)](#)

Initializes a new instance of compressing stream with the specified byte array content.

[BrotliContent\(Stream\)](#)

Initializes a new instance of compressing stream with the specified stream content.

[BrotliContent\(HttpContext\)](#)

Initializes a new instance of compressing stream with the specified inner HTTP content.

Methods

[GetCompressingStream\(Stream\)](#)

Gets a stream that compresses the output stream.

[Setup\(\)](#)

Represents the method that is invoked once within the constructor to setup this compressor. This method is indeeded to add the missing Content-Encoding headers used by this compressor.

Constructor BrotliContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

BrotliContent(HttpContent)

Initializes a new instance of compressing stream with the specified inner HTTP content.

```
public BrotliContent(HttpContent innerContent)
```

Parameters

innerContent [HttpContent](#)

The inner HTTP content.

BrotliContent(byte[])

Initializes a new instance of compressing stream with the specified byte array content.

```
public BrotliContent(byte[] byteArrayContent)
```

Parameters

byteArrayContent [byte](#)[]

The byte array content.

BrotliContent(Stream)

Initializes a new instance of compressing stream with the specified stream content.

```
public BrotliContent(Stream baseContent)
```

Parameters

baseContent Stream

The stream content.

Method GetCompressingStream

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetCompressingStream(Stream)

Gets a stream that compresses the output stream.

```
public override Stream GetCompressingStream(Stream outputStream)
```

Parameters

outputStream [Stream](#)

The output stream to compress.

Returns

[Stream](#)

A stream that compresses the output stream.

Method Setup

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Setup()

Represents the method that is invoked once within the constructor to setup this compressor. This method is indeeded to add the missing Content-Encoding headers used by this compressor.

```
public override void Setup()
```

Class CompressedContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Represents a base class for HTTP contents served over an compressing stream.

```
public abstract class CompressedContent : HttpContent, IDisposable
```

Implements

[IDisposable](#) ↗

Constructors

[CompressedContent\(byte\[\]\)](#)

Initializes a new instance of compressing stream with the specified byte array content.

[CompressedContent\(Stream\)](#)

Initializes a new instance of compressing stream with the specified stream content.

[CompressedContent\(HttpContent\)](#)

Initializes a new instance of compressing stream with the specified inner HTTP content.

Properties

[InnerContent](#)

Gets the inner HTTP content.

Methods

[Dispose\(bool\)](#)

[GetCompressingStream\(Stream\)](#)

Gets a stream that compresses the output stream.

[SerializeToStream\(Stream, TransportContext?, CancellationToken\)](#)

[SerializeToStreamAsync\(Stream, TransportContext?\)](#)

[SerializeToStreamAsync\(Stream, TransportContext?, CancellationToken\)](#)

[Setup\(\)](#)

Represents the method that is invoked once within the constructor to setup this compressor. This method is indeeded to add the missing Content-Encoding headers used by this compressor.

[TryComputeLength\(out long\)](#)

Constructor CompressedContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

CompressedContent(HttpContent)

Initializes a new instance of compressing stream with the specified inner HTTP content.

```
public CompressedContent(HttpContent innerContent)
```

Parameters

innerContent [HttpContent](#)

The inner HTTP content.

CompressedContent(byte[])

Initializes a new instance of compressing stream with the specified byte array content.

```
public CompressedContent(byte[] byteArrayContent)
```

Parameters

byteArrayContent [byte](#)

The byte array content.

CompressedContent(Stream)

Initializes a new instance of compressing stream with the specified stream content.

```
public CompressedContent(Stream baseContent)
```

Parameters

baseContent Stream

The stream content.

Property InnerContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

InnerContent

Gets the inner HTTP content.

```
public HttpContent InnerContent { get; }
```

Property Value

[HttpContent](#) ↗

Method Dispose

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Dispose(bool)

```
protected override void Dispose(bool disposing)
```

Parameters

disposing [bool](#)

Method GetCompressingStream

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetCompressingStream(Stream)

Gets a stream that compresses the output stream.

```
public abstract Stream GetCompressingStream(Stream outputStream)
```

Parameters

outputStream [Stream](#)

The output stream to compress.

Returns

[Stream](#)

A stream that compresses the output stream.

Method SerializeToStream

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

SerializeToStream(Stream, TransportContext?, CancellationToken)

```
protected override sealed void SerializeToStream(Stream stream, TransportContext? context,  
CancellationToken cancellationToken)
```

Parameters

stream Stream

context TransportContext

cancellationToken CancellationToken

Method SerializeToStreamAsync

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

SerializeToStreamAsync(Stream, TransportContext?)

```
protected override sealed Task SerializeToStreamAsync(Stream stream, TransportContext? context)
```

Parameters

stream Stream

context TransportContext

Returns

Task

SerializeToStreamAsync(Stream, TransportContext?, CancellationToken)

```
protected override sealed Task SerializeToStreamAsync(Stream stream, TransportContext? context,  
CancellationToken cancellationToken)
```

Parameters

stream Stream

context TransportContext

cancellationToken CancellationToken

Returns

Task

Method Setup

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Setup()

Represents the method that is invoked once within the constructor to setup this compressor. This method is indeeded to add the missing Content-Encoding headers used by this compressor.

```
public abstract void Setup()
```

Method TryComputeLength

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

TryComputeLength(out long)

```
protected override sealed bool TryComputeLength(out long length)
```

Parameters

length [long](#)

Returns

[bool](#)

Class DefaultMessagePage

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Provides methods for creating informative static pages used by Sisk.

```
public static class DefaultMessagePage
```

Properties

[DefaultPageCSS](#)

Gets or sets the page CSS string used by the page code.

Methods

[CreateDefaultPageHtml\(string, string\)](#)

Creates an static default page with given header and description.

[CreateDefaultResponse\(in HttpStatusInformation, string\)](#)

Creates an static default page with given status code and description.

Property DefaultPageCSS

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

DefaultPageCSS

Gets or sets the page CSS string used by the page code.

```
public static string DefaultPageCSS { get; set; }
```

Property Value

[string](#)

Method CreateDefaultPageHtml

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

CreateDefaultPageHtml(string, string)

Creates an static default page with given header and description.

```
public static string CreateDefaultPageHtml(string firstHeader, string description)
```

Parameters

firstHeader [string](#)

The static page header text.

description [string](#)

The static page description text.

Returns

[string](#)

Method CreateDefaultResponse

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

CreateDefaultResponse(in HttpStatusInformation, string)

Creates an static default page with given status code and description.

```
public static HttpResponse CreateDefaultResponse(in HttpStatusInformation status,  
string longDescription)
```

Parameters

status [HttpStatusInformation](#)

The static page status code.

longDescription [string](#)

The static page description text.

Returns

[HttpResponse](#)

Class DeflateContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Represents an HTTP content that is compressed using the Deflate algorithm.

```
public sealed class DeflateContent : CompressedContent, IDisposable
```

Implements

[IDisposable](#) ↗

Constructors

[DeflateContent\(byte\[\]\)](#)

Initializes a new instance of compressing stream with the specified byte array content.

[DeflateContent\(Stream\)](#)

Initializes a new instance of compressing stream with the specified stream content.

[DeflateContent\(HttpContent\)](#)

Initializes a new instance of compressing stream with the specified inner HTTP content.

Methods

[GetCompressingStream\(Stream\)](#)

Gets a stream that compresses the output stream.

[Setup\(\)](#)

Represents the method that is invoked once within the constructor to setup this compressor. This method is indeeded to add the missing Content-Encoding headers used by this compressor.

Constructor DeflateContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

DeflateContent(HttpContent)

Initializes a new instance of compressing stream with the specified inner HTTP content.

```
public DeflateContent(HttpContent innerContent)
```

Parameters

innerContent [HttpContent](#)

The inner HTTP content.

DeflateContent(byte[])

Initializes a new instance of compressing stream with the specified byte array content.

```
public DeflateContent(byte[] byteArrayContent)
```

Parameters

byteArrayContent [byte](#)

The byte array content.

DeflateContent(Stream)

Initializes a new instance of compressing stream with the specified stream content.

```
public DeflateContent(Stream baseContent)
```

Parameters

baseContent Stream

The stream content.

Method GetCompressingStream

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetCompressingStream(Stream)

Gets a stream that compresses the output stream.

```
public override Stream GetCompressingStream(Stream outputStream)
```

Parameters

outputStream [Stream](#)

The output stream to compress.

Returns

[Stream](#)

A stream that compresses the output stream.

Method Setup

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Setup()

Represents the method that is invoked once within the constructor to setup this compressor. This method is indeeded to add the missing Content-Encoding headers used by this compressor.

```
public override void Setup()
```

Class ForwardingResolver

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Provides HTTP forwarding resolving methods that can be used to resolving the client remote address, host and protocol of a proxy, load balancer or CDN, through the HTTP request.

```
public abstract class ForwardingResolver
```

Constructors

[ForwardingResolver\(\)](#)

Methods

[OnResolveClientAddress\(HttpRequest, IPEndPoint\)](#)

Method that is called when resolving the IP address of the client in the request.

[OnResolveRequestHost\(HttpRequest, string\)](#)

Method that is called when resolving the client request host.

[OnResolveSecureConnection\(HttpRequest, bool\)](#)

Method that is called when resolving whether the HTTP request is using HTTPS or HTTP.

Constructor ForwardingResolver

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ForwardingResolver()

```
protected ForwardingResolver()
```

Method OnResolveClientAddress

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

OnResolveClientAddress(HttpRequest, IPPEndPoint)

Method that is called when resolving the IP address of the client in the request.

```
public virtual IPAddress OnResolveClientAddress(HttpRequest request,  
IPPEndPoint connectingEndpoint)
```

Parameters

request [HttpRequest](#)

The [HttpRequest](#) object which contains parameters of the request.

connectingEndpoint [IPPEndPoint](#)

The original connecting endpoint.

Returns

[IPAddress](#)

Method OnResolveRequestHost

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

OnResolveRequestHost(HttpRequest, string)

Method that is called when resolving the client request host.

```
public virtual string OnResolveRequestHost(HttpRequest request, string requestedHost)
```

Parameters

request [HttpRequest](#)

The [HttpRequest](#) object which contains parameters of the request.

requestedHost [string](#)

The original requested host.

Returns

[string](#)

Method OnResolveSecureConnection

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

OnResolveSecureConnection(HttpRequest, bool)

Method that is called when resolving whether the HTTP request is using HTTPS or HTTP.

```
public virtual bool OnResolveSecureConnection(HttpRequest request, bool isSecure)
```

Parameters

request [HttpRequest](#)

The [HttpRequest](#) object which contains parameters of the request.

isSecure [bool](#)

The original security state of the request.

Returns

[bool](#)

Class GZipContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Represents an HTTP content that is compressed using the GZip algorithm.

```
public sealed class GZipContent : CompressedContent, IDisposable
```

Implements

[IDisposable](#) ↗

Constructors

[GZipContent\(byte\[\]\)](#)

Initializes a new instance of compressing stream with the specified byte array content.

[GZipContent\(Stream\)](#)

Initializes a new instance of compressing stream with the specified stream content.

[GZipContent\(HttpContent\)](#)

Initializes a new instance of compressing stream with the specified inner HTTP content.

Methods

[GetCompressingStream\(Stream\)](#)

Gets a stream that compresses the output stream.

[Setup\(\)](#)

Represents the method that is invoked once within the constructor to setup this compressor. This method is indeeded to add the missing Content-Encoding headers used by this compressor.

Constructor GZipContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GZipContent(HttpContent)

Initializes a new instance of compressing stream with the specified inner HTTP content.

```
public GZipContent(HttpContent innerContent)
```

Parameters

innerContent [HttpContent](#)

The inner HTTP content.

GZipContent(byte[])

Initializes a new instance of compressing stream with the specified byte array content.

```
public GZipContent(byte[] byteArrayContent)
```

Parameters

byteArrayContent [byte](#)[]

The byte array content.

GZipContent(Stream)

Initializes a new instance of compressing stream with the specified stream content.

```
public GZipContent(Stream baseContent)
```

Parameters

baseContent Stream

The stream content.

Method GetCompressingStream

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetCompressingStream(Stream)

Gets a stream that compresses the output stream.

```
public override Stream GetCompressingStream(Stream outputStream)
```

Parameters

outputStream [Stream](#)

The output stream to compress.

Returns

[Stream](#)

A stream that compresses the output stream.

Method Setup

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Setup()

Represents the method that is invoked once within the constructor to setup this compressor. This method is indeeded to add the missing Content-Encoding headers used by this compressor.

```
public override void Setup()
```

Class HtmlContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Provides HTTP content based on HTML contents.

```
public class HtmlContent : StringContent, IDisposable
```

Implements

[IDisposable](#) ↗

Constructors

[HtmlContent\(ReadOnlySpan<byte>\)](#)

Creates a new [HtmlContent](#) class with given HTML content as a UTF-8 encoded byte span.

[HtmlContent\(ReadOnlySpan<byte>, Encoding\)](#)

Creates a new [HtmlContent](#) class with given HTML content as a byte span and encoding.

[HtmlContent\(string\)](#)

Creates an new [HtmlContent](#) class with given HTML content, using the environment default encoding.

[HtmlContent\(string, Encoding\)](#)

Creates an new [HtmlContent](#) class with given HTML content and encoding.

Constructor HtmlContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

HtmlContent(string, Encoding)

Creates an new [HtmlContent](#) class with given HTML content and encoding.

```
public HtmlContent(string content, Encoding encoding)
```

Parameters

content [string](#)

The HTML content string.

encoding [Encoding](#)

The encoding which will encode the HTML contents.

HtmlContent(string)

Creates an new [HtmlContent](#) class with given HTML content, using the environment default encoding.

```
public HtmlContent(string content)
```

Parameters

content [string](#)

The HTML content string.

HtmlContent(ReadOnlySpan<byte>, Encoding)

Creates a new [HtmlContent](#) class with given HTML content as a byte span and encoding.

```
public HtmlContent(ReadOnlySpan<byte> contents, Encoding encoding)
```

Parameters

contents [ReadOnlySpan](#)<byte>

The HTML content as a byte span.

encoding [Encoding](#)

The encoding which will decode the HTML contents.

HtmlContent(ReadOnlySpan<byte>)

Creates a new [HtmlContent](#) class with given HTML content as a UTF-8 encoded byte span.

```
public HtmlContent(ReadOnlySpan<byte> utf8Contents)
```

Parameters

utf8Contents [ReadOnlySpan](#)<byte>

The HTML content as a UTF-8 encoded byte span.

Class HttpContext

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Represents an context that is shared in a entire HTTP session.

```
public sealed class HttpContext
```

Properties

[Current](#)

Gets the current running [HttpContext](#).

[ExtraHeaders](#)

Gets or sets the [HttpHeaderCollection](#) indicating HTTP headers which will be added (not overwritten) in the final response.

[HttpServer](#)

Gets the context [HttpServer](#) instance.

[IsRequestContext](#)

Gets whether the current thread context is running inside an HTTP context.

[ListeningHost](#)

Gets the [ListeningHost](#) instance of this HTTP context.

[LogMode](#)

Gets or sets an [LogOutput](#) mode for this context, which will overwrite the matched route log mode option.

[MatchedRoute](#)

Gets the matched [Route](#) for this context.

[OverrideHeaders](#)

Gets or sets an [HttpHeaderCollection](#) indicating HTTP headers which will overwrite headers set by CORS, router response or request handlers.

[Request](#)

Gets the [HttpRequest](#) which is contained in this HTTP context.

[RequestBag](#)

Gets or sets a managed collection for this HTTP context.

[Router](#)

Gets the [Router](#) where this context was created.

[RouterResponse](#)

Gets the [HttpResponse](#) for this context. This property accessible when a post-executing [IRequestHandler](#) was executed for this router context.

Methods

[GetCurrentContext\(\)](#)

Gets the current running [HttpContext](#).

Property Current

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Current

Gets the current running [HttpContext](#).

```
public static HttpContext Current { get; }
```

Property Value

[HttpContext](#)

Remarks

This property is only accessible during an HTTP session, within the executing HTTP code.

Property ExtraHeaders

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ExtraHeaders

Gets or sets the [HttpHeaderCollection](#) indicating HTTP headers which will be added (not overwritten) in the final response.

```
public HttpHeadersCollection ExtraHeaders { get; set; }
```

Property Value

[HttpHeaderCollection](#)

Property HttpServer

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

HttpServer

Gets the context [HttpServer](#) instance.

```
public HttpServer HttpServer { get; }
```

Property Value

[HttpServer](#)

Property IsRequestContext

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

IsRequestContext

Gets whether the current thread context is running inside an HTTP context.

```
public static bool IsRequestContext { get; }
```

Property Value

bool

Property ListeningHost

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ListeningHost

Gets the [ListeningHost](#) instance of this HTTP context.

```
public ListeningHost? ListeningHost { get; }
```

Property Value

[ListeningHost](#)

Property LogMode

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

LogMode

Gets or sets an [LogOutput](#) mode for this context, which will overwrite the matched route log mode option.

```
public LogOutput? LogMode { get; set; }
```

Property Value

[LogOutput?](#)

Property MatchedRoute

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

MatchedRoute

Gets the matched [Route](#) for this context.

```
public Route? MatchedRoute { get; }
```

Property Value

[Route](#)

Property OverrideHeaders

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

OverrideHeaders

Gets or sets an [HttpHeaderCollection](#) indicating HTTP headers which will overwrite headers set by CORS, router response or request handlers.

```
public HttpHeadersOverride Headers { get; set; }
```

Property Value

[HttpHeaderCollection](#)

Remarks

This property replaces existing headers in the final response. Use [ExtraHeaders](#) to add headers without replacing existing ones.

Property Request

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Request

Gets the [HttpRequest](#) which is contained in this HTTP context.

```
public HttpRequest Request { get; }
```

Property Value

[HttpRequest](#)

Property RequestBag

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RequestBag

Gets or sets a managed collection for this HTTP context.

```
public TypedValueDictionary RequestBag { get; set; }
```

Property Value

[TypedValueDictionary](#)

Property Router

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Router

Gets the [Router](#) where this context was created.

```
public Router? Router { get; }
```

Property Value

[Router](#)

Property RouterResponse

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RouterResponse

Gets the [HttpResponse](#) for this context. This property accessible when a post-executing [IRequestHandler](#) was executed for this router context.

```
public HttpResponse? RouterResponse { get; }
```

Property Value

[HttpResponse](#)

Method GetCurrentContext

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetCurrentContext()

Gets the current running [HttpContext](#).

```
public static HttpContext? GetCurrentContext()
```

Returns

[HttpContext](#)

The current [HttpContext](#) instance, otherwise, [null](#).

Class HttpKnownHeaderNames

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Provides most of the most commonly known HTTP headers for constants.

```
public static class HttpKnownHeaderNames
```

Fields

[Accept](#)

The HTTP Accept header.

Specifies the media types that are acceptable for the response, allowing the client to indicate its preferences.

[AcceptCH](#)

The HTTP Accept-CH header.

Specifies the client hints that the server supports, allowing clients to provide additional information about their capabilities.

[AcceptCharset](#)

The HTTP Accept-Charset header.

Indicates the character sets that are acceptable for the response, allowing the client to specify its preferred encoding.

[AcceptEncoding](#)

The HTTP Accept-Encoding header.

Specifies the content encodings that are acceptable for the response, allowing the client to indicate its preferences for compression.

[AcceptLanguage](#)

The HTTP Accept-Language header.

Indicates the natural languages that are preferred for the response, allowing the client to specify its language preferences.

[AcceptPatch](#)

The HTTP Accept-Patch header.

Indicates the patch document formats that are acceptable for the response, allowing the client to specify its preferences for patching resources.

[AcceptPost](#)

The HTTP Accept-Post header.

Specifies which media types are accepted by the server in a POST request.

[AcceptRanges](#)

The HTTP Accept-Ranges header.

Indicates that the server supports range requests for the resource, allowing clients to request specific byte ranges.

[AccessControlAllowCredentials](#)

The HTTP Access-Control-Allow-Credentials header.

Indicates whether the response to the request can expose credentials, allowing cross-origin requests to include credentials.

[AccessControlAllowHeaders](#)

The HTTP Access-Control-Allow-Headers header.

Specifies which headers can be used when making the actual request in a cross-origin resource sharing (CORS) context.

[AccessControlAllowMethods](#)

The HTTP Access-Control-Allow-Methods header.

Specifies the methods that are allowed when accessing the resource in a CORS context.

[AccessControlAllowOrigin](#)

The HTTP Access-Control-Allow-Origin header.

Specifies which origins are allowed to access the resource in a CORS context, helping to control cross-origin requests.

[AccessControlExposeHeaders](#)

The HTTP Access-Control-Expose-Headers header.

Indicates which headers can be exposed as part of the response to a cross-origin request.

[AccessControlMaxAge](#)

The HTTP Access-Control-Max-Age header.

Specifies how long the results of a preflight request can be cached, reducing the number of preflight requests made.

[AccessControlRequestHeaders](#)

The HTTP Access-Control-Request-Headers header.

[AccessControlRequestMethod](#)

The HTTP Access-Control-Request-Method header.

[Age](#)

The HTTP Age header.

Indicates the age of the object in a cache, helping clients understand how fresh the cached response is.

[Allow](#)

The HTTP Allow header.

Lists the HTTP methods that are supported by the resource, informing clients about the available actions.

[AltSvc](#)

The HTTP Alt-Svc header.

Indicates that an alternative service is available for the resource, allowing clients to connect to a different server or protocol.

[Authorization](#)

The HTTP Authorization header.

Contains credentials for authenticating the client with the server, often used for basic or bearer token authentication.

[CacheControl](#)

The HTTP Cache-Control header.

Directs caching mechanisms on how to cache the response, including directives for expiration and revalidation.

[Connection](#)

The HTTP Connection header.

Controls whether the network connection stays open after the current transaction finishes, allowing for persistent connections.

[ContentDisposition](#)

The HTTP Content-Disposition header.

Indicates if the content should be displayed inline in the browser or treated as an attachment to be downloaded.

[ContentEncoding](#)

The HTTP Content-Encoding header.

Specifies the encoding transformations that have been applied to the response body, such as gzip or deflate.

[ContentLanguage](#)

The HTTP Content-Language header.

Indicates the natural language(s) of the intended audience for the response, helping clients understand the content's language.

[ContentLength](#)

The HTTP Content-Length header.

Indicates the size of the response body in bytes, allowing the client to know how much data to expect.

[ContentLocation](#)

The HTTP Content-Location header.

Indicates an alternate location for the returned data, often used for redirecting clients to a different resource.

[ContentMD5](#)

The HTTP Content-MD5 header.

Contains the MD5 hash of the response body, allowing clients to verify the integrity of the received data.

[ContentRange](#)

The HTTP Content-Range header.

Indicates the part of a document that the server is returning, used in range requests to specify byte ranges.

[ContentSecurityPolicy](#)

The HTTP Content-Security-Policy header.

Defines security policies for the content, helping to prevent cross-site scripting (XSS) and other code injection attacks.

[ContentType](#)

The HTTP Content-Type header.

Indicates the media type of the resource, allowing the client to understand how to process the response body.

[Cookie](#)

The HTTP Cookie header.

Contains stored HTTP cookies previously sent by the server, allowing the server to identify the client on subsequent requests.

[Cookie2](#)

The HTTP Cookie2 header.

Used to send cookies in a more advanced format, primarily for compatibility with older versions of HTTP.

[Date](#)

The HTTP Date header.

Indicates the date and time at which the message was sent, helping clients understand the freshness of the response.

[Dnt](#)

The HTTP DNT header.

The HTTP DNT (Do Not Track) request header indicates the user's tracking preference.

[ETag](#)

The HTTP ETag header.

Provides a unique identifier for a specific version of a resource, allowing clients to cache and validate resources efficiently.

[Expect](#)

The HTTP Expect header.

Indicates that the client expects certain behaviors from the server, such as support for specific features or conditions.

[Expires](#)

The HTTP Expires header.

Indicates the date and time after which the response is considered stale, helping clients manage caching.

[Host](#)

The HTTP Host header.

Specifies the domain name of the server and the TCP port number on which the server is listening, allowing for virtual hosting.

[IfMatch](#)

The HTTP If-Match header.

Used to make a conditional request, allowing the client to specify that the request should only be processed if the resource matches the given ETag.

IfModifiedSince

The HTTP If-Modified-Since header.

Used to make a conditional request, allowing the client to specify that the resource should only be returned if it has been modified since the given date.

IfNoneMatch

The HTTP If-None-Match header.

Used to make a conditional request, allowing the client to specify that the resource should only be returned if it does not match the given ETag.

IfRange

The HTTP If-Range header.

Used to make a conditional range request, allowing the client to specify that the range should only be returned if the resource has not changed.

IfUnmodifiedSince

The HTTP If-Unmodified-Since header.

Used to make a conditional request, allowing the client to specify that the resource should only be returned if it has not been modified since the given date.

KeepAlive

The HTTP Keep-Alive header.

Used to specify parameters for persistent connections, allowing the client and server to maintain an open connection for multiple requests.

LastModified

The HTTP Last-Modified header.

Indicates the date and time at which the resource was last modified, helping clients determine if they need to refresh their cached version.

Link

The HTTP Link header.

Used to provide relationships between the current resource and other resources, often used for navigation and linking.

Location

The HTTP Location header.

Used in redirection responses to indicate the URL to which the client should redirect.

MaxForwards

The HTTP Max-Forwards header.

Used in OPTIONS requests to limit the number of times the request can be forwarded by proxies.

Origin

The HTTP Origin header.

Indicates the origin of the request, helping servers implement CORS and manage cross-origin requests.

P3P

The HTTP P3P header.

Used to indicate the privacy policy of the server, allowing clients to understand how their data will be handled.

Pragma

The HTTP Pragma header.

Used to include implementation-specific directives that might apply to any recipient along the request/response chain.

ProxyAuthenticate

The HTTP Proxy-Authenticate header.

Used by a proxy server to request authentication from the client, indicating the authentication method required.

ProxyAuthorization

The HTTP Proxy-Authorization header.

Contains credentials for authenticating the client with a proxy server, allowing access to the requested resource.

ProxyConnection

The HTTP Proxy-Connection header.

Used to control whether the network connection to the proxy server should be kept open after the current transaction.

PublicKeyPins

The HTTP Public-Key-Pins header.

Used to prevent man-in-the-middle attacks by specifying which public keys are valid for the server's certificate.

Range

The HTTP Range header.

Used to request a specific range of bytes from a resource, allowing clients to download large files in parts.

Referer

The HTTP Referer header.

Indicates the URL of the resource from which the request originated, helping servers understand the source of traffic.

RetryAfter

The HTTP Retry-After header.

Indicates how long the client should wait before making a follow-up request, often used in rate limiting scenarios.

SecGPC

The HTTP Sec-GPC header.

The HTTP Sec-GPC request header is part of the Global Privacy Control (GPC) mechanism to indicate whether the user consents to a website or service selling or sharing their personal information with third parties.

SecWebSocketAccept

The HTTP Sec-WebSocket-Accept header.

Used in the WebSocket handshake to confirm the server's acceptance of the connection request.

SecWebSocketExtensions

The HTTP Sec-WebSocket-Extensions header.

Used to negotiate WebSocket extensions during the handshake, allowing for additional features and capabilities.

SecWebSocketKey

The HTTP Sec-WebSocket-Key header.

Contains a base64-encoded value used to establish a WebSocket connection, ensuring the request is valid.

SecWebSocketProtocol

The HTTP Sec-WebSocket-Protocol header.

Used to specify subprotocols that the client wishes to use during the WebSocket connection.

SecWebSocketVersion

The HTTP Sec-WebSocket-Version header.

Indicates the version of the WebSocket protocol that the client wishes to use.

Server

The HTTP Server header.

Contains information about the server software handling the request, often used for informational purposes.

SetCookie

The HTTP Set-Cookie header.

Used to send cookies from the server to the client, allowing the server to store state information on the client.

SetCookie2

The HTTP Set-Cookie2 header.

Used to send cookies in a more advanced format, primarily for compatibility with older versions of HTTP.

StrictTransportSecurity

The HTTP Strict-Transport-Security header.

Enforces secure (HTTPS) connections to the server, helping to prevent man-in-the-middle attacks.

TE

The HTTP TE header.

Indicates the transfer encodings that are acceptable for the response, allowing for content negotiation.

TSV

The HTTP TSV header.

Used to indicate the type of data being sent in a transaction, often used in specific applications or protocols.

Trailer

The HTTP Trailer header.

Indicates that the sender will include additional fields in the message trailer, which can be used for metadata.

TransferEncoding

The HTTP Transfer-Encoding header.

Specifies the form of encoding used to safely transfer the payload body to the user.

Upgrade

The HTTP Upgrade header.

Indicates that the client prefers to upgrade to a different protocol, such as switching from HTTP/1.1 to HTTP/2.

UpgradelnsecureRequests

The HTTP Upgrade-Insecure-Requests header.

Indicates that the client prefers to receive an upgraded version of the resource over HTTPS instead of HTTP.

User-Agent

The HTTP User-Agent header.

Contains information about the user agent (browser or application) making the request, including its version and platform.

Vary

The HTTP Vary header.

Indicates that the response varies based on the value of the specified request headers, allowing for content negotiation.

Via

The HTTP Via header.

Used to track message forwards and proxies, indicating the intermediate protocols and recipients involved in the request/response chain.

WWWAuthenticate

The HTTP WWW-Authenticate header.

Used in response to a request for authentication, indicating the authentication method that should be used to access the resource.

Warning

The HTTP Warning header.

Provides additional information about the status or transformation of a message, often used for caching and validation.

XContentDuration

The HTTP X-Content-Duration header.

Specifies the duration of the content in seconds, often used for media files.

XContentTypeOptions

The HTTP X-Content-Type-Options header.

Used to prevent MIME type sniffing, ensuring that the browser respects the declared content type.

XForwardedFor

The HTTP X-Forwarded-For header.

Used to identify the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer.

XForwardedHost

The HTTP X-Forwarded-Host header.

Used to identify the original host requested by the client in the Host HTTP request header, often used in proxy setups.

XFrameOptions

The HTTP X-Frame-Options header.

Used to control whether a browser should be allowed to render a page in a iframe, frame, embed or object tag, helping to prevent clickjacking attacks.

XPoweredBy

The HTTP X-Powered-By header.

Indicates the technology or framework that powers the web application, often used for informational purposes.

XRequestID

The HTTP X-Request-ID header.

Used to uniquely identify a request for tracking and debugging purposes, often generated by the client or server.

XUACompatible

The HTTP X-UA-Compatible header.

Used to specify the document mode that Internet Explorer should use to render the page, helping to ensure compatibility with older versions.

Field Accept

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Accept header.

Specifies the media types that are acceptable for the response, allowing the client to indicate its preferences.

```
public const string Accept = "Accept"
```

Returns

[string](#)

The HTTP Accept header. Specifies the media types that are acceptable for the response, allowing the client to indicate its preferences.

Field AcceptCH

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Accept-CH header.

Specifies the client hints that the server supports, allowing clients to provide additional information about their capabilities.

```
public const string AcceptCH = "Accept-CH"
```

Returns

[string](#)

The HTTP Accept-CH header. Specifies the client hints that the server supports, allowing clients to provide additional information about their capabilities.

Field AcceptCharset

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Accept-Charset header.

Indicates the character sets that are acceptable for the response, allowing the client to specify its preferred encoding.

```
public const string AcceptCharset = "Accept-Charset"
```

Returns

[string](#)

The HTTP Accept-Charset header. Indicates the character sets that are acceptable for the response, allowing the client to specify its preferred encoding.

Field AcceptEncoding

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Accept-Encoding header.

Specifies the content encodings that are acceptable for the response, allowing the client to indicate its preferences for compression.

```
public const string AcceptEncoding = "Accept-Encoding"
```

Returns

[string](#)

The HTTP Accept-Encoding header. Specifies the content encodings that are acceptable for the response, allowing the client to indicate its preferences for compression.

Field AcceptLanguage

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Accept-Language header.

Indicates the natural languages that are preferred for the response, allowing the client to specify its language preferences.

```
public const string AcceptLanguage = "Accept-Language"
```

Returns

[string](#)

The HTTP Accept-Language header. Indicates the natural languages that are preferred for the response, allowing the client to specify its language preferences.

Field AcceptPatch

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Accept-Patch header.

Indicates the patch document formats that are acceptable for the response, allowing the client to specify its preferences for patching resources.

```
public const string AcceptPatch = "Accept-Patch"
```

Returns

[string](#)

The HTTP Accept-Patch header. Indicates the patch document formats that are acceptable for the response, allowing the client to specify its preferences for patching resources.

Field AcceptPost

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Accept-Post header.

Specifies which media types are accepted by the server in a POST request.

```
public const string AcceptPost = "Accept-Post"
```

Returns

[string](#)

The HTTP Accept-Post header. Specifies which media types are accepted by the server in a POST request.

Field AcceptRanges

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Accept-Ranges header.

Indicates that the server supports range requests for the resource, allowing clients to request specific byte ranges.

```
public const string AcceptRanges = "Accept-Ranges"
```

Returns

[string](#)

The HTTP Accept-Ranges header. Indicates that the server supports range requests for the resource, allowing clients to request specific byte ranges.

Field AccessControlAllowCredentials

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Access-Control-Allow-Credentials header.

Indicates whether the response to the request can expose credentials, allowing cross-origin requests to include credentials.

```
public const string AccessControlAllowCredentials = "Access-Control-Allow-Credentials"
```

Returns

[string](#)

The HTTP Access-Control-Allow-Credentials header. Indicates whether the response to the request can expose credentials, allowing cross-origin requests to include credentials.

Field AccessControlAllowHeaders

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Access-Control-Allow-Headers header.

Specifies which headers can be used when making the actual request in a cross-origin resource sharing (CORS) context.

```
public const string AccessControlAllowHeaders = "Access-Control-Allow-Headers"
```

Returns

[string](#)

The HTTP Access-Control-Allow-Headers header. Specifies which headers can be used when making the actual request in a cross-origin resource sharing (CORS) context.

Field AccessControlAllowMethods

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Access-Control-Allow-Methods header.

Specifies the methods that are allowed when accessing the resource in a CORS context.

```
public const string AccessControlAllowMethods = "Access-Control-Allow-Methods"
```

Returns

[string](#)

The HTTP Access-Control-Allow-Methods header. Specifies the methods that are allowed when accessing the resource in a CORS context.

Field AccessControlAllowOrigin

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Access-Control-Allow-Origin header.

Specifies which origins are allowed to access the resource in a CORS context, helping to control cross-origin requests.

```
public const string AccessControlAllowOrigin = "Access-Control-Allow-Origin"
```

Returns

[string](#)

The HTTP Access-Control-Allow-Origin header. Specifies which origins are allowed to access the resource in a CORS context, helping to control cross-origin requests.

Field AccessControlExposeHeaders

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Access-Control-Expose-Headers header.

Indicates which headers can be exposed as part of the response to a cross-origin request.

```
public const string AccessControlExposeHeaders = "Access-Control-Expose-Headers"
```

Returns

[string](#)

The HTTP Access-Control-Expose-Headers header. Indicates which headers can be exposed as part of the response to a cross-origin request.

Field AccessControlMaxAge

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Access-Control-Max-Age header.

Specifies how long the results of a preflight request can be cached, reducing the number of preflight requests made.

```
public const string AccessControlMaxAge = "Access-Control-Max-Age"
```

Returns

[string](#)

The HTTP Access-Control-Max-Age header. Specifies how long the results of a preflight request can be cached, reducing the number of preflight requests made.

Field AccessControlRequestHeaders

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Access-Control-Request-Headers header.

```
public const string AccessControlRequestHeaders = "Access-Control-Request-Headers"
```

Returns

[string](#)

The HTTP Access-Control-Request-Headers header.

Field AccessControlRequestMethod

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Access-Control-Request-Method header.

```
public const string AccessControlRequestMethod = "Access-Control-Request-Method"
```

Returns

[string](#)

The HTTP Access-Control-Request-Method header.

Field Age

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Age header.

Indicates the age of the object in a cache, helping clients understand how fresh the cached response is.

```
public const string Age = "Age"
```

Returns

[string](#)

The HTTP Age header. Indicates the age of the object in a cache, helping clients understand how fresh the cached response is.

Field Allow

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Allow header.

Lists the HTTP methods that are supported by the resource, informing clients about the available actions.

```
public const string Allow = "Allow"
```

Returns

[string](#)

The HTTP Allow header. Lists the HTTP methods that are supported by the resource, informing clients about the available actions.

Field AltSvc

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Alt-Svc header.

Indicates that an alternative service is available for the resource, allowing clients to connect to a different server or protocol.

```
public const string AltSvc = "Alt-Svc"
```

Returns

[string](#)

The HTTP Alt-Svc header. Indicates that an alternative service is available for the resource, allowing clients to connect to a different server or protocol.

Field Authorization

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Authorization header.

Contains credentials for authenticating the client with the server, often used for basic or bearer token authentication.

```
public const string Authorization = "Authorization"
```

Returns

[string](#)

The HTTP Authorization header. Contains credentials for authenticating the client with the server, often used for basic or bearer token authentication.

Field CacheControl

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Cache-Control header.

Directs caching mechanisms on how to cache the response, including directives for expiration and revalidation.

```
public const string CacheControl = "Cache-Control"
```

Returns

[string](#)

The HTTP Cache-Control header. Directs caching mechanisms on how to cache the response, including directives for expiration and revalidation.

Field Connection

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Connection header.

Controls whether the network connection stays open after the current transaction finishes, allowing for persistent connections.

```
public const string Connection = "Connection"
```

Returns

[string](#)

The HTTP Connection header. Controls whether the network connection stays open after the current transaction finishes, allowing for persistent connections.

Field ContentDisposition

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Content-Disposition header.

Indicates if the content should be displayed inline in the browser or treated as an attachment to be downloaded.

```
public const string ContentDisposition = "Content-Disposition"
```

Returns

[string](#)

The HTTP Content-Disposition header. Indicates if the content should be displayed inline in the browser or treated as an attachment to be downloaded.

Field ContentEncoding

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Content-Encoding header.

Specifies the encoding transformations that have been applied to the response body, such as gzip or deflate.

```
public const string ContentEncoding = "Content-Encoding"
```

Returns

[string](#)

The HTTP Content-Encoding header. Specifies the encoding transformations that have been applied to the response body, such as gzip or deflate.

Field ContentLanguage

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Content-Language header.

Indicates the natural language(s) of the intended audience for the response, helping clients understand the content's language.

```
public const string ContentLanguage = "Content-Language"
```

Returns

[string](#)

The HTTP Content-Language header. Indicates the natural language(s) of the intended audience for the response, helping clients understand the content's language.

Field ContentLength

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Content-Length header.

Indicates the size of the response body in bytes, allowing the client to know how much data to expect.

```
public const string ContentLength = "Content-Length"
```

Returns

[string](#)

The HTTP Content-Length header. Indicates the size of the response body in bytes, allowing the client to know how much data to expect.

Field ContentLocation

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Content-Location header.

Indicates an alternate location for the returned data, often used for redirecting clients to a different resource.

```
public const string ContentLocation = "Content-Location"
```

Returns

[string](#)

The HTTP Content-Location header. Indicates an alternate location for the returned data, often used for redirecting clients to a different resource.

Field ContentMD5

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Content-MD5 header.

Contains the MD5 hash of the response body, allowing clients to verify the integrity of the received data.

```
public const string ContentMD5 = "Content-MD5"
```

Returns

[string](#)

The HTTP Content-MD5 header. Contains the MD5 hash of the response body, allowing clients to verify the integrity of the received data.

Field ContentRange

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Content-Range header.

Indicates the part of a document that the server is returning, used in range requests to specify byte ranges.

```
public const string ContentRange = "Content-Range"
```

Returns

[string](#)

The HTTP Content-Range header. Indicates the part of a document that the server is returning, used in range requests to specify byte ranges.

Field ContentSecurityPolicy

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Content-Security-Policy header.

Defines security policies for the content, helping to prevent cross-site scripting (XSS) and other code injection attacks.

```
public const string ContentSecurityPolicy = "Content-Security-Policy"
```

Returns

[string](#)

The HTTP Content-Security-Policy header. Defines security policies for the content, helping to prevent cross-site scripting (XSS) and other code injection attacks.

Field ContentType

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Content-Type header.

Indicates the media type of the resource, allowing the client to understand how to process the response body.

```
public const string ContentType = "Content-Type"
```

Returns

[string](#)

The HTTP Content-Type header. Indicates the media type of the resource, allowing the client to understand how to process the response body.

Field Cookie

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Cookie header.

Contains stored HTTP cookies previously sent by the server, allowing the server to identify the client on subsequent requests.

```
public const string Cookie = "Cookie"
```

Returns

[string](#)

The HTTP Cookie header. Contains stored HTTP cookies previously sent by the server, allowing the server to identify the client on subsequent requests.

Field Cookie2

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Cookie2 header.

Used to send cookies in a more advanced format, primarily for compatibility with older versions of HTTP.

```
public const string Cookie2 = "Cookie2"
```

Returns

[string](#)

The HTTP Cookie2 header. Used to send cookies in a more advanced format, primarily for compatibility with older versions of HTTP.

Field Date

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Date header.

Indicates the date and time at which the message was sent, helping clients understand the freshness of the response.

```
public const string Date = "Date"
```

Returns

[string](#)

The HTTP Date header. Indicates the date and time at which the message was sent, helping clients understand the freshness of the response.

Field Dnt

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP DNT header.

The HTTP DNT (Do Not Track) request header indicates the user's tracking preference.

```
public const string Dnt = "DNT"
```

Returns

[string](#)

The HTTP DNT header. The HTTP DNT (Do Not Track) request header indicates the user's tracking preference.

Field ETag

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP ETag header.

Provides a unique identifier for a specific version of a resource, allowing clients to cache and validate resources efficiently.

```
public const string ETag = "ETag"
```

Returns

[string](#)

The HTTP ETag header. Provides a unique identifier for a specific version of a resource, allowing clients to cache and validate resources efficiently.

Field Expect

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Expect header.

Indicates that the client expects certain behaviors from the server, such as support for specific features or conditions.

```
public const string Expect = "Expect"
```

Returns

[string](#)

The HTTP Expect header. Indicates that the client expects certain behaviors from the server, such as support for specific features or conditions.

Field Expires

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Expires header.

Indicates the date and time after which the response is considered stale, helping clients manage caching.

```
public const string Expires = "Expires"
```

Returns

[string](#)

The HTTP Expires header. Indicates the date and time after which the response is considered stale, helping clients manage caching.

Field Host

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Host header.

Specifies the domain name of the server and the TCP port number on which the server is listening, allowing for virtual hosting.

```
public const string Host = "Host"
```

Returns

[string](#)

The HTTP Host header. Specifies the domain name of the server and the TCP port number on which the server is listening, allowing for virtual hosting.

Field IfMatch

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP If-Match header.

Used to make a conditional request, allowing the client to specify that the request should only be processed if the resource matches the given ETag.

```
public const string IfMatch = "If-Match"
```

Returns

[string](#)

The HTTP If-Match header. Used to make a conditional request, allowing the client to specify that the request should only be processed if the resource matches the given ETag.

Field IfModifiedSince

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP If-Modified-Since header.

Used to make a conditional request, allowing the client to specify that the resource should only be returned if it has been modified since the given date.

```
public const string IfModifiedSince = "If-Modified-Since"
```

Returns

[string](#)

The HTTP If-Modified-Since header. Used to make a conditional request, allowing the client to specify that the resource should only be returned if it has been modified since the given date.

Field IfNoneMatch

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP If-None-Match header.

Used to make a conditional request, allowing the client to specify that the resource should only be returned if it does not match the given ETag.

```
public const string IfNoneMatch = "If-None-Match"
```

Returns

[string](#)

The HTTP If-None-Match header. Used to make a conditional request, allowing the client to specify that the resource should only be returned if it does not match the given ETag.

Field IfRange

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP If-Range header.

Used to make a conditional range request, allowing the client to specify that the range should only be returned if the resource has not changed.

```
public const string IfRange = "If-Range"
```

Returns

[string](#)

The HTTP If-Range header. Used to make a conditional range request, allowing the client to specify that the range should only be returned if the resource has not changed.

Field IfUnmodifiedSince

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP If-Unmodified-Since header.

Used to make a conditional request, allowing the client to specify that the resource should only be returned if it has not been modified since the given date.

```
public const string IfUnmodifiedSince = "If-Unmodified-Since"
```

Returns

[string](#)

The HTTP If-Unmodified-Since header. Used to make a conditional request, allowing the client to specify that the resource should only be returned if it has not been modified since the given date.

Field KeepAlive

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Keep-Alive header.

Used to specify parameters for persistent connections, allowing the client and server to maintain an open connection for multiple requests.

```
public const string KeepAlive = "Keep-Alive"
```

Returns

[string](#)

The HTTP Keep-Alive header. Used to specify parameters for persistent connections, allowing the client and server to maintain an open connection for multiple requests.

Field LastModified

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Last-Modified header.

Indicates the date and time at which the resource was last modified, helping clients determine if they need to refresh their cached version.

```
public const string LastModified = "Last-Modified"
```

Returns

[string](#)

The HTTP Last-Modified header. Indicates the date and time at which the resource was last modified, helping clients determine if they need to refresh their cached version.

Field Link

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Link header.

Used to provide relationships between the current resource and other resources, often used for navigation and linking.

```
public const string Link = "Link"
```

Returns

[string](#)

The HTTP Link header. Used to provide relationships between the current resource and other resources, often used for navigation and linking.

Field Location

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Location header.

Used in redirection responses to indicate the URL to which the client should redirect.

```
public const string Location = "Location"
```

Returns

[string](#)

The HTTP Location header. Used in redirection responses to indicate the URL to which the client should redirect.

Field MaxForwards

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Max-Forwards header.

Used in OPTIONS requests to limit the number of times the request can be forwarded by proxies.

```
public const string MaxForwards = "Max-Forwards"
```

Returns

[string](#)

The HTTP Max-Forwards header. Used in OPTIONS requests to limit the number of times the request can be forwarded by proxies.

Field Origin

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Origin header.

Indicates the origin of the request, helping servers implement CORS and manage cross-origin requests.

```
public const string Origin = "Origin"
```

Returns

[string](#)

The HTTP Origin header. Indicates the origin of the request, helping servers implement CORS and manage cross-origin requests.

Field P3P

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP P3P header.

Used to indicate the privacy policy of the server, allowing clients to understand how their data will be handled.

```
public const string P3P = "P3P"
```

Returns

[string](#)

The HTTP P3P header. Used to indicate the privacy policy of the server, allowing clients to understand how their data will be handled.

Field Pragma

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Pragma header.

Used to include implementation-specific directives that might apply to any recipient along the request/response chain.

```
public const string Pragma = "Pragma"
```

Returns

[string](#)

The HTTP Pragma header. Used to include implementation-specific directives that might apply to any recipient along the request/response chain.

Field ProxyAuthenticate

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Proxy-Authenticate header.

Used by a proxy server to request authentication from the client, indicating the authentication method required.

```
public const string ProxyAuthenticate = "Proxy-Authenticate"
```

Returns

[string](#)

The HTTP Proxy-Authenticate header. Used by a proxy server to request authentication from the client, indicating the authentication method required.

Field ProxyAuthorization

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Proxy-Authorization header.

Contains credentials for authenticating the client with a proxy server, allowing access to the requested resource.

```
public const string ProxyAuthorization = "Proxy-Authorization"
```

Returns

[string](#)

The HTTP Proxy-Authorization header. Contains credentials for authenticating the client with a proxy server, allowing access to the requested resource.

Field ProxyConnection

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Proxy-Connection header.

Used to control whether the network connection to the proxy server should be kept open after the current transaction.

```
public const string ProxyConnection = "Proxy-Connection"
```

Returns

[string](#)

The HTTP Proxy-Connection header. Used to control whether the network connection to the proxy server should be kept open after the current transaction.

Field PublicKeyPins

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Public-Key-Pins header.

Used to prevent man-in-the-middle attacks by specifying which public keys are valid for the server's certificate.

```
public const string PublicKeyPins = "Public-Key-Pins"
```

Returns

[string](#)

The HTTP Public-Key-Pins header. Used to prevent man-in-the-middle attacks by specifying which public keys are valid for the server's certificate.

Field Range

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Range header.

Used to request a specific range of bytes from a resource, allowing clients to download large files in parts.

```
public const string Range = "Range"
```

Returns

[string](#)

The HTTP Range header. Used to request a specific range of bytes from a resource, allowing clients to download large files in parts.

Field Referer

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Referer header.

Indicates the URL of the resource from which the request originated, helping servers understand the source of traffic.

```
public const string Referer = "Referer"
```

Returns

[string](#)

The HTTP Referer header. Indicates the URL of the resource from which the request originated, helping servers understand the source of traffic.

Field RetryAfter

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Retry-After header.

Indicates how long the client should wait before making a follow-up request, often used in rate limiting scenarios.

```
public const string RetryAfter = "Retry-After"
```

Returns

[string](#)

The HTTP Retry-After header. Indicates how long the client should wait before making a follow-up request, often used in rate limiting scenarios.

Field SecGPC

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Sec-GPC header.

The HTTP Sec-GPC request header is part of the Global Privacy Control (GPC) mechanism to indicate whether the user consents to a website or service selling or sharing their personal information with third parties.

```
public const string SecGPC = "Sec-GPC"
```

Returns

[string](#)

The HTTP Sec-GPC header. The HTTP Sec-GPC request header is part of the Global Privacy Control (GPC) mechanism to indicate whether the user consents to a website or service selling or sharing their personal information with third parties.

Field SecWebSocketAccept

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Sec-WebSocket-Accept header.

Used in the WebSocket handshake to confirm the server's acceptance of the connection request.

```
public const string SecWebSocketAccept = "Sec-WebSocket-Accept"
```

Returns

[string](#)

The HTTP Sec-WebSocket-Accept header. Used in the WebSocket handshake to confirm the server's acceptance of the connection request.

Field SecWebSocketExtensions

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Sec-WebSocket-Extensions header.

Used to negotiate WebSocket extensions during the handshake, allowing for additional features and capabilities.

```
public const string SecWebSocketExtensions = "Sec-WebSocket-Extensions"
```

Returns

[string](#)

The HTTP Sec-WebSocket-Extensions header. Used to negotiate WebSocket extensions during the handshake, allowing for additional features and capabilities.

Field SecWebSocketKey

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Sec-WebSocket-Key header.

Contains a base64-encoded value used to establish a WebSocket connection, ensuring the request is valid.

```
public const string SecWebSocketKey = "Sec-WebSocket-Key"
```

Returns

[string](#)

The HTTP Sec-WebSocket-Key header. Contains a base64-encoded value used to establish a WebSocket connection, ensuring the request is valid.

Field SecWebSocketProtocol

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Sec-WebSocket-Protocol header.

Used to specify subprotocols that the client wishes to use during the WebSocket connection.

```
public const string SecWebSocketProtocol = "Sec-WebSocket-Protocol"
```

Returns

[string](#)

The HTTP Sec-WebSocket-Protocol header. Used to specify subprotocols that the client wishes to use during the WebSocket connection.

Field SecWebSocketVersion

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Sec-WebSocket-Version header.

Indicates the version of the WebSocket protocol that the client wishes to use.

```
public const string SecWebSocketVersion = "Sec-WebSocket-Version"
```

Returns

[string](#)

The HTTP Sec-WebSocket-Version header. Indicates the version of the WebSocket protocol that the client wishes to use.

Field Server

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Server header.

Contains information about the server software handling the request, often used for informational purposes.

```
public const string Server = "Server"
```

Returns

[string](#)

The HTTP Server header. Contains information about the server software handling the request, often used for informational purposes.

Field SetCookie

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Set-Cookie header.

Used to send cookies from the server to the client, allowing the server to store state information on the client.

```
public const string SetCookie = "Set-Cookie"
```

Returns

[string](#)

The HTTP Set-Cookie header. Used to send cookies from the server to the client, allowing the server to store state information on the client.

Field SetCookie2

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Set-Cookie2 header.

Used to send cookies in a more advanced format, primarily for compatibility with older versions of HTTP.

```
public const string SetCookie2 = "Set-Cookie2"
```

Returns

[string](#)

The HTTP Set-Cookie2 header. Used to send cookies in a more advanced format, primarily for compatibility with older versions of HTTP.

Field StrictTransportSecurity

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Strict-Transport-Security header.

Enforces secure (HTTPS) connections to the server, helping to prevent man-in-the-middle attacks.

```
public const string StrictTransportSecurity = "Strict-Transport-Security"
```

Returns

[string](#)

The HTTP Strict-Transport-Security header. Enforces secure (HTTPS) connections to the server, helping to prevent man-in-the-middle attacks.

Field TE

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP TE header.

Indicates the transfer encodings that are acceptable for the response, allowing for content negotiation.

```
public const string TE = "TE"
```

Returns

[string](#)

The HTTP TE header. Indicates the transfer encodings that are acceptable for the response, allowing for content negotiation.

Field TSV

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP TSV header.

Used to indicate the type of data being sent in a transaction, often used in specific applications or protocols.

```
public const string TSV = "TSV"
```

Returns

[string](#)

The HTTP TSV header. Used to indicate the type of data being sent in a transaction, often used in specific applications or protocols.

Field Trailer

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Trailer header.

Indicates that the sender will include additional fields in the message trailer, which can be used for metadata.

```
public const string Trailer = "Trailer"
```

Returns

[string](#)

The HTTP Trailer header. Indicates that the sender will include additional fields in the message trailer, which can be used for metadata.

Field TransferEncoding

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Transfer-Encoding header.

Specifies the form of encoding used to safely transfer the payload body to the user.

```
public const string TransferEncoding = "Transfer-Encoding"
```

Returns

[string](#)

The HTTP Transfer-Encoding header. Specifies the form of encoding used to safely transfer the payload body to the user.

Field Upgrade

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Upgrade header.

Indicates that the client prefers to upgrade to a different protocol, such as switching from HTTP/1.1 to HTTP/2.

```
public const string Upgrade = "Upgrade"
```

Returns

[string](#)

The HTTP Upgrade header. Indicates that the client prefers to upgrade to a different protocol, such as switching from HTTP/1.1 to HTTP/2.

Field UpgradeInsecureRequests

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Upgrade-Insecure-Requests header.

Indicates that the client prefers to receive an upgraded version of the resource over HTTPS instead of HTTP.

```
public const string UpgradeInsecureRequests = "Upgrade-Insecure-Requests"
```

Returns

[string](#)

The HTTP Upgrade-Insecure-Requests header. Indicates that the client prefers to receive an upgraded version of the resource over HTTPS instead of HTTP.

Field UserAgent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP User-Agent header.

Contains information about the user agent (browser or application) making the request, including its version and platform.

```
public const string UserAgent = "User-Agent"
```

Returns

[string](#)

The HTTP User-Agent header. Contains information about the user agent (browser or application) making the request, including its version and platform.

Field Vary

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Vary header.

Indicates that the response varies based on the value of the specified request headers, allowing for content negotiation.

```
public const string Vary = "Vary"
```

Returns

[string](#)

The HTTP Vary header. Indicates that the response varies based on the value of the specified request headers, allowing for content negotiation.

Field Via

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Via header.

Used to track message forwards and proxies, indicating the intermediate protocols and recipients involved in the request/response chain.

```
public const string Via = "Via"
```

Returns

[string](#)

The HTTP Via header. Used to track message forwards and proxies, indicating the intermediate protocols and recipients involved in the request/response chain.

Field WWWAuthenticate

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP WWW-Authenticate header.

Used in response to a request for authentication, indicating the authentication method that should be used to access the resource.

```
public const string WWWAuthenticate = "WWW-Authenticate"
```

Returns

[string](#)

The HTTP WWW-Authenticate header. Used in response to a request for authentication, indicating the authentication method that should be used to access the resource.

Field Warning

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP Warning header.

Provides additional information about the status or transformation of a message, often used for caching and validation.

```
public const string Warning = "Warning"
```

Returns

[string](#)

The HTTP Warning header. Provides additional information about the status or transformation of a message, often used for caching and validation.

Field XContentDuration

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP X-Content-Duration header.

Specifies the duration of the content in seconds, often used for media files.

```
public const string XContentDuration = "X-Content-Duration"
```

Returns

[string](#)

The HTTP X-Content-Duration header. Specifies the duration of the content in seconds, often used for media files.

Field XContentTypeOptions

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP X-Content-Type-Options header.

Used to prevent MIME type sniffing, ensuring that the browser respects the declared content type.

```
public const string XContentTypeOptions = "X-Content-Type-Options"
```

Returns

[string](#)

The HTTP X-Content-Type-Options header. Used to prevent MIME type sniffing, ensuring that the browser respects the declared content type.

Field XForwardedFor

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP X-Forwarded-For header.

Used to identify the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer.

```
public const string XForwardedFor = "X-Forwarded-For"
```

Returns

[string](#)

The HTTP X-Forwarded-For header. Used to identify the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer.

Field XForwardedHost

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP X-Forwarded-Host header.

Used to identify the original host requested by the client in the Host HTTP request header, often used in proxy setups.

```
public const string XForwardedHost = "X-Forwarded-Host"
```

Returns

[string](#)

The HTTP X-Forwarded-Host header. Used to identify the original host requested by the client in the Host HTTP request header, often used in proxy setups.

Field XFrameOptions

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP X-Frame-Options header.

Used to control whether a browser should be allowed to render a page in a iframe, frame, embed or object tag, helping to prevent clickjacking attacks.

```
public const string XFrameOptions = "X-Frame-Options"
```

Returns

[string](#)

The HTTP X-Frame-Options header. Used to control whether a browser should be allowed to render a page in a iframe, frame, embed or object tag, helping to prevent clickjacking attacks.

Field XPoweredBy

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP X-Powered-By header.

Indicates the technology or framework that powers the web application, often used for informational purposes.

```
public const string XPoweredBy = "X-Powered-By"
```

Returns

[string](#)

The HTTP X-Powered-By header. Indicates the technology or framework that powers the web application, often used for informational purposes.

Field XRequestID

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP X-Request-ID header.

Used to uniquely identify a request for tracking and debugging purposes, often generated by the client or server.

```
public const string XRequestID = "X-Request-ID"
```

Returns

[string](#)

The HTTP X-Request-ID header. Used to uniquely identify a request for tracking and debugging purposes, often generated by the client or server.

Field XUACOMPATIBLE

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

The HTTP X-UA-Compatible header.

Used to specify the document mode that Internet Explorer should use to render the page, helping to ensure compatibility with older versions.

```
public const string XUACOMPATIBLE = "X-UA-Compatible"
```

Returns

[string](#)

The HTTP X-UA-Compatible header. Used to specify the document mode that Internet Explorer should use to render the page, helping to ensure compatibility with older versions.

Class HttpRequest

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Represents an HTTP request received by a Sisk server.

```
public sealed class HttpRequest : IDisposable
```

Implements

[IDisposable](#) ↗

Properties

[Authority](#)

Get the requested host header with the port from this HTTP request.

[Bag](#)

Gets the managed object which holds data for an entire HTTP session.

[Body](#)

Gets the HTTP request body as string, decoded by the request content encoding.

[ContentLength](#)

Gets the content length in bytes count.

[Context](#)

Gets the [HttpContext](#) for this request.

[Cookies](#)

Gets an [StringKeyStoreCollection](#) object with all cookies set in this request.

[DefaultJsonSerializerOptions](#)

Gets or sets the default options used for JSON serialization.

[DisconnectToken](#)

Gets a cancellation token that is signaled when the client disconnects.

FullPath

Gets the raw, full HTTP request path with the query string.

FullUrl

Gets the full URL for this request, with scheme, host, port, path and query.

HasContents

Gets a boolean indicating whether this request has body contents.

Headers

Gets the HTTP request headers.

Host

Get the requested host (without port) for this [HttpRequest](#).

IsContentAvailable

Gets a boolean indicating whether this request has body contents and whether it has already been read into memory by the server.

IsSecure

Gets a boolean indicating whether this request was locally made by an secure transport context (SSL/TLS) or not.

Method

Gets the HTTP request method.

Path

Gets the HTTP request path without the query string.

Query

Gets the HTTP request query value collection.

QueryString

Gets the HTTP request URL raw query string, including the '?' char.

RawBody

Gets the HTTP request body as a byte array.

RemoteAddress

Gets the incoming local IP address from the request.

RequestEncoding

Gets an string [Encoding](#) that can be used to decode text in this HTTP request.

RequestId

Gets a unique random ID for this request.

RequestedAt

Gets the moment which the request was received by the server.

RouteParameters

Gets the [StringValueCollection](#) object which represents the current route parameters.

Uri

Gets the [System.Uri](#) component for this HTTP request requested URL.

Methods

Abort()

Immediately closes the connection with the client and does not send any response.

~HttpRequest()

GetBodyContents()

Gets the request contents of the body as a byte array.

GetBodyContentsAsync(CancellationToken)

Asynchronously reads the request contents as a memory byte array.

GetEventSource(string?)

Gets an Event Source interface for this request. Calling this method will put this [HttpRequest](#) instance in its event source listening state.

GetEventSourceAsync(string?)

Asynchronously gets an Event Source interface for this request. Calling this method will put this [HttpRequest](#) instance in its event source listening state.

GetFormContent()

Reads the request body and extracts form data parameters from it.

GetFormContentAsync(CancellationToken)

Asynchronously reads the request body and extracts form data parameters from it.

GetJsonContentAsync<T>(JsonSerializerOptions?, CancellationToken)

Asynchronously deserializes the request body into an object of type `T` using the provided [JsonSerializerOptions](#) [↗](#).

GetJsonContentAsync<T>(JsonTypeInfo<T>, CancellationToken)

Asynchronously deserializes the request body into an object of type `T` using the provided [JsonTypeInfo<T>](#) [↗](#).

[GetJsonContentAsync<T>\(CancellationToken\)](#)

Asynchronously deserializes the request body into an object of type `T` using the default [JsonSerializerOptions](#).

[GetJsonContent<T>\(\)](#)

Deserializes the request body into an object of type `T` using the default [JsonSerializerOptions](#) from [DefaultJsonSerializerOptions](#).

[GetJsonContent<T>\(JsonSerializerOptions?\)](#)

Deserializes the request body into an object of type `T` using the provided [JsonSerializerOptions](#).

[GetJsonContent<T>\(JsonTypeInfo<T>\)](#)

Deserializes the request body into an object of type `T` using the provided [JsonTypeInfo<T>](#).

[GetMultipartFormContent\(\)](#)

Reads the request body and obtains a [MultipartFormCollection](#) from it.

[GetMultipartFormContentAsync\(CancellationToken\)](#)

Asynchronously reads the request body and obtains a [MultipartFormCollection](#) from it.

[GetRawHttpRequest\(bool, bool\)](#)

Gets a visual representation of this request.

[GetRequestStream\(\)](#)

Gets the HTTP request content stream. This property is only available while the content has not been imported by the HTTP server and will invalidate the body content cached in this object.

[GetResponseStream\(\)](#)

Gets an HTTP response stream for this HTTP request.

[GetWebSocket\(string?, string?\)](#)

Accepts and acquires a websocket for this request. Calling this method will put this [HttpRequest](#) instance in streaming state.

[GetWebSocketAsync\(string?, string?\)](#)

Asynchronously accepts and acquires a websocket for this request. Calling this method will put this [HttpRequest](#) instance in streaming state.

[SendTo\(RouteAction\)](#)

Calls another handler for this request, preserving the current call-stack frame, and then returns the response from it. This method manages to prevent possible stack overflows.

[ToString\(\)](#)

Gets an string representation of this [HttpRequest](#) object.

Property Authority

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Authority

Get the requested host header with the port from this HTTP request.

```
public string Authority { get; }
```

Property Value

[string](#)

Remarks

This property brings local request data, so it may not reflect the original client request when used with proxy or CDNs.

Property Bag

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Bag

Gets the managed object which holds data for an entire HTTP session.

```
public TypedValueDictionary Bag { get; }
```

Property Value

[TypedValueDictionary](#)

Remarks

This property is an shortcut for [RequestBag](#) property.

Property Body

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Body

Gets the HTTP request body as string, decoded by the request content encoding.

```
public string Body { get; }
```

Property Value

[string](#)

Remarks

When calling this property, the entire content of the request is read into memory and stored in [RawBody](#).

Property ContentLength

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ContentLength

Gets the content length in bytes count.

```
public long ContentLength { get; }
```

Property Value

[long](#)

Remarks

This value can be negative if the content length is unknown.

Property Context

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Context

Gets the [HttpContext](#) for this request.

```
public HttpContext Context { get; }
```

Property Value

[HttpContext](#)

Property Cookies

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Cookies

Gets an [StringKeyStoreCollection](#) object with all cookies set in this request.

```
public StringKeyStoreCollection Cookies { get; }
```

Property Value

[StringKeyStoreCollection](#)

Property DefaultJsonSerializerOptions

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

DefaultJsonSerializerOptions

Gets or sets the default options used for JSON serialization.

```
public static JsonSerializerOptions? DefaultJsonSerializerOptions { get; set; }
```

Property Value

[JsonSerializerOptions](#)

Remarks

These options are used by default when serializing or deserializing JSON data through [GetJsonContent<T>\(\)](#), unless custom options are provided. See [JsonSerializerOptions](#) for more information on available options.

Property DisconnectToken

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

DisconnectToken

Gets a cancellation token that is signaled when the client disconnects.

```
public CancellationToken DisconnectToken { get; }
```

Property Value

[CancellationToken](#)

Property FullPath

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

FullPath

Gets the raw, full HTTP request path with the query string.

```
public string FullPath { get; }
```

Property Value

[string](#)

Property FullUrl

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

FullUrl

Gets the full URL for this request, with scheme, host, port, path and query.

```
public string FullUrl { get; }
```

Property Value

[string](#)

Remarks

This property brings local request data, so it may not reflect the original client request when used with proxy or CDNs.

Property HasContents

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

HasContents

Gets a boolean indicating whether this request has body contents.

```
public bool HasContents { get; }
```

Property Value

[bool](#)

Property Headers

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Headers

Gets the HTTP request headers.

```
public HttpHeaders Headers { get; }
```

Property Value

[HttpHeaderCollection](#)

Property Host

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Host

Get the requested host (without port) for this [HttpRequest](#).

```
public string? Host { get; }
```

Property Value

[string](#)

Property IsContentAvailable

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

IsContentAvailable

Gets a boolean indicating whether this request has body contents and whether it has already been read into memory by the server.

```
public bool IsContentAvailable { get; }
```

Property Value

[bool](#)

Property IsSecure

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

IsSecure

Gets a boolean indicating whether this request was locally made by an secure transport context (SSL/TLS) or not.

```
public bool IsSecure { get; }
```

Property Value

[bool](#)

Remarks

This property brings local request data, so it may not reflect the original client request when used with proxy or CDNs.

Property Method

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Method

Gets the HTTP request method.

```
public HttpMethod Method { get; }
```

Property Value

[HttpMethod](#)

Property Path

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Path

Gets the HTTP request path without the query string.

```
public string Path { get; }
```

Property Value

[string](#)

Property Query

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Query

Gets the HTTP request query value collection.

```
public StringValueCollection Query { get; }
```

Property Value

[StringValueCollection](#)

Property QueryString

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

QueryString

Gets the HTTP request URL raw query string, including the '?' char.

```
public string QueryString { get; }
```

Property Value

string↗

Property RawBody

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RawBody

Gets the HTTP request body as a byte array.

```
public byte[] RawBody { get; }
```

Property Value

[byte](#) 

Remarks

When calling this property, the entire content of the request is read into memory.

Property RemoteAddress

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RemoteAddress

Gets the incoming local IP address from the request.

```
public IPAddress RemoteAddress { get; }
```

Property Value

[IPAddress](#) ↗

Property RequestEncoding

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RequestEncoding

Gets an string [Encoding](#) that can be used to decode text in this HTTP request.

```
public Encoding RequestEncoding { get; }
```

Property Value

[Encoding](#)

Property RequestId

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RequestId

Gets a unique random ID for this request.

```
public Guid RequestId { get; }
```

Property Value

[Guid](#) ↗

Property RequestedAt

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RequestedAt

Gets the moment which the request was received by the server.

```
public DateTime RequestedAt { get; }
```

Property Value

[DateTime](#)

Property RouteParameters

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RouteParameters

Gets the [StringValueCollection](#) object which represents the current route parameters.

```
public StringValueCollection RouteParameters { get; }
```

Property Value

[StringValueCollection](#)

Property Uri

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Uri

Gets the System.Uri component for this HTTP request requested URL.

```
public Uri Uri { get; }
```

Property Value

Uri

Method Abort

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Abort()

Immediately closes the connection with the client and does not send any response.

```
public HttpResponse Abort()
```

Returns

[HttpResponse](#)

Remarks

This method returns an [HttpResponse](#) indicated to exit outside the scope of the request context. However, when calling this method, the connection is interrupted instantly.

Method GetBodyContents

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetBodyContents()

Gets the request contents of the body as a byte array.

```
public byte[] GetBodyContents()
```

Returns

[byte](#) ↗

A byte array containing the body contents.

Method GetBodyContentsAsync

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetBodyContentsAsync(CancellationToken)

Asynchronously reads the request contents as a memory byte array.

```
public Task<Memory<byte>> GetBodyContentsAsync(CancellationToken cancellation = default)
```

Parameters

cancellation [CancellationToken](#)

A [CancellationToken](#) to cancel the operation.

Returns

[Task](#)< [Memory](#)< [byte](#)>>

A [Task](#) that returns a [Memory](#)<T> of bytes containing the body contents.

Method GetEventSource

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetEventSource(string?)

Gets an Event Source interface for this request. Calling this method will put this [HttpRequest](#) instance in it's event source listening state.

```
public HttpRequestEventSource GetEventSource(string? identifier = null)
```

Parameters

identifier string ↗

Optional. Defines an label to the EventStream connection, useful for finding this connection's reference later.

Returns

[HttpRequestEventSource](#)

Method GetEventSourceAsync

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetEventSourceAsync(string?)

Asynchronously gets an Event Source interface for this request. Calling this method will put this [HttpRequest](#) instance in its event source listening state.

```
public Task<HttpRequestEventSource> GetEventSourceAsync(string? identifier = null)
```

Parameters

identifier [string](#)

Optional. Defines a label to the EventStream connection, useful for finding this connection's reference later.

Returns

[Task](#)<[HttpRequestEventSource](#)>

A [Task](#) that represents the asynchronous operation, containing an [HttpRequestEventSource](#) instance representing the event source for this request.

Method GetFormContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetFormContent()

Reads the request body and extracts form data parameters from it.

```
public StringKeyStoreCollection GetFormContent()
```

Returns

[StringKeyStoreCollection](#)

Method GetFormContentAsync

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetFormContentAsync(CancellationToken)

Asynchronously reads the request body and extracts form data parameters from it.

```
public Task<StringKeyStoreCollection> GetFormContentAsync(CancellationToken cancellation  
= default)
```

Parameters

cancellation [CancellationToken](#)

A [CancellationToken](#) to cancel the asynchronous operation.

Returns

[Task](#)< [StringKeyStoreCollection](#)>

Method GetJsonContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetJsonContent<T>(JsonTypeInfo<T>)

Deserializes the request body into an object of type `T` using the provided [JsonTypeInfo<T>](#).

```
public T? GetJsonContent<T>(JsonTypeInfo<T> typeInfo)
```

Parameters

typeInfo [JsonTypeInfo](#)<T>

The [JsonTypeInfo<T>](#) to use for deserialization.

Returns

`T`

The deserialized object, or `null` if the request body is empty.

Type Parameters

`T`

The type of the object to deserialize into.

GetJsonContent<T>(JsonSerializerOptions?)

Deserializes the request body into an object of type `T` using the provided [JsonSerializerOptions](#).

```
public T? GetJsonContent<T>(JsonSerializerOptions? jsonOptions = null)
```

Parameters

jsonOptions [JsonSerializerOptions](#)

The [JsonSerializerOptions](#) to use for deserialization.

Returns

T

The deserialized object, or `null` if the request body is empty.

Type Parameters

T

The type of the object to deserialize into.

GetJsonContent<T>()

Deserializes the request body into an object of type T using the default [JsonSerializerOptions](#) from [DefaultJsonSerializerOptions](#).

```
public T? GetJsonContent<T>()
```

Returns

T

The deserialized object, or `null` if the request body is empty.

Type Parameters

T

The type of the object to deserialize into.

Method GetJsonContentAsync

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetJsonContentAsync<T>(JsonTypeInfo<T>, CancellationToken)

Asynchronously deserializes the request body into an object of type `T` using the provided [JsonTypeInfo<T>](#).

```
public ValueTask<T?> GetJsonContentAsync<T>(JsonTypeInfo<T> typeInfo, CancellationToken cancellation = default)
```

Parameters

typeInfo [JsonTypeInfo](#)<T>

The [JsonTypeInfo](#)<T> to use for deserialization.

cancellation [CancellationToken](#)

A [CancellationToken](#) to cancel the asynchronous operation.

Returns

[ValueTask](#)<T>

A [ValueTask](#)<TResult> that represents the asynchronous deserialization operation.

Type Parameters

T

The type of the object to deserialize into.

GetJsonContentAsync<T>(JsonSerializerOptions?, CancellationToken)

Asynchronously deserializes the request body into an object of type `T` using the provided [JsonSerializerOptions](#).

```
public ValueTask<T?> GetJsonContentAsync<T>(JsonSerializerOptions? jsonOptions,  
CancellationToken cancellation = default)
```

Parameters

jsonOptions [JsonSerializerOptions](#)

The [JsonSerializerOptions](#) to use for deserialization.

cancellation [CancellationToken](#)

A [CancellationToken](#) to cancel the asynchronous operation.

Returns

[ValueTask](#)<T>

A [ValueTask](#)<TResult> that represents the asynchronous deserialization operation.

Type Parameters

T

The type of the object to deserialize into.

GetJsonContentAsync<T>(CancellationToken)

Asynchronously deserializes the request body into an object of type T using the default [JsonSerializerOptions](#).

```
public ValueTask<T?> GetJsonContentAsync<T>(CancellationToken cancellation = default)
```

Parameters

cancellation [CancellationToken](#)

A [CancellationToken](#) to cancel the asynchronous operation.

Returns

[ValueTask](#)<T>

A [ValueTask](#)<TResult> that represents the asynchronous deserialization operation.

Type Parameters

T

The type of the object to deserialize into.

Method GetMultipartFormContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetMultipartFormContent()

Reads the request body and obtains a [MultipartFormCollection](#) from it.

```
public MultipartFormCollection GetMultipartFormContent()
```

Returns

[MultipartFormCollection](#)

Method GetMultipartFormContentAsync

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetMultipartFormContentAsync(CancellationToken)

Asynchronously reads the request body and obtains a [MultipartFormCollection](#) from it.

```
public Task<MultipartFormCollection> GetMultipartFormContentAsync(CancellationToken cancellation  
= default)
```

Parameters

cancellation [CancellationToken](#)

A [CancellationToken](#) to cancel the asynchronous operation.

Returns

[Task](#)<MultipartFormCollection>

A [Task](#) that represents the asynchronous operation, containing a [MultipartFormCollection](#) instance representing the multipart form content of the request.

Exceptions

[HttpRequestException](#)

If an error occurs while parsing the multipart form content.

Method GetRawHttpRequest

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetRawHttpRequest(bool, bool)

Gets a visual representation of this request.

```
public string GetRawHttpRequest(bool includeBody = true, bool appendExtraInfo = false)
```

Parameters

includeBody [bool](#)

Optional. Defines if the body should be included in the output.

appendExtraInfo [bool](#)

Optional. Appends extra information, such as request id and date into the output.

Returns

[string](#)

Method GetRequestStream

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetRequestStream()

Gets the HTTP request content stream. This property is only available while the content has not been imported by the HTTP server and will invalidate the body content cached in this object.

```
public Stream GetRequestStream()
```

Returns

[Stream](#)

Method GetResponseStream

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetResponseStream()

Gets an HTTP response stream for this HTTP request.

```
public HttpResponseMessage GetResponseStream()
```

Returns

[HttpResponseStreamManager](#)

Method GetWebSocket

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetWebSocket(string?, string?)

Accepts and acquires a websocket for this request. Calling this method will put this [HttpRequest](#) instance in streaming state.

```
public HttpWebSocket GetWebSocket(string? subprotocol = null, string? identifier = null)
```

Parameters

subprotocol [string](#)

Optional. Determines the sub-protocol to plug the websocket in.

identifier [string](#)

Optional. Defines an label to the Web Socket connection, useful for finding this connection's reference later.

Returns

[HttpWebSocket](#)

Method GetWebSocketAsync

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetWebSocketAsync(string?, string?)

Asynchronously accepts and acquires a websocket for this request. Calling this method will put this [Http Request](#) instance in streaming state.

```
public Task<HttpWebSocket> GetWebSocketAsync(string? subprotocol = null, string? identifier = null)
```

Parameters

subprotocol [string](#)

Optional. Determines the sub-protocol to plug the websocket in.

identifier [string](#)

Optional. Defines an label to the Web Socket connection, useful for finding this connection's reference later.

Returns

[Task](#)<[HttpWebSocket](#)>

A task that represents the asynchronous operation, returning an instance of [HttpWebSocket](#) representing the accepted websocket connection.

Method SendTo

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

SendTo(RouteAction)

Calls another handler for this request, preserving the current call-stack frame, and then returns the response from it. This method manages to prevent possible stack overflows.

```
public object SendTo(RouteAction otherCallback)
```

Parameters

otherCallback [RouteAction](#)

Defines the [RouteAction](#) method which will handle this request.

Returns

[object](#) ↗

Method ToString

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ToString()

Gets an string representation of this [HttpRequest](#) object.

```
public override string ToString()
```

Returns

[string](#) ↗

Method ~HttpRequest

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

~HttpRequest()

```
protected ~HttpRequest()
```

Class HttpRequestException

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Represents an exception that is thrown while a request is being interpreted by the HTTP server.

```
public sealed class HttpRequestException : Exception, ISerializable
```

Implements

[ISerializable](#) ↗

Class HttpResponse

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Represents an HTTP Response.

```
public class HttpResponse
```

Extension Methods

[HttpResponseExtensions.WithContent<THttpResponse>\(THttpResponse, HttpContent\)](#) ,
[HttpResponseExtensions.WithContent<THttpResponse>\(THttpResponse, string\)](#) ,
[HttpResponseExtensions.WithContent<THttpResponse>\(THttpResponse, string, Encoding?, string\)](#) ,
[HttpResponseExtensions.WithCookie<THttpResponse>\(THttpResponse, Cookie\)](#) ,
[HttpResponseExtensions.WithCookie<THttpResponse>\(THttpResponse, string, string, DateTime?, TimeSpan?, string?, string?, bool?, bool?, string?\)](#) ,
[HttpResponseExtensions.WithHeader<THttpResponse>\(THttpResponse, StringKeyStoreCollection\)](#) ,
[HttpResponseExtensions.WithHeader<THttpResponse>\(THttpResponse, NameValueCollection\)](#) ,
[HttpResponseExtensions.WithHeader<THttpResponse>\(THttpResponse, string, string\)](#) ,
[HttpResponseExtensions.WithStatus<THttpResponse>\(THttpResponse, in HttpStatusInformation\)](#) ,
[HttpResponseExtensions.WithStatus<THttpResponse>\(THttpResponse, int\)](#) ,
[HttpResponseExtensions.WithStatus<THttpResponse>\(THttpResponse, HttpStatusCode\)](#)

Constructors

[HttpResponse\(\)](#)

Creates an new [HttpResponse](#) instance with HTTP OK status code and no content.

[HttpResponse\(in HttpStatusInformation\)](#)

Creates an new [HttpResponse](#) instance with given status code.

[HttpResponse\(int\)](#)

Creates an new [HttpResponse](#) instance with given status code.

[HttpResponse\(int, HttpContent?\)](#)

Creates an new [HttpResponse](#) instance with given status code and HTTP content.

[HttpResponse\(HttpContent?\)](#)

Creates an new [HttpResponse](#) instance with given HTTP content, with default status code as 200 OK.

[HttpResponse\(HttpStatusCode, HttpContent?\)](#)

Creates an new [HttpResponse](#) instance with given status code and HTTP contents.

[HttpResponse\(HttpStatusCode, string\)](#)

Creates an new [HttpResponse](#) instance with given status code and string content.

[HttpResponse\(string\)](#)

Creates an new [HttpResponse](#) instanec with given string content and status code as 200 OK.

Properties

[Content](#)

Gets or sets the HTTP response body contents.

[Headers](#)

Gets or sets the [HttpHeaderCollection](#) instance of the HTTP response headers.

[SendChunked](#)

Gets or sets whether the HTTP response will be sent chunked. When setting this property to [true](#), the Content-Length header is automatically omitted.

[Status](#)

Gets or sets the HTTP status code and description for this HTTP response.

Methods

[Equals\(object?\)](#)

[GetHashCode\(\)](#)

[GetHeaderValue\(string\)](#)

Gets the value of a specific header from the request and content headers.

[GetRawHttpResponse\(bool\)](#)

Gets a visual representation of this HTTP response.

[Refuse\(\)](#)

Creates an [HttpResponse](#) object which closes the connection with the client immediately (ECONNRESET).

[SetCookie\(Cookie\)](#)

Sets a cookie and sends it in the response to be set by the client.

[SetCookie\(string, string\)](#)

Sets a cookie and sends it in the response to be set by the client.

[SetCookie\(string, string, DateTime?, TimeSpan?, string?, string?, bool?, bool?, string?\)](#)

Sets a cookie and sends it in the response to be set by the client.

[ToString\(\)](#)

Constructor HttpResponseMessage

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

HttpResponse()

Creates an new [HttpResponse](#) instance with HTTP OK status code and no content.

```
public HttpResponseMessage()
```

HttpResponse(int)

Creates an new [HttpResponse](#) instance with given status code.

```
public HttpResponseMessage(int status)
```

Parameters

status int ↗

The status code of this HTTP response.

HttpResponse(int, HttpContent?)

Creates an new [HttpResponse](#) instance with given status code and HTTP content.

```
public HttpResponseMessage(int status, HttpContent? content)
```

Parameters

status int ↗

The status code of this HTTP response.

content [HttpContent](#)

The response content, if any.

HttpResponse(HttpContent?)

Creates an new [HttpResponse](#) instance with given HTTP content, with default status code as 200 OK.

```
public HttpResponse(HttpContent? content)
```

Parameters

content [HttpContent](#)

The response content, if any.

HttpResponse(string)

Creates an new [HttpResponse](#) instance with given string content and status code as 200 OK.

```
public HttpResponse(string stringContent)
```

Parameters

stringContent [string](#)

The UTF-8 string content.

HttpResponse(HttpStatusCode, string)

Creates an new [HttpResponse](#) instance with given status code and string content.

```
public HttpResponse(HttpStatusCode status, string stringContent)
```

Parameters

status [HttpStatusCode](#)

The [HttpStatusCode](#) of this HTTP response.

stringContent [string](#)

The UTF-8 string content.

HttpResponse(HttpStatusCode, HttpContent?)

Creates an new [HttpResponse](#) instance with given status code and HTTP contents.

```
public HttpResponse(HttpStatusCode status, HttpContent? content)
```

Parameters

status [HttpStatusCode](#)

The [HttpStatusCode](#) of this HTTP response.

content [HttpContent](#)

The response content, if any.

HttpResponse(in HttpStatusCode)

Creates an new [HttpResponse](#) instance with given status code.

```
public HttpResponse(in HttpStatusCode status)
```

Parameters

status [HttpStatusCode](#)

The [HttpStatusCode](#) of this HTTP response.

Property Content

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Content

Gets or sets the HTTP response body contents.

```
public HttpContent? Content { get; set; }
```

Property Value

[HttpContent](#) ↗

Property Headers

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Headers

Gets or sets the [HttpHeaderCollection](#) instance of the HTTP response headers.

```
public HttpHeaders Headers { get; set; }
```

Property Value

[HttpHeaderCollection](#)

Property SendChunked

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

SendChunked

Gets or sets whether the HTTP response will be sent chunked. When setting this property to [true](#), the Content-Length header is automatically omitted.

```
public bool SendChunked { get; set; }
```

Property Value

[bool](#)

Remarks

The response is always sent as chunked when it is not possible to determine the size of the content to send.

Property Status

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Status

Gets or sets the HTTP status code and description for this HTTP response.

```
public HttpStatusInformation Status { get; set; }
```

Property Value

[HttpStatusInformation](#)

Method Equals

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Equals(object?)

```
public override bool Equals(object? obj)
```

Parameters

obj object↗

Returns

bool↗

Method GetHashCode

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetHashCode()

```
public override int GetHashCode()
```

Returns

[int](#)

Method GetHeaderValue

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetHeaderValue(string)

Gets the value of a specific header from the request and content headers.

```
public string? GetHeaderValue(string headerName)
```

Parameters

headerName [string](#)

The name of the header to retrieve.

Returns

[string](#)

The header value, or [null](#) if the header is not found.

Method GetRawHttpResponse

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetRawHttpResponse(bool)

Gets a visual representation of this HTTP response.

```
public string GetRawHttpResponse(bool includeBody = true)
```

Parameters

includeBody [bool](#)

Determines whether the message content will also be included in the return from this function.

Returns

[string](#)

Method Refuse

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Refuse()

Creates an [HttpResponse](#) object which closes the connection with the client immediately (ECONNRESET).

```
public static HttpResponse Refuse()
```

Returns

[HttpResponse](#)

Method SetCookie

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

SetCookie(Cookie)

Sets a cookie and sends it in the response to be set by the client.

```
public void SetCookie(Cookie cookie)
```

Parameters

cookie [Cookie](#)

The cookie object.

SetCookie(string, string)

Sets a cookie and sends it in the response to be set by the client.

```
public void SetCookie(string name, string value)
```

Parameters

name [string](#)

The cookie name.

value [string](#)

The cookie value.

SetCookie(string, string, DateTime?, TimeSpan?, string?, string?, bool?, bool?, string?)

Sets a cookie and sends it in the response to be set by the client.

```
public void SetCookie(string name, string value, DateTime? expires = null, TimeSpan? maxAge = null, string? domain = null, string? path = null, bool? secure = null, bool? httpOnly = null, string? sameSite = null)
```

Parameters

name `string`

The cookie name.

value `string`

The cookie value.

expires `DateTime?`

The cookie expiry date.

maxAge `TimeSpan?`

The cookie max duration after being set.

domain `string`

The domain where the cookie will be valid.

path `string`

The path where the cookie will be valid.

secure `bool?`

Determines if the cookie will only be stored in a secure context.

httpOnly `bool?`

Determines if the cookie will be only available in the HTTP context.

sameSite `string`

The cookie SameSite parameter.

Method ToString

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ToString()

```
public override string ToString()
```

Returns

[string](#)

Class HttpResponseExtensions

Namespace:[Sisk.Core.Http](#)

Assembly:[Sisk.Core.dll](#)

Provides useful extensions for [HttpResponse](#) objects.

```
public static class HttpResponseExtensions
```

Methods

[WithContent<THttpResponse>\(THttpResponse, HttpContent\)](#)

Sets an [HttpContent](#) as the HTTP content body in this [HttpResponse](#).

[WithContent<THttpResponse>\(THttpResponse, string\)](#)

Sets an UTF-8 string as the HTTP response content in this [HttpResponse](#).

[WithContent<THttpResponse>\(THttpResponse, string, Encoding?, string\)](#)

Sets an string as the HTTP response content in this [HttpResponse](#).

[WithCookie<THttpResponse>\(THttpResponse, Cookie\)](#)

Sets a cookie and sends it in the response to be set by the client.

[WithCookie<THttpResponse>\(THttpResponse, string, string, DateTime?, TimeSpan?, string?, string?, bool?, bool?, string?\)](#)

Sets a cookie and sends it in the response to be set by the client.

[WithHeader<THttpResponse>\(THttpResponse, StringKeyStoreCollection\)](#)

Sets an list of HTTP headers in this [HttpResponse](#).

[WithHeader<THttpResponse>\(THttpResponse, NameValueCollection\)](#)

Sets an list of HTTP headers in this [HttpResponse](#).

[WithHeader<THttpResponse>\(THttpResponse, string, string\)](#)

Sets an HTTP header in this [HttpResponse](#).

[WithStatus<THttpResponse>\(THttpResponse, in HttpStatusInformation\)](#)

Sets the HTTP status code of this [HttpResponse](#).

`WithStatus<THttpResponse>(THttpResponse, int)`

Sets the HTTP status code of this [HttpResponse](#).

`WithStatus<THttpResponse>(THttpResponse, HttpStatusCode)`

Sets the HTTP status code of this [HttpResponse](#).

Method WithContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

WithContent<THttpResponse>(THttpResponse, string)

Sets an UTF-8 string as the HTTP response content in this [HttpResponse](#).

```
public static THttpResponse WithContent<THttpResponse>(this THttpResponse response, string content) where THttpResponse : HttpResponse
```

Parameters

response THttpResponse

The [HttpResponse](#) object.

content string ↗

The UTF-8 string containing the response body.

Returns

THttpResponse

The self THttpResponse object.

Type Parameters

THttpResponse

The type which implements [HttpResponse](#).

WithContent<THttpResponse>(THttpResponse, string, Encoding?, string)

Sets an string as the HTTP response content in this [HttpResponse](#).

```
public static THttpResponse WithContent<THttpResponse>(this THttpResponse response, string content, Encoding? encoding, string mimeType) where THttpResponse : HttpResponse
```

Parameters

response `THttpResponse`

The `HttpResponse` object.

content `string`

The string containing the response body.

encoding `Encoding`

The encoding to encode the string message.

mimeType `string`

The mime-type of the response content.

Returns

`THttpResponse`

The self `THttpResponse` object.

Type Parameters

THttpResponse

The type which implements `HttpResponse`.

WithContent<THttpResponse>(THttpResponse, HttpContent)

Sets an `HttpContent` as the HTTP content body in this `HttpResponse`.

```
public static THttpResponse WithContent<THttpResponse>(this THttpResponse response, HttpContent content) where THttpResponse : HttpResponse
```

Parameters

response `THttpResponse`

The [HttpResponse](#) object.

content [HttpContent](#) ↗

The HTTP content object.

Returns

[THttpResponse](#)

The self [THttpResponse](#) object.

Type Parameters

THttpResponse

The type which implements [HttpResponse](#).

Method WithCookie

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

`WithCookie<THttpResponse>(THttpResponse, string, string, DateTime?, TimeSpan?, string?, string?, bool?, bool?, string?)`

Sets a cookie and sends it in the response to be set by the client.

```
public static THttpResponse WithCookie<THttpResponse>(this THttpResponse response, string name,
string value, DateTime? expires = null, TimeSpan? maxAge = null, string? domain = null, string?
path = null, bool? secure = null, bool? httpOnly = null, string? sameSite = null) where
THttpResponse : HttpResponse
```

Parameters

response [THttpResponse](#)

The [HttpResponse](#) object.

name [string](#)

The cookie name.

value [string](#)

The cookie value.

expires [DateTime](#)?

The cookie expiry date.

maxAge [TimeSpan](#)?

The cookie max duration after being set.

domain [string](#)

The domain where the cookie will be valid.

path [string](#)

The path where the cookie will be valid.

secure [bool](#)?

Determines if the cookie will only be stored in an secure context.

httpOnly [bool](#)?

Determines if the cookie will be only available in the HTTP context.

sameSite [string](#)

The cookie SameSite parameter.

Returns

[THttpResponse](#)

Type Parameters

THttpResponse

The type which implements [HttpResponse](#).

WithCookie<THttpResponse>(THttpResponse, Cookie)

Sets a cookie and sends it in the response to be set by the client.

```
public static THttpResponse WithCookie<THttpResponse>(this THttpResponse response, Cookie  
cookie) where THttpResponse : HttpResponse
```

Parameters

response [THttpResponse](#)

The [HttpResponse](#) object.

cookie [Cookie](#)?

The cookie object.

Returns

THttpResponse

Type Parameters

THttpResponse

The type which implements [HttpResponse](#).

Method WithHeader

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

WithHeader<THttpResponse>(THttpResponse, string, string)

Sets an HTTP header in this [HttpResponse](#).

```
public static THttpResponse WithHeader<THttpResponse>(this THttpResponse response, string
headerName, string headerValue) where THttpResponse : HttpResponse
```

Parameters

response THttpResponse

The [HttpResponse](#) object.

headerName string ↗

The name of the header.

headerValue string ↗

The header value.

Returns

THttpResponse

The self THttpResponse object.

Type Parameters

THttpResponse

The type which implements [HttpResponse](#).

WithHeader<THttpResponse>(THttpResponse, NameValueCollection)

Sets an list of HTTP headers in this [HttpResponse](#).

```
public static THttpResponse WithHeader<THttpResponse>(this THttpResponse response,  
NameValueCollection headers) where THttpResponse : HttpResponse
```

Parameters

response THttpResponse

The [HttpResponse](#) object.

headers NameValueCollection

The collection of HTTP headers.

Returns

THttpResponse

The self THttpResponse object.

Type Parameters

THttpResponse

The type which implements [HttpResponse](#).

WithHeader<THttpResponse>(THttpResponse, StringKeyStoreCollection)

Sets an list of HTTP headers in this [HttpResponse](#).

```
public static THttpResponse WithHeader<THttpResponse>(this THttpResponse response,  
StringKeyStoreCollection headers) where THttpResponse : HttpResponse
```

Parameters

response THttpResponse

The [HttpResponse](#) object.

headers StringKeyStoreCollection

The collection of HTTP headers.

Returns

`THttpResponse`

The self `THttpResponse` object.

Type Parameters

THttpResponse

The type which implements [HttpResponse](#).

Method WithStatus

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

WithStatus<THttpResponse>(THttpResponse, int)

Sets the HTTP status code of this [HttpResponse](#).

```
public static THttpResponse WithStatus<THttpResponse>(this THttpResponse response, int  
httpStatusCode) where THttpResponse : HttpResponse
```

Parameters

response THttpResponse

The [HttpResponse](#) object.

httpStatusCode [int](#)

The HTTP status code.

Returns

THttpResponse

The self THttpResponse object.

Type Parameters

THttpResponse

The type which implements [HttpResponse](#).

WithStatus<THttpResponse>(THttpResponse, HttpStatusCode)

Sets the HTTP status code of this [HttpResponse](#).

```
public static THttpResponse WithStatus<THttpResponse>(this THttpResponse response,  
HttpStatusCode statusCode) where THttpResponse : HttpResponse
```

Parameters

response THttpResponse

The [HttpResponse](#) object.

statusCode HttpStatusCode

The HTTP status code.

Returns

THttpResponse

The self THttpResponse object.

Type Parameters

THttpResponse

The type which implements [HttpResponse](#).

WithStatus<THttpResponse>(THttpResponse, in HttpStatusCodeInformation)

Sets the HTTP status code of this [HttpResponse](#).

```
public static THttpResponse WithStatus<THttpResponse>(this THttpResponse response, in  
HttpStatusCodeInformation statusInformation) where THttpResponse : HttpResponse
```

Parameters

response THttpResponse

The [HttpResponse](#) object.

statusInformation HttpStatusCodeInformation

The HTTP status information.

Returns

THttpResponse

The self THttpResponse object.

Type Parameters

THttpResponse

The type which implements [HttpResponse](#).

Class HttpServer

Namespace:[Sisk.Core.Http](#)

Assembly:[Sisk.Core.dll](#)

Provides an lightweight HTTP server powered by Sisk.

```
public sealed class HttpServer : IDisposable
```

Implements

[IDisposable](#)

Constructors

[HttpServer\(\)](#)

Creates an new [HttpServer](#) instance with no predefined configuration.

[HttpServer\(HttpServerConfiguration\)](#)

Creates a new default configuration [HttpServer](#) instance with the given Route and server configuration.

Properties

[EventSources](#)

Gets an [HttpEventSourceCollection](#) with active event source connections in this HTTP server.

[IsDynamicCodeSupported](#)

Gets an [bool](#) indicating if the current environment supports dynamic code or it's running in an AOT assembly.

[IsListening](#)

Gets an boolean indicating if this HTTP server is running and listening.

[IsSupported](#)

Gets an [bool](#) indicating if Sisk can be used with the current environment.

[ListeningPrefixes](#)

Gets an string array containing all URL prefixes which this HTTP server is listening to.

[PoweredBy](#)

Gets the X-Powered-By Sisk header value.

[ServerConfiguration](#)

Gets or sets the Server Configuration object.

[SiskVersion](#)

Gets the current Sisk version.

[WebSockets](#)

Gets an [HttpWebSocketConnectionCollection](#) with active Web Sockets connections in this HTTP server.

Methods

[CreateBuilder\(\)](#)

Builds an empty [HttpServerHostContext](#) context.

[CreateBuilder\(Action<HttpServerHostContextBuilder>\)](#)

Builds an [HttpServerHostContext](#) context invoking the handler on it.

[CreateBuilder\(string\)](#)

Builds an empty [HttpServerHostContext](#) context with predefined listening host string.

[CreateBuilder\(ushort\)](#)

Builds an empty [HttpServerHostContext](#) context with predefined listening port.

[CreateListener\(\)](#)

Gets an listening and running HTTP server in an random port.

[CreateListener\(ushort\)](#)

Gets an listening and running HTTP server in the specified port.

[CreateListener\(ushort, out HttpServerConfiguration, out ListeningHost, out Router\)](#)

Gets an listening and running HTTP server in the specified port.

[Dispose\(\)](#)

Invalidates this class and releases the resources used by it, and permanently closes the HTTP server.

[Emit\(ushort, out HttpServerConfiguration, out ListeningHost, out Router\)](#)

Gets an non-listening HTTP server with configuration, listening host, and router.

[RegisterHandler\(HttpServerHandler\)](#)

Associate an [HttpServerHandler](#) in this HttpServer to handle functions such as requests, routers and contexts.

[RegisterHandler<T>\(\)](#)

Associate an [HttpServerHandler](#) in this HttpServer to handle functions such as requests, routers and contexts.

[Restart\(\)](#)

Restarts this HTTP server, sending all processing responses and starting them again, reading the listening ports again.

[Start\(\)](#)

Starts listening to the set port and handling requests on this server.

[Stop\(\)](#)

Stops the server from listening and stops the request handler.

[WaitNext\(\)](#)

Waits for the next execution result from the server. This method obtains the next completed context from the HTTP server, both with the request and its response. This method does not interrupt the asynchronous processing of requests.

[WaitNextAsync\(\)](#)

Waits for the next execution result from the server asynchronously. This method obtains the next completed context from the HTTP server, both with the request and its response. This method does not interrupt the asynchronous processing of requests.

Constructor HttpServer

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

HttpServer()

Creates an new [HttpServer](#) instance with no predefined configuration.

```
public HttpServer()
```

HttpServer(HttpServerConfiguration)

Creates a new default configuration [HttpServer](#) instance with the given Route and server configuration.

```
public HttpServer(HttpServerConfiguration configuration)
```

Parameters

configuration [HttpServerConfiguration](#)

The configuration object of the server.

Property EventSources

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

EventSources

Gets an [HttpEventSourceCollection](#) with active event source connections in this HTTP server.

```
public HttpEventSourceCollection EventSources { get; }
```

Property Value

[HttpEventSourceCollection](#)

Property IsDynamicCodeSupported

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

IsDynamicCodeSupported

Gets an [bool](#) indicating if the current environment supports dynamic code or it's running in an AOT assembly.

```
public static bool IsDynamicCodeSupported { get; }
```

Property Value

[bool](#)

Property IsListening

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

IsListening

Gets an boolean indicating if this HTTP server is running and listening.

```
public bool IsListening { get; }
```

Property Value

bool ↗

Property IsSupported

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

IsSupported

Gets an [bool](#) indicating if Sisk can be used with the current environment.

```
[Obsolete("This property is no longer supported. To find out if HttpListener is supported,  
use HttpListener.IsSupported.")]  
public static bool IsSupported { get; }
```

Property Value

[bool](#)

Property ListeningPrefixes

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ListeningPrefixes

Gets an string array containing all URL prefixes which this HTTP server is listening to.

```
public string[] ListeningPrefixes { get; }
```

Property Value

[string](#) []

Property PoweredBy

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

PoweredBy

Gets the X-Powered-By Sisk header value.

```
public static string PoweredBy { get; }
```

Property Value

[string](#)

Property ServerConfiguration

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ServerConfiguration

Gets or sets the Server Configuration object.

```
public HttpServerConfiguration ServerConfiguration { get; set; }
```

Property Value

[HttpServerConfiguration](#)

Property SiskVersion

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

SiskVersion

Gets the current Sisk version.

```
public static Version SiskVersion { get; }
```

Property Value

[Version](#)

Property WebSockets

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

WebSockets

Gets an [HttpWebSocketConnectionCollection](#) with active Web Sockets connections in this HTTP server.

```
public HttpWebSocketConnectionCollection WebSockets { get; }
```

Property Value

[HttpWebSocketConnectionCollection](#)

Method CreateBuilder

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

CreateBuilder(Action<HttpServerHostContextBuilder>)

Builds an [HttpServerHostContext](#) context invoking the handler on it.

```
public static HttpServerHostContextBuilder CreateBuilder(Action<HttpServerHostContextBuilder> handler)
```

Parameters

handler [Action](#)<HttpServerHostContextBuilder>

The action which will configure the host context.

Returns

[HttpServerHostContextBuilder](#)

CreateBuilder(ushort)

Builds an empty [HttpServerHostContext](#) context with predefined listening port.

```
public static HttpServerHostContextBuilder CreateBuilder(ushort port)
```

Parameters

port [ushort](#)

Returns

[HttpServerHostContextBuilder](#)

CreateBuilder(string)

Builds an empty [HttpServerHostContext](#) context with predefined listening host string.

```
public static HttpServerHostContextBuilder CreateBuilder(string listeningHost)
```

Parameters

listeningHost [string](#)

Returns

[HttpServerHostContextBuilder](#)

CreateBuilder()

Builds an empty [HttpServerHostContext](#) context.

```
public static HttpServerHostContextBuilder CreateBuilder()
```

Returns

[HttpServerHostContextBuilder](#)

Method CreateListener

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

CreateListener()

Gets an listening and running HTTP server in an random port.

```
public static HttpServer CreateListener()
```

Returns

[HttpServer](#)

CreateListener(ushort)

Gets an listening and running HTTP server in the specified port.

```
public static HttpServer CreateListener(ushort port)
```

Parameters

port [ushort](#)

The listening port of the HTTP server.

Returns

[HttpServer](#)

CreateListener(ushort, out HttpServerConfiguration, out ListeningHost, out Router)

Gets an listening and running HTTP server in the specified port.

```
public static HttpServer CreateListener(ushort insecureHttpPort, out HttpServerConfiguration configuration, out ListeningHost host, out Router router)
```

Parameters

insecureHttpPort [ushort](#)

The insecure port where the HTTP server will listen.

configuration [HttpServerConfiguration](#)

The [HttpServerConfiguration](#) object issued from this method.

host [ListeningHost](#)

The [ListeningHost](#) object issued from this method.

router [Router](#)

The [Router](#) object issued from this method.

Returns

[HttpServer](#)

Method Dispose

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Dispose()

Invalidates this class and releases the resources used by it, and permanently closes the HTTP server.

```
public void Dispose()
```

Method Emit

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Emit(ushort, out HttpServerConfiguration, out ListeningHost, out Router)

Gets an non-listening HTTP server with configuration, listening host, and router.

```
public static HttpServer Emit(ushort insecureHttpPort, out HttpServerConfiguration  
configuration, out ListeningHost host, out Router router)
```

Parameters

insecureHttpPort [ushort](#)

The insecure port where the HTTP server will listen.

configuration [HttpServerConfiguration](#)

The [HttpServerConfiguration](#) object issued from this method.

host [ListeningHost](#)

The [ListeningHost](#) object issued from this method.

router [Router](#)

The [Router](#) object issued from this method.

Returns

[HttpServer](#)

Method RegisterHandler

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RegisterHandler<T>()

Associate an [HttpServerHandler](#) in this HttpServer to handle functions such as requests, routers and contexts.

```
public void RegisterHandler<T>() where T : HttpServerHandler, new()
```

Type Parameters

T

The handler which implements [HttpServerHandler](#).

RegisterHandler(HttpServerHandler)

Associate an [HttpServerHandler](#) in this HttpServer to handle functions such as requests, routers and contexts.

```
public void RegisterHandler(HttpServerHandler obj)
```

Parameters

obj [HttpServerHandler](#)

The instance of the server handler.

Method Restart

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Restart()

Restarts this HTTP server, sending all processing responses and starting them again, reading the listening ports again.

```
public void Restart()
```

Method Start

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Start()

Starts listening to the set port and handling requests on this server.

```
public void Start()
```

Method Stop

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Stop()

Stops the server from listening and stops the request handler.

```
public void Stop()
```

Method WaitNext

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

WaitNext()

Waits for the next execution result from the server. This method obtains the next completed context from the HTTP server, both with the request and its response. This method does not interrupt the asynchronous processing of requests.

```
public HttpServerExecutionResult WaitNext()
```

Returns

[HttpServerExecutionResult](#)

Remarks

Calling this method, it starts the HTTP server if it isn't started yet.

Method WaitNextAsync

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

WaitNextAsync()

Waits for the next execution result from the server asynchronously. This method obtains the next completed context from the HTTP server, both with the request and its response. This method does not interrupt the asynchronous processing of requests.

```
public Task<HttpServerExecutionResult> WaitNextAsync()
```

Returns

[Task](#)<HttpServerExecutionResult>

Remarks

Calling this method, it starts the HTTP server if it isn't started yet.

Class HttpServerConfiguration

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Provides execution parameters for an [HttpServer](#).

```
public sealed class HttpServerConfiguration : IDisposable
```

Implements

[IDisposable](#) ↗

Constructors

[HttpServerConfiguration\(\)](#)

Creates an new [HttpServerConfiguration](#) instance with no parameters.

Fields

[DefaultAccessLogFormat](#)

Represents the default access logging format for incoming HTTP requests.

Properties

[AccessLogsFormat](#)

Gets or sets the access logging format for incoming HTTP requests.

[AccessLogsStream](#)

Gets or sets the [LogStream](#) object which the HTTP server will write HTTP server access messages to.

[AsyncRequestProcessing](#)

Gets or sets whether the HTTP server should handle requests asynchronously or if it should limit the request processing to one request per time.

[ConvertIAsyncEnumerableIntoEnumerable](#)

Gets or sets whether the HTTP server should convert [IAsyncEnumerable<T>](#) object responses into an blocking [IEnumerable<T>](#).

[DisposeDisposableContextValues](#)

Gets or sets whether the HTTP server should dispose all [IDisposable](#) values in the [HttpContext](#) bag when an HTTP session is closed.

[EnableAutomaticResponseCompression](#)

Gets or sets whether the HTTP server should automatically compress response content bodies using request-allowed encoding algorithms when possible.

[Engine](#)

Gets or sets the HTTP server processing engine.

[ErrorsLogsStream](#)

Gets or sets the [LogStream](#) object which the HTTP server will write HTTP server error transcriptions to.

[ForceTrailingSlash](#)

Gets or sets whether the HTTP server should automatically rewrite GET requests to end their path with / . This is applicable only to non-Regex routes.

[ForwardingResolver](#)

Gets or sets an object that is responsible for resolving the client address, host and protocol of a proxy, load balancer or CDN, through the HTTP request.

[IdleConnectionTimeout](#)

Gets or sets the maximum time allowed for an idle connection.

[IncludeRequestIdHeader](#)

Gets or sets whether the server should include the "X-Request-Id" header in response headers.

[KeepAlive](#)

Gets or sets whether the client should maintain an persistent connection with the HTTP server.

[ListeningHosts](#)

Gets or sets the listening hosts repository that the [HttpServer](#) instance will listen to.

[MaximumContentLength](#)

Gets or sets the maximum size of a request body before it is closed by the socket.

[NormalizeHeadersEncodings](#)

Gets or sets whether the HTTP server should convert request headers encoding to the content encoding.

[OptionsLogMode](#)

Gets or sets the log mode that the HTTP server should use to log OPTIONS requests.

[RemoteRequestsAction](#)

Gets or sets the server's action when it receives an HTTP request outside the local host.

[SendSiskHeader](#)

Gets or sets whether the HTTP server should send the X-Powered-By header in all responses.

[ThrowExceptions](#)

Gets or sets whether the server should throw exceptions instead of reporting it on [HttpServerExecutionStatus](#) if any is thrown while processing requests.

Methods

[Dispose\(\)](#)

Frees the resources and invalidates this instance.

Constructor HttpServerConfiguration

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

HttpServerConfiguration()

Creates an new [HttpServerConfiguration](#) instance with no parameters.

```
public HttpServerConfiguration()
```

Field DefaultAccessLogFormat

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Represents the default access logging format for incoming HTTP requests.

```
public const string DefaultAccessLogFormat = "%ri - [%dd/%dmm/%dy %tH:%ti:%ts %tz] \"%%rm\n%rz%%rq\" %sc %lou"
```

Returns

[string](#)

Represents the default access logging format for incoming HTTP requests.

Property AccessLogsFormat

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

AccessLogsFormat

Gets or sets the access logging format for incoming HTTP requests.

```
public string AccessLogsFormat { get; set; }
```

Property Value

[string](#)

Property AccessLogsStream

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

AccessLogsStream

Gets or sets the [LogStream](#) object which the HTTP server will write HTTP server access messages to.

```
public LogStream? AccessLogsStream { get; set; }
```

Property Value

[LogStream](#)

Property AsyncRequestProcessing

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

AsyncRequestProcessing

Gets or sets whether the HTTP server should handle requests asynchronously or if it should limit the request processing to one request per time.

```
public bool AsyncRequestProcessing { get; set; }
```

Property Value

[bool](#)

Property ConvertIAsyncEnumerableIntoEnumerable

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ConvertIAsyncEnumerableIntoEnumerable

Gets or sets whether the HTTP server should convert [IAsyncEnumerable<T>](#) object responses into an blocking [IEnumerable<T>](#).

```
public bool ConvertIAsyncEnumerableIntoEnumerable { get; set; }
```

Property Value

[bool](#)

Property DisposeDisposableContextValues

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

DisposeDisposableContextValues

Gets or sets whether the HTTP server should dispose all [IDisposable](#) values in the [HttpContext](#) bag when an HTTP session is closed.

```
public bool DisposeDisposableContextValues { get; set; }
```

Property Value

[bool](#)

Property EnableAutomaticResponseCompression

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

EnableAutomaticResponseCompression

Gets or sets whether the HTTP server should automatically compress response content bodies using request-allowed encoding algorithms when possible.

```
public bool EnableAutomaticResponseCompression { get; set; }
```

Property Value

[bool](#)

Property Engine

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Engine

Gets or sets the HTTP server processing engine.

```
public HttpServerEngine Engine { get; set; }
```

Property Value

[HttpServerEngine](#)

Property ErrorsLogsStream

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ErrorsLogsStream

Gets or sets the [LogStream](#) object which the HTTP server will write HTTP server error transcriptions to.

```
public LogStream? ErrorsLogsStream { get; set; }
```

Property Value

[LogStream](#)

Property ForceTrailingSlash

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ForceTrailingSlash

Gets or sets whether the HTTP server should automatically rewrite GET requests to end their path with / . This is applicable only to non-Regex routes.

```
public bool ForceTrailingSlash { get; set; }
```

Property Value

[bool](#)

Property ForwardingResolver

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ForwardingResolver

Gets or sets an object that is responsible for resolving the client address, host and protocol of a proxy, load balancer or CDN, through the HTTP request.

```
public ForwardingResolver? ForwardingResolver { get; set; }
```

Property Value

[ForwardingResolver](#)

Property IdleConnectionTimeout

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

IdleConnectionTimeout

Gets or sets the maximum time allowed for an idle connection.

```
public TimeSpan IdleConnectionTimeout { get; set; }
```

Property Value

[TimeSpan](#)

Property IncludeRequestIdHeader

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

IncludeRequestIdHeader

Gets or sets whether the server should include the "X-Request-Id" header in response headers.

```
public bool IncludeRequestIdHeader { get; set; }
```

Property Value

bool 

Property KeepAlive

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

KeepAlive

Gets or sets whether the client should maintain a persistent connection with the HTTP server.

```
public bool KeepAlive { get; set; }
```

Property Value

[bool](#)

Property ListeningHosts

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ListeningHosts

Gets or sets the listening hosts repository that the [HttpServer](#) instance will listen to.

```
public ListeningHostRepository ListeningHosts { get; set; }
```

Property Value

[ListeningHostRepository](#)

Property MaximumContentLength

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

MaximumContentLength

Gets or sets the maximum size of a request body before it is closed by the socket.

```
public long MaximumContentLength { get; set; }
```

Property Value

[long](#)

Remarks

Leave it as "0" to set the maximum content length to unlimited.

Property NormalizeHeadersEncodings

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

NormalizeHeadersEncodings

Gets or sets whether the HTTP server should convert request headers encoding to the content encoding.

```
public bool NormalizeHeadersEncodings { get; set; }
```

Property Value

bool 

Property OptionsLogMode

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

OptionsLogMode

Gets or sets the log mode that the HTTP server should use to log OPTIONS requests.

```
public LogOutput OptionsLogMode { get; set; }
```

Property Value

[LogOutput](#)

Property RemoteRequestsAction

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RemoteRequestsAction

Gets or sets the server's action when it receives an HTTP request outside the local host.

```
public RequestListenAction RemoteRequestsAction { get; set; }
```

Property Value

[RequestListenAction](#)

Remarks

It is recommended to use [Drop](#) in this property when working with a reverse proxy or in environments where the service is not directly exposed to the internet.

Property SendSiskHeader

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

SendSiskHeader

Gets or sets whether the HTTP server should send the X-Powered-By header in all responses.

```
public bool SendSiskHeader { get; set; }
```

Property Value

bool

Property ThrowExceptions

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ThrowExceptions

Gets or sets whether the server should throw exceptions instead of reporting it on [HttpServerExecutionStatus](#) if any is thrown while processing requests.

```
public bool ThrowExceptions { get; set; }
```

Property Value

[bool](#)

Method Dispose

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Dispose()

Frees the resources and invalidates this instance.

```
public void Dispose()
```

Class HttpServerExecutionResult

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Represents the results of an request execution on the HTTP server.

```
public sealed class HttpServerExecutionResult
```

Properties

[Context](#)

Gets the [HttpContext](#) of this execution result.

[Elapsed](#)

Gets the total processing time of the HTTP session.

[IsSuccessStatus](#)

Gets an boolean indicating if this execution status is an success status.

[Request](#)

Gets the [HttpRequest](#) received in this diagnosis.

[RequestSize](#)

Gets the estimated request size in bytes.

[Response](#)

Gets the resulted [HttpResponse](#) send by the router, if any. This object can be null if the server didn't sent any response to the client.

[ResponseSize](#)

Gets the estimated response size in bytes, if any.

[ServerErrorException](#)

Gets the exception that was thrown when executing the route, if any.

[Status](#)

Gets the status of server operation.

Property Context

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Context

Gets the [HttpContext](#) of this execution result.

```
public HttpContext Context { get; }
```

Property Value

[HttpContext](#)

Property Elapsed

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Elapsed

Gets the total processing time of the HTTP session.

```
public TimeSpan Elapsed { get; }
```

Property Value

[TimeSpan](#) ↗

Property IsSuccessStatus

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

IsSuccessStatus

Gets an boolean indicating if this execution status is an success status.

```
public bool IsSuccessStatus { get; }
```

Property Value

bool

Property Request

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Request

Gets the [HttpRequest](#) received in this diagnosis.

```
public HttpRequest Request { get; }
```

Property Value

[HttpRequest](#)

Property RequestSize

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RequestSize

Gets the estimated request size in bytes.

```
public long RequestSize { get; }
```

Property Value

[long](#)

Property Response

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Response

Gets the resulted [HttpResponse](#) send by the router, if any. This object can be null if the server didn't sent any response to the client.

```
public HttpResponse? Response { get; }
```

Property Value

[HttpResponse](#)

Property ResponseSize

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ResponseSize

Gets the estimated response size in bytes, if any.

```
public long ResponseSize { get; }
```

Property Value

[long](#)

Property ServerException

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ServerException

Gets the exception that was thrown when executing the route, if any.

```
public Exception? ServerException { get; }
```

Property Value

[Exception](#)

Property Status

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Status

Gets the status of server operation.

```
public HttpServerExecutionStatus Status { get; }
```

Property Value

[HttpServerExecutionStatus](#)

Enum HttpServerExecutionStatus

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Represents the status of an execution of a request on an [HttpServer](#).

```
public enum HttpServerExecutionStatus
```

Fields

ConnectionClosed = 3

Represents that the connection stream was closed by the client.

ContentServedOnIllegalMethod = 1

Represents that the request has sent an request body with an with a HTTP method that is not indicated for receiving request contents.

ContentTooLarge = 2

Represents that the content of the request is too large than what was configured on the server, or it's bigger than the max supported size (2GB).

[Obsolete("This field is now obsolete and is not used anywhere.")] DnsFailed = 5

Represents that the client did not correctly specify a host in the request.

DnsUnknownHost = 6

Represents that the client requested an host that's not been set up on this server.

ExceptionThrown = 7

Indicates that the server encountered an exception while processing the request.

Executed = 0

Represents that the request was closed by the HTTP server and executed by a router and its response was successfully delivered.

ListeningHostNotReady = 9

Indicates that the DNS was successful, however the matched [ListeningHost](#) does not have an valid initialized router.

MalformedRequest = 10

Indicates that the server cannot or will not process the request due to something that is perceived to be a client error.

NoResponse = 4

Represents that the router did not deliver a response to the received request.

RemoteRequestDropped = 11

Indicates that the HTTP server closed an unwanted remote connection.

UncaughtExceptionThrown = 8

Indicates that the router encontered an uncaught exception while calling it's action function.

Struct HttpStatusInformation

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Represents a value that holds an HTTP response status information, with it's status code and description.

```
public readonly struct HttpStatusInformation : IEquatable<HttpStatusInformation>,
IEquatable<HttpStatusCode>, IEquatable<int>
```

Implements

[IEquatable](#)<[HttpStatusInformation](#)>, [IEquatable](#)<[HttpStatusCode](#)>, [IEquatable](#)<int>

Constructors

[HttpStatusInformation\(\)](#)

Creates an new [HttpStatusInformation](#) with default parameters (200 OK) status.

[HttpStatusInformation\(int\)](#)

Creates an new [HttpStatusInformation](#) instance with given parameters.

[HttpStatusInformation\(int, string\)](#)

Creates an new [HttpStatusInformation](#) instance with given parameters.

[HttpStatusInformation\(HttpStatusCode\)](#)

Creates an new [HttpStatusInformation](#) instance with given parameters.

Properties

[Accepted](#)

Gets an [HttpStatusInformation](#) with an HTTP 202 Accepted status.

[AlreadyReported](#)

Gets an [HttpStatusInformation](#) with an HTTP 208 Already Reported status.

[BadGateway](#)

Gets an [HttpStatusInformation](#) with an HTTP 502 Bad Gateway status.

[BadRequest](#)

Gets an [HttpStatusInformation](#) with an HTTP 400 Bad Request status.

[Conflict](#)

Gets an [HttpStatusInformation](#) with an HTTP 409 Conflict status.

[Continue](#)

Gets an [HttpStatusInformation](#) with an HTTP 100 Continue status.

[Created](#)

Gets an [HttpStatusInformation](#) with an HTTP 201 Created status.

[Description](#)

Gets the short description of the HTTP message.

[EarlyHints](#)

Gets an [HttpStatusInformation](#) with an HTTP 103 Early Hints status.

[ExpectationFailed](#)

Gets an [HttpStatusInformation](#) with an HTTP 417 Expectation Failed status.

[FailedDependency](#)

Gets an [HttpStatusInformation](#) with an HTTP 424 Failed Dependency status.

[Forbidden](#)

Gets an [HttpStatusInformation](#) with an HTTP 403 Forbidden status.

[Found](#)

Gets an [HttpStatusInformation](#) with an HTTP 302 Found status.

[GatewayTimeout](#)

Gets an [HttpStatusInformation](#) with an HTTP 504 Gateway Timeout status.

[Gone](#)

Gets an [HttpStatusInformation](#) with an HTTP 410 Gone status.

[HttpVersionNotSupported](#)

Gets an [HttpStatusInformation](#) with an HTTP 505 HTTP Version Not Supported status.

[ImATeapot](#)

Gets an [HttpStatusInformation](#) with an HTTP 418 I'm a teapot status.

[ImUsed](#)

Gets an [HttpStatusInformation](#) with an HTTP 226 IM Used status.

[InsufficientStorage](#)

Gets an [HttpStatusInformation](#) with an HTTP 507 Insufficient Storage status.

[InternalServerError](#)

Gets an [HttpStatusInformation](#) with an HTTP 500 Internal Server Error status.

[LengthRequired](#)

Gets an [HttpStatusInformation](#) with an HTTP 411 Length Required status.

[Locked](#)

Gets an [HttpStatusInformation](#) with an HTTP 423 Locked status.

[LoopDetected](#)

Gets an [HttpStatusInformation](#) with an HTTP 508 Loop Detected status.

[MethodNotAllowed](#)

Gets an [HttpStatusInformation](#) with an HTTP 405 Method Not Allowed status.

[MisdirectedRequest](#)

Gets an [HttpStatusInformation](#) with an HTTP 421 Misdirected Request status.

[MovedPermanently](#)

Gets an [HttpStatusInformation](#) with an HTTP 301 Moved Permanently status.

[MultiStatus](#)

Gets an [HttpStatusInformation](#) with an HTTP 207 Multi-Status status.

[MultipleChoices](#)

Gets an [HttpStatusInformation](#) with an HTTP 300 Multiple Choices status.

[NoContent](#)

Gets an [HttpStatusInformation](#) with an HTTP 204 No Content status.

[NonAuthoritativeInformation](#)

Gets an [HttpStatusInformation](#) with an HTTP 203 Non-Authoritative Information status.

[NotAcceptable](#)

Gets an [HttpStatusInformation](#) with an HTTP 406 Not Acceptable status.

[NotExtended](#)

Gets an [HttpStatusInformation](#) with an HTTP 510 Not Extended status.

[NotFound](#)

Gets an [HttpStatusInformation](#) with an HTTP 404 Not Found status.

[NotImplemented](#)

Gets an [HttpStatusInformation](#) with an HTTP 501 Not Implemented status.

[NotModified](#)

Gets an [HttpStatusInformation](#) with an HTTP 304 Not Modified status.

[Ok](#)

Gets an [HttpStatusInformation](#) with an HTTP 200 OK status.

[PartialContent](#)

Gets an [HttpStatusInformation](#) with an HTTP 206 Partial Content status.

[PayloadTooLarge](#)

Gets an [HttpStatusInformation](#) with an HTTP 413 Payload Too Large status.

[PaymentRequired](#)

Gets an [HttpStatusInformation](#) with an HTTP 402 Payment Required status.

[PermanentRedirect](#)

Gets an [HttpStatusInformation](#) with an HTTP 308 Permanent Redirect status.

[PreconditionFailed](#)

Gets an [HttpStatusInformation](#) with an HTTP 412 Precondition Failed status.

[PreconditionRequired](#)

Gets an [HttpStatusInformation](#) with an HTTP 428 Precondition Required status.

[Processing](#)

Gets an [HttpStatusInformation](#) with an HTTP 102 Processing status.

[ProxyAuthenticationRequired](#)

Gets an [HttpStatusInformation](#) with an HTTP 407 Proxy Authentication Required status.

[RangeNotSatisfiable](#)

Gets an [HttpStatusInformation](#) with an HTTP 416 Range Not Satisfiable status.

[RequestHeaderFieldsTooLarge](#)

Gets an [HttpStatusInformation](#) with an HTTP 431 Request Header Fields Too Large status.

[RequestTimeout](#)

Gets an [HttpStatusInformation](#) with an HTTP 408 Request Timeout status.

[ResetContent](#)

Gets an [HttpStatusInformation](#) with an HTTP 205 Reset Content status.

[SeeOther](#)

Gets an [HttpStatusInformation](#) with an HTTP 303 See Other status.

[ServiceUnavailable](#)

Gets an [HttpStatusInformation](#) with an HTTP 503 Service Unavailable status.

[StatusCode](#)

Gets the numeric HTTP status code of the HTTP message.

[SwitchProxy](#)

Gets an [HttpStatusInformation](#) with an HTTP 306 Switch Proxy status.

[SwitchingProtocols](#)

Gets an [HttpStatusInformation](#) with an HTTP 101 Switching Protocols status.

[TemporaryRedirect](#)

Gets an [HttpStatusInformation](#) with an HTTP 307 Temporary Redirect status.

[TooManyRequests](#)

Gets an [HttpStatusInformation](#) with an HTTP 429 Too Many Requests status.

[Unauthorized](#)

Gets an [HttpStatusInformation](#) with an HTTP 401 Unauthorized status.

[UnavailableForLegalReasons](#)

Gets an [HttpStatusInformation](#) with an HTTP 451 Unavailable For Legal Reasons status.

[UnprocessableEntity](#)

Gets an [HttpStatusInformation](#) with an HTTP 422 Unprocessable Entity status.

[UnsupportedMediaType](#)

Gets an [HttpStatusInformation](#) with an HTTP 415 Unsupported Media Type status.

[UpgradeRequired](#)

Gets an [HttpStatusInformation](#) with an HTTP 426 Upgrade Required status.

[UriTooLong](#)

Gets an [HttpStatusInformation](#) with an HTTP 414 URI Too Long status.

[UseProxy](#)

Gets an [HttpStatusInformation](#) with an HTTP 305 Use Proxy status.

[VariantAlsoNegotiates](#)

Gets an [HttpStatusInformation](#) with an HTTP 506 Variant Also Negotiates status.

Methods

[GetHttpStatusCode\(\)](#)

Gets an [HttpStatusCode](#) corresponding to this instance, or null if the HTTP status does not match any value.

[GetStatusCodeDescription\(int\)](#)

Gets the description of the specified HTTP status code.

[GetStatusCodeDescription\(HttpStatusCode\)](#)

Gets the description of the specified HTTP status code.

[ToString\(\)](#)

Gets an string representation of this HTTP Status Code.

Operators

[operator ==\(HttpStatusInformation, int?\)](#)

Constructor HttpStatusInformation

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

HttpStatusInformation()

Creates an new [HttpStatusInformation](#) with default parameters (200 OK) status.

```
public HttpStatusInformation()
```

HttpStatusInformation(int)

Creates an new [HttpStatusInformation](#) instance with given parameters.

```
public HttpStatusInformation(int statusCode)
```

Parameters

statusCode [int](#)

Sets the numeric HTTP status code of the HTTP message.

HttpStatusInformation(HttpStatusCode)

Creates an new [HttpStatusInformation](#) instance with given parameters.

```
public HttpStatusInformation(HttpStatusCode statusCode)
```

Parameters

statusCode [HttpStatusCode](#)

Sets the numeric HTTP status code of the HTTP message.

HttpStatusInformation(int, string)

Creates an new [HttpStatusInformation](#) instance with given parameters.

```
public HttpStatusInformation(int statusCode, string description)
```

Parameters

statusCode [int](#)

Sets the numeric HTTP status code of the HTTP message.

description [string](#)

Sets the short description of the HTTP message.

Remarks

Custom status descriptions is only supported for plain HTTP/1.1 and 1.0 transfers.

Exceptions

[ArgumentNullException](#)

Property Accepted

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Accepted

Gets an [HttpStatusInformation](#) with an HTTP 202 Accepted status.

```
public static HttpStatusInformation Accepted { get; }
```

Property Value

[HttpStatusInformation](#)

Property AlreadyReported

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

AlreadyReported

Gets an [HttpStatusInformation](#) with an HTTP 208 Already Reported status.

```
public static HttpStatusInformation AlreadyReported { get; }
```

Property Value

[HttpStatusInformation](#)

Property BadGateway

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

BadGateway

Gets an [HttpStatusInformation](#) with an HTTP 502 Bad Gateway status.

```
public static HttpStatusInformation BadGateway { get; }
```

Property Value

[HttpStatusInformation](#)

Property BadRequest

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

BadRequest

Gets an [HttpStatusInformation](#) with an HTTP 400 Bad Request status.

```
public static HttpStatusInformation BadRequest { get; }
```

Property Value

[HttpStatusInformation](#)

Property Conflict

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Conflict

Gets an [HttpStatusInformation](#) with an HTTP 409 Conflict status.

```
public static HttpStatusInformation Conflict { get; }
```

Property Value

[HttpStatusInformation](#)

Property Continue

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Continue

Gets an [HttpStatusInformation](#) with an HTTP 100 Continue status.

```
public static HttpStatusInformation Continue { get; }
```

Property Value

[HttpStatusInformation](#)

Property Created

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Created

Gets an [HttpStatusInformation](#) with an HTTP 201 Created status.

```
public static HttpStatusInformation Created { get; }
```

Property Value

[HttpStatusInformation](#)

Property Description

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Description

Gets the short description of the HTTP message.

```
public string Description { get; }
```

Property Value

[string](#)

Remarks

Custom status descriptions is only supported for plain HTTP/1.1 and 1.0 transfers.

Property EarlyHints

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

EarlyHints

Gets an [HttpStatusInformation](#) with an HTTP 103 Early Hints status.

```
public static HttpStatusInformation EarlyHints { get; }
```

Property Value

[HttpStatusInformation](#)

Property ExpectationFailed

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ExpectationFailed

Gets an [HttpStatusInformation](#) with an HTTP 417 Expectation Failed status.

```
public static HttpStatusInformation ExpectationFailed { get; }
```

Property Value

[HttpStatusInformation](#)

Property FailedDependency

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

FailedDependency

Gets an [HttpStatusInformation](#) with an HTTP 424 Failed Dependency status.

```
public static HttpStatusInformation FailedDependency { get; }
```

Property Value

[HttpStatusInformation](#)

Property Forbidden

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Forbidden

Gets an [HttpStatusInformation](#) with an HTTP 403 Forbidden status.

```
public static HttpStatusInformation Forbidden { get; }
```

Property Value

[HttpStatusInformation](#)

Property Found

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Found

Gets an [HttpStatusInformation](#) with an HTTP 302 Found status.

```
public static HttpStatusInformation Found { get; }
```

Property Value

[HttpStatusInformation](#)

Property GatewayTimeout

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GatewayTimeout

Gets an [HttpStatusInformation](#) with an HTTP 504 Gateway Timeout status.

```
public static HttpStatusInformation GatewayTimeout { get; }
```

Property Value

[HttpStatusInformation](#)

Property Gone

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Gone

Gets an [HttpStatusInformation](#) with an HTTP 410 Gone status.

```
public static HttpStatusInformation Gone { get; }
```

Property Value

[HttpStatusInformation](#)

Property HttpVersionNotSupported

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

HttpVersionNotSupported

Gets an [HttpStatusInformation](#) with an HTTP 505 HTTP Version Not Supported status.

```
public static HttpStatusInformation HttpVersionNotSupported { get; }
```

Property Value

[HttpStatusInformation](#)

Property ImATeapot

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ImATeapot

Gets an [HttpStatusInformation](#) with an HTTP 418 I'm a teapot status.

```
public static HttpStatusInformation ImATeapot { get; }
```

Property Value

[HttpStatusInformation](#)

Property ImUsed

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ImUsed

Gets an [HttpStatusInformation](#) with an HTTP 226 IM Used status.

```
public static HttpStatusInformation ImUsed { get; }
```

Property Value

[HttpStatusInformation](#)

Property InsufficientStorage

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

InsufficientStorage

Gets an [HttpStatusInformation](#) with an HTTP 507 Insufficient Storage status.

```
public static HttpStatusInformation InsufficientStorage { get; }
```

Property Value

[HttpStatusInformation](#)

Property InternalServerError

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

InternalServerError

Gets an [HttpStatusInformation](#) with an HTTP 500 Internal Server Error status.

```
public static HttpStatusInformation InternalServerError { get; }
```

Property Value

[HttpStatusInformation](#)

Property LengthRequired

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

LengthRequired

Gets an [HttpStatusInformation](#) with an HTTP 411 Length Required status.

```
public static HttpStatusInformation LengthRequired { get; }
```

Property Value

[HttpStatusInformation](#)

Property Locked

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Locked

Gets an [HttpStatusInformation](#) with an HTTP 423 Locked status.

```
public static HttpStatusInformation Locked { get; }
```

Property Value

[HttpStatusInformation](#)

Property LoopDetected

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

LoopDetected

Gets an [HttpStatusInformation](#) with an HTTP 508 Loop Detected status.

```
public static HttpStatusInformation LoopDetected { get; }
```

Property Value

[HttpStatusInformation](#)

Property MethodNotAllowed

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

MethodNotAllowed

Gets an [HttpStatusInformation](#) with an HTTP 405 Method Not Allowed status.

```
public static HttpStatusInformation MethodNotAllowed { get; }
```

Property Value

[HttpStatusInformation](#)

Property MisdirectedRequest

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

MisdirectedRequest

Gets an [HttpStatusInformation](#) with an HTTP 421 Misdirected Request status.

```
public static HttpStatusInformation MisdirectedRequest { get; }
```

Property Value

[HttpStatusInformation](#)

Property MovedPermanently

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

MovedPermanently

Gets an [HttpStatusInformation](#) with an HTTP 301 Moved Permanently status.

```
public static HttpStatusInformation MovedPermanently { get; }
```

Property Value

[HttpStatusInformation](#)

Property MultiStatus

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

MultiStatus

Gets an [HttpStatusInformation](#) with an HTTP 207 Multi-Status status.

```
public static HttpStatusInformation MultiStatus { get; }
```

Property Value

[HttpStatusInformation](#)

Property MultipleChoices

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

MultipleChoices

Gets an [HttpStatusInformation](#) with an HTTP 300 Multiple Choices status.

```
public static HttpStatusInformation MultipleChoices { get; }
```

Property Value

[HttpStatusInformation](#)

Property NoContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

NoContent

Gets an [HttpStatusInformation](#) with an HTTP 204 No Content status.

```
public static HttpStatusInformation NoContent { get; }
```

Property Value

[HttpStatusInformation](#)

Property NonAuthoritativeInformation

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

NonAuthoritativeInformation

Gets an [HttpStatusInformation](#) with an HTTP 203 Non-Authoritative Information status.

```
public static HttpStatusInformation NonAuthoritativeInformation { get; }
```

Property Value

[HttpStatusInformation](#)

Property NotAcceptable

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

NotAcceptable

Gets an [HttpStatusInformation](#) with an HTTP 406 Not Acceptable status.

```
public static HttpStatusInformation NotAcceptable { get; }
```

Property Value

[HttpStatusInformation](#)

Property NotExtended

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

NotExtended

Gets an [HttpStatusInformation](#) with an HTTP 510 Not Extended status.

```
public static HttpStatusInformation NotExtended { get; }
```

Property Value

[HttpStatusInformation](#)

Property NotFound

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

NotFound

Gets an [HttpStatusInformation](#) with an HTTP 404 Not Found status.

```
public static HttpStatusInformation NotFound { get; }
```

Property Value

[HttpStatusInformation](#)

Property NotImplemented

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

NotImplemented

Gets an [HttpStatusInformation](#) with an HTTP 501 Not Implemented status.

```
public static HttpStatusInformation NotImplemented { get; }
```

Property Value

[HttpStatusInformation](#)

Property NotModified

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

NotModified

Gets an [HttpStatusInformation](#) with an HTTP 304 Not Modified status.

```
public static HttpStatusInformation NotModified { get; }
```

Property Value

[HttpStatusInformation](#)

Property Ok

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Ok

Gets an [HttpStatusInformation](#) with an HTTP 200 OK status.

```
public static HttpStatusInformation Ok { get; }
```

Property Value

[HttpStatusInformation](#)

Property PartialContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

PartialContent

Gets an [HttpStatusInformation](#) with an HTTP 206 Partial Content status.

```
public static HttpStatusInformation PartialContent { get; }
```

Property Value

[HttpStatusInformation](#)

Property PayloadTooLarge

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

PayloadTooLarge

Gets an [HttpStatusInformation](#) with an HTTP 413 Payload Too Large status.

```
public static HttpStatusInformation PayloadTooLarge { get; }
```

Property Value

[HttpStatusInformation](#)

Property PaymentRequired

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

PaymentRequired

Gets an [HttpStatusInformation](#) with an HTTP 402 Payment Required status.

```
public static HttpStatusInformation PaymentRequired { get; }
```

Property Value

[HttpStatusInformation](#)

Property PermanentRedirect

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

PermanentRedirect

Gets an [HttpStatusInformation](#) with an HTTP 308 Permanent Redirect status.

```
public static HttpStatusInformation PermanentRedirect { get; }
```

Property Value

[HttpStatusInformation](#)

Property PreconditionFailed

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

PreconditionFailed

Gets an [HttpStatusInformation](#) with an HTTP 412 Precondition Failed status.

```
public static HttpStatusInformation PreconditionFailed { get; }
```

Property Value

[HttpStatusInformation](#)

Property PreconditionRequired

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

PreconditionRequired

Gets an [HttpStatusInformation](#) with an HTTP 428 Precondition Required status.

```
public static HttpStatusInformation PreconditionRequired { get; }
```

Property Value

[HttpStatusInformation](#)

Property Processing

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Processing

Gets an [HttpStatusInformation](#) with an HTTP 102 Processing status.

```
public static HttpStatusInformation Processing { get; }
```

Property Value

[HttpStatusInformation](#)

Property ProxyAuthenticationRequired

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ProxyAuthenticationRequired

Gets an [HttpStatusInformation](#) with an HTTP 407 Proxy Authentication Required status.

```
public static HttpStatusInformation ProxyAuthenticationRequired { get; }
```

Property Value

[HttpStatusInformation](#)

Property RangeNotSatisfiable

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RangeNotSatisfiable

Gets an [HttpStatusInformation](#) with an HTTP 416 Range Not Satisfiable status.

```
public static HttpStatusInformation RangeNotSatisfiable { get; }
```

Property Value

[HttpStatusInformation](#)

Property RequestHeaderFieldsTooLarge

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RequestHeaderFieldsTooLarge

Gets an [HttpStatusInformation](#) with an HTTP 431 Request Header Fields Too Large status.

```
public static HttpStatusInformation RequestHeaderFieldsTooLarge { get; }
```

Property Value

[HttpStatusInformation](#)

Property RequestTimeout

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RequestTimeout

Gets an [HttpStatusInformation](#) with an HTTP 408 Request Timeout status.

```
public static HttpStatusInformation RequestTimeout { get; }
```

Property Value

[HttpStatusInformation](#)

Property ResetContent

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ResetContent

Gets an [HttpStatusInformation](#) with an HTTP 205 Reset Content status.

```
public static HttpStatusInformation ResetContent { get; }
```

Property Value

[HttpStatusInformation](#)

Property SeeOther

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

SeeOther

Gets an [HttpStatusInformation](#) with an HTTP 303 See Other status.

```
public static HttpStatusInformation SeeOther { get; }
```

Property Value

[HttpStatusInformation](#)

Property ServiceUnavailable

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ServiceUnavailable

Gets an [HttpStatusInformation](#) with an HTTP 503 Service Unavailable status.

```
public static HttpStatusInformation ServiceUnavailable { get; }
```

Property Value

[HttpStatusInformation](#)

Property StatusCode

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

StatusCode

Gets the numeric HTTP status code of the HTTP message.

```
public int StatusCode { get; }
```

Property Value

[int](#)

Property SwitchProxy

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

SwitchProxy

Gets an [HttpStatusInformation](#) with an HTTP 306 Switch Proxy status.

```
public static HttpStatusInformation SwitchProxy { get; }
```

Property Value

[HttpStatusInformation](#)

Property SwitchingProtocols

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

SwitchingProtocols

Gets an [HttpStatusInformation](#) with an HTTP 101 Switching Protocols status.

```
public static HttpStatusInformation SwitchingProtocols { get; }
```

Property Value

[HttpStatusInformation](#)

Property TemporaryRedirect

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

TemporaryRedirect

Gets an [HttpStatusInformation](#) with an HTTP 307 Temporary Redirect status.

```
public static HttpStatusInformation TemporaryRedirect { get; }
```

Property Value

[HttpStatusInformation](#)

Property TooManyRequests

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

TooManyRequests

Gets an [HttpStatusInformation](#) with an HTTP 429 Too Many Requests status.

```
public static HttpStatusInformation TooManyRequests { get; }
```

Property Value

[HttpStatusInformation](#)

Property Unauthorized

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Unauthorized

Gets an [HttpStatusInformation](#) with an HTTP 401 Unauthorized status.

```
public static HttpStatusInformation Unauthorized { get; }
```

Property Value

[HttpStatusInformation](#)

Property UnavailableForLegalReasons

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

UnavailableForLegalReasons

Gets an [HttpStatusInformation](#) with an HTTP 451 Unavailable For Legal Reasons status.

```
public static HttpStatusInformation UnavailableForLegalReasons { get; }
```

Property Value

[HttpStatusInformation](#)

Property UnprocessableEntity

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

UnprocessableEntity

Gets an [HttpStatusInformation](#) with an HTTP 422 Unprocessable Entity status.

```
public static HttpStatusInformation UnprocessableEntity { get; }
```

Property Value

[HttpStatusInformation](#)

Property UnsupportedMediaType

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

UnsupportedMediaType

Gets an [HttpStatusInformation](#) with an HTTP 415 Unsupported Media Type status.

```
public static HttpStatusInformation UnsupportedMediaType { get; }
```

Property Value

[HttpStatusInformation](#)

Property UpgradeRequired

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

UpgradeRequired

Gets an [HttpStatusInformation](#) with an HTTP 426 Upgrade Required status.

```
public static HttpStatusInformation UpgradeRequired { get; }
```

Property Value

[HttpStatusInformation](#)

Property UriTooLong

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

UriTooLong

Gets an [HttpStatusInformation](#) with an HTTP 414 URI Too Long status.

```
public static HttpStatusInformation UriTooLong { get; }
```

Property Value

[HttpStatusInformation](#)

Property UseProxy

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

UseProxy

Gets an [HttpStatusInformation](#) with an HTTP 305 Use Proxy status.

```
public static HttpStatusInformation UseProxy { get; }
```

Property Value

[HttpStatusInformation](#)

Property VariantAlsoNegotiates

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

VariantAlsoNegotiates

Gets an [HttpStatusInformation](#) with an HTTP 506 Variant Also Negotiates status.

```
public static HttpStatusInformation VariantAlsoNegotiates { get; }
```

Property Value

[HttpStatusInformation](#)

Method GetHttpStatusCode

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetHttpStatusCode()

Gets an [HttpStatusCode](#) corresponding to this instance, or null if the HTTP status does not match any value.

```
public HttpStatusCode? GetHttpStatusCode()
```

Returns

[HttpStatusCode](#)?

An [HttpStatusCode](#) or null if the HTTP status matches no entry on it.

Method GetStatusCodeDescription

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetStatusCodeDescription(int)

Gets the description of the specified HTTP status code.

```
public static string GetStatusCodeDescription(int statusCode)
```

Parameters

statusCode [int](#)

The HTTP status code.

Returns

[string](#)

The description of the HTTP status code.

GetStatusCodeDescription(HttpStatusCode)

Gets the description of the specified HTTP status code.

```
public static string GetStatusCodeDescription(HttpStatusCode statusCode)
```

Parameters

statusCode [HttpStatusCode](#)

The [HttpStatusCode](#) value.

Returns

string 

The description of the HTTP status code.

Method ToString

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ToString()

Gets an string representation of this HTTP Status Code.

```
public override string ToString()
```

Returns

[string](#)

Operator operator ==

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

operator ==(HttpStatusInformation, int?)

```
public static bool operator ==(HttpStatusInformation a, int? b)
```

Parameters

a [HttpStatusInformation](#)

b [int?](#)

Returns

[bool](#)

Class ListeningHost

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Provides a structure to contain the fields needed by an http server host.

```
public sealed class ListeningHost
```

Constructors

[ListeningHost\(\)](#)

Creates an new empty [ListeningHost](#) instance.

[ListeningHost\(params ListeningPort\[\]\)](#)

Creates an new [ListeningHost](#) instance with given array of [ListeningPort](#).

[ListeningHost\(string, Router\)](#)

Creates an new [ListeningHost](#) instance with given URL.

Properties

[CanListen](#)

Gets whether this [ListeningHost](#) can be listened by it's host [HttpServer](#).

[CrossOriginResourceSharingPolicy](#)

Gets or sets the CORS sharing policy object.

[Label](#)

Gets or sets a label for this Listening Host.

[Ports](#)

Gets or sets the list of [ListeningPort](#) that this host will listen on.

Router

Gets or sets the [Router](#) for this [ListeningHost](#) instance.

Methods

[Equals\(object?\)](#)

Determines if another object is equals to this class instance.

[GetHashCode\(\)](#)

Gets the hash code for this listening host.

Constructor ListeningHost

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ListeningHost()

Creates an new empty [ListeningHost](#) instance.

```
public ListeningHost()
```

ListeningHost(params ListeningPort[])

Creates an new [ListeningHost](#) instance with given array of [ListeningPort](#).

```
public ListeningHost(params ListeningPort[] ports)
```

Parameters

ports [ListeningPort](#)[]

The array of [ListeningPort](#) to listen in the [ListeningHost](#).

ListeningHost(string, Router)

Creates an new [ListeningHost](#) instance with given URL.

```
public ListeningHost(string uri, Router r)
```

Parameters

uri [string](#)

The well formatted URL with scheme, hostname and port.

Router

The router which will handle this listener requests.

Property CanListen

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

CanListen

Gets whether this [ListeningHost](#) can be listened by it's host [HttpServer](#).

```
public bool CanListen { get; }
```

Property Value

bool ↗

Property CrossOriginResourceSharingPolicy

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

CrossOriginResourceSharingPolicy

Gets or sets the CORS sharing policy object.

```
public CrossOriginResourceSharingHeaders CrossOriginResourceSharingPolicy { get; set; }
```

Property Value

[CrossOriginResourceSharingHeaders](#)

Property Label

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Label

Gets or sets a label for this Listening Host.

```
public string? Label { get; set; }
```

Property Value

[string](#)

Property Ports

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Ports

Gets or sets the list of [ListeningPort](#) that this host will listen on.

```
public IList<ListeningPort> Ports { get; set; }
```

Property Value

[IList](#)<[ListeningPort](#)>

Property Router

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Router

Gets or sets the [Router](#) for this [ListeningHost](#) instance.

```
public Router? Router { get; set; }
```

Property Value

[Router](#)

Method Equals

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Equals(object?)

Determines if another object is equals to this class instance.

```
public override bool Equals(object? obj)
```

Parameters

obj [object](#)

The another object which will be used to compare.

Returns

[bool](#)

Method GetHashCode

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetHashCode()

Gets the hash code for this listening host.

```
public override int GetHashCode()
```

Returns

[int](#)

Class ListeningHostRepository

Namespace:[Sisk.Core.Http](#)

Assembly:[Sisk.Core.dll](#)

Represents an fluent repository of [ListeningHost](#) that can add, modify, or remove listening hosts while an [Http Server](#) is running.

```
public sealed class ListeningHostRepository : IList<ListeningHost>, ICollection<ListeningHost>,  
IEnumerable<ListeningHost>, IEnumerable
```

Implements

[IList](#)<ListeningHost>, [ICollection](#)<ListeningHost>, [IEnumerable](#)<ListeningHost>, [IEnumerable](#)

Constructors

[ListeningHostRepository\(\)](#)

Creates a new instance of an empty [ListeningHostRepository](#).

[ListeningHostRepository\(IEnumerable<ListeningHost>\)](#)

Creates a new instance of an [ListeningHostRepository](#) copying the items from another collection of [Listening Host](#).

Properties

[Count](#)

Gets the number of elements contained in this [ListeningHostRepository](#).

[IsReadOnly](#)

Gets an boolean indicating if this [ListeningHostRepository](#) is read only. This property always returns [false](#).

[this\[int\]](#)

Methods

[Add\(ListeningHost\)](#)

Adds a listeninghost to this repository. If this listeninghost already exists in this class, an exception will be thrown.

[Clear\(\)](#)

Removes all listeninghosts from this repository.

[Contains\(ListeningHost\)](#)

Determines if an [ListeningHost](#) is present in this repository.

[CopyTo\(ListeningHost\[\], int\)](#)

Copies all elements from this repository to another compatible repository.

[GetEnumerator\(\)](#)

Returns an enumerator that iterates through this [ListeningHostRepository](#).

[IndexOf\(ListeningHost\)](#)

[Insert\(int, ListeningHost\)](#)

[Remove\(ListeningHost\)](#)

Try to remove a [ListeningHost](#) from this repository. If the item is removed, this methods returns [true](#)✓.

[RemoveAt\(int\)](#)

Constructor ListeningHostRepository

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ListeningHostRepository()

Creates a new instance of an empty [ListeningHostRepository](#).

```
public ListeningHostRepository()
```

ListeningHostRepository(IEnumerable<ListeningHost>)

Creates a new instance of an [ListeningHostRepository](#) copying the items from another collection of [Listening Host](#).

```
public ListeningHostRepository(IEnumerable<ListeningHost> hosts)
```

Parameters

hosts [IEnumerable](#)<ListeningHost>

The collection which stores the [ListeningHost](#) which will be copied to this repository.

Property Count

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Count

Gets the number of elements contained in this [ListeningHostRepository](#).

```
public int Count { get; }
```

Property Value

[int](#)

Property IsReadOnly

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

IsReadOnly

Gets an boolean indicating if this [ListeningHostRepository](#) is read only. This property always returns [false](#).

```
public bool IsReadOnly { get; }
```

Property Value

[bool](#)

Property this

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

this[int]

```
public ListeningHost this[int index] { get; set; }
```

Parameters

index int ↗

Property Value

[ListeningHost](#)

Method Add

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Add(ListeningHost)

Adds a listeninghost to this repository. If this listeninghost already exists in this class, an exception will be thrown.

```
public void Add(ListeningHost item)
```

Parameters

item [ListeningHost](#)

The [ListeningHost](#) to add to this collection.

Method Clear

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Clear()

Removes all listeninghosts from this repository.

```
public void Clear()
```

Method Contains

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Contains(ListeningHost)

Determines if an [ListeningHost](#) is present in this repository.

```
public bool Contains(ListeningHost item)
```

Parameters

item [ListeningHost](#)

The [ListeningHost](#) to check if is present in this repository.

Returns

[bool](#) ↗

Method CopyTo

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

CopyTo(ListeningHost[], int)

Copies all elements from this repository to another compatible repository.

```
public void CopyTo(ListeningHost[] array, int arrayIndex)
```

Parameters

array [ListeningHost\[\]](#)

The one-dimensional System.Array that is the destination of the elements copied.

arrayIndex [int](#)

The zero-based index in array at which copying begins.

Method GetEnumerator

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetEnumerator()

Returns an enumerator that iterates through this [ListeningHostRepository](#).

```
public IEnumarator<ListeningHost> GetEnumerator()
```

Returns

[IEnumarator](#)<ListeningHost>

Method IndexOf

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

IndexOf(ListeningHost)

```
public int IndexOf(ListeningHost item)
```

Parameters

item [ListeningHost](#)

Returns

[int](#)

Method Insert

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

`Insert(int, ListeningHost)`

```
public void Insert(int index, ListeningHost item)
```

Parameters

`index` [int](#)

`item` [ListeningHost](#)

Method Remove

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Remove(ListeningHost)

Try to remove a [ListeningHost](#) from this repository. If the item is removed, this methods returns [true](#).

```
public bool Remove(ListeningHost item)
```

Parameters

item [ListeningHost](#)

The [ListeningHost](#) to be removed.

Returns

[bool](#)

Method RemoveAt

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RemoveAt(int)

```
public void RemoveAt(int index)
```

Parameters

index int ↗

Struct ListeningPort

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Provides a structure to contain a listener port for an [ListeningHost](#) instance.

```
public readonly struct ListeningPort : IEquatable<ListeningPort>, IParsable<ListeningPort>
```

Implements

[IEquatable](#)<ListeningPort>, [IParsable](#)<ListeningPort>

Examples

A listener port represents an access point on the HTTP server. It consists of an indicator that it should use a secure connection (HTTPS), its hostname and port.

It must start with https:// or http://, and must terminate with an /.

It is represented by the syntax:

```
[http|https]://[hostname]:[port]/
```

Examples:

```
1  
2  
http://localhost:80/  
https://subdomain.domain.net:443/  
http://182.32.112.223:5251/
```

Constructors

[ListeningPort\(\)](#)

Creates an new [ListeningPort](#) instance with default parameters.

[ListeningPort\(bool, string, ushort\)](#)

Creates an new [ListeningPort](#) instance with the specified port, secure context and hostname.

[ListeningPort\(bool, string, ushort, string\)](#)

Creates an new [ListeningPort](#) instance with the specified port, secure context, hostname and path.

[ListeningPort\(string\)](#)

Creates an new [ListeningPort](#) instance with the specified URI.

[ListeningPort\(ushort\)](#)

Creates an new [ListeningPort](#) instance with the specified port at the loopback host.

[ListeningPort\(ushort, bool\)](#)

Creates an new [ListeningPort](#) instance with the specified port and secure context at the loopback host.

Properties

[Hostname](#)

Gets the DNS hostname pattern where this listening port will refer.

[IsPathRoot](#)

Gets an boolean indicating if this listening port has an non-rooted path.

[Path](#)

Gets where this listening port prefix is listening to.

[Port](#)

Gets the port where this listening port will refer.

[Secure](#)

Gets whether the server should listen to this port securely (SSL).

Methods

[Equals\(ListeningPort\)](#)

Determines if this [ListeningPort](#) is equals to another [ListeningPort](#).

[Equals\(object?\)](#)

Determines if another object is equals to this class instance.

[GetHashCode\(\)](#)

Gets the hash code for this listening port.

[GetRandomPort\(\)](#)

Gets an [ListeningPort](#) object with an random insecure port at the default loopback address.

[Parse\(string\)](#)

Parses a string into a [ListeningPort](#).

[ToString\(\)](#)

Gets an string representation of this [ListeningPort](#).

[ToString\(bool\)](#)

Gets an string representation of this [ListeningPort](#).

Operators

[operator ==\(ListeningPort, ListeningPort\)](#)

[operator !=\(ListeningPort, ListeningPort\)](#)

Explicit Interface Implementations

[Parse\(string, IFormatProvider?\)](#)

Parses a string into a [ListeningPort](#).

[TryParse\(string?, IFormatProvider?, out ListeningPort\)](#)

Tries to parse a string into a [ListeningPort](#).

Constructor ListeningPort

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ListeningPort()

Creates an new [ListeningPort](#) instance with default parameters.

```
public ListeningPort()
```

ListeningPort(ushort)

Creates an new [ListeningPort](#) instance with the specified port at the loopback host.

```
public ListeningPort(ushort port)
```

Parameters

port [ushort](#)

The port the server will listen on. If this port is the default HTTPS port (443), the class will have the property [Secure](#) to true.

ListeningPort(ushort, bool)

Creates an new [ListeningPort](#) instance with the specified port and secure context at the loopback host.

```
public ListeningPort(ushort port, bool secure)
```

Parameters

port [ushort](#)

The port the server will listen on.

secure [bool](#)

Indicates whether the server should listen to this port securely (SSL).

ListeningPort(bool, string, ushort)

Creates an new [ListeningPort](#) instance with the specified port, secure context and hostname.

```
public ListeningPort(bool secure, string hostname, ushort port)
```

Parameters

secure [bool](#)

Indicates whether the server should listen to this port securely (SSL).

hostname [string](#)

The hostname DNS pattern the server will listen to.

port [ushort](#)

The port the server will listen on.

ListeningPort(bool, string, ushort, string)

Creates an new [ListeningPort](#) instance with the specified port, secure context, hostname and path.

```
public ListeningPort(bool secure, string hostname, ushort port, string path)
```

Parameters

secure [bool](#)

Indicates whether the server should listen to this port securely (SSL).

hostname [string](#)

The hostname DNS pattern the server will listen to.

port `ushort`

The port the server will listen on.

path `string`

The prefix path.

ListeningPort(string)

Creates an new [ListeningPort](#) instance with the specified URI.

```
public ListeningPort(string uri)
```

Parameters

uri `string`

The URI component that will be parsed to the listening port format.

Property Hostname

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Hostname

Gets the DNS hostname pattern where this listening port will refer.

```
public string Hostname { get; }
```

Property Value

[string](#)

Property IsPathRoot

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

IsPathRoot

Gets an boolean indicating if this listening port has an non-rooted path.

```
public bool IsPathRoot { get; }
```

Property Value

bool ↗

Property Path

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Path

Gets where this listening port prefix is listening to.

```
public string Path { get; }
```

Property Value

[string](#)

Property Port

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Port

Gets the port where this listening port will refer.

```
public ushort Port { get; }
```

Property Value

[ushort](#)

Property Secure

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Secure

Gets whether the server should listen to this port securely (SSL).

```
public bool Secure { get; }
```

Property Value

[bool](#)

Method Equals

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Equals(object?)

Determines if another object is equals to this class instance.

```
public override bool Equals(object? obj)
```

Parameters

obj [object](#)

The another object which will be used to compare.

Returns

[bool](#)

Equals(ListeningPort)

Determines if this [ListeningPort](#) is equals to another [ListeningPort](#).

```
public bool Equals(ListeningPort other)
```

Parameters

other [ListeningPort](#)

The another object which will be used to compare.

Returns

[bool](#)

Method GetHashCode

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetHashCode()

Gets the hash code for this listening port.

```
public override int GetHashCode()
```

Returns

[int](#)

Method GetRandomPort

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

GetRandomPort()

Gets an [ListeningPort](#) object with an random insecure port at the default loopback address.

```
public static ListeningPort GetRandomPort()
```

Returns

[ListeningPort](#)

Method Parse

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Parse(string)

Parses a string into a [ListeningPort](#).

```
public static ListeningPort Parse(string s)
```

Parameters

s [string](#)

The string to parse.

Returns

[ListeningPort](#)

Parse(string, IFormatProvider?)

Parses a string into a [ListeningPort](#).

```
static ListeningPort Parse(string s, IFormatProvider? provider)
```

Parameters

s [string](#)

The string to parse.

provider [IFormatProvider](#)

An object that provides culture-specific formatting information about s.

Returns

[ListeningPort](#)

Method ToString

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ToString()

Gets an string representation of this [ListeningPort](#).

```
public override string ToString()
```

Returns

[string](#)

ToString(bool)

Gets an string representation of this [ListeningPort](#).

```
public string ToString(bool includePath = true)
```

Parameters

includePath [bool](#)

Optional. Defines whether the path should be included in the result string.

Returns

[string](#)

Operator operator !=

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

operator !=(ListeningPort, ListeningPort)

```
public static bool operator !=(ListeningPort left, ListeningPort right)
```

Parameters

left [ListeningPort](#)

right [ListeningPort](#)

Returns

[bool](#)

Operator operator ==

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

operator ==(ListeningPort, ListeningPort)

```
public static bool operator ==(ListeningPort left, ListeningPort right)
```

Parameters

left [ListeningPort](#)

right [ListeningPort](#)

Returns

[bool](#)

Method TryParse

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

TryParse(string?, IFormatProvider?, out ListeningPort)

Tries to parse a string into a [ListeningPort](#).

```
static bool TryParse(string? s, IFormatProvider? provider, out ListeningPort result)
```

Parameters

s [string](#)

The string to parse.

provider [IFormatProvider](#)

An object that provides culture-specific formatting information about s.

result [ListeningPort](#)

When this method returns, contains the result of successfully parsing s or an undefined value on failure.

Returns

[bool](#)

Class LogStream

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Provides a managed, asynchronous log writer which supports writing safe data to log files or text streams.

```
public class LogStream : IDisposable, IAsyncDisposable
```

Implements

[IDisposable](#), [IAsyncDisposable](#)

Constructors

[LogStream\(\)](#)

Creates an new [LogStream](#) instance with no predefined outputs.

[LogStream\(TextWriter\)](#)

Creates an new [LogStream](#) instance with the given [TextWriter](#) object.

[LogStream\(string\)](#)

Creates an new [LogStream](#) instance with the given relative or absolute file path.

[LogStream\(string?, TextWriter?\)](#)

Creates an new [LogStream](#) instance which writes text to an file and an [TextWriter](#).

Properties

[ConsoleOutput](#)

Gets a shared [LogStream](#) that writes its output to the [Out](#) stream.

[Disposed](#)

Gets an boolean indicating if this [LogStream](#) was disposed.

[Empty](#)

Gets a shared [LogStream](#) without any output stream.

[Encoding](#)

Gets or sets the encoding used for writing data to the output file. This property is only applicable if this instance is using an file-based output.

[FilePath](#)

Gets or sets the absolute path to the file where the log is being written to.

[IsBuffering](#)

Gets an boolean indicating if this [LogStream](#) is buffering output messages to their internal message buffer.

[NormalizeEntries](#)

Gets or sets a boolean that indicates that every input must be trimmed and have their line endings normalized before being written to the output stream.

[RotatingPolicy](#)

Gets the defined [RotatingLogPolicy](#) for this [LogStream](#).

[TextWriter](#)

Gets the [TextWriter](#) object where the log is being written to.

Methods

[Close\(\)](#)

Writes all pending logs from the queue and closes all resources used by this object.

[ConfigureRotatingPolicy\(long, TimeSpan\)](#)

Defines the time interval and size threshold for starting the task, and then starts the task. This method is a shortcut for calling [Configure\(long, TimeSpan\)](#) of this defined [RotatingPolicy](#) method.

[Dispose\(\)](#)

Writes all pending logs from the queue and closes all resources used by this object.

[DisposeAsync\(\)](#)

Asynchronously writes all pending logs from the queue and closes all resources used by this object.

[~LogStream\(\)](#)

[Flush\(\)](#)

Blocks the current thread until all currently enqueued content is written to the underlying streams.

[FlushAsync\(\)](#)

Asynchronously waits until all currently enqueued content is written to the underlying streams.

[Peek\(\)](#)

Reads the output buffer. To use this method, it's required to set this [LogStream](#) buffering with [StartBuffering\(int\)](#).

[StartBuffering\(int\)](#)

Start buffering all output to an alternate stream in memory for readability with [Peek\(\)](#) later.

[StopBuffering\(\)](#)

Stops buffering output.

[WriteException\(Exception\)](#)

Writes an exception description in the log.

[WriteException\(Exception, string?\)](#)

Writes an exception description in the log.

[WriteExceptionAsync\(Exception\)](#)

Writes an exception description in the log.

[WriteExceptionAsync\(Exception, string?\)](#)

Writes an exception description in the log.

[WriteLine\(\)](#)

Writes an line-break at the end of the output.

[WriteLine\(IFormatProvider?, string, params object?\[\]\)](#)

Writes the text format and arguments and appends a line-break at the end into the output, using the specified format provider.

[WriteLine\(object?\)](#)

Writes the text and concats an line-break at the end into the output.

[WriteLine\(string\)](#)

Writes the text and concats an line-break at the end into the output.

[WriteLine\(string, IEnumerable<object?>\)](#)

Writes the text format and arguments and concats an line-break at the end into the output.

[WriteLine\(string, scoped ReadOnlySpan<object?>\)](#)

Writes the text format and arguments and concats an line-break at the end into the output.

[WriteLineAsync\(\)](#)

Writes an line-break at the end of the output.

[WriteLineAsync\(IFormatProvider?, string, params object?\[\]\)](#)

Writes the text format and arguments and appends a line-break at the end into the output, using the specified format provider.

[WriteLineAsync\(object?\)](#)

Writes the text and concats an line-break at the end into the output.

[WriteLineAsync\(string\)](#)

Writes the text and concats an line-break at the end into the output.

[WriteLineAsync\(string, IEnumerable<object?>\)](#)

Writes the text format and arguments and concats an line-break at the end into the output.

[WriteLineInternal\(string\)](#)

Represents the method that intercepts the line that will be written to an output log before being queued for writing. This method will block if the log queue is full.

[WriteLineInternalAsync\(string\)](#)

Represents the asynchronous method that intercepts the line that will be written to an output log before being queued for writing.

Constructor LogStream

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

LogStream()

Creates an new [LogStream](#) instance with no predefined outputs.

```
public LogStream()
```

LogStream(TextWriter)

Creates an new [LogStream](#) instance with the given TextWriter object.

```
public LogStream(TextWriter tw)
```

Parameters

tw [TextWriter](#)

The [TextWriter](#) instance which this instance will write log to.

LogStream(string)

Creates an new [LogStream](#) instance with the given relative or absolute file path.

```
public LogStream(string filename)
```

Parameters

filename [string](#)

The file path where this instance will write log to.

LogStream(string?, TextWriter?)

Creates an new [LogStream](#) instance which writes text to an file and an [TextWriter](#).

```
public LogStream(string? filename, TextWriter? tw)
```

Parameters

filename [string](#)

The file path where this instance will write log to.

tw [TextWriter](#)

The text writer which this instance will write log to.

Property ConsoleOutput

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ConsoleOutput

Gets a shared [LogStream](#) that writes its output to the [Out](#) stream.

```
public static LogStream ConsoleOutput { get; }
```

Property Value

[LogStream](#)

Property Disposed

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Disposed

Gets an boolean indicating if this [LogStream](#) was disposed.

```
public bool Disposed { get; }
```

Property Value

bool ↗

Property Empty

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Empty

Gets a shared [LogStream](#) without any output stream.

```
public static LogStream Empty { get; }
```

Property Value

[LogStream](#)

Property Encoding

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Encoding

Gets or sets the encoding used for writing data to the output file. This property is only applicable if this instance is using an file-based output.

```
public Encoding Encoding { get; set; }
```

Property Value

[Encoding](#)

Property FilePath

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

FilePath

Gets or sets the absolute path to the file where the log is being written to.

```
public string? FilePath { get; set; }
```

Property Value

[string](#)

Remarks

When setting this method, if the file directory doesn't exists, it is created.

Property IsBuffering

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

IsBuffering

Gets an boolean indicating if this [LogStream](#) is buffering output messages to their internal message buffer.

```
public bool IsBuffering { get; }
```

Property Value

bool 

Property NormalizeEntries

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

NormalizeEntries

Gets or sets a boolean that indicates that every input must be trimmed and have their line endings normalized before being written to the output stream.

```
public bool NormalizeEntries { get; set; }
```

Property Value

[bool](#)

Property RotatingPolicy

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RotatingPolicy

Gets the defined [RotatingLogPolicy](#) for this [LogStream](#).

```
public RotatingLogPolicy RotatingPolicy { get; }
```

Property Value

[RotatingLogPolicy](#)

Property TextWriter

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

TextWriter

Gets the [TextWriter](#) object where the log is being written to.

```
public TextWriter? TextWriter { get; set; }
```

Property Value

[TextWriter](#)

Method Close

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Close()

Writes all pending logs from the queue and closes all resources used by this object.

```
public void Close()
```

Method ConfigureRotatingPolicy

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

ConfigureRotatingPolicy(long, TimeSpan)

Defines the time interval and size threshold for starting the task, and then starts the task. This method is an shortcut for calling [Configure\(long, TimeSpan\)](#) of this defined [RotatingPolicy](#) method.

```
public LogStream ConfigureRotatingPolicy(long maximumSize, TimeSpan dueTime)
```

Parameters

maximumSize [long](#)

The non-negative size threshold of the log file size in byte count.

dueTime [TimeSpan](#)

The time interval between checks.

Returns

[LogStream](#)

Remarks

The first run is performed immediately after calling this method.

Method Dispose

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Dispose()

Writes all pending logs from the queue and closes all resources used by this object.

```
public void Dispose()
```

Method DisposeAsync

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

DisposeAsync()

Asynchronously writes all pending logs from the queue and closes all resources used by this object.

```
public ValueTask DisposeAsync()
```

Returns

[ValueTask](#)

Method Flush

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Flush()

Blocks the current thread until all currently enqueued content is written to the underlying streams.

```
public void Flush()
```

Method FlushAsync

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

FlushAsync()

Asynchronously waits until all currently enqueued content is written to the underlying streams.

```
public Task FlushAsync()
```

Returns

[Task](#)

Method Peek

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Peek()

Reads the output buffer. To use this method, it's required to set this [LogStream](#) buffering with [StartBuffering\(int\)](#).

```
public string Peek()
```

Returns

[string](#)

Exceptions

[InvalidOperationException](#)

Thrown when this LogStream is not buffering.

Method StartBuffering

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

StartBuffering(int)

Start buffering all output to an alternate stream in memory for readability with [Peek\(\)](#) later.

```
public void StartBuffering(int lines)
```

Parameters

lines int ↗

The amount of lines to store in the buffer.

Method StopBuffering

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

StopBuffering()

Stops buffering output.

```
public void StopBuffering()
```

Method WriteException

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

WriteException(Exception)

Writes an exception description in the log.

```
public void WriteException(Exception exp)
```

Parameters

exp [Exception](#)

The exception which will be written.

WriteException(Exception, string?)

Writes an exception description in the log.

```
public void WriteException(Exception exp, string? extraContext = null)
```

Parameters

exp [Exception](#)

The exception which will be written.

extraContext [string](#)

Extra context message to append to the exception message.

Method WriteExceptionAsync

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

WriteExceptionAsync(Exception)

Writes an exception description in the log.

```
public Task WriteExceptionAsync(Exception exp)
```

Parameters

exp [Exception](#)

The exception which will be written.

Returns

[Task](#)

WriteExceptionAsync(Exception, string?)

Writes an exception description in the log.

```
public Task WriteExceptionAsync(Exception exp, string? extraContext = null)
```

Parameters

exp [Exception](#)

The exception which will be written.

extraContext [string](#)

Extra context message to append to the exception message.

Returns

Task ↗

Method WriteLine

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

WriteLine()

Writes an line-break at the end of the output.

```
public void WriteLine()
```

WriteLine(object?)

Writes the text and concats an line-break at the end into the output.

```
public void WriteLine(object? message)
```

Parameters

message object

The text that will be written in the output.

WriteLine(string)

Writes the text and concats an line-break at the end into the output.

```
public void WriteLine(string message)
```

Parameters

message string

The text that will be written in the output.

WriteLine(string, scoped ReadOnlySpan<object?>)

Writes the text format and arguments and concats an line-break at the end into the output.

```
public void WriteLine(string format, scoped ReadOnlySpan<object?> args)
```

Parameters

format string

The string format that represents the arguments positions.

args ReadOnlySpan<object>

An array of objects that represents the string format slots values.

WriteLine(string, IEnumerable<object?>)

Writes the text format and arguments and concats an line-break at the end into the output.

```
public void WriteLine(string format, IEnumerable<object?> args)
```

Parameters

format string

The string format that represents the arguments positions.

args IEnumerable<object>

An array of objects that represents the string format slots values.

WriteLine(IFormatProvider?, string, params object?[])

Writes the text format and arguments and appends a line-break at the end into the output, using the specified format provider.

```
public void WriteLine(IFormatProvider? formatProvider, string format, params object?[] args)
```

Parameters

formatProvider [IFormatProvider](#)

The format provider to use when formatting the string. If null, the current culture is used.

format [string](#)

The string format that represents the arguments positions.

args [object](#)[]

An array of objects that represents the string format slots values.

Method WriteLineAsync

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

WriteLineAsync()

Writes an line-break at the end of the output.

```
public Task WriteLineAsync()
```

Returns

[Task](#)

WriteLineAsync(object?)

Writes the text and concats an line-break at the end into the output.

```
public Task WriteLineAsync(object? message)
```

Parameters

message [object](#)

The text that will be written in the output.

Returns

[Task](#)

WriteLineAsync(string)

Writes the text and concats an line-break at the end into the output.

```
public Task WriteLineAsync(string message)
```

Parameters

message [string](#)

The text that will be written in the output.

Returns

[Task](#)

WriteLineAsync(string, IEnumerable<object?>)

Writes the text format and arguments and concats an line-break at the end into the output.

```
public Task WriteLineAsync(string format, IEnumerable<object?> args)
```

Parameters

format [string](#)

The string format that represents the arguments positions.

args [IEnumerable](#)<[object](#)>

An array of objects that represents the string format slots values.

Returns

[Task](#)

WriteLineAsync(IFormatProvider?, string, params object?[])

Writes the text format and arguments and appends a line-break at the end into the output, using the specified format provider.

```
public Task WriteLineAsync(IFormatProvider? formatProvider, string format, params object?[] args)
```

Parameters

formatProvider [IFormatProvider](#)

The format provider to use when formatting the string. If null, the current culture is used.

format [string](#)

The string format that represents the arguments positions.

args [object](#)[]

An array of objects that represents the string format slots values.

Returns

[Task](#)

Method WriteLineInternal

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

WriteLineInternal(string)

Represents the method that intercepts the line that will be written to an output log before being queued for writing. This method will block if the log queue is full.

```
protected virtual void WriteLineInternal(string line)
```

Parameters

line [string](#)

The line which will be written to the log stream.

Method WriteLineInternalAsync

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

WriteLineInternalAsync(string)

Represents the asynchronous method that intercepts the line that will be written to an output log before being queued for writing.

```
protected virtual ValueTask WriteLineInternalAsync(string line)
```

Parameters

line [string](#)

The line which will be written to the log stream.

Returns

[ValueTask](#)

A [ValueTask](#) that represents the asynchronous operation.

Method ~LogStream

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

`~LogStream()`

```
protected ~LogStream()
```

Class PrefixedLogStream

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Represents a log stream that prefixes log messages with a custom string.

```
public sealed class PrefixedLogStream : LogStream, IDisposable, IAsyncDisposable
```

Implements

[IDisposable](#), [IAsyncDisposable](#)

Constructors

[PrefixedLogStream\(Func<string>\)](#)

Initializes a new instance of the [PrefixedLogStream](#) class with the specified prefix function.

[PrefixedLogStream\(Func<string>, TextWriter\)](#)

Initializes a new instance of the [PrefixedLogStream](#) class with the specified prefix function and text writer.

[PrefixedLogStream\(Func<string>, string\)](#)

Initializes a new instance of the [PrefixedLogStream](#) class with the specified prefix function and file name.

[PrefixedLogStream\(Func<string>, string?, TextWriter?\)](#)

Initializes a new instance of the [PrefixedLogStream](#) class with the specified prefix function, file name, and text writer.

Properties

[PrefixFunction](#)

Gets or sets a function that returns the prefix to be added to log messages.

Methods

[WriteLineInternal\(string\)](#)

Represents the method that intercepts the line that will be written to an output log before being queued for writing. This method will block if the log queue is full.

[WriteLineInternalAsync\(string\)](#)

Represents the asynchronous method that intercepts the line that will be written to an output log before being queued for writing.

Constructor PrefixLogStream

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

PrefixLogStream(Func<string>)

Initializes a new instance of the [PrefixLogStream](#) class with the specified prefix function.

```
public PrefixLogStream(Func<string> prefixFunction)
```

Parameters

prefixFunction [Func<string>](#)

A function that returns the prefix to be added to log messages.

PrefixLogStream(Func<string>, TextWriter)

Initializes a new instance of the [PrefixLogStream](#) class with the specified prefix function and text writer.

```
public PrefixLogStream(Func<string> prefixFunction, TextWriter tw)
```

Parameters

prefixFunction [Func<string>](#)

A function that returns the prefix to be added to log messages.

tw [TextWriter](#)

The text writer to write log messages to.

PrefixLogStream(Func<string>, string)

Initializes a new instance of the [PrefixedLogStream](#) class with the specified prefix function and file name.

```
public PrefixedLogStream(Func<string> prefixFunction, string filename)
```

Parameters

prefixFunction [Func<string>](#)

A function that returns the prefix to be added to log messages.

filename [string](#)

The name of the file to write log messages to.

PrefixedLogStream(Func<string>, string?, TextWriter?)

Initializes a new instance of the [PrefixedLogStream](#) class with the specified prefix function, file name, and text writer.

```
public PrefixedLogStream(Func<string> prefixFunction, string? filename, TextWriter? tw)
```

Parameters

prefixFunction [Func<string>](#)

A function that returns the prefix to be added to log messages.

filename [string](#)

The name of the file to write log messages to, or `null` to write to the text writer.

tw [TextWriter](#)

The text writer to write log messages to, or `null` to write to the file.

Property PrefixFunction

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

PrefixFunction

Gets or sets a function that returns the prefix to be added to log messages.

```
public Func<string> PrefixFunction { get; set; }
```

Property Value

[Func<string>](#)

Method WriteLineInternal

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

WriteLineInternal(string)

Represents the method that intercepts the line that will be written to an output log before being queued for writing. This method will block if the log queue is full.

```
protected override void WriteLineInternal(string line)
```

Parameters

line [string](#)

The line which will be written to the log stream.

Method WriteLineInternalAsync

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

WriteLineInternalAsync(string)

Represents the asynchronous method that intercepts the line that will be written to an output log before being queued for writing.

```
protected override ValueTask WriteLineInternalAsync(string line)
```

Parameters

line [string](#)

The line which will be written to the log stream.

Returns

[ValueTask](#)

A [ValueTask](#) that represents the asynchronous operation.

Enum RequestListenAction

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Represents the HTTP server action when receiving an request.

```
public enum RequestListenAction
```

Fields

Accept = 1

The server must accept and route the request.

Drop = 2

The server must reject the request and close the connection with the client.

Class RotatingLogPolicy

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Provides a managed utility for rotating log files by their file size.

```
public sealed class RotatingLogPolicy : IDisposable
```

Implements

[IDisposable](#) ↗

Constructors

[RotatingLogPolicy\(LogStream\)](#)

Creates an new [RotatingLogPolicy](#) instance with the given [LogStream](#) object to watch.

Properties

[Due](#)

Gets the time interval between checks.

[MaximumSize](#)

Gets the file size threshold in bytes for when the file will be compressed and then cleared.

Methods

[Configure\(long, TimeSpan\)](#)

Defines the time interval and size threshold for starting the task, and then starts the task.

[Dispose\(\)](#)

Constructor RotatingLogPolicy

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

RotatingLogPolicy(LogStream)

Creates an new [RotatingLogPolicy](#) instance with the given [LogStream](#) object to watch.

```
public RotatingLogPolicy(LogStream ls)
```

Parameters

ls [LogStream](#)

Property Due

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Due

Gets the time interval between checks.

```
public TimeSpan Due { get; }
```

Property Value

[TimeSpan](#) ↗

Property MaximumSize

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

MaximumSize

Gets the file size threshold in bytes for when the file will be compressed and then cleared.

```
public long MaximumSize { get; }
```

Property Value

[long](#)

Method Configure

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Configure(long, TimeSpan)

Defines the time interval and size threshold for starting the task, and then starts the task.

```
public void Configure(long maximumSize, TimeSpan due)
```

Parameters

maximumSize [long](#)

The non-negative size threshold of the log file size in byte count.

due [TimeSpan](#)

The time interval between checks.

Remarks

The first run is performed immediately after calling this method.

Method Dispose

Namespace:[Sisk.Core.Http](#)

Assembly:Sisk.Core.dll

Dispose()

```
public void Dispose()
```

Namespace Sisk.Core.Routing

Classes

[AsyncRequestHandler](#)

Represents a class that implements [IRequestHandler](#) and its execution method is asynchronous.

[RegexRoute](#)

Represents an [Route](#) which it's implementation already enables [UseRegex](#).

[RegexRouteAttribute](#)

Represents a mapping to an route, which it's path is defined by an regular expression. This attribute is an shorthand from [RouteAttribute](#).

[RequestHandler](#)

Represents an abstract class which implements [IRequestHandler](#).

[RequestHandlerAttribute](#)

Specifies that the method or class, when used on this attribute, will instantiate the type and call the [IRequestHandler](#) with given parameters.

[RequestHandlerAttribute<T>](#)

Specifies that the method or class, when used on this attribute, will instantiate the type and call the [IRequestHandler](#) with given parameters.

[Route](#)

Represents an HTTP route to be matched by an [Router](#).

[RouteAttribute](#)

Represents an class that, when applied to a method, will be recognized by a router as a route.

[RouteDeleteAttribute](#)

Represents a mapping to an HTTP DELETE route. This attribute is an shorthand from [RouteAttribute](#).

[RouteGetAttribute](#)

Represents a mapping to an HTTP GET route. This attribute is an shorthand from [RouteAttribute](#).

[RouteMatch](#)

Represents the result of a route matching operation.

[RoutePatchAttribute](#)

Represents a mapping to an HTTP PATCH route. This attribute is an shorthand from [RouteAttribute](#).

[RoutePostAttribute](#)

Represents a mapping to an HTTP POST route. This attribute is an shorthand from [RouteAttribute](#).

[RoutePrefixAttribute](#)

Represents an attribute that, when applied to an class containing routes, all child routes will start with the specified prefix.

[RoutePutAttribute](#)

Represents a mapping to an HTTP PUT route. This attribute is an shorthand from [RouteAttribute](#).

[Router](#)

Represents a collection of [Route](#) and main executor of actions in the [HttpServer](#).

[RouterModule](#)

Indicates that extended class supports router modules, which allows the management of routes, request handlers and prefixes.

[ValueResult<T>](#)

Represents a mutable type for boxing objects by value or reference in a response from a router.

Interfaces

[IRequestHandler](#)

Represents an interface that is executed before a request.

Enums

[LogOutput](#)

Determines the way the server can write log messages. This enumerator is for giving permissions for certain contexts to be able or not to write to the server logs, such as [AccessLogsStream](#) and [ErrorsLogsStream](#).

[RequestHandlerExecutionMode](#)

Defines when the [IRequestHandler](#) should be executed.

[RouteMethod](#)

Represents an HTTP method to be matched in an [Route](#).

Delegates

[ExceptionErrorCallback](#)

Represents the function that is called after the route action threw an exception.

[ParameterlessRouteAction](#)

Represents the function that is called after the route is matched with the request.

[RouteAction](#)

Represents the function that is called after the route is matched with the request.

[RouterActionHandlerCallback<T>](#)

Represents the function that receives an object of the `T` and returns an [HttpResponse](#) response from the informed object.

[RoutingErrorCallback](#)

Represents the function that is called when an request reaches an error on the router.

Class AsyncRequestHandler

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents a class that implements [IRequestHandler](#) and its execution method is asynchronous.

```
public abstract class AsyncRequestHandler : IRequestHandler
```

Implements

[IRequestHandler](#)

Constructors

[AsyncRequestHandler\(\)](#)

Properties

[ExecutionMode](#)

Gets or sets when this [IRequestHandler](#) should run.

Methods

[Create\(Func<HttpRequest, HttpContext, Task<HttpResponse?>\), RequestHandlerExecutionMode\)](#)

Gets an inline [AsyncRequestHandler](#) that resolves to the specified function.

[ExecuteAsync\(HttpRequest, HttpContext\)](#)

This method is called by the [Router](#) before executing a request when the [Route](#) instantiates an object that implements this interface. If it returns a [HttpResponse](#) object, the route action is not called and all execution of the route is stopped. If it returns "null", the execution is continued.

[Next\(\)](#)

Returns an null [HttpResponse](#) reference, which points to the next request handler or route action.

Constructor AsyncRequestHandler

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

AsyncRequestHandler()

```
protected AsyncRequestHandler()
```

Property ExecutionMode

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

ExecutionMode

Gets or sets when this [IRequestHandler](#) should run.

```
public virtual RequestHandlerExecutionMode ExecutionMode { get; init; }
```

Property Value

[RequestHandlerExecutionMode](#)

Method Create

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Create(Func<HttpRequest, HttpContext, Task<HttpResponse?>>, RequestHandlerExecutionMode)

Gets an inline [AsyncRequestHandler](#) that resolves to the specified function.

```
public static AsyncRequestHandler Create(Func<HttpRequest, HttpContext, Task<HttpResponse?>>
execute, RequestHandlerExecutionMode executionMode = RequestHandlerExecutionMode.BeforeResponse)
```

Parameters

execute [Func<HttpRequest, HttpContext, Task<HttpResponse?>>](#)

The function that the [AsyncRequestHandler](#) will run.

executionMode [RequestHandlerExecutionMode](#)

Optional. Determines where the request handler will be executed.

Returns

[AsyncRequestHandler](#)

Method ExecuteAsync

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

ExecuteAsync([HttpRequest](#), [HttpContext](#))

This method is called by the [Router](#) before executing a request when the [Route](#) instantiates an object that implements this interface. If it returns a [HttpResponse](#) object, the route action is not called and all execution of the route is stopped. If it returns "null", the execution is continued.

```
public abstract Task<HttpResponse?> ExecuteAsync(HttpRequest request, HttpContext context)
```

Parameters

request [HttpRequest](#)

The entry HTTP request.

context [HttpContext](#)

The HTTP request context. It may contain information from other [IRequestHandler](#).

Returns

[Task](#) ↗<[HttpResponse](#)>

Method Next

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Next()

Returns an null [HttpResponse](#) reference, which points to the next request handler or route action.

```
public HttpResponse? Next()
```

Returns

[HttpResponse](#)

Delegate ExceptionErrorCallback

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents the function that is called after the route action threw an exception.

```
public delegate HttpResponse ExceptionErrorCallback(Exception ex, HttpContext context)
```

Parameters

ex [Exception](#)

Represents the function that is called after the route action threw an exception.

context [HttpContext](#)

Represents the function that is called after the route action threw an exception.

Returns

[HttpResponse](#)

Represents the function that is called after the route action threw an exception.

Constructors

[ExceptionErrorCallback\(object, nint\)](#)

Methods

[BeginInvoke\(Exception, HttpContext, AsyncCallback, object\)](#)

[EndInvoke\(IAsyncResult\)](#)

[Invoke\(Exception, HttpContext\)](#)

Constructor ExceptionErrorCallback

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

ExceptionErrorCallback(object, nint)

```
public ExceptionErrorCallback(object @object, nint method)
```

Parameters

object object ↗

method nint ↗

Method BeginInvoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

BeginInvoke(Exception, HttpContext, AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(Exception ex, HttpContext context, AsyncCallback  
callback, object @object)
```

Parameters

ex [Exception](#)

context [HttpContext](#)

callback [AsyncCallback](#)

object [object](#)

Returns

[IAsyncResult](#)

Method EndInvoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

EndInvoke(IAsyncResult)

```
public virtual HttpResponse EndInvoke(IAsyncResult result)
```

Parameters

result [IAsyncResult](#) ↗

Returns

[HttpResponse](#)

Method Invoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Invoke(Exception, HttpContext)

```
public virtual HttpResponse Invoke(Exception ex, HttpContext context)
```

Parameters

ex [Exception](#)

context [HttpContext](#)

Returns

[HttpResponse](#)

Interface IRequestHandler

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents an interface that is executed before a request.

```
public interface IRequestHandler
```

Properties

[ExecutionMode](#)

Gets or sets when this [IRequestHandler](#) should run.

Methods

[Execute\(HttpContext\)](#)

This method is called by the [Router](#) before executing a request when the [Route](#) instantiates an object that implements this interface. If it returns a [HttpResponse](#) object, the route action is not called and all execution of the route is stopped. If it returns "null", the execution is continued.

Property ExecutionMode

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

ExecutionMode

Gets or sets when this [IRequestHandler](#) should run.

```
RequestHandlerExecutionMode ExecutionMode { get; init; }
```

Property Value

[RequestHandlerExecutionMode](#)

Method Execute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Execute([HttpRequest](#), [HttpContext](#))

This method is called by the [Router](#) before executing a request when the [Route](#) instantiates an object that implements this interface. If it returns a [HttpResponse](#) object, the route action is not called and all execution of the route is stopped. If it returns "null", the execution is continued.

```
HttpResponse? Execute(HttpRequest request, HttpContext context)
```

Parameters

request [HttpRequest](#)

The entry HTTP request.

context [HttpContext](#)

The HTTP request context. It may contain information from other [IRequestHandler](#).

Returns

[HttpResponse](#)

Enum LogOutput

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Determines the way the server can write log messages. This enumerator is for giving permissions for certain contexts to be able or not to write to the server logs, such as [AccessLogsStream](#) and [ErrorsLogsStream](#).

[Flags]

```
public enum LogOutput
```

Fields

AccessLog = 2

Determines that the context or the route can write log messages only to the access logs through [AccessLogs Stream](#).

Both = AccessLog | ErrorLog

Determines that the context or the route can write log messages to both error and access logs.

ErrorLog = 4

Determines that the context or the route can write error messages only to the error logs through [ErrorsLogs Stream](#).

None = 0

Determines that the context or the route cannot write any log messages.

Delegate ParameterlessRouteAction

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents the function that is called after the route is matched with the request.

```
public delegate object ParameterlessRouteAction()
```

Returns

[object](#)

Represents the function that is called after the route is matched with the request.

Constructors

[ParameterlessRouteAction\(object, nint\)](#)

Methods

[BeginInvoke\(AsyncCallback, object\)](#)

[EndInvoke\(IAsyncResult\)](#)

[Invoke\(\)](#)

Constructor ParameterlessRouteAction

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

ParameterlessRouteAction(object, nint)

```
public ParameterlessRouteAction(object @object, nint method)
```

Parameters

object object

method nint

Method BeginInvoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

BeginInvoke(AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(AsyncCallback callback, object @object)
```

Parameters

callback [AsyncCallback](#)

object [object](#)

Returns

[IAsyncResult](#)

Method EndInvoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

EndInvoke(IAsyncResult)

```
public virtual object EndInvoke(IAsyncResult result)
```

Parameters

result [IAsyncResult](#)

Returns

[object](#)

Method Invoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Invoke()

```
public virtual object Invoke()
```

Returns

[object](#)

Class RegexRoute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents an [Route](#) which it's implementation already enables [UseRegex](#).

```
public sealed class RegexRoute : Route
```

Constructors

[RegexRoute\(RouteMethod, string, RouteAction\)](#)

Initializes a new instance of the [RegexRoute](#) class.

[RegexRoute\(RouteMethod, string, string?, RouteAction, IRequestHandler\[\]?\)](#)

Initializes a new instance of the [RegexRoute](#) class.

Constructor RegexRoute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RegexRoute(RouteMethod, string, RouteAction)

Initializes a new instance of the [RegexRoute](#) class.

```
public RegexRoute(RouteMethod method, string pattern, RouteAction action)
```

Parameters

method [RouteMethod](#)

The HTTP method for this route.

pattern [string](#)

The regular expression pattern for this route.

action [RouteAction](#)

The action to be executed when this route is matched.

RegexRoute(RouteMethod, string, string?, RouteAction, IRequestHandler[]?)

Initializes a new instance of the [RegexRoute](#) class.

```
public RegexRoute(RouteMethod method, string pattern, string? name, RouteAction action,
IRequestHandler[]? beforeCallback)
```

Parameters

method [RouteMethod](#)

The HTTP method for this route.

pattern [string](#)

The regular expression pattern for this route.

name [string](#)

The name of this route.

action [RouteAction](#)

The action to be executed when this route is matched.

beforeCallback [IRequestHandler\[\]](#)

The callback to be executed before the action.

Class RegexRouteAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents a mapping to an route, which it's path is defined by an regular expression. This attribute is an shorthand from [RouteAttribute](#).

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = true, Inherited = true)]
public sealed class RegexRouteAttribute : RouteAttribute
```

Constructors

[RegexRouteAttribute\(RouteMethod, string\)](#)

Creates an new [RouteGetAttribute](#) attribute instance with given path.

Constructor RegexRouteAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RegexRouteAttribute(RouteMethod, string)

Creates an new [RouteGetAttribute](#) attribute instance with given path.

```
public RegexRouteAttribute(RouteMethod method, string pattern)
```

Parameters

method [RouteMethod](#)

The route entry point method.

pattern [string](#)

The Regex pattern which will match the route.

Class RequestHandler

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents an abstract class which implements [IRequestHandler](#).

```
public abstract class RequestHandler : IRequestHandler
```

Implements

[IRequestHandler](#)

Constructors

[RequestHandler\(\)](#)

Methods

[Create\(Func<HttpRequest, HttpContext, HttpResponse?>, RequestHandlerExecutionMode\)](#)

Gets an inline [RequestHandler](#) that resolves to the specified function.

[Next\(\)](#)

Returns an null [HttpResponse](#) reference, which points to the next request handler or route action.

Constructor RequestHandler

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RequestHandler()

```
protected RequestHandler()
```

Method Create

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Create(Func<HttpRequest, HttpContext, HttpResponseMessage?>, RequestHandlerExecutionMode)

Gets an inline [RequestHandler](#) that resolves to the specified function.

```
public static RequestHandler Create(Func<HttpRequest, HttpContext, HttpResponseMessage?> execute,  
RequestHandlerExecutionMode executionMode = RequestHandlerExecutionMode.BeforeResponse)
```

Parameters

execute [Func<HttpRequest, HttpContext, HttpResponseMessage>](#)

The function that the [RequestHandler](#) will run.

executionMode [RequestHandlerExecutionMode](#)

Optional. Determines where the request handler will be executed.

Returns

[RequestHandler](#)

Method Next

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Next()

Returns an null [HttpResponse](#) reference, which points to the next request handler or route action.

```
public HttpResponse? Next()
```

Returns

[HttpResponse](#)

Class RequestHandlerAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Specifies that the method or class, when used on this attribute, will instantiate the type and call the [IRequestHandler](#) with given parameters.

```
[AttributeUsage(AttributeTargets.Class|AttributeTargets.Method, AllowMultiple = true)]
public class RequestHandlerAttribute : Attribute
```

Constructors

[RequestHandlerAttribute\(Type\)](#)

Creates a new instance of this attribute with the informed parameters.

Properties

[ConstructorArguments](#)

Specifies parameters for the given type's constructor.

[RequestHandlerType](#)

Gets or sets the type that implements [IRequestHandler](#) which will be instantiated.

Constructor RequestHandlerAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RequestHandlerAttribute(Type)

Creates a new instance of this attribute with the informed parameters.

```
public RequestHandlerAttribute(Type handledBy)
```

Parameters

handledBy [Type](#)

The type that implements [IRequestHandler](#) which will be instantiated.

Property ConstructorArguments

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

ConstructorArguments

Specifies parameters for the given type's constructor.

```
public object?[] ConstructorArguments { get; set; }
```

Property Value

[object](#) []

Property RequestHandlerType

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RequestHandlerType

Gets or sets the type that implements [IRequestHandler](#) which will be instantiated.

```
public Type RequestHandlerType { get; set; }
```

Property Value

Type[!\[\]\(d0ad113d5b11063a350a67b43ee25999_img.jpg\)](#)

Class RequestHandlerAttribute<T>

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Specifies that the method or class, when used on this attribute, will instantiate the type and call the [IRequestHandler](#) with given parameters.

```
public class RequestHandlerAttribute<T> : RequestHandlerAttribute where T : IRequestHandler
```

Type Parameters

T

Constructors

[RequestHandlerAttribute\(\)](#)

Creates an new instance of this [RequestHandlerAttribute<T>](#) class.

[RequestHandlerAttribute\(params object?\[\]\)](#)

Creates an new instance of this [RequestHandlerAttribute<T>](#) class with the specified constructor arguments for T .

Constructor RequestHandlerAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RequestHandlerAttribute()

Creates an new instance of this [RequestHandlerAttribute<T>](#) class.

```
public RequestHandlerAttribute()
```

RequestHandlerAttribute(params object?[])

Creates an new instance of this [RequestHandlerAttribute<T>](#) class with the specified constructor arguments for T .

```
public RequestHandlerAttribute(params object?[] constructorArguments)
```

Parameters

constructorArguments [object](#)[]

An optional array of objects which is passed to the request handler constructor.

Enum RequestHandlerExecutionMode

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Defines when the [IRequestHandler](#) should be executed.

```
[Flags]  
public enum RequestHandlerExecutionMode
```

Fields

AfterResponse = 4

Indicates that the request handler should be executed after the route action.

BeforeResponse = 2

Indicates that the request handler should be executed before the route action.

Both = BeforeResponse | AfterResponse

Indicates that the request handler should be executed before and after the route action.

Class Route

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents an HTTP route to be matched by an [Router](#).

```
public class Route
```

Constructors

[Route\(\)](#)

Creates an new [Route](#) instance with no parameters.

[Route\(RouteMethod, string, Delegate?\)](#)

Creates an new [Route](#) instance with given parameters.

[Route\(RouteMethod, string, string?, Delegate?, IRequestHandler\[\]?\)](#)

Creates an new [Route](#) instance with given parameters.

Fields

[AnyPath](#)

Represents an route path which captures any URL path.

Properties

[Action](#)

Gets or sets the function that is called after the route is matched with the request.

[Bag](#)

Gets or sets an [TypedValueDictionary](#) for this route, which can hold contextual variables for this [Route](#) object.

[BypassGlobalRequestHandlers](#)

Gets or sets the global request handlers instances that will not run on this route.

[IsAsync](#)

Gets an boolean indicating if this [Route](#) action return is an asynchronous [Task](#).

[LogMode](#)

Gets or sets how this route can write messages to log files on the server.

[Method](#)

Gets or sets the matching HTTP method.

[Name](#)

Gets or sets the route name.

[Path](#)

Gets or sets the path expression that will be interpreted by the router and validated by the requests.

[RequestHandlers](#)

Gets or sets the request handlers instances to run before the route action.

[UseCors](#)

Gets or sets whether this route should send Cross-Origin Resource Sharing headers in the response.

[UseRegex](#)

Get or sets if this route should use regex to be interpreted instead of predefined templates.

Methods

[Any\(string, RouteAction\)](#)

Creates a route that responds to any HTTP request method.

[Any\(string, Delegate?\)](#)

Creates a route that responds to any HTTP request method.

[Delete\(string, RouteAction\)](#)

Creates a route that responds to HTTP DELETE requests.

[Delete\(string, Delegate?\)](#)

Creates a route that responds to HTTP DELETE requests.

[Get\(string, RouteAction\)](#)

Creates a route that responds to HTTP GET requests.

[Get\(string, Delegate?\)](#)

Creates a route that responds to HTTP GET requests.

[Head\(string, RouteAction\)](#)

Creates a route that responds to HTTP HEAD requests.

[Head\(string, Delegate?\)](#)

Creates a route that responds to HTTP HEAD requests.

[Options\(string, RouteAction\)](#)

Creates a route that responds to HTTP OPTIONS requests.

[Options\(string, Delegate?\)](#)

Creates a route that responds to HTTP OPTIONS requests.

[Patch\(string, RouteAction\)](#)

Creates a route that responds to HTTP PATCH requests.

[Patch\(string, Delegate?\)](#)

Creates a route that responds to HTTP PATCH requests.

[Post\(string, RouteAction\)](#)

Creates a route that responds to HTTP POST requests.

[Post\(string, Delegate?\)](#)

Creates a route that responds to HTTP POST requests.

[Put\(string, RouteAction\)](#)

Creates a route that responds to HTTP PUT requests.

[Put\(string, Delegate?\)](#)

Creates a route that responds to HTTP PUT requests.

[ToString\(\)](#)

Gets an string notation for this [Route](#) object.

Constructor Route

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Route(RouteMethod, string, Delegate?)

Creates an new [Route](#) instance with given parameters.

```
public Route(RouteMethod method, string path, Delegate? action)
```

Parameters

method [RouteMethod](#)

The matching HTTP method. If it is "Any", the route will just use the path expression to be matched, not the HTTP method.

path [string](#)

The path expression that will be interpreted by the router and validated by the requests.

action [Delegate](#)

The function that is called after the route is matched with the request.

Route(RouteMethod, string, string?, Delegate?, IRequestHandler[]?)

Creates an new [Route](#) instance with given parameters.

```
public Route(RouteMethod method, string path, string? name, Delegate? action,
IRequestHandler[]? handlers)
```

Parameters

method [RouteMethod](#)

The matching HTTP method. If it is "Any", the route will just use the path expression to be matched, not the HTTP method.

path [string](#)

The path expression that will be interpreted by the router and validated by the requests.

name [string](#)

The route name. It allows it to be found by other routes and makes it easier to create links.

action [Delegate](#)

The function that is called after the route is matched with the request.

handlers [IRequestHandler\[\]](#)

The RequestHandlers to run before the route's Action.

Route()

Creates an new [Route](#) instance with no parameters.

```
public Route()
```

Field AnyPath

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents an route path which captures any URL path.

```
public const string AnyPath = "/<>ANY>"
```

Returns

[string](#)

Represents an route path which captures any URL path.

Property Action

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Action

Gets or sets the function that is called after the route is matched with the request.

```
public Delegate? Action { get; set; }
```

Property Value

[Delegate](#)↗

Property Bag

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Bag

Gets or sets an [TypedValueDictionary](#) for this route, which can hold contextual variables for this [Route](#) object.

```
public TypedValueDictionary Bag { get; set; }
```

Property Value

[TypedValueDictionary](#)

Property BypassGlobalRequestHandlers

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

BypassGlobalRequestHandlers

Gets or sets the global request handlers instances that will not run on this route.

```
public IRequestHandler[] BypassGlobalRequestHandlers { get; set; }
```

Property Value

[IRequestHandler\[\]](#)

Property IsAsync

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

IsAsync

Gets an boolean indicating if this [Route](#) action return is an asynchronous [Task](#).

```
public bool IsAsync { get; }
```

Property Value

[bool](#)

Property LogMode

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

LogMode

Gets or sets how this route can write messages to log files on the server.

```
public LogOutput LogMode { get; set; }
```

Property Value

[LogOutput](#)

Property Method

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Method

Gets or sets the matching HTTP method.

```
public RouteMethod Method { get; set; }
```

Property Value

[RouteMethod](#)

Property Name

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Name

Gets or sets the route name.

```
public string? Name { get; set; }
```

Property Value

[string](#)

Property Path

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Path

Gets or sets the path expression that will be interpreted by the router and validated by the requests.

```
public string Path { get; set; }
```

Property Value

[string](#)

Property RequestHandlers

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RequestHandlers

Gets or sets the request handlers instances to run before the route action.

```
public IRequestHandler[] RequestHandlers { get; set; }
```

Property Value

[IRequestHandler\[\]](#)

Property UseCors

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

UseCors

Gets or sets whether this route should send Cross-Origin Resource Sharing headers in the response.

```
public bool UseCors { get; set; }
```

Property Value

[bool](#)

Property UseRegex

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

UseRegex

Get or sets if this route should use regex to be interpreted instead of predefined templates.

```
public bool UseRegex { get; set; }
```

Property Value

[bool](#)

Method Any

Namespace:[Sisk.Core.Routing](#)

Assembly:[Sisk.Core.dll](#)

Any(string, Delegate?)

Creates a route that responds to any HTTP request method.

```
public static Route Any(string path, Delegate? action)
```

Parameters

path [string](#)

The URL path for the route.

action [Delegate](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for any request method.

Any(string, RouteAction)

Creates a route that responds to any HTTP request method.

```
public static Route Any(string path, RouteAction action)
```

Parameters

path [string](#)

The URL path for the route.

action [RouteAction](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for any request method.

Method Delete

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Delete(string, Delegate?)

Creates a route that responds to HTTP DELETE requests.

```
public static Route Delete(string path, Delegate? action)
```

Parameters

path [string](#)

The URL path for the route.

action [Delegate](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for DELETE requests.

Delete(string, RouteAction)

Creates a route that responds to HTTP DELETE requests.

```
public static Route Delete(string path, RouteAction action)
```

Parameters

path [string](#)

The URL path for the route.

action [RouteAction](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for DELETE requests.

Method Get

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Get(string, Delegate?)

Creates a route that responds to HTTP GET requests.

```
public static Route Get(string path, Delegate? action)
```

Parameters

path [string](#)

The URL path for the route.

action [Delegate](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for GET requests.

Get(string, RouteAction)

Creates a route that responds to HTTP GET requests.

```
public static Route Get(string path, RouteAction action)
```

Parameters

path [string](#)

The URL path for the route.

action [RouteAction](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for GET requests.

Method Head

Namespace:[Sisk.Core.Routing](#)

Assembly:[Sisk.Core.dll](#)

Head(string, Delegate?)

Creates a route that responds to HTTP HEAD requests.

```
public static Route Head(string path, Delegate? action)
```

Parameters

path [string](#)

The URL path for the route.

action [Delegate](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for HEAD requests.

Head(string, RouteAction)

Creates a route that responds to HTTP HEAD requests.

```
public static Route Head(string path, RouteAction action)
```

Parameters

path [string](#)

The URL path for the route.

action [RouteAction](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for HEAD requests.

Method Options

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Options(string, Delegate?)

Creates a route that responds to HTTP OPTIONS requests.

```
public static Route Options(string path, Delegate? action)
```

Parameters

path [string](#)

The URL path for the route.

action [Delegate](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for OPTIONS requests.

Options(string, RouteAction)

Creates a route that responds to HTTP OPTIONS requests.

```
public static Route Options(string path, RouteAction action)
```

Parameters

path [string](#)

The URL path for the route.

action [RouteAction](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for OPTIONS requests.

Method Patch

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Patch(string, Delegate?)

Creates a route that responds to HTTP PATCH requests.

```
public static Route Patch(string path, Delegate? action)
```

Parameters

path [string](#)

The URL path for the route.

action [Delegate](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for PATCH requests.

Patch(string, RouteAction)

Creates a route that responds to HTTP PATCH requests.

```
public static Route Patch(string path, RouteAction action)
```

Parameters

path [string](#)

The URL path for the route.

action [RouteAction](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for PATCH requests.

Method Post

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Post(string, Delegate?)

Creates a route that responds to HTTP POST requests.

```
public static Route Post(string path, Delegate? action)
```

Parameters

path [string](#)

The URL path for the route.

action [Delegate](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for POST requests.

Post(string, RouteAction)

Creates a route that responds to HTTP POST requests.

```
public static Route Post(string path, RouteAction action)
```

Parameters

path [string](#)

The URL path for the route.

action [RouteAction](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for POST requests.

Method Put

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Put(string, Delegate?)

Creates a route that responds to HTTP PUT requests.

```
public static Route Put(string path, Delegate? action)
```

Parameters

path [string](#)

The URL path for the route.

action [Delegate](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for PUT requests.

Put(string, RouteAction)

Creates a route that responds to HTTP PUT requests.

```
public static Route Put(string path, RouteAction action)
```

Parameters

path [string](#)

The URL path for the route.

action [RouteAction](#)

The action to be executed when the route is matched.

Returns

[Route](#)

A [Route](#) object configured for PUT requests.

Method ToString

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

ToString()

Gets an string notation for this [Route](#) object.

```
public override string ToString()
```

Returns

[string](#) ↗

Delegate RouteAction

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents the function that is called after the route is matched with the request.

```
public delegate object RouteAction(HttpRequest request)
```

Parameters

request [HttpRequest](#)

The received request on the router.

Returns

[object](#) ↗

Represents the function that is called after the route is matched with the request.

Constructors

[RouteAction\(object, nint\)](#)

Methods

[BeginInvoke\(HttpRequest, AsyncCallback, object\)](#)

[EndInvoke\(IAsyncResult\)](#)

[Invoke\(HttpRequest\)](#)

Constructor RouteAction

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RouteAction(object, nint)

```
public RouteAction(object @object, nint method)
```

Parameters

object object ↗

method nint ↗

Method BeginInvoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

BeginInvoke(HttpRequest, AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(HttpContext request, AsyncCallback callback,
object @object)
```

Parameters

request [HttpRequest](#)

callback [AsyncCallback](#)

object [object](#)

Returns

[IAsyncResult](#)

Method EndInvoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

EndInvoke(IAsyncResult)

```
public virtual object EndInvoke(IAsyncResult result)
```

Parameters

result [IAsyncResult](#)

Returns

[object](#)

Method Invoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Invoke(HttpRequest)

```
public virtual object Invoke(HttpContext request)
```

Parameters

request [HttpRequest](#)

Returns

[object](#)

Class RouteAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents an class that, when applied to a method, will be recognized by a router as a route.

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = true)]
public class RouteAttribute : Attribute
```

Constructors

[RouteAttribute\(RouteMethod, string\)](#)

Creates an new [RouteAttribute](#) instance with given route method and path pattern.

Properties

[LogMode](#)

Gets or sets how this route can write messages to log files on the server.

[Method](#)

Gets or sets the matching HTTP method. If it is "Any", the route will just use the path expression to be matched, not the HTTP method.

[Name](#)

Gets or sets the route name. It allows it to be found by other routes and makes it easier to create links.

[Path](#)

Gets or sets the path expression that will be interpreted by the router and validated by the requests.

[UseCors](#)

Gets or sets whether this route should send Cross-Origin Resource Sharing headers in the response.

[UseRegex](#)

Get or sets if this route should use regex to be interpreted instead of predefined templates.

Constructor RouteAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RouteAttribute(RouteMethod, string)

Creates an new [RouteAttribute](#) instance with given route method and path pattern.

```
public RouteAttribute(RouteMethod method, string path)
```

Parameters

method [RouteMethod](#)

The route entry point method.

path [string](#) ↗

The route path.

Property LogMode

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

LogMode

Gets or sets how this route can write messages to log files on the server.

```
public LogOutput LogMode { get; set; }
```

Property Value

[LogOutput](#)

Property Method

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Method

Gets or sets the matching HTTP method. If it is "Any", the route will just use the path expression to be matched, not the HTTP method.

```
public RouteMethod Method { get; set; }
```

Property Value

[RouteMethod](#)

Property Name

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Name

Gets or sets the route name. It allows it to be found by other routes and makes it easier to create links.

```
public string? Name { get; set; }
```

Property Value

[string](#)

Property Path

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Path

Gets or sets the path expression that will be interpreted by the router and validated by the requests.

```
public string Path { get; set; }
```

Property Value

[string](#)

Property UseCors

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

UseCors

Gets or sets whether this route should send Cross-Origin Resource Sharing headers in the response.

```
public bool UseCors { get; set; }
```

Property Value

[bool](#)

Property UseRegex

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

UseRegex

Get or sets if this route should use regex to be interpreted instead of predefined templates.

```
public bool UseRegex { get; set; }
```

Property Value

[bool](#)

Class RouteDeleteAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents a mapping to an HTTP DELETE route. This attribute is an shorthand from [RouteAttribute](#).

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = true, Inherited = true)]
public sealed class RouteDeleteAttribute : RouteAttribute
```

Constructors

[RouteDeleteAttribute\(\)](#)

Creates an new [RouteDeleteAttribute](#) attribute instance with an root path (/).

[RouteDeleteAttribute\(string\)](#)

Creates an new [RouteDeleteAttribute](#) attribute instance with given path.

Constructor RouteDeleteAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RouteDeleteAttribute(string)

Creates an new [RouteDeleteAttribute](#) attribute instance with given path.

```
public RouteDeleteAttribute(string path)
```

Parameters

path string ↗

The DELETE route path.

RouteDeleteAttribute()

Creates an new [RouteDeleteAttribute](#) attribute instance with an root path (/).

```
public RouteDeleteAttribute()
```

Class RouteGetAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents a mapping to an HTTP GET route. This attribute is an shorthand from [RouteAttribute](#).

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = true, Inherited = true)]
public sealed class RouteGetAttribute : RouteAttribute
```

Constructors

[RouteGetAttribute\(\)](#)

Creates an new [RouteGetAttribute](#) attribute instance with an root path (/).

[RouteGetAttribute\(string\)](#)

Creates an new [RouteGetAttribute](#) attribute instance with given path.

Constructor RouteGetAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RouteGetAttribute(string)

Creates an new [RouteGetAttribute](#) attribute instance with given path.

```
public RouteGetAttribute(string path)
```

Parameters

path string ↗

The GET route path.

RouteGetAttribute()

Creates an new [RouteGetAttribute](#) attribute instance with an root path (/).

```
public RouteGetAttribute()
```

Class RouteMatch

Namespace:[Sisk.Core.Routing](#)

Assembly:[Sisk.Core.dll](#)

Represents the result of a route matching operation.

```
public sealed class RouteMatch
```

Properties

[Parameters](#)

Gets a collection of parameters extracted from the route, or [null](#) if the route matching operation was not successful.

[Success](#)

Gets a value indicating whether the route matching operation was successful.

Property Parameters

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Parameters

Gets a collection of parameters extracted from the route, or [null](#) if the route matching operation was not successful.

```
public NameValueCollection? Parameters { get; }
```

Property Value

[NameValueCollection](#)

Property Success

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Success

Gets a value indicating whether the route matching operation was successful.

```
public bool Success { get; }
```

Property Value

[bool](#)

Enum RouteMethod

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents an HTTP method to be matched in an [Route](#).

[Flags]

```
public enum RouteMethod
```

Fields

Any = Get | Post | Put | Patch | Delete | Head | Options

Represents any HTTP method.

Delete = 32

Represents the HTTP DELETE method.

Get = 2

Represents the HTTP GET method.

Head = 128

Represents the HTTP HEAD method.

Options = 256

Represents the HTTP OPTIONS method.

Patch = 16

Represents the HTTP PATCH method.

Post = 4

Represents the HTTP POST method.

Put = 8

Represents the HTTP PUT method.

Class RoutePatchAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents a mapping to an HTTP PATCH route. This attribute is an shorthand from [RouteAttribute](#).

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = true, Inherited = true)]
public sealed class RoutePatchAttribute : RouteAttribute
```

Constructors

[RoutePatchAttribute\(\)](#)

Creates an new [RoutePatchAttribute](#) attribute instance with an root path (/).

[RoutePatchAttribute\(string\)](#)

Creates an new [RoutePatchAttribute](#) attribute instance with given path.

Constructor RoutePatchAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RoutePatchAttribute(string)

Creates an new [RoutePatchAttribute](#) attribute instance with given path.

```
public RoutePatchAttribute(string path)
```

Parameters

path string ↗

The PATCH route path.

RoutePatchAttribute()

Creates an new [RoutePatchAttribute](#) attribute instance with an root path (/).

```
public RoutePatchAttribute()
```

Class RoutePostAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents a mapping to an HTTP POST route. This attribute is an shorthand from [RouteAttribute](#).

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = true, Inherited = true)]
public sealed class RoutePostAttribute : RouteAttribute
```

Constructors

[RoutePostAttribute\(\)](#)

Creates an new [RoutePostAttribute](#) attribute instance with an root path (/).

[RoutePostAttribute\(string\)](#)

Creates an new [RoutePostAttribute](#) attribute instance with given path.

Constructor RoutePostAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RoutePostAttribute(string)

Creates an new [RoutePostAttribute](#) attribute instance with given path.

```
public RoutePostAttribute(string path)
```

Parameters

path string ↗

The POST route path.

RoutePostAttribute()

Creates an new [RoutePostAttribute](#) attribute instance with an root path (/).

```
public RoutePostAttribute()
```

Class RoutePrefixAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents an attribute that, when applied to an class containing routes, all child routes will start with the specified prefix.

```
[AttributeUsage(AttributeTargets.Class, AllowMultiple = false)]
public sealed class RoutePrefixAttribute : Attribute
```

Constructors

[RoutePrefixAttribute\(string\)](#)

Initializes an new [RoutePrefixAttribute](#) with given prefix.

Properties

[Prefix](#)

Gets or sets the route prefix.

Constructor RoutePrefixAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RoutePrefixAttribute(string)

Initializes an new [RoutePrefixAttribute](#) with given prefix.

```
public RoutePrefixAttribute(string prefix)
```

Parameters

prefix string ↗

Property Prefix

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Prefix

Gets or sets the route prefix.

```
public string Prefix { get; set; }
```

Property Value

[string](#)

Class RoutePutAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents a mapping to an HTTP PUT route. This attribute is an shorthand from [RouteAttribute](#).

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = true, Inherited = true)]
public sealed class RoutePutAttribute : RouteAttribute
```

Constructors

[RoutePutAttribute\(\)](#)

Creates an new [RoutePutAttribute](#) attribute instance with an root path (/).

[RoutePutAttribute\(string\)](#)

Creates an new [RoutePutAttribute](#) attribute instance with given path.

Constructor RoutePutAttribute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RoutePutAttribute(string)

Creates an new [RoutePutAttribute](#) attribute instance with given path.

```
public RoutePutAttribute(string path)
```

Parameters

path string ↗

The PUT route path.

RoutePutAttribute()

Creates an new [RoutePutAttribute](#) attribute instance with an root path (/).

```
public RoutePutAttribute()
```

Class Router

Namespace:[Sisk.Core.Routing](#)

Assembly:[Sisk.Core.dll](#)

Represents a collection of [Route](#) and main executor of actions in the [HttpServer](#).

```
public sealed class Router : IReadOnlyCollection<Route>, IEnumerable<Route>, IEnumerable
```

Implements

[IReadOnlyCollection](#)<Route>, [IEnumerable](#)<Route>, [IEnumerable](#)

Constructors

[Router\(\)](#)

Creates an new [Router](#) instance with default values.

[Router\(IEnumerable<Route>\)](#)

Creates an new [Router](#) instance with given route collection.

Properties

[CallbackErrorHandler](#)

Gets or sets the Router action exception handler. The response handler for this property will send an HTTP response to the client when an exception is caught during execution. This property is only called when [Throw Exceptions](#) is disabled.

[CheckForRouteCollisions](#)

Gets or sets whether this [Router](#) should check for possible routing collisions before starting the HTTP server.

[Count](#)

[GlobalRequestHandlers](#)

Gets or sets the global requests handlers that will be executed in all matched routes.

[IsReadOnly](#)

Gets an boolean indicating where this [Router](#) is read-only or not.

[MatchRoutesIgnoreCase](#)

Gets or sets whether this [Router](#) will match routes ignoring case.

[MethodNotAllowedErrorHandler](#)

Gets or sets the Router "405 Method Not Allowed" handler.

[NotFoundErrorHandler](#)

Gets or sets the Router "404 Not Found" handler.

[Prefix](#)

Gets or sets the prefix which will be applied to all next defining routes in this router.

Methods

[AutoScanModules\(Type, Assembly\)](#)

Scans for all types that implements the specified module type and associates an instance of each type to the router.

[AutoScanModules<TModule>\(\)](#)

Scans for all types that implements `TModule` and associates an instance of each type to the router. Note that, `TModule` must be an [RouterModule](#) type and an accessible constructor for each type must be present.

[AutoScanModules<TModule>\(Assembly\)](#)

Scans for all types that implements `TModule` and associates an instance of each type to the router. Note that, `TModule` must be an [RouterModule](#) type and an accessible constructor for each type must be present.

[Combine\(params string\[\]\)](#)

Combines an array of string parts into a single path.

[GetDefinedRoutes\(\)](#)

Gets all routes defined on this router instance.

[GetEnumerator\(\)](#)

[GetRouteFromName\(string\)](#)

Gets an defined [Route](#) by their name property.

[GetRouteFromPath\(RouteMethod, string\)](#)

Gets the first matched [Route](#) by their HTTP method and path.

[GetRouteFromPath\(string\)](#)

Gets the first matched [Route](#) by their URL path.

[IsDefined\(RouteMethod, string\)](#)

Gets an boolean indicating if there are any route that matches the specified method and route path.

[IsRouteExpressionsOverlap\(in ReadOnlySpan<char>, in ReadOnlySpan<char>, StringComparison\)](#)

Determines whether two route expressions overlap.

[IsRouteExpressionsOverlap\(string, string, StringComparison\)](#)

Determines whether two route expressions overlap.

[MapAny\(string, RouteAction\)](#)

Maps an route which matches any HTTP method, using the specified path and action function.

[MapAny\(string, Delegate\)](#)

Maps an route which matches any HTTP method, using the specified path and action function.

[MapDelete\(string, RouteAction\)](#)

Maps an DELETE route using the specified path and action function.

[MapDelete\(string, Delegate\)](#)

Maps an DELETE route using the specified path and action function.

[MapGet\(string, RouteAction\)](#)

Maps an GET route using the specified path and action function.

[MapGet\(string, Delegate\)](#)

Maps an GET route using the specified path and action function.

[MapHead\(string, RouteAction\)](#)

Maps an HEAD route using the specified path and action function.

[MapHead\(string, Delegate\)](#)

Maps an HEAD route using the specified path and action function.

[MapOptions\(string, RouteAction\)](#)

Maps an OPTIONS route using the specified path and action function.

[MapOptions\(string, Delegate\)](#)

Maps an OPTIONS route using the specified path and action function.

[MapPatch\(string, RouteAction\)](#)

Maps an PATCH route using the specified path and action function.

[MapPatch\(string, Delegate\)](#)

Maps an PATCH route using the specified path and action function.

[MapPost\(string, RouteAction\)](#)

Maps an POST route using the specified path and action function.

[MapPost\(string, Delegate\)](#)

Maps an POST route using the specified path and action function.

[MapPut\(string, RouteAction\)](#)

Maps an PUT route using the specified path and action function.

[MapPut\(string, Delegate\)](#)

Maps an PUT route using the specified path and action function.

[MatchRouteExpression\(in ReadOnlySpan<char>, in ReadOnlySpan<char>, StringComparison\)](#)

Attempts to match the specified route expression against the given path.

[MatchRouteExpression\(string, string, StringComparison\)](#)

Attempts to match the specified route expression against the given path.

[RegisterValueHandler<T>\(RouterActionHandlerCallback<T>\)](#)

Register an type handling association to converting it to an [HttpResponse](#) object.

[ResolveActionResult\(object?\)](#)

Resolves the specified object into an valid [HttpResponse](#) using the defined value handlers or throws an exception if not possible.

[Rewrite\(string, string\)](#)

Maps a rewrite route, which redirects all requests that match the given path to another path, keeping the body and headers of the original request.

[SetObject\(object\)](#)

Searches for all instance and static methods that are marked with an attribute of type [RouteAttribute](#) in the specified object and creates routes for these methods.

[SetObject\(Type\)](#)

Searches for all instance and static methods that are marked with an attribute of type [RouteAttribute](#) in the specified object and creates routes for these methods.

[SetObject\(Type, object\)](#)

Searches for all instance and static methods that are marked with an attribute of type [RouteAttribute](#) in the specified object and creates routes for these methods.

[SetObject< TObject >\(\)](#)

Searches for all instance and static methods that are marked with an attribute of type [RouteAttribute](#) in the specified object and creates routes for these methods.

[SetObject< TObject >\(TObject\)](#)

Searches for all instance and static methods that are marked with an attribute of type [RouteAttribute](#) in the specified object and creates routes for these methods.

[SetRoute\(Route\)](#)

Defines an route in this Router instance.

[SetRoute\(RouteMethod, string, RouteAction\)](#)

Defines an route with their method, path and action function.

[SetRoute\(RouteMethod, string, Delegate\)](#)

Defines an route with their method, path and action function.

[SetRoute\(RouteMethod, string, Delegate, string?\)](#)

Defines an route with their method, path, action function and name.

[SetRoute\(RouteMethod, string, Delegate, string?, IRequestHandler\[\]\)](#)

Defines an route with their method, path, action function, name and request handlers.

[SetRoutes\(IEnumerable<Route>\)](#)

Defines the specified collection of routes.

[TryResolveActionResult\(object?, out HttpResponse?\)](#)

Tries to resolve the specified object into an valid [HttpResponse](#) using the defined value handlers.

Constructor Router

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Router()

Creates an new [Router](#) instance with default values.

```
public Router()
```

Router(IEnumerable<Route>)

Creates an new [Router](#) instance with given route collection.

```
public Router(IEnumerable<Route> routes)
```

Parameters

routes [IEnumerable](#)<Route>

The route collection to import in this router.

Property CallbackErrorHandler

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

CallbackErrorHandler

Gets or sets the Router action exception handler. The response handler for this property will send an HTTP response to the client when an exception is caught during execution. This property is only called when [Throw Exceptions](#) is disabled.

```
public ExceptionErrorCallback? CallbackErrorHandler { get; set; }
```

Property Value

[ExceptionErrorCallback](#)

Property CheckForRouteCollisions

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

CheckForRouteCollisions

Gets or sets whether this [Router](#) should check for possible routing collisions before starting the HTTP server.

```
public bool CheckForRouteCollisions { get; set; }
```

Property Value

bool 

Property Count

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Count

```
public int Count { get; }
```

Property Value

[int](#)

Property GlobalRequestHandlers

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

GlobalRequestHandlers

Gets or sets the global requests handlers that will be executed in all matched routes.

```
public IRequestHandler[] GlobalRequestHandlers { get; set; }
```

Property Value

[IRequestHandler\[\]](#)

Property IsReadOnly

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

IsReadOnly

Gets an boolean indicating where this [Router](#) is read-only or not.

```
public bool IsReadOnly { get; }
```

Property Value

[bool](#)

Property MatchRoutesIgnoreCase

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

MatchRoutesIgnoreCase

Gets or sets whether this [Router](#) will match routes ignoring case.

```
public bool MatchRoutesIgnoreCase { get; set; }
```

Property Value

bool 

Property MethodNotAllowedErrorHandler

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

MethodNotAllowedErrorHandler

Gets or sets the Router "405 Method Not Allowed" handler.

```
public RoutingErrorCallback? MethodNotAllowedErrorHandler { get; set; }
```

Property Value

[RoutingErrorCallback](#)

Property NotFoundErrorHandler

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

NotFoundErrorHandler

Gets or sets the Router "404 Not Found" handler.

```
public RoutingErrorCallback? NotFoundErrorHandler { get; set; }
```

Property Value

[RoutingErrorCallback](#)

Property Prefix

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Prefix

Gets or sets the prefix which will be applied to all next defining routes in this router.

```
public string? Prefix { get; set; }
```

Property Value

[string](#)

Method AutoScanModules

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

AutoScanModules(Type, Assembly)

Scans for all types that implements the specified module type and associates an instance of each type to the router.

```
public void AutoScanModules(Type moduleType, Assembly searchAssembly)
```

Parameters

moduleType [Type](#)

An class which implements [RouterModule](#), or the router module itself.

searchAssembly [Assembly](#)

The assembly to search the module type in.

AutoScanModules<TModule>(Assembly)

Scans for all types that implements `TModule` and associates an instance of each type to the router. Note that, `TModule` must be an [RouterModule](#) type and an accessible constructor for each type must be present.

```
public void AutoScanModules<TModule>(Assembly assembly) where TModule : RouterModule
```

Parameters

assembly [Assembly](#)

The assembly to search `TModule` in.

Type Parameters

TModule

An class which implements [RouterModule](#), or the router module itself.

AutoScanModules<TModule>()

Scans for all types that implements `TModule` and associates an instance of each type to the router. Note that, `TModule` must be an [RouterModule](#) type and an accessible constructor for each type must be present.

```
public void AutoScanModules<TModule>() where TModule : RouterModule
```

Type Parameters

TModule

An class which implements [RouterModule](#), or the router module itself.

Method Combine

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Combine(params string[])

Combines an array of string parts into a single path.

```
public static string Combine(params string[] parts)
```

Parameters

parts [string](#)[]

An array of string parts to combine.

Returns

[string](#)

A string representing the combined path.

Method GetDefinedRoutes

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

GetDefinedRoutes()

Gets all routes defined on this router instance.

```
public Route[] GetDefinedRoutes()
```

Returns

[Route\[\]](#)

Method GetEnumerator

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

GetEnumerator()

```
public IEnumarator<Route> GetEnumerator()
```

Returns

[IEnumarator](#)<Route>

Method GetRouteFromName

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

GetRouteFromName(string)

Gets an defined [Route](#) by their name property.

```
public Route? GetRouteFromName(string name)
```

Parameters

name [string](#)

The route name.

Returns

[Route](#)

Method GetRouteFromPath

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

GetRouteFromPath(RouteMethod, string)

Gets the first matched [Route](#) by their HTTP method and path.

```
public Route? GetRouteFromPath(RouteMethod method, string uri)
```

Parameters

method [RouteMethod](#)

The HTTP method to match.

uri [string](#) ↗

The URL expression.

Returns

[Route](#)

GetRouteFromPath(string)

Gets the first matched [Route](#) by their URL path.

```
public Route? GetRouteFromPath(string uri)
```

Parameters

uri [string](#) ↗

The URL expression.

Returns

Route

Method IsDefined

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

IsDefined(RouteMethod, string)

Gets an boolean indicating if there are any route that matches the specified method and route path.

```
public bool IsDefined(RouteMethod method, string path)
```

Parameters

method [RouteMethod](#)

The route method.

path [string](#)

The route path.

Returns

[bool](#)

Method IsRouteExpressionsOverlap

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

IsRouteExpressionsOverlap(in ReadOnlySpan<char>, in ReadOnlySpan<char>, StringComparison)

Determines whether two route expressions overlap.

```
public static bool IsRouteExpressionsOverlap(in ReadOnlySpan<char> routeExpression1, in  
    ReadOnlySpan<char> routeExpression2, StringComparison stringComparer = StringComparison.Ordinal)
```

Parameters

routeExpression1 [ReadOnlySpan<char>](#)

The first route expression to compare.

routeExpression2 [ReadOnlySpan<char>](#)

The second route expression to compare.

stringComparer [StringComparison](#)

The string comparison to use when comparing the route expressions. Defaults to [Ordinal](#).

Returns

[bool](#)

`true` if the route expressions overlap; otherwise, `false`.

IsRouteExpressionsOverlap(string, string, StringComparison)

Determines whether two route expressions overlap.

```
public static bool IsRouteExpressionsOverlap(string routeExpression1, string routeExpression2,  
StringComparison stringComparer = StringComparison.Ordinal)
```

Parameters

routeExpression1 [string](#)

The first route expression to compare.

routeExpression2 [string](#)

The second route expression to compare.

stringComparer [StringComparison](#)

The string comparison to use when comparing the route expressions. Defaults to [Ordinal](#).

Returns

[bool](#)

true if the route expressions overlap; otherwise, false .

Method MapAny

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

MapAny(string, Delegate)

Maps an route which matches any HTTP method, using the specified path and action function.

```
public void MapAny(string path, Delegate action)
```

Parameters

path [string](#)

The route path.

action [Delegate](#)

The route function to be called after matched.

MapAny(string, RouteAction)

Maps an route which matches any HTTP method, using the specified path and action function.

```
public void MapAny(string path, RouteAction action)
```

Parameters

path [string](#)

The route path.

action [RouteAction](#)

The route function to be called after matched.

Method MapDelete

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

MapDelete(string, Delegate)

Maps an DELETE route using the specified path and action function.

```
public void MapDelete(string path, Delegate action)
```

Parameters

path [string](#)

The route path.

action [Delegate](#)

The route function to be called after matched.

MapDelete(string, RouteAction)

Maps an DELETE route using the specified path and action function.

```
public void MapDelete(string path, RouteAction action)
```

Parameters

path [string](#)

The route path.

action [RouteAction](#)

The route function to be called after matched.

Method MapGet

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

MapGet(string, Delegate)

Maps an GET route using the specified path and action function.

```
public void MapGet(string path, Delegate action)
```

Parameters

path [string](#)

The route path.

action [Delegate](#)

The route function to be called after matched.

MapGet(string, RouteAction)

Maps an GET route using the specified path and action function.

```
public void MapGet(string path, RouteAction action)
```

Parameters

path [string](#)

The route path.

action [RouteAction](#)

The route function to be called after matched.

Method MapHead

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

MapHead(string, Delegate)

Maps an HEAD route using the specified path and action function.

```
public void MapHead(string path, Delegate action)
```

Parameters

path [string](#)

The route path.

action [Delegate](#)

The route function to be called after matched.

MapHead(string, RouteAction)

Maps an HEAD route using the specified path and action function.

```
public void MapHead(string path, RouteAction action)
```

Parameters

path [string](#)

The route path.

action [RouteAction](#)

The route function to be called after matched.

Method MapOptions

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

MapOptions(string, Delegate)

Maps an OPTIONS route using the specified path and action function.

```
public void MapOptions(string path, Delegate action)
```

Parameters

path [string](#)

The route path.

action [Delegate](#)

The route function to be called after matched.

MapOptions(string, RouteAction)

Maps an OPTIONS route using the specified path and action function.

```
public void MapOptions(string path, RouteAction action)
```

Parameters

path [string](#)

The route path.

action [RouteAction](#)

The route function to be called after matched.

Method MapPatch

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

MapPatch(string, Delegate)

Maps an PATCH route using the specified path and action function.

```
public void MapPatch(string path, Delegate action)
```

Parameters

path [string](#)

The route path.

action [Delegate](#)

The route function to be called after matched.

MapPatch(string, RouteAction)

Maps an PATCH route using the specified path and action function.

```
public void MapPatch(string path, RouteAction action)
```

Parameters

path [string](#)

The route path.

action [RouteAction](#)

The route function to be called after matched.

Method MapPost

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

MapPost(string, Delegate)

Maps an POST route using the specified path and action function.

```
public void MapPost(string path, Delegate action)
```

Parameters

path [string](#)

The route path.

action [Delegate](#)

The route function to be called after matched.

MapPost(string, RouteAction)

Maps an POST route using the specified path and action function.

```
public void MapPost(string path, RouteAction action)
```

Parameters

path [string](#)

The route path.

action [RouteAction](#)

The route function to be called after matched.

Method MapPut

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

MapPut(string, Delegate)

Maps an PUT route using the specified path and action function.

```
public void MapPut(string path, Delegate action)
```

Parameters

path [string](#)

The route path.

action [Delegate](#)

The route function to be called after matched.

MapPut(string, RouteAction)

Maps an PUT route using the specified path and action function.

```
public void MapPut(string path, RouteAction action)
```

Parameters

path [string](#)

The route path.

action [RouteAction](#)

The route function to be called after matched.

Method MatchRouteExpression

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

MatchRouteExpression(in ReadOnlySpan<char>, in ReadOnlySpan<char>, StringComparison)

Attempts to match the specified route expression against the given path.

```
public static RouteMatch MatchRouteExpression(in ReadOnlySpan<char> routeExpression, in  
ReadOnlySpan<char> path, StringComparison stringComparer = StringComparison.Ordinal)
```

Parameters

routeExpression [ReadOnlySpan<char>](#)

The route expression to match.

path [ReadOnlySpan<char>](#)

The path to match against the route expression.

stringComparer [StringComparison](#)

The string comparison to use when matching the route expression. Defaults to [Ordinal](#).

Returns

[RouteMatch](#)

A [RouteMatch](#) object indicating the result of the match.

MatchRouteExpression(string, string, StringComparison)

Attempts to match the specified route expression against the given path.

```
public static RouteMatch MatchRouteExpression(string routeExpression, string path,  
StringComparison stringComparer = StringComparison.OrdinalIgnoreCase)
```

Parameters

routeExpression [string](#)

The route expression to match.

path [string](#)

The path to match against the route expression.

stringComparer [StringComparison](#)

The string comparison to use when matching the route expression. Defaults to [Ordinal](#).

Returns

[RouteMatch](#)

A [RouteMatch](#) object indicating the result of the match.

Method RegisterValueHandler

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RegisterValueHandler<T>(RouterActionHandlerCallback<T>)

Register an type handling association to converting it to an [HttpResponse](#) object.

```
public void RegisterValueHandler<T>(RouterActionHandlerCallback<T> actionHandler) where T : notnull
```

Parameters

actionHandler [RouterActionHandlerCallback<T>](#)

The function that receives an object of the `T` and returns an [HttpResponse](#) response from the informed object.

Type Parameters

`T`

Method ResolveActionResult

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

ResolveActionResult(object?)

Resolves the specified object into an valid [HttpResponse](#) using the defined value handlers or throws an exception if not possible.

```
public HttpResponse ResolveActionResult(object? result)
```

Parameters

result [object](#)↗

The object that will be converted to an valid [HttpResponse](#).

Returns

[HttpResponse](#)

Remarks

This method can throw exceptions. To avoid exceptions while trying to convert the specified object into an [HttpResponse](#), consider using [TryResolveActionResult\(object?, out HttpResponse?\)](#).

Method Rewrite

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Rewrite(string, string)

Maps a rewrite route, which redirects all requests that match the given path to another path, keeping the body and headers of the original request.

```
public void Rewrite(string rewritePath, string rewriteInto)
```

Parameters

rewritePath [string](#)

The incoming HTTP request path.

rewriteInto [string](#)

The rewritten URL.

Method SetObject

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

SetObject(object)

Searches for all instance and static methods that are marked with an attribute of type [RouteAttribute](#) in the specified object and creates routes for these methods.

```
public void SetObject(object attrClassInstance)
```

Parameters

attrClassInstance [object](#)

The instance of the class where the methods are. The routing methods must be marked with any [RouteAttribute](#).

Exceptions

[Exception](#)

An exception is thrown when a method has an erroneous signature.

SetObject(Type)

Searches for all instance and static methods that are marked with an attribute of type [RouteAttribute](#) in the specified object and creates routes for these methods.

```
public void SetObject(Type attrClassType)
```

Parameters

attrClassType [Type](#)

The type of the class where the methods are. The routing methods must be marked with any [RouteAttribute](#).

SetObject(Type, object)

Searches for all instance and static methods that are marked with an attribute of type [RouteAttribute](#) in the specified object and creates routes for these methods.

```
public void SetObject(Type attrClassType, object instance)
```

Parameters

attrClassType [Type](#)

The type of the class where the methods are. The routing methods must be marked with any [RouteAttribute](#).

instance [object](#)

The instance of the object where the route methods are.

SetObject<TObject>()

Searches for all instance and static methods that are marked with an attribute of type [RouteAttribute](#) in the specified object and creates routes for these methods.

```
public void SetObject<TObject>()
```

Type Parameters

TObject

The type of the class where the methods are. The routing methods must be marked with any [RouteAttribute](#).

Exceptions

[Exception](#)

An exception is thrown when a method has an erroneous signature.

SetObject<TObject>(TObject)

Searches for all instance and static methods that are marked with an attribute of type [RouteAttribute](#) in the specified object and creates routes for these methods.

```
public void SetObject<TObject>(TObject instance) where TObject : notnull
```

Parameters

instance TObject

The instance of TObject to invoke the instance methods on.

Type Parameters

TObject

The type of the class where the methods are. The routing methods must be marked with any [RouteAttribute](#).

Exceptions

[Exception](#) ↗

An exception is thrown when a method has an erroneous signature.

Method SetRoute

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

SetRoute(RouteMethod, string, RouteAction)

Defines an route with their method, path and action function.

```
public void SetRoute(RouteMethod method, string path, RouteAction action)
```

Parameters

method [RouteMethod](#)

The route method to be matched. "Any" means any method that matches their path.

path [string](#)

The route path.

action [RouteAction](#)

The route function to be called after matched.

SetRoute(RouteMethod, string, Delegate)

Defines an route with their method, path and action function.

```
public void SetRoute(RouteMethod method, string path, Delegate action)
```

Parameters

method [RouteMethod](#)

The route method to be matched. "Any" means any method that matches their path.

path `string`

The route path.

action `Delegate`

The route function to be called after matched.

SetRoute(RouteMethod, string, Delegate, string?)

Defines an route with their method, path, action function and name.

```
public void SetRoute(RouteMethod method, string path, Delegate action, string? name)
```

Parameters

method `RouteMethod`

The route method to be matched. "Any" means any method that matches their path.

path `string`

The route path.

action `Delegate`

The route function to be called after matched.

name `string`

The route name.

SetRoute(RouteMethod, string, Delegate, string?, IRequestHandler[])

Defines an route with their method, path, action function, name and request handlers.

```
public void SetRoute(RouteMethod method, string path, Delegate action, string? name,
IRequestHandler[] middlewares)
```

Parameters

method `RouteMethod`

The route method to be matched. "Any" means any method that matches their path.

path [string](#)

The route path.

action [Delegate](#)

The route function to be called after matched.

name [string](#)

The route name.

middlewares [IRequestHandler\[\]](#)

Handlers that run before calling your route action.

SetRoute(Route)

Defines an route in this Router instance.

```
public void SetRoute(Route r)
```

Parameters

r [Route](#)

The route to be defined in the Router.

Method SetRoutes

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

SetRoutes(IEnumerable<Route>)

Defines the specified collection of routes.

```
public void SetRoutes(IEnumerable<Route> routes)
```

Parameters

routes [IEnumerable<Route>](#)

The routes to be defined.

Method TryResolveActionResult

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

TryResolveActionResult(object?, out HttpResponse?)

Tries to resolve the specified object into an valid [HttpResponse](#) using the defined value handlers.

```
public bool TryResolveActionResult(object? result, out HttpResponse? response)
```

Parameters

result [object](#)

The object that will be converted to an valid [HttpResponse](#).

response [HttpResponse](#)

When this method returns, the response object. This parameter is not initialized.

Returns

[bool](#)

When this method returns, the [HttpResponse](#) object.

Delegate RouterActionHandlerCallback<T>

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents the function that receives an object of the `T` and returns an [HttpResponse](#) response from the informed object.

```
public delegate HttpResponse RouterActionHandlerCallback<T>(T input) where T : notnull
```

Parameters

input `T`

The result router object.

Returns

[HttpResponse](#)

Represents the function that receives an object of the `T` and returns an response from the informed object.

Type Parameters

`T`

The input object type. Cannot be nullable.

Constructors

[RouterActionHandlerCallback\(object, nint\)](#)

Methods

[BeginInvoke\(T, AsyncCallback, object\)](#)

EndInvoke(IAsyncResult)

Invoke(T)

Constructor RouterActionHandlerCallback

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RouterActionHandlerCallback(object, nint)

```
public RouterActionHandlerCallback(object @object, nint method)
```

Parameters

object object

method nint

Method BeginInvoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

BeginInvoke(T, AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(T input, AsyncCallback callback, object @object)
```

Parameters

input T

callback AsyncCallback

object object

Returns

IAsyncResult

Method EndInvoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

EndInvoke(IAsyncResult)

```
public virtual HttpResponse EndInvoke(IAsyncResult result)
```

Parameters

result [IAsyncResult](#) ↗

Returns

[HttpResponse](#)

Method Invoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Invoke(T)

```
public virtual HttpResponse Invoke(T input)
```

Parameters

input T

Returns

[HttpResponse](#)

Class RouterModule

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Indicates that extended class supports router modules, which allows the management of routes, request handlers and prefixes.

```
public abstract class RouterModule
```

Constructors

[RouterModule\(\)](#)

Properties

[Prefix](#)

Gets or sets the router prefix for this class. This property overrides any value defined by [RoutePrefixAttribute](#) set in this class.

[RequestHandlers](#)

Gets or sets an list of [IRequestHandler](#) this [RouterModule](#) runs.

Methods

[HasPrefix\(string\)](#)

Specifies a prefix for all routes defined by this module.

[HasRequestHandler\(IRequestHandler\)](#)

Registers an [IRequestHandler](#) on all routes defined by this module.

[OnRouteCreating\(Route\)](#)

This method is called before a route is defined in the router and after it is created in this class, so its attributes and parameters can be modified. This method must be overloaded in the extending class and must not be called directly.

[OnSetup\(Router\)](#)

Method that is called when an [Router](#) is defining routes from the current [RouterModule](#).

Constructor RouterModule

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RouterModule()

```
protected RouterModule()
```

Property Prefix

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Prefix

Gets or sets the router prefix for this class. This property overrides any value defined by [RoutePrefixAttribute](#) set in this class.

```
public string? Prefix { get; set; }
```

Property Value

[string](#)

Property RequestHandlers

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RequestHandlers

Gets or sets an list of [IRequestHandler](#) this [RouterModule](#) runs.

```
public IList<IRequestHandler> RequestHandlers { get; set; }
```

Property Value

[IList](#)<[IRequestHandler](#)>

Method HasPrefix

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

HasPrefix(string)

Specifies a prefix for all routes defined by this module.

```
protected void HasPrefix(string prefix)
```

Parameters

prefix string ↗

The prefix to be applied to all registered routes of this class.

Remarks

This method allows for the specification of a common prefix for all routes defined by this module, which can be useful for organizing and structuring routes in a large application.

Method HasRequestHandler

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

HasRequestHandler(IRequestHandler)

Registers an [IRequestHandler](#) on all routes defined by this module.

```
protected void HasRequestHandler(IRequestHandler handler)
```

Parameters

handler [IRequestHandler](#)

The [IRequestHandler](#) instance which will be applied to all registered routes of this class.

Method OnRouteCreating

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

OnRouteCreating(Route)

This method is called before a route is defined in the router and after it is created in this class, so its attributes and parameters can be modified. This method must be overloaded in the extending class and must not be called directly.

```
protected virtual void OnRouteCreating(Route configuringRoute)
```

Parameters

configuringRoute [Route](#)

The route being defined on the router.

Method OnSetup

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

OnSetup(Router)

Method that is called when an [Router](#) is defining routes from the current [RouterModule](#).

```
protected virtual void OnSetup(Router parentRouter)
```

Parameters

parentRouter Router

The [Router](#) which is defining routes from the current [RouterModule](#).

Remarks

The base method [OnSetup\(Router\)](#) is mandatory to be called on all derived methods.

Delegate RoutingErrorCallback

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents the function that is called when an request reaches an error on the router.

```
public delegate HttpResponse RoutingErrorCallback(HttpContext context)
```

Parameters

context [HttpContext](#)

Represents the function that is called when an request reaches an error on the router.

Returns

[HttpResponse](#)

Represents the function that is called when an request reaches an error on the router.

Constructors

[RoutingErrorCallback\(object, nint\)](#)

Methods

[BeginInvoke\(HttpContext, AsyncCallback, object\)](#)

[EndInvoke\(IAsyncResult\)](#)

[Invoke\(HttpContext\)](#)

Constructor RoutingErrorCallback

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

RoutingErrorCallback(object, nint)

```
public RoutingErrorCallback(object @object, nint method)
```

Parameters

object object ↗

method nint ↗

Method BeginInvoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

BeginInvoke(HttpContext, AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(HttpContext context, AsyncCallback callback,
object @object)
```

Parameters

context [HttpContext](#)

callback [AsyncCallback](#)

object [object](#)

Returns

[IAsyncResult](#)

Method EndInvoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

EndInvoke(IAsyncResult)

```
public virtual HttpResponse EndInvoke(IAsyncResult result)
```

Parameters

result [IAsyncResult](#) ↗

Returns

[HttpResponse](#)

Method Invoke

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Invoke(HttpContext)

```
public virtual HttpResponse Invoke(HttpContext context)
```

Parameters

context [HttpContext](#)

Returns

[HttpResponse](#)

Class ValueResult<T>

Namespace:[Sisk.Core.Routing](#)

Assembly:Sisk.Core.dll

Represents a mutable type for boxing objects by value or reference in a response from a router.

```
public sealed class ValueResult<T> where T : notnull
```

Type Parameters

T

The type of object to be boxed.

Namespace Sisk.Documenting

Namespaces

[Sisk.Documenting.Annotations](#)

[Sisk.Documenting.Html](#)

Classes

[ApiDocumentation](#)

Represents the API documentation, including application details and endpoints.

[ApiEndpoint](#)

Represents an API endpoint, including its metadata, request and response details.

[ApiEndpointHeader](#)

Represents a header for an API endpoint, including its name and requirements.

[ApiEndpointParameter](#)

Represents a parameter for an API endpoint, including its name, type, and requirements.

[ApiEndpointPathParameter](#)

Represents a path parameter for an API endpoint, including its name, type, and description.

[ApiEndpointRequestExample](#)

Represents an example request for an API endpoint, including its description and example content.

[ApiEndpointResponse](#)

Represents a response for an API endpoint, including the status code and example content.

[Apildentifier](#)

Represents an identifier for an API, including application details such as name, version, and description.

Interfaces

[IApiDocumentationExporter](#)

Defines a contract for exporting API documentation content.

Namespace Sisk.Documenting.Annotations

Classes

[ApiEndpointAttribute](#)

Specifies an attribute for an API endpoint, allowing metadata such as name, description, and group to be associated with methods.

[ApiHeaderAttribute](#)

Specifies an attribute for an API header, allowing metadata such as header name, description, and requirement status to be associated with methods.

[ApiParameterAttribute](#)

Specifies an attribute for an API parameter, allowing metadata such as name, type, description, and requirement status to be associated with methods.

[ApiPathParameterAttribute](#)

Specifies an attribute for an API path parameter, allowing metadata such as name, type, and description to be associated with methods.

[ApiRequestAttribute](#)

Specifies an attribute for an API request, allowing metadata such as description, example language, and example content to be associated with methods.

[ApiResponseAttribute](#)

Specifies an attribute for an API response, allowing metadata such as status code, description, example content, and example language to be associated with methods.

Class ApiEndpointAttribute

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Specifies an attribute for an API endpoint, allowing metadata such as name, description, and group to be associated with methods.

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = false)]
public sealed class ApiEndpointAttribute : Attribute
```

Constructors

[ApiEndpointAttribute\(string\)](#)

Initializes a new instance of the [ApiEndpointAttribute](#) class with the specified endpoint name.

Properties

[Description](#)

Gets or sets the description of the API endpoint.

[Group](#)

Gets or sets the group to which the API endpoint belongs.

[Name](#)

Gets or sets the name of the API endpoint.

Constructor ApiEndpointAttribute

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

ApiEndpointAttribute(string)

Initializes a new instance of the [ApiEndpointAttribute](#) class with the specified endpoint name.

```
public ApiEndpointAttribute(string name)
```

Parameters

name [string](#)

The name of the API endpoint.

Property Description

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Description

Gets or sets the description of the API endpoint.

```
public string? Description { get; set; }
```

Property Value

[string](#)

Property Group

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Group

Gets or sets the group to which the API endpoint belongs.

```
public string? Group { get; set; }
```

Property Value

[string](#)

Property Name

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Name

Gets or sets the name of the API endpoint.

```
public string Name { get; set; }
```

Property Value

[string](#)

Class ApiHeaderAttribute

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Specifies an attribute for an API header, allowing metadata such as header name, description, and requirement status to be associated with methods.

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = true)]
public sealed class ApiHeaderAttribute : Attribute
```

Constructors

[ApiHeaderAttribute\(string\)](#)

Initializes a new instance of the [ApiHeaderAttribute](#) class with the specified header name.

Properties

[Description](#)

Gets or sets the description of the header.

[HeaderName](#)

Gets the name of the header.

[IsRequired](#)

Gets or sets a value indicating whether the header is required.

Constructor ApiHeaderAttribute

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

ApiHeaderAttribute(string)

Initializes a new instance of the [ApiHeaderAttribute](#) class with the specified header name.

```
public ApiHeaderAttribute(string headerName)
```

Parameters

headerName [string](#)

The name of the header.

Property Description

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Description

Gets or sets the description of the header.

```
public string? Description { get; set; }
```

Property Value

[string](#)

Property HeaderName

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

HeaderName

Gets the name of the header.

```
public string HeaderName { get; }
```

Property Value

[string](#)

Property IsRequired

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

IsRequired

Gets or sets a value indicating whether the header is required.

```
public bool IsRequired { get; set; }
```

Property Value

[bool](#)

Class ApiParameterAttribute

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Specifies an attribute for an API parameter, allowing metadata such as name, type, description, and requirement status to be associated with methods.

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = true)]
public sealed class ApiParameterAttribute : Attribute
```

Constructors

[ApiParameterAttribute\(string, string\)](#)

Initializes a new instance of the [ApiParameterAttribute](#) class with the specified name and type name.

Properties

[Description](#)

Gets or sets the description of the parameter.

[IsRequired](#)

Gets or sets a value indicating whether the parameter is required.

[Name](#)

Gets the name of the parameter.

[TypeName](#)

Gets the type name of the parameter.

Constructor ApiParameterAttribute

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

ApiParameterAttribute(string, string)

Initializes a new instance of the [ApiParameterAttribute](#) class with the specified name and type name.

```
public ApiParameterAttribute(string name, string typeName)
```

Parameters

name [string](#)

The name of the parameter.

typeName [string](#)

The type name of the parameter.

Property Description

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Description

Gets or sets the description of the parameter.

```
public string? Description { get; set; }
```

Property Value

[string](#)

Property IsRequired

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

IsRequired

Gets or sets a value indicating whether the parameter is required.

```
public bool IsRequired { get; set; }
```

Property Value

bool

Property Name

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Name

Gets the name of the parameter.

```
public string Name { get; }
```

Property Value

[string](#)

Property TypeName

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

TypeName

Gets the type name of the parameter.

```
public string TypeName { get; }
```

Property Value

[string](#)

Class ApiPathParameterAttribute

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Specifies an attribute for an API path parameter, allowing metadata such as name, type, and description to be associated with methods.

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = true)]
public sealed class ApiPathParameterAttribute : Attribute
```

Constructors

[ApiPathParameterAttribute\(string\)](#)

Initializes a new instance of the [ApiPathParameterAttribute](#) class with the specified name.

Properties

Description

Gets or sets the description of the path parameter.

Name

Gets the name of the path parameter.

Type

Gets or sets the type of the path parameter.

Constructor ApiPathParameterAttribute

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

ApiPathParameterAttribute(string)

Initializes a new instance of the [ApiPathParameterAttribute](#) class with the specified name.

```
public ApiPathParameterAttribute(string name)
```

Parameters

name [string](#)

The name of the path parameter.

Property Description

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Description

Gets or sets the description of the path parameter.

```
public string? Description { get; set; }
```

Property Value

[string](#)

Property Name

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Name

Gets the name of the path parameter.

```
public string Name { get; }
```

Property Value

[string](#)

Property Type

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Type

Gets or sets the type of the path parameter.

```
public string? Type { get; set; }
```

Property Value

[string](#)

Class ApiRequestAttribute

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Specifies an attribute for an API request, allowing metadata such as description, example language, and example content to be associated with methods.

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = true)]
public sealed class ApiRequestAttribute : Attribute
```

Constructors

[ApiRequestAttribute\(string\)](#)

Initializes a new instance of the [ApiRequestAttribute](#) class with the specified description.

Properties

Description

Gets or sets the description of the API request.

Example

Gets or sets the actual example request content.

ExampleLanguage

Gets or sets the programming language used in the example, if applicable.

Constructor ApiRequestAttribute

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

ApiRequestAttribute(string)

Initializes a new instance of the [ApiRequestAttribute](#) class with the specified description.

```
public ApiRequestAttribute(string description)
```

Parameters

description string ↗

The description of the API request.

Property Description

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Description

Gets or sets the description of the API request.

```
public string Description { get; set; }
```

Property Value

[string](#)

Property Example

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Example

Gets or sets the actual example request content.

```
public string? Example { get; set; }
```

Property Value

[string](#)

Property ExampleLanguage

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

ExampleLanguage

Gets or sets the programming language used in the example, if applicable.

```
public string? ExampleLanguage { get; set; }
```

Property Value

[string](#)

Class ApiResponseAttribute

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Specifies an attribute for an API response, allowing metadata such as status code, description, example content, and example language to be associated with methods.

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = true)]
public sealed class ApiResponseAttribute : Attribute
```

Constructors

[ApiResponseAttribute\(HttpStatusCode\)](#)

Initializes a new instance of the [ApiResponseAttribute](#) class with the specified status code.

Properties

[Description](#)

Gets or sets the description of the response.

[Example](#)

Gets or sets the example response content.

[ExampleLanguage](#)

Gets or sets the programming language used in the example, if applicable.

[StatusCode](#)

Gets the HTTP status code for the response.

Constructor ApiResponseAttribute

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

ApiResponseAttribute(HttpStatusCode)

Initializes a new instance of the [ApiResponseAttribute](#) class with the specified status code.

```
public ApiResponseAttribute(HttpStatusCode statusCode)
```

Parameters

statusCode [HttpStatusCode](#)

The HTTP status code for the response.

Property Description

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Description

Gets or sets the description of the response.

```
public string? Description { get; set; }
```

Property Value

[string](#)

Property Example

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

Example

Gets or sets the example response content.

```
public string? Example { get; set; }
```

Property Value

[string](#)

Property ExampleLanguage

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

ExampleLanguage

Gets or sets the programming language used in the example, if applicable.

```
public string? ExampleLanguage { get; set; }
```

Property Value

[string](#)

Property StatusCode

Namespace:[Sisk.Documenting.Annotations](#)

Assembly:Sisk.Documenting.dll

StatusCode

Gets the HTTP status code for the response.

```
public HttpStatusCode StatusCode { get; }
```

Property Value

[HttpStatusCode](#)

Namespace Sisk.Documenting.Html

Classes

[HtmlDocumentationExporter](#)

Represents a class for exporting API documentation to HTML format.

Class HtmlDocumentationExporter

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

Represents a class for exporting API documentation to HTML format.

```
public class HtmlDocumentationExporter : IApiDocumentationExporter
```

Implements

[IApiDocumentationExporter](#)

Constructors

[HtmlDocumentationExporter\(\)](#)

Creates a new instance of the [HtmlDocumentationExporter](#) class.

Properties

[Footer](#)

Gets or sets an optional object to append after the generated contents, right at the end of the `<main>` tag of the generated page.

[FormatEndpointHeaders](#)

Gets or sets the format string for endpoint headers.

[FormatEndpointParameters](#)

Gets or sets the format string for endpoint request parameters.

[FormatEndpointPathParameters](#)

Gets or sets the format string for endpoint path parameters.

[FormatEndpointRequestExamples](#)

Gets or sets the format string for endpoint request examples.

[FormatEndpointResponses](#)

Gets or sets the format string for endpoint responses.

[FormatMainTitleServiceVersion](#)

Gets or sets the format string for the main title service version.

[FormatRequiredText](#)

Gets or sets the format string for required text.

[Head](#)

Gets or sets an optional object to append inside the `<head>` tag of the generated page.

[Header](#)

Gets or sets an optional object to append after the main title, at the beginning of the `<main>` tag of the generated page.

[PageTitle](#)

Gets or sets the title of the HTML page.

[Script](#)

Gets or sets the JavaScript script to be included in the HTML page.

[Style](#)

Gets or sets the CSS styles of the HTML page.

Methods

[CreateCodeBlock\(string, string?\)](#)

Creates an HTML code block element from the provided code and language.

[CreateEndpointBadge\(RouteMethod, string?\)](#)

Creates an HTML badge element for an API endpoint.

[CreateParagraphs\(string?\)](#)

Creates one or more HTML paragraph elements from the provided text.

[Ellipsis\(string?, int\)](#)

Truncates a string to the specified size, appending an ellipsis if necessary.

[ExportDocumentationContent\(ApiDocumentation\)](#)

Exports the API documentation as HTTP content.

[ExportHtml\(ApiDocumentation\)](#)

Exports the API documentation as an HTML string.

GetRouteMethodHexColor(RouteMethod)

Gets the hex color code associated with the specified route method.

TransformId(string)

Transforms an unsafe ID into a safe and valid HTML ID.

WriteEndpointDescription(ApiEndpoint)

Writes the description of an API endpoint.

WriteMainTitle(ApiDocumentation)

Writes the main title of the API documentation.

Constructor HtmlDocumentationExporter

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

HtmlDocumentationExporter()

Creates an new instance of the [HtmlDocumentationExporter](#) class.

```
public HtmlDocumentationExporter()
```

Property Footer

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

Footer

Gets or sets an optional object to append after the generated contents, right at the end of the `<main>` tag of the generated page.

```
public object? Footer { get; set; }
```

Property Value

[object](#)

Property FormatEndpointHeaders

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

FormatEndpointHeaders

Gets or sets the format string for endpoint headers.

```
public string FormatEndpointHeaders { get; set; }
```

Property Value

[string](#)

Property FormatEndpointParameters

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

FormatEndpointParameters

Gets or sets the format string for endpoint request parameters.

```
public string FormatEndpointParameters { get; set; }
```

Property Value

[string](#)

Property FormatEndpointPathParameters

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

FormatEndpointPathParameters

Gets or sets the format string for endpoint path parameters.

```
public string FormatEndpointPathParameters { get; set; }
```

Property Value

[string](#)

Property FormatEndpointRequestExamples

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

FormatEndpointRequestExamples

Gets or sets the format string for endpoint request examples.

```
public string FormatEndpointRequestExamples { get; set; }
```

Property Value

[string](#)

Property FormatEndpointResponses

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

FormatEndpointResponses

Gets or sets the format string for endpoint responses.

```
public string FormatEndpointResponses { get; set; }
```

Property Value

[string](#)

Property FormatMainTitleServiceVersion

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

FormatMainTitleServiceVersion

Gets or sets the format string for the main title service version.

```
public string FormatMainTitleServiceVersion { get; set; }
```

Property Value

[string](#)

Property FormatRequiredText

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

FormatRequiredText

Gets or sets the format string for required text.

```
public string FormatRequiredText { get; set; }
```

Property Value

[string](#)

Property Head

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

Head

Gets or sets an optional object to append inside the `<head>` tag of the generated page.

```
public object? Head { get; set; }
```

Property Value

[object](#)

Property Header

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

Header

Gets or sets an optional object to append after the main title, at the beginning of the `<main>` tag of the generated page.

```
public object? Header { get; set; }
```

Property Value

[object](#)

Property PageTitle

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

PageTitle

Gets or sets the title of the HTML page.

```
public string PageTitle { get; set; }
```

Property Value

[string](#)

Property Script

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

Script

Gets or sets the JavaScript script to be included in the HTML page.

```
public string? Script { get; set; }
```

Property Value

[string](#)

Property Style

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

Style

Gets or sets the CSS styles of the HTML page.

```
public string Style { get; set; }
```

Property Value

[string](#)

Method CreateCodeBlock

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

CreateCodeBlock(string, string?)

Creates an HTML code block element from the provided code and language.

```
protected virtual HtmlElement? CreateCodeBlock(string code, string? language)
```

Parameters

code [string](#)

The code to display in the code block.

language [string](#)

The programming language of the code, or null for no language highlighting.

Returns

[HtmlElement](#)

The HTML element representing the code block, or null if no code is provided.

Method CreateEndpointBadge

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

CreateEndpointBadge(RouteMethod, string?)

Creates an HTML badge element for an API endpoint.

```
protected virtual HtmlElement? CreateEndpointBadge(RouteMethod method, string? path)
```

Parameters

method [RouteMethod](#)

The HTTP method of the endpoint (e.g. GET, POST, PUT, etc.).

path [string](#) ↗

The path of the endpoint, or null for no path display.

Returns

[HtmlElement](#)

The HTML element representing the endpoint badge, or null if no badge is applicable.

Method CreateParagraphs

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

CreateParagraphs(string?)

Creates one or more HTML paragraph elements from the provided text.

```
protected virtual object? CreateParagraphs(string? text)
```

Parameters

text [string](#)

The text to display in the paragraphs, or null for no paragraphs.

Returns

[object](#)

The HTML element representing the paragraphs, or null if no text is provided.

Method Ellipsis

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

Ellipsis(string?, int)

Truncates a string to the specified size, appending an ellipsis if necessary.

```
protected string? Ellipsis(string? s, int size)
```

Parameters

s [string](#)

The string to truncate, or null for no truncation.

size [int](#)

The maximum length of the string, including the ellipsis.

Returns

[string](#)

The truncated string, or the original string if it is already within the size limit, or null if the input string is null.

Method ExportDocumentationContent

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

ExportDocumentationContent(ApiDocumentation)

Exports the API documentation as HTTP content.

```
public IHttpContent ExportDocumentationContent(ApiDocumentation documentation)
```

Parameters

documentation [ApiDocumentation](#)

The API documentation to export.

Returns

[HttpContent](#) ↗

The exported API documentation as HTTP content.

Method ExportHtml

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

ExportHtml(ApiDocumentation)

Exports the API documentation as an HTML string.

```
public string ExportHtml(ApiDocumentation d)
```

Parameters

d [ApiDocumentation](#)

The API documentation to export.

Returns

[string](#) ↗

The exported HTML string.

Method GetRouteMethodHexColor

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

GetRouteMethodHexColor(RouteMethod)

Gets the hex color code associated with the specified route method.

```
protected virtual string GetRouteMethodHexColor(RouteMethod rm)
```

Parameters

rm [RouteMethod](#)

The route method to get the color for.

Returns

[string](#)

The hex color code as a string.

Method TransformId

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

TransformId(string)

Transforms an unsafe ID into a safe and valid HTML ID.

```
protected string TransformId(string unsafeId)
```

Parameters

unsafeId [string](#)

The ID to transform, which may contain invalid characters.

Returns

[string](#)

The transformed ID, which is safe for use in HTML.

Method WriteEndpointDescription

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

WriteEndpointDescription(ApiEndpoint)

Writes the description of an API endpoint.

```
protected virtual HtmlElement? WriteEndpointDescription(ApiEndpoint endpoint)
```

Parameters

endpoint [ApiEndpoint](#)

The API endpoint to write the description for.

Returns

[HtmlElement](#)

The HTML element representing the endpoint description, or null if no description is available.

Method WriteMainTitle

Namespace:[Sisk.Documenting.Html](#)

Assembly:Sisk.Documenting.Html.dll

WriteMainTitle(ApiDocumentation)

Writes the main title of the API documentation.

```
protected virtual HtmlElement? WriteMainTitle(ApiDocumentation documentation)
```

Parameters

documentation [ApiDocumentation](#)

The API documentation to write the title for.

Returns

[HtmlElement](#)

The HTML element representing the main title.

Class ApiDocumentation

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Represents the API documentation, including application details and endpoints.

```
public sealed class ApiDocumentation
```

Properties

[ApiVersion](#)

Gets or sets the version of the API.

[ApplicationDescription](#)

Gets or sets the description of the application.

[ApplicationName](#)

Gets or sets the name of the application.

[Endpoints](#)

Gets or sets the array of API endpoints.

Methods

[Generate\(Router, Apildentifier\)](#)

Generates an instance of [ApiDocumentation](#) by reading documentation from the specified router and identifier.

Property ApiVersion

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

ApiVersion

Gets or sets the version of the API.

```
public string? ApiVersion { get; }
```

Property Value

[string](#)

Property ApplicationDescription

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

ApplicationDescription

Gets or sets the description of the application.

```
public string? ApplicationDescription { get; }
```

Property Value

[string](#)

Property ApplicationName

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

ApplicationName

Gets or sets the name of the application.

```
public string? ApplicationName { get; }
```

Property Value

[string](#)

Property Endpoints

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Endpoints

Gets or sets the array of API endpoints.

```
public ApiEndpoint[] Endpoints { get; }
```

Property Value

[ApiEndpoint\[\]](#)

Method Generate

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Generate(Router, Apildentifier)

Generates an instance of [ApiDocumentation](#) by reading documentation from the specified router and identifier.

```
public static ApiDocumentation Generate(Router router, ApiIdentifier identifier)
```

Parameters

router Router

The router used to generate the documentation.

identifier Apildentifier

The identifier for the API documentation.

Returns

[ApiDocumentation](#)

An instance of [ApiDocumentation](#).

Class ApiEndpoint

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Represents an API endpoint, including its metadata, request and response details.

```
public sealed class ApiEndpoint
```

Properties

Description

Gets the description of the API endpoint.

Group

Gets the group to which the API endpoint belongs.

Headers

Gets the headers associated with the API endpoint.

Name

Gets the name of the API endpoint.

Parameters

Gets the parameters accepted by the API endpoint.

Path

Gets the path of the API endpoint.

PathParameters

Gets the path parameters for the API endpoint.

RequestExamples

Gets the parameters accepted by the API endpoint.

Responses

Gets the possible responses from the API endpoint.

RouteMethod

Gets the route method used for the API endpoint.

Property Description

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Description

Gets the description of the API endpoint.

```
public string? Description { get; }
```

Property Value

[string](#)

Property Group

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Group

Gets the group to which the API endpoint belongs.

```
public string? Group { get; }
```

Property Value

[string](#)

Property Headers

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Headers

Gets the headers associated with the API endpoint.

```
public ApiEndpointHeader[] Headers { get; }
```

Property Value

[ApiEndpointHeader](#)[]

Property Name

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Name

Gets the name of the API endpoint.

```
public string Name { get; }
```

Property Value

[string](#)

Property Parameters

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Parameters

Gets the parameters accepted by the API endpoint.

```
public ApiEndpointParameter[] Parameters { get; }
```

Property Value

[ApiEndpointParameter\[\]](#)

Property Path

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Path

Gets the path of the API endpoint.

```
public string Path { get; }
```

Property Value

[string](#)

Property PathParameters

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

PathParameters

Gets the path parameters for the API endpoint.

```
public ApiEndpointPathParameter[] PathParameters { get; }
```

Property Value

[ApiEndpointPathParameter\[\]](#)

Property RequestExamples

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

RequestExamples

Gets the parameters accepted by the API endpoint.

```
public ApiEndpointRequestExample[] RequestExamples { get; }
```

Property Value

[ApiEndpointRequestExample\[\]](#)

Property Responses

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Responses

Gets the possible responses from the API endpoint.

```
public ApiEndpointResponse[] Responses { get; }
```

Property Value

[ApiEndpointResponse\[\]](#)

Property RouteMethod

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

RouteMethod

Gets the route method used for the API endpoint.

```
public RouteMethod RouteMethod { get; }
```

Property Value

[RouteMethod](#)

Class ApiEndpointHeader

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Represents a header for an API endpoint, including its name and requirements.

```
public sealed class ApiEndpointHeader
```

Properties

Description

Gets or sets the description of the header.

HeaderName

Gets or sets the name of the header.

IsRequired

Gets or sets a value indicating whether the header is required.

Property Description

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Description

Gets or sets the description of the header.

```
public string? Description { get; }
```

Property Value

[string](#) ↗

Property HeaderName

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

HeaderName

Gets or sets the name of the header.

```
public string HeaderName { get; }
```

Property Value

[string](#)

Property IsRequired

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

IsRequired

Gets or sets a value indicating whether the header is required.

```
public bool IsRequired { get; }
```

Property Value

[bool](#)

Class ApiEndpointParameter

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Represents a parameter for an API endpoint, including its name, type, and requirements.

```
public sealed class ApiEndpointParameter
```

Properties

[Description](#)

Gets the description of the parameter.

[IsRequired](#)

Gets a value indicating whether the parameter is required.

[Name](#)

Gets the name of the parameter.

[TypeName](#)

Gets the type name of the parameter.

Property Description

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Description

Gets the description of the parameter.

```
public string? Description { get; }
```

Property Value

[string](#)

Property IsRequired

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

IsRequired

Gets a value indicating whether the parameter is required.

```
public bool IsRequired { get; }
```

Property Value

[bool](#)

Property Name

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Name

Gets the name of the parameter.

```
public string Name { get; }
```

Property Value

[string](#)

Property TypeName

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

TypeName

Gets the type name of the parameter.

```
public string TypeName { get; }
```

Property Value

[string](#)

Class ApiEndpointPathParameter

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Represents a path parameter for an API endpoint, including its name, type, and description.

```
public sealed class ApiEndpointPathParameter
```

Properties

Description

Gets the description of the path parameter.

Name

Gets the name of the path parameter.

Type

Gets the type of the path parameter.

Property Description

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Description

Gets the description of the path parameter.

```
public string? Description { get; }
```

Property Value

[string](#)

Property Name

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Name

Gets the name of the path parameter.

```
public string Name { get; }
```

Property Value

[string](#)

Property Type

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Type

Gets the type of the path parameter.

```
public string? Type { get; }
```

Property Value

[string](#)

Class ApiEndpointRequestExample

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Represents an example request for an API endpoint, including its description and example content.

```
public sealed class ApiEndpointRequestExample
```

Properties

Description

Gets the description of the request example.

Example

Gets the actual example request content.

ExampleLanguage

Gets the programming language used in the example, if applicable.

Property Description

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Description

Gets the description of the request example.

```
public string Description { get; }
```

Property Value

[string](#)

Property Example

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Example

Gets the actual example request content.

```
public string? Example { get; }
```

Property Value

[string](#)

Property ExampleLanguage

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

ExampleLanguage

Gets the programming language used in the example, if applicable.

```
public string? ExampleLanguage { get; }
```

Property Value

[string](#)

Class ApiEndpointResponse

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Represents a response for an API endpoint, including the status code and example content.

```
public sealed class ApiEndpointResponse
```

Properties

Description

Gets the description of the response.

Example

Gets the example response content.

ExampleLanguage

Gets the programming language used in the example, if applicable.

StatusCode

Gets the HTTP status code for the response.

Property Description

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Description

Gets the description of the response.

```
public string? Description { get; }
```

Property Value

[string](#)

Property Example

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Example

Gets the example response content.

```
public string? Example { get; }
```

Property Value

[string](#)

Property ExampleLanguage

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

ExampleLanguage

Gets the programming language used in the example, if applicable.

```
public string? ExampleLanguage { get; }
```

Property Value

[string](#)

Property StatusCode

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

StatusCode

Gets the HTTP status code for the response.

```
public HttpStatusCode StatusCode { get; }
```

Property Value

[HttpStatusCode](#)

Class Apildentifier

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Represents an identifier for an API, including application details such as name, version, and description.

```
public sealed class ApiIdentifier
```

Constructors

[Apildentifier\(\)](#)

Properties

[ApplicationDescription](#)

Gets or sets the description of the application.

[ApplicationName](#)

Gets or sets the name of the application.

[ApplicationVersion](#)

Gets or sets the version of the application.

Constructor Apildentifier

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Apildentifier()

```
public ApiIdentifier()
```

Property ApplicationDescription

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

ApplicationDescription

Gets or sets the description of the application.

```
public string? ApplicationDescription { get; set; }
```

Property Value

[string](#)

Property ApplicationName

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

ApplicationName

Gets or sets the name of the application.

```
public string? ApplicationName { get; set; }
```

Property Value

[string](#)

Property ApplicationVersion

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

ApplicationVersion

Gets or sets the version of the application.

```
public string? ApplicationVersion { get; set; }
```

Property Value

[string](#)

Interface IApiDocumentationExporter

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

Defines a contract for exporting API documentation content.

```
public interface IApiDocumentationExporter
```

Methods

[ExportDocumentationContent\(ApiDocumentation\)](#)

Exports the specified API documentation content.

Method ExportDocumentationContent

Namespace:[Sisk.Documenting](#)

Assembly:Sisk.Documenting.dll

ExportDocumentationContent(ApiDocumentation)

Exports the specified API documentation content.

```
HttpContent ExportDocumentationContent(ApiDocumentation documentation)
```

Parameters

documentation [ApiDocumentation](#)

The API documentation to export.

Returns

[HttpContent](#)

An [HttpContent](#) representing the exported documentation.

Namespace Sisk.Helpers

Namespaces

[Sisk.Helpers.Mitmproxy](#)

[Sisk.Helpers.mitmproxy](#)

Namespace Sisk.Helpers.Mitmproxy

Classes

[MitmproxyProvider](#)

Provides a MITM proxy server handler.

Class MitmproxyProvider

Namespace:[Sisk.Helpers.Mitmproxy](#)

Assembly:Sisk.Helpers.mitmproxy.dll

Provides a MITM proxy server handler.

```
public sealed class MitmproxyProvider : HttpServerHandler
```

Constructors

[MitmproxyProvider\(\)](#)

Initializes a new instance of the [MitmproxyProvider](#) class.

[MitmproxyProvider\(ushort, Action<ChildProcessStartInfo>?\)](#)

Initializes a new instance of the [MitmproxyProvider](#) class with a specified proxy port and optional process setup action.

Properties

[MitmdumpProcess](#)

Gets the mitmdump process.

[ProxyPort](#)

Gets or sets the proxy port.

[Silent](#)

Gets or sets a value indicating whether to run the mitmdump process silently.

Methods

[OnServerStarted\(HttpServer\)](#)

Event that is called immediately after starting the [HttpServer](#), when it's ready and listening.

[OnServerStarting\(HttpServer\)](#)

Event that is called immediately before starting the [HttpServer](#).

[OnServerStopped\(HttpServer\)](#)

Event that is called after the [HttpServer](#) is stopped, meaning it has stopped from listening to requests.

[OnServerStopping\(HttpServer\)](#)

Event that is called before the [HttpServer](#) stop, when it is stopping from listening requests.

Constructor MitmproxyProvider

Namespace:[Sisk.Helpers.Mitmproxy](#)

Assembly:Sisk.Helpers.mitmproxy.dll

MitmproxyProvider()

Initializes a new instance of the [MitmproxyProvider](#) class.

```
public MitmproxyProvider()
```

MitmproxyProvider(ushort, Action<ChildProcessStartInfo>?)

Initializes a new instance of the [MitmproxyProvider](#) class with a specified proxy port and optional process setup action.

```
public MitmproxyProvider(ushort proxyPort, Action<ChildProcessStartInfo>? processSetupAction  
= null)
```

Parameters

proxyPort [ushort](#)

The port on which the mitmproxy will listen.

processSetupAction [Action](#)<ChildProcessStartInfo>

Optional. An action to configure the child process start information.

Property MitmdumpProcess

Namespace:[Sisk.Helpers.Mitmproxy](#)

Assembly:Sisk.Helpers.mitmproxy.dll

MitmdumpProcess

Gets the mitmdump process.

```
public IChildProcess MitmdumpProcess { get; }
```

Property Value

IChildProcess

Property ProxyPort

Namespace:[Sisk.Helpers.Mitmproxy](#)

Assembly:Sisk.Helpers.mitmproxy.dll

ProxyPort

Gets or sets the proxy port.

```
public ushort ProxyPort { get; }
```

Property Value

[ushort](#)

Property Silent

Namespace:[Sisk.Helpers.Mitmproxy](#)

Assembly:Sisk.Helpers.mitmproxy.dll

Silent

Gets or sets a value indicating whether to run the mitmdump process silently.

```
public bool Silent { get; set; }
```

Property Value

[bool](#)

Method OnServerStarted

Namespace:[Sisk.Helpers.Mitmproxy](#)

Assembly:Sisk.Helpers.mitmproxy.dll

OnServerStarted(HttpServer)

Event that is called immediately after starting the [HttpServer](#), when it's ready and listening.

```
protected override void OnServerStarted(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which is ready.

Method OnServerStarting

Namespace:[Sisk.Helpers.Mitmproxy](#)

Assembly:Sisk.Helpers.mitmproxy.dll

OnServerStarting(HttpServer)

Event that is called immediately before starting the [HttpServer](#).

```
protected override void OnServerStarting(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which is starting.

Method OnServerStopped

Namespace:[Sisk.Helpers.Mitmproxy](#)

Assembly:Sisk.Helpers.mitmproxy.dll

OnServerStopped(HttpServer)

Event that is called after the [HttpServer](#) is stopped, meaning it has stopped from listening to requests.

```
protected override void OnServerStopped(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which has stopped.

Method OnServerStopping

Namespace:[Sisk.Helpers.Mitmproxy](#)

Assembly:Sisk.Helpers.mitmproxy.dll

OnServerStopping(HttpServer)

Event that is called before the [HttpServer](#) stop, when it is stopping from listening requests.

```
protected override void OnServerStopping(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which is stopping.

Namespace Sisk.Helpers.mitmproxy

Classes

[MitmproxyHelper](#)

Provides extension methods for configuring an HTTP server to use mitmproxy.

Class MitmproxyHelper

Namespace:[Sisk.Helpers.mitmproxy](#)

Assembly:Sisk.Helpers.mitmproxy.dll

Provides extension methods for configuring an HTTP server to use mitmproxy.

```
public static class MitmproxyHelper
```

Methods

[UseMitmproxy\(HttpServerHostContextBuilder\)](#)

Configures the specified [HttpServerHostContextBuilder](#) to use mitmproxy with a random proxy port.

[UseMitmproxy\(HttpServerHostContextBuilder, ushort, bool, Action<ChildProcessStartInfo>?\)](#)

Configures the specified [HttpServerHostContextBuilder](#) to use mitmproxy with the specified options.

Method UseMitmproxy

Namespace:[Sisk.Helpers.mitmproxy](#)

Assembly:Sisk.Helpers.mitmproxy.dll

UseMitmproxy([HttpServerHostContextBuilder](#))

Configures the specified [HttpServerHostContextBuilder](#) to use mitmproxy with a random proxy port.

```
public static HttpServerHostContextBuilder UseMitmproxy(this HttpServerHostContextBuilder builder)
```

Parameters

builder [HttpServerHostContextBuilder](#)

The [HttpServerHostContextBuilder](#) instance to configure.

Returns

[HttpServerHostContextBuilder](#)

The updated [HttpServerHostContextBuilder](#) instance.

UseMitmproxy([HttpServerHostContextBuilder](#), ushort, bool, Action<ChildProcessStartInfo>?)

Configures the specified [HttpServerHostContextBuilder](#) to use mitmproxy with the specified options.

```
public static HttpServerHostContextBuilder UseMitmproxy(this HttpServerHostContextBuilder builder, ushort proxyPort = 0, bool silent = false, Action<ChildProcessStartInfo>? setupAction = null)
```

Parameters

builder [HttpServerHostContextBuilder](#)

The [HttpServerHostContextBuilder](#) instance to configure.

proxyPort `ushort`

The port on which the mitmproxy will listen.

silent `bool`

Indicates whether the mitmproxy should run in silent mode. Default is false.

setupAction `Action<ChildProcessStartInfo>`

An optional action to configure the child process start information.

Returns

[HttpServerHostContextBuilder](#)

The updated [HttpServerHostContextBuilder](#) instance.

Namespace Sisk.IniConfiguration

Namespaces

[Sisk.IniConfiguration.Core](#)

Classes

[IniConfigurationReader](#)

Provides an INI-Document based configuration-reader pipeline.

Namespace Sisk.IniConfiguration.Core

Namespaces

[Sisk.IniConfiguration.Core.Serialization](#)

Classes

[IniDocument](#)

Represents an INI document.

[IniSection](#)

Represents an INI section, which contains it's own properties.

[IniSectionCollection](#)

Represents an collection of [IniSection](#).

Namespace Sisk.IniConfiguration.Core.Serialization

Classes

[IniReader](#)

Provides an INI-document reader and parser.

[IniWriter](#)

Represents a writer for INI files.

Enums

[IniWriting.NewLineBehavior](#)

Specifies the behavior for writing new lines in an INI file.

Class IniReader

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

Provides an INI-document reader and parser.

```
public sealed class IniReader : IDisposable
```

Implements

[IDisposable](#)

Constructors

[IniReader\(TextReader\)](#)

Creates an new [IniReader](#) with the specified text reader.

Properties

[IniNamingComparer](#)

Gets or sets the default [StringComparer](#) used by the INI reader and instances to compare key names.

[Reader](#)

Gets the [TextReader](#) which is providing data to this INI reader.

Methods

[Dispose\(\)](#)

[Read\(\)](#)

Reads the INI document from the input stream.

Constructor IniReader

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

IniReader(TextReader)

Creates an new [IniReader](#) with the specified text reader.

```
public IniReader(TextReader reader)
```

Parameters

reader [TextReader](#)

The [TextReader](#) instace to read the INI document.

Property IniNamingComparer

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

IniNamingComparer

Gets or sets the default [StringComparer](#) used by the INI reader and instances to compare key names.

```
public static StringComparer IniNamingComparer { get; set; }
```

Property Value

[StringComparer](#)

Property Reader

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

Reader

Gets the [TextReader](#) which is providing data to this INI reader.

```
public TextReader Reader { get; }
```

Property Value

[TextReader](#)

Method Dispose

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

Dispose()

```
public void Dispose()
```

Method Read

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

Read()

Reads the INI document from the input stream.

```
public IniDocument Read()
```

Returns

[IniDocument](#)

An [IniDocument](#) file containing all properties and data from the input stream.

Class IniWriter

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

Represents a writer for INI files.

```
public sealed class IniWriter : IDisposable
```

Implements

[IDisposable](#) ↗

Constructors

[IniWriter\(TextWriter\)](#)

Initializes a new instance of the [IniWriter](#) class.

Properties

[CommentChar](#)

Gets or sets the default comment character.

[NewLineBehavior](#)

Gets or sets the behavior for writing new lines inside properties values.

[Writer](#)

Gets the underlying text writer.

Methods

[Dispose\(\)](#)

Releases all resources used by the [IniWriter](#) object.

[**Write\(IniDocument\)**](#)

Writes an INI document to the INI file.

[**Write\(IniSection\)**](#)

Writes an INI section to the INI file.

[**Write\(in KeyValuePair<string, string\[\]>\)**](#)

Writes a key-value pair to the INI file, where the value is an array of strings.

[**Write\(string, string?\)**](#)

Writes a key-value pair to the INI file.

[**WriteComment\(string\)**](#)

Writes a comment to the INI file.

Constructor IniWriter

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

IniWriter(TextWriter)

Initializes a new instance of the [IniWriter](#) class.

```
public IniWriter(TextWriter writer)
```

Parameters

writer [TextWriter](#)

The underlying text writer.

Property CommentChar

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

CommentChar

Gets or sets the default comment character.

```
public char CommentChar { get; set; }
```

Property Value

char ↗

Property NewLineBehavior

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

NewLineBehavior

Gets or sets the behavior for writing new lines inside properties values.

```
public IniWriting.NewLineBehavior NewLineBehavior { get; set; }
```

Property Value

[IniWriting.NewLineBehavior](#)

Property Writer

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

Writer

Gets the underlying text writer.

```
public TextWriter Writer { get; }
```

Property Value

[TextWriter](#)

Method Dispose

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

Dispose()

Releases all resources used by the [IniWriter](#) object.

```
public void Dispose()
```

Method Write

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

Write(string, string?)

Writes a key-value pair to the INI file.

```
public void Write(string key, string? value)
```

Parameters

key `string`

The key to write.

value `string`

The value to write.

Write(in KeyValuePair<string, string[]>)

Writes a key-value pair to the INI file, where the value is an array of strings.

```
public void Write(in KeyValuePair<string, string[]> value)
```

Parameters

value `KeyValuePair<string, string[]>`

The key-value pair to write.

Write(IniSection)

Writes an INI section to the INI file.

```
public void Write(IniSection section)
```

Parameters

section `IniSection`

The section to write.

Write(IniDocument)

Writes an INI document to the INI file.

```
public void Write(IniDocument document)
```

Parameters

document `IniDocument`

The document to write.

Method WriteComment

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

WriteComment(string)

Writes a comment to the INI file.

```
public void WriteComment(string commentString)
```

Parameters

commentString [string](#)

The comment to write.

Enum IniWriting.NewLineBehavior

Namespace:[Sisk.IniConfiguration.Core.Serialization](#)

Assembly:Sisk.IniConfiguration.Core.dll

Specifies the behavior for writing new lines in an INI file.

[Flags]

```
public enum IniWriting.NewLineBehavior
```

Fields

Escape = 4

Escapes the new line characters when writing new lines.

Quote = 1

Quotes the value when writing new lines.

Split = 2

Splits the value into multiple lines when writing new lines.

Class IniDocument

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:[Sisk.IniConfiguration.Core.dll](#)

Represents an INI document.

```
public sealed class IniDocument
```

Constructors

[IniDocument\(\)](#)

Creates an new empty [IniDocument](#) instance with no INI sections added to it.

[IniDocument\(IEnumerable<IniSection>\)](#)

Creates an new [IniDocument](#) instance from the specified [IniSection](#) collection.

Properties

[Global](#)

Gets the global INI section, which is the primary section in the document.

[Sections](#)

Gets all INI sections defined in this INI document.

Methods

[FromFile\(string, Encoding?, bool\)](#)

Creates an new [IniDocument](#) document from the specified file using the specified encoding.

[FromStream\(Stream, Encoding?\)](#)

Creates an new [IniDocument](#) document from the specified stream using the specified encoding.

[FromStream\(TextReader\)](#)

Creates an new [IniDocument](#) document from the specified [TextReader](#).

[FromString\(string\)](#)

Creates an new [IniDocument](#) document from the specified string, reading it as an UTF-8 string.

[GetEntries\(\)](#)

Retrieves all entries in the INI document.

[GetEntry\(string, StringComparison\)](#)

Retrieves the values of a specific entry in the INI document.

[GetSection\(string\)](#)

Gets an defined INI section from this document. The search is case-insensitive.

[ToString\(\)](#)

Gets the INI document string from this [IniDocument](#).

Constructor IniDocument

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

IniDocument(IEnumerable<IniSection>)

Creates an new [IniDocument](#) instance from the specified [IniSection](#) collection.

```
public IniDocument(IEnumerable<IniSection> sections)
```

Parameters

sections [IEnumerable](#)<[IniSection](#)>

The list of [IniSection](#).

IniDocument()

Creates an new empty [IniDocument](#) instance with no INI sections added to it.

```
public IniDocument()
```

Property Global

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Global

Gets the global INI section, which is the primary section in the document.

```
public IniSection Global { get; }
```

Property Value

[IniSection](#)

Property Sections

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Sections

Gets all INI sections defined in this INI document.

```
public IniSectionCollection Sections { get; }
```

Property Value

[IniSectionCollection](#)

Method FromFile

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

FromFile(string, Encoding?, bool)

Creates an new [IniDocument](#) document from the specified file using the specified encoding.

```
public static IniDocument FromFile(string filePath, Encoding? encoding = null, bool throwIfNotExists = true)
```

Parameters

filePath [string](#)

The absolute or relative file path to the INI document.

encoding [Encoding](#)

Optional. The encoding used to read the file. Defaults to UTF-8.

throwIfNotExists [bool](#)

Optional. Defines whether this method should throw if the specified file doesn't exists or return an empty INI document.

Returns

[IniDocument](#)

Method FromStream

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

FromStream(Stream, Encoding?)

Creates an new [IniDocument](#) document from the specified stream using the specified encoding.

```
public static IniDocument FromStream(Stream stream, Encoding? encoding = null)
```

Parameters

stream [Stream](#)

The input stream where the INI document is.

encoding [Encoding](#)

Optional. The encoding used to read the stream. Defaults to UTF-8.

Returns

[IniDocument](#)

FromStream(TextReader)

Creates an new [IniDocument](#) document from the specified [TextReader](#).

```
public static IniDocument FromStream(TextReader reader)
```

Parameters

reader [TextReader](#)

The [TextReader](#) instance.

Returns

IniDocument

Method FromString

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

FromString(string)

Creates an new [IniDocument](#) document from the specified string, reading it as an UTF-8 string.

```
public static IniDocument FromString(string iniConfiguration)
```

Parameters

iniConfiguration [string](#)

The UTF-8 string.

Returns

[IniDocument](#)

Method GetEntries

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

GetEntries()

Retrieves all entries in the INI document.

```
public IEnumerable<KeyValuePair<string, string[]>> GetEntries()
```

Returns

[IEnumerable](#)<[KeyValuePair](#)<string, string[]>>

An enumerable collection of key-value pairs, where each key is the entry name and each value is an array of strings representing the entry values.

Method GetEntry

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

GetEntry(string, StringComparison)

Retrieves the values of a specific entry in the INI document.

```
public string[] GetEntry(string name, StringComparison stringComparison =  
StringComparison.OrdinalIgnoreCase)
```

Parameters

name [string](#)

The name of the entry to retrieve.

stringComparison [StringComparison](#)

The string comparison to use when searching for the entry. Defaults to [OrdinalIgnoreCase](#).

Returns

[string](#)[]

An array of strings representing the values of the entry, or an empty array if the entry is not found.

Method GetSection

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

GetSection(string)

Gets an defined INI section from this document. The search is case-insensitive.

```
public IniSection? GetSection(string sectionName)
```

Parameters

sectionName [string](#)

The section name.

Returns

[IniSection](#)

The [IniSection](#) object if found, or null if not defined.

Method ToString

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

ToString()

Gets the INI document string from this [IniDocument](#).

```
public override string ToString()
```

Returns

[string](#)

Class IniSection

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Represents an INI section, which contains it's own properties.

```
public sealed class IniSection : IDictionary<string, string[]>, ICollection<KeyValuePair<string, string[]>>, IEnumerable<KeyValuePair<string, string[]>>, IEnumerable, IEquatable<IniSection>
```

Implements

[IDictionary](#)<string[], string[]>, [ICollection](#)<KeyValuePair<string[], string[]>>,
[IEnumerable](#)<KeyValuePair<string[], string[]>>, [IEnumerable](#), [IEquatable](#)<IniSection>

Constructors

[IniSection\(string\)](#)

Initializes a new instance of the [IniSection](#) class with the specified name.

[IniSection\(string, IEnumerable<KeyValuePair<string, string>>\)](#)

Initializes a new instance of the [IniSection](#) class with the specified name and items.

Properties

[Count](#)

Gets the number of properties in this INI section.

[IsReadOnly](#)

[this\[string\]](#)

Gets all values associated with the specified property name, performing an case-insensitive search.

[Keys](#)

Gets all keys defined in this INI section, without duplicates.

Name

Gets the INI section name.

Values

Gets all values defined in this INI section.

Methods

[Add\(KeyValuePair<string, string\[\]>\)](#)

[Add\(string, string?\)](#)

Adds a new key-value pair to the INI section.

[Add\(string, string\[\]\)](#)

[Clear\(\)](#)

[Contains\(KeyValuePair<string, string\[\]>\)](#)

[ContainsKey\(string\)](#)

Gets an boolean indicating if the specified key/property name is defined in this [IniSection](#).

[CopyTo\(KeyValuePair<string, string\[\]>\[\], int\)](#)

This method is not supported and will throw an [NotSupportedException](#).

[Equals\(IniSection?\)](#)

[Equals\(object?\)](#)

[GetEnumerator\(\)](#)

[GetHashCode\(\)](#)

[GetMany\(string\)](#)

Gets all values defined in this INI section by their property name.

[GetOne\(string\)](#)

Gets the last value defined in this INI section by their property name.

[Remove\(KeyValuePair<string, string\[\]>\)](#)

[Remove\(string\)](#)

[TryGetValue\(string, out string\[\]\)](#)

Constructor IniSection

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

IniSection(string)

Initializes a new instance of the [IniSection](#) class with the specified name.

```
public IniSection(string name)
```

Parameters

name [string](#)

The name of the INI section.

IniSection(string, IEnumerable<KeyValuePair<string, string>>)

Initializes a new instance of the [IniSection](#) class with the specified name and items.

```
public IniSection(string name, IEnumerable<KeyValuePair<string, string>> items)
```

Parameters

name [string](#)

The name of the INI section.

items [IEnumerable<KeyValuePair<string, string>>](#)

A collection of key-value pairs to be added to the section.

Property Count

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Count

Gets the number of properties in this INI section.

```
public int Count { get; }
```

Property Value

[int](#)

Property IsReadOnly

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

IsReadOnly

```
public bool IsReadOnly { get; }
```

Property Value

[bool](#)

Property Keys

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Keys

Gets all keys defined in this INI section, without duplicates.

```
public ICollection<string> Keys { get; }
```

Property Value

[ICollection](#)<[string](#)>

Property Name

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Name

Gets the INI section name.

```
public string Name { get; }
```

Property Value

[string](#)

Property Values

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Values

Gets all values defined in this INI section.

```
public ICollection<string[]> Values { get; }
```

Property Value

[ICollection](#)<[string](#)[]>

Property this

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

this[string]

Gets all values associated with the specified property name, performing an case-insensitive search.

```
public string[] this[string key] { get; set; }
```

Parameters

key [string](#)

The property name.

Property Value

[string](#)

Method Add

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Add(string, string[])

```
public void Add(string key, string[] value)
```

Parameters

key [string](#)

value [string\[\]](#)

Add(string, string?)

Adds a new key-value pair to the INI section.

```
public void Add(string key, string? value)
```

Parameters

key [string](#)

The key to be added.

value [string](#)

The value associated with the key, or `null` to set an empty value.

Add(KeyValuePair<string, string[]>)

```
public void Add(KeyValuePair<string, string[]> item)
```

Parameters

item `KeyValuePair<string, string[]>`

Method Clear

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Clear()

```
public void Clear()
```

Method Contains

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Contains(KeyValuePair<string, string[]>)

```
public bool Contains(KeyValuePair<string, string[]> item)
```

Parameters

item [KeyValuePair<string, string\[\]>](#)

Returns

[bool](#)

Method ContainsKey

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

ContainsKey(string)

Gets an boolean indicating if the specified key/property name is defined in this [IniSection](#).

```
public bool ContainsKey(string key)
```

Parameters

key [string](#)

The property name.

Returns

[bool](#)

An [bool](#) indicating if the specified property name is defined or not.

Method CopyTo

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

CopyTo(KeyValuePair<string, string>[][], int)

This method is not supported and will throw an [NotSupportedException](#).

```
public void CopyTo(KeyValuePair<string, string>[][], int arrayIndex)
```

Parameters

array KeyValuePair<string, string>[][],

arrayIndex int

Method Equals

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Equals(object?)

```
public override bool Equals(object? obj)
```

Parameters

obj object

Returns

bool

Equals(IniSection?)

```
public bool Equals(IniSection? other)
```

Parameters

other IniSection

Returns

bool

Method GetEnumerator

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

GetEnumerator()

```
public IEnumarator<KeyValuePair<string, string[]>> GetEnumerator()
```

Returns

[IEnumarator](#)<[KeyValuePair](#)<string, string[]>>

Method GetHashCode

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

GetHashCode()

```
public override int GetHashCode()
```

Returns

[int](#)

Method GetMany

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

GetMany(string)

Gets all values defined in this INI section by their property name.

```
public string[] GetMany(string key)
```

Parameters

key [string](#)

The property name.

Returns

[string](#)

All values associated with the specified property name.

Method GetOne

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

GetOne(string)

Gets the last value defined in this INI section by their property name.

```
public string? GetOne(string key)
```

Parameters

key [string](#) ↗

The property name.

Returns

[string](#) ↗

The last value associated with the specified property name, or null if nothing is found.

Method Remove

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Remove(string)

```
public bool Remove(string key)
```

Parameters

key [string](#)

Returns

[bool](#)

Remove(KeyValuePair<string, string[]>)

```
public bool Remove(KeyValuePair<string, string[]> item)
```

Parameters

item [KeyValuePair](#)<string, string[]>

Returns

[bool](#)

Method TryGetValue

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

TryGetValue(string, out string[])

```
public bool TryGetValue(string key, out string[] value)
```

Parameters

key [string](#)

value [string\[\]](#)

Returns

[bool](#)

Class IniSectionCollection

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:[Sisk.IniConfiguration.Core.dll](#)

Represents an collection of [IniSection](#).

```
public sealed class IniSectionCollection : IList<IniSection>, ICollection<IniSection>,
IReadOnlyList<IniSection>, IReadOnlyCollection<IniSection>, IEnumerable<IniSection>,
ICollection, IEnumerable
```

Implements

[IList](#)<IniSection>, [ICollection](#)<IniSection>, [IReadOnlyList](#)<IniSection>, [IReadOnlyCollection](#)<IniSection>,
[IEnumerable](#)<IniSection>, [ICollection](#), [IEnumerable](#)

Properties

[Count](#)

[IsReadOnly](#)

[this\[int\]](#)

Methods

[Add\(IniSection\)](#)

[Clear\(\)](#)

[Contains\(IniSection\)](#)

[CopyTo\(IniSection\[\], int\)](#)

[GetEnumerator\(\)](#)

[GetGlobal\(\)](#)

Gets the global [IniSection](#) in this collection or creates a new one if it doesn't exists.

[IndexOf\(IniSection\)](#)

[Insert\(int, IniSection\)](#)

[Remove\(IniSection\)](#)

[RemoveAt\(int\)](#)

Property Count

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Count

```
public int Count { get; }
```

Property Value

[int](#)

Property IsReadOnly

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

IsReadOnly

```
public bool IsReadOnly { get; }
```

Property Value

[bool](#)

Property this

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

this[int]

```
public IniSection this[int index] { get; set; }
```

Parameters

index int ↗

Property Value

[IniSection](#)

Method Add

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Add(IniSection)

```
public void Add(IniSection item)
```

Parameters

item [IniSection](#)

Method Clear

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Clear()

```
public void Clear()
```

Method Contains

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Contains(IniSection)

```
public bool Contains(IniSection item)
```

Parameters

item [IniSection](#)

Returns

[bool](#)

Method CopyTo

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

CopyTo(IniSection[], int)

```
public void CopyTo(IniSection[] array, int arrayIndex)
```

Parameters

array [IniSection\[\]](#)

arrayIndex [int](#)

Method GetEnumerator

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

GetEnumerator()

```
public IEnumarator<IniSection> GetEnumerator()
```

Returns

[IEnumarator](#)<IniSection>

Method GetGlobal

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

GetGlobal()

Gets the global [IniSection](#) in this collection or creates a new one if it doesn't exists.

```
public IniSection GetGlobal()
```

Returns

[IniSection](#)

The global [IniSection](#).

Method IndexOf

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

IndexOf(IniSection)

```
public int IndexOf(IniSection item)
```

Parameters

item [IniSection](#)

Returns

[int](#)

Method Insert

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Insert(int, IniSection)

```
public void Insert(int index, IniSection item)
```

Parameters

index int ↗

item IniSection

Method Remove

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

Remove(IniSection)

```
public bool Remove(IniSection item)
```

Parameters

item [IniSection](#)

Returns

[bool](#)

Method RemoveAt

Namespace:[Sisk.IniConfiguration.Core](#)

Assembly:Sisk.IniConfiguration.Core.dll

RemoveAt(int)

```
public void RemoveAt(int index)
```

Parameters

index int ↗

Class IniConfigurationReader

Namespace:[Sisk.IniConfiguration](#)

Assembly:Sisk.IniConfiguration.dll

Provides an INI-Document based configuration-reader pipeline.

```
public sealed class IniConfigurationReader : IConfigurationReader
```

Implements

[IConfigurationReader](#)

Constructors

[IniConfigurationReader\(\)](#)

Methods

[ReadConfiguration\(ConfigurationContext\)](#)

Represents the method that reads and applies settings from a settings file.

Constructor IniConfigurationReader

Namespace:[Sisk.IniConfiguration](#)

Assembly:Sisk.IniConfiguration.dll

IniConfigurationReader()

```
public IniConfigurationReader()
```

Method ReadConfiguration

Namespace:[Sisk.IniConfiguration](#)

Assembly:Sisk.IniConfiguration.dll

ReadConfiguration(ConfigurationContext)

Represents the method that reads and applies settings from a settings file.

```
public void ReadConfiguration(ConfigurationContext context)
```

Parameters

context [ConfigurationContext](#)

The configuration context object.

Namespace Sisk.JsonRPC

Namespaces

[Sisk.JsonRPC.Annotations](#)

[Sisk.JsonRPC.Documentation](#)

Classes

[JsonRpcException](#)

Represents an error that occur during the JSON-RPC application execution.

[JsonRpcHandler](#)

Represents a handler for JSON-RPC requests.

[JsonRpcMethodCollection](#)

Represents a collection of JSON-RPC methods, allowing for dynamic addition and removal of methods.

[JsonRpcRequest](#)

Represents an JSON-RPC request message.

[JsonRpcResponse](#)

Represents an JSON-RPC response message.

[JsonRpcServerConfigurationEventArgs](#)

Represents the class which contains event data for the JSON-RPC configuration event.

[JsonRpcServerHandler](#)

Provides an [HttpServerHandler](#) for configuring the JSON-RPC handler for the HTTP server.

[JsonRpcServerHandlerExtensions](#)

Provides extensions methods for the HTTP server and JSON-RPC handler.

[JsonRpcTransportLayer](#)

Provides transport layers for handling JSON-RPC communications.

Structs

[JsonRpcError](#)

Represents an JSON-RPC error.

Namespace Sisk.JsonRPC.Annotations

Classes

[MethodDescriptionAttribute](#)

Specifies a description for a method.

[ParamDescriptionAttribute](#)

Specifies a description for a method parameter.

[WebMethodAttribute](#)

Represents an JSON-RPC method.

[WebNameAttribute](#)

Represents an attribute which holds the class name for a group of JSON-RPC methods.

Class MethodDescriptionAttribute

Namespace:[Sisk.JsonRPC.Annotations](#)

Assembly:Sisk.JsonRPC.dll

Specifies a description for a method.

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = false)]
public sealed class MethodDescriptionAttribute : Attribute
```

Constructors

[MethodDescriptionAttribute\(string\)](#)

Initializes a new instance of the [MethodDescriptionAttribute](#) class with the specified description.

Properties

[Category](#)

Optional. Gets or sets the web method category.

[Description](#)

Gets the description of the method.

Constructor MethodDescriptionAttribute

Namespace:[Sisk.JsonRPC.Annotations](#)

Assembly:Sisk.JsonRPC.dll

MethodDescriptionAttribute(string)

Initializes a new instance of the [MethodDescriptionAttribute](#) class with the specified description.

```
public MethodDescriptionAttribute(string description)
```

Parameters

description string ↗

The description of the method.

Property Category

Namespace:[Sisk.JsonRPC.Annotations](#)

Assembly:Sisk.JsonRPC.dll

Category

Optional. Gets or sets the web method category.

```
public string? Category { get; set; }
```

Property Value

[string](#)

Property Description

Namespace:[Sisk.JsonRPC.Annotations](#)

Assembly:Sisk.JsonRPC.dll

Description

Gets the description of the method.

```
public string Description { get; }
```

Property Value

[string](#)

Class ParamDescriptionAttribute

Namespace:[Sisk.JsonRPC.Annotations](#)

Assembly:Sisk.JsonRPC.dll

Specifies a description for a method parameter.

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = true)]
public sealed class ParamDescriptionAttribute : Attribute
```

Constructors

[ParamDescriptionAttribute\(string, string\)](#)

Initializes a new instance of the [ParamDescriptionAttribute](#) class with the specified description.

Properties

[Description](#)

Gets the description of the method parameter.

[ParameterName](#)

Gets the target parameter name.

Constructor ParamDescriptionAttribute

Namespace:[Sisk.JsonRPC.Annotations](#)

Assembly:Sisk.JsonRPC.dll

ParamDescriptionAttribute(string, string)

Initializes a new instance of the [ParamDescriptionAttribute](#) class with the specified description.

```
public ParamDescriptionAttribute(string paramName, string description)
```

Parameters

paramName [string](#)

The parameter name.

description [string](#)

The description of the method parameter.

Property Description

Namespace:[Sisk.JsonRPC.Annotations](#)

Assembly:Sisk.JsonRPC.dll

Description

Gets the description of the method parameter.

```
public string Description { get; }
```

Property Value

[string](#)

Property ParameterName

Namespace:[Sisk.JsonRPC.Annotations](#)

Assembly:Sisk.JsonRPC.dll

ParameterName

Gets the target parameter name.

```
public string ParameterName { get; }
```

Property Value

[string](#)

Class WebMethodAttribute

Namespace:[Sisk.JsonRPC.Annotations](#)

Assembly:Sisk.JsonRPC.dll

Represents an JSON-RPC method.

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = false, Inherited = false)]
public sealed class WebMethodAttribute : Attribute
```

Constructors

[WebMethodAttribute\(\)](#)

Creates an new [WebMethodAttribute](#) with no parameters.

[WebMethodAttribute\(string\)](#)

Creates an new [WebMethodAttribute](#) with given parameters.

Properties

[Name](#)

Gets or sets the method name.

Constructor WebMethodAttribute

Namespace:[Sisk.JsonRPC.Annotations](#)

Assembly:Sisk.JsonRPC.dll

WebMethodAttribute()

Creates an new [WebMethodAttribute](#) with no parameters.

```
public WebMethodAttribute()
```

WebMethodAttribute(string)

Creates an new [WebMethodAttribute](#) with given parameters.

```
public WebMethodAttribute(string methodName)
```

Parameters

methodName [string](#)

The method name.

Property Name

Namespace:[Sisk.JsonRPC.Annotations](#)

Assembly:Sisk.JsonRPC.dll

Name

Gets or sets the method name.

```
public string? Name { get; set; }
```

Property Value

[string](#)

Class WebNameAttribute

Namespace:[Sisk.JsonRPC.Annotations](#)

Assembly:Sisk.JsonRPC.dll

Represents an attribute which holds the class name for a group of JSON-RPC methods.

```
[AttributeUsage(AttributeTargets.Class)]
public sealed class WebNameAttribute : Attribute
```

Constructors

[WebNameAttribute\(string\)](#)

Creates an new instance of the [WebMethodAttribute](#) attribute.

Properties

[Name](#)

Gets or sets the name associated with the method group.

Constructor WebNameAttribute

Namespace:[Sisk.JsonRPC.Annotations](#)

Assembly:Sisk.JsonRPC.dll

WebNameAttribute(string)

Creates an new instance of the [WebMethodAttribute](#) attribute.

```
public WebNameAttribute(string name)
```

Parameters

name [string](#)

The method-group name.

Property Name

Namespace:[Sisk.JsonRPC.Annotations](#)

Assembly:Sisk.JsonRPC.dll

Name

Gets or sets the name associated with the method group.

```
public string Name { get; set; }
```

Property Value

[string](#)

Namespace Sisk.JsonRPC.Documentation

Classes

[JsonRpcDocumentation](#)

Represents the documentation for JSON-RPC methods.

[JsonRpcDocumentationMetadata](#)

Represents the documentation metadata for JSON-RPC documentation.

[JsonRpcDocumentationMethod](#)

Represents the documentation for a single JSON-RPC method.

[JsonRpcDocumentationParameter](#)

Represents the documentation for a parameter of a JSON-RPC method.

[JsonRpcHtmlExport](#)

Provides an HTML-based [IJsonRpcDocumentationExporter](#).

[JsonRpcJsonExport](#)

Provides an JSON-based [IJsonRpcDocumentationExporter](#).

Interfaces

[IJsonRpcDocumentationExporter](#)

Defines a method to export JSON-RPC documentation to a byte array.

Interface IJsonRpcDocumentationExporter

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Defines a method to export JSON-RPC documentation to a byte array.

```
public interface IJsonRpcDocumentationExporter
```

Methods

[ExportDocumentBytes\(JsonRpcDocumentation\)](#)

Exports the JSON-RPC documentation to a byte array.

Method ExportDocumentBytes

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

ExportDocumentBytes(JsonRpcDocumentation)

Exports the JSON-RPC documentation to a byte array.

```
byte[ ] ExportDocumentBytes(JsonRpcDocumentation documentation)
```

Parameters

documentation [JsonRpcDocumentation](#)

The JSON-RPC documentation to export.

Returns

[byte](#)[]

A byte array containing the exported documentation.

Class JsonRpcDocumentation

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Represents the documentation for JSON-RPC methods.

```
public sealed class JsonRpcDocumentation
```

Properties

[Metadata](#)

Gets the used [JsonRpcDocumentationMetadata](#) for this [JsonRpcDocumentation](#).

[Methods](#)

Gets the collection of JSON-RPC methods.

Methods

[Export\(IJsonRpcDocumentationExporter\)](#)

Exports this [JsonRpcDocumentation](#) with the specified [IJsonRpcDocumentationExporter](#).

[ExportToJson\(\)](#)

Gets an JSON string representation of this [JsonRpcDocumentation](#).

[ExportToJson\(JsonOptions\)](#)

Gets an JSON string representation of this [JsonRpcDocumentation](#).

Property Metadata

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Metadata

Gets the used [JsonRpcDocumentationMetadata](#) for this [JsonRpcDocumentation](#).

```
public JsonRpcDocumentationMetadata? Metadata { get; }
```

Property Value

[JsonRpcDocumentationMetadata](#)

Property Methods

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Methods

Gets the collection of JSON-RPC methods.

```
public JsonRpcDocumentationMethod[] Methods { get; }
```

Property Value

[JsonRpcDocumentationMethod\[\]](#)

Method Export

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Export(IJsonRpcDocumentationExporter)

Exports this [JsonRpcDocumentation](#) with the specified [IJsonRpcDocumentationExporter](#).

```
public byte[] Export(IJsonRpcDocumentationExporter exporter)
```

Parameters

exporter [IJsonRpcDocumentationExporter](#)

The [IJsonRpcDocumentationExporter](#) instance.

Returns

[byte](#) ↗
[]

Method ExportToJson

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

ExportToJson(JsonOptions)

Gets an JSON string representation of this [JsonRpcDocumentation](#).

```
public string ExportToJson(JsonOptions options)
```

Parameters

options JsonOptions

The LightJson.JsonOptions used to encode this documentation.

Returns

string↗

ExportToJson()

Gets an JSON string representation of this [JsonRpcDocumentation](#).

```
public string ExportToJson()
```

Returns

string↗

Class JsonRpcDocumentationMetadata

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:[Sisk.JsonRPC.dll](#)

Represents the documentation metadata for JSON-RPC documentation.

```
public sealed class JsonRpcDocumentationMetadata
```

Constructors

[JsonRpcDocumentationMetadata\(\)](#)

Properties

[AllowedMethods](#)

Gets or sets an array of [HttpMethod](#) that are allowed for the JSON-RPC service at [ServicePath](#).

[ApplicationDescription](#)

Gets or sets the description of the application.

[ApplicationName](#)

Gets or sets the name of the application.

[ServicePath](#)

Gets or sets the path where the JSON-RPC service can receive remote procedures.

Constructor JsonRpcDocumentationMetadata

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

JsonRpcDocumentationMetadata()

```
public JsonRpcDocumentationMetadata()
```

Property AllowedMethods

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

AllowedMethods

Gets or sets an array of [HttpMethod](#) that are allowed for the JSON-RPC service at [ServicePath](#).

```
public string[] AllowedMethods { get; set; }
```

Property Value

[string](#)[]

Property ApplicationDescription

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

ApplicationDescription

Gets or sets the description of the application.

```
public string? ApplicationDescription { get; set; }
```

Property Value

[string](#)

Property ApplicationName

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

ApplicationName

Gets or sets the name of the application.

```
public string? ApplicationName { get; set; }
```

Property Value

[string](#)

Property ServicePath

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

ServicePath

Gets or sets the path where the JSON-RPC service can receive remote procedures.

```
public string? ServicePath { get; set; }
```

Property Value

[string](#)

Class JsonRpcDocumentationMethod

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Represents the documentation for a single JSON-RPC method.

```
public sealed class JsonRpcDocumentationMethod
```

Properties

[Category](#)

Gets the category of the JSON-RPC method.

[Description](#)

Gets the description of the JSON-RPC method.

[MethodName](#)

Gets the name of the JSON-RPC method.

[Parameters](#)

Gets the parameters of this JSON-RPC method.

[ReturnType](#)

Gets the return type of the JSON-RPC method.

Property Category

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Category

Gets the category of the JSON-RPC method.

```
public string? Category { get; }
```

Property Value

[string](#)

Property Description

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Description

Gets the description of the JSON-RPC method.

```
public string? Description { get; }
```

Property Value

[string](#)

Property MethodName

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

MethodName

Gets the name of the JSON-RPC method.

```
public string MethodName { get; }
```

Property Value

[string](#)

Property Parameters

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Parameters

Gets the parameters of this JSON-RPC method.

```
public JsonRpcDocumentationParameter[] Parameters { get; }
```

Property Value

[JsonRpcDocumentationParameter\[\]](#)

Property ReturnType

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

ReturnType

Gets the return type of the JSON-RPC method.

```
public Type ReturnType { get; }
```

Property Value

Type 

Class JsonRpcDocumentationParameter

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Represents the documentation for a parameter of a JSON-RPC method.

```
public sealed class JsonRpcDocumentationParameter
```

Properties

[Description](#)

Gets the description of the parameter.

[IsOptional](#)

Gets a value indicating whether the parameter is optional.

[ParameterName](#)

Gets the name of the parameter.

[ParameterType](#)

Gets the type of the parameter.

Property Description

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Description

Gets the description of the parameter.

```
public string? Description { get; }
```

Property Value

[string](#)

Property IsOptional

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

IsOptional

Gets a value indicating whether the parameter is optional.

```
public bool IsOptional { get; }
```

Property Value

bool ↗

Property ParameterName

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

ParameterName

Gets the name of the parameter.

```
public string ParameterName { get; }
```

Property Value

[string](#)

Property ParameterType

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

ParameterType

Gets the type of the parameter.

```
public Type ParameterType { get; }
```

Property Value

Type ↗

Class JsonRpcHtmlExport

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Provides an HTML-based [IJsonRpcDocumentationExporter](#).

```
public class JsonRpcHtmlExport : IJsonRpcDocumentationExporter
```

Implements

[IJsonRpcDocumentationExporter](#)

Constructors

[JsonRpcHtmlExport\(\)](#)

Properties

[ExportMetadata](#)

Gets or sets an boolean indicating if the documentation metadata should be exported in the HTML.

[ExportSummary](#)

Gets or sets an boolean indicating if an summary should be exported in the HTML.

[Header](#)

Gets or sets an optional object to append to the header of the exported HTML.

[Style](#)

Gets or sets the CSS styles used in the HTML export.

Methods

[EncodeDocumentationHtml\(JsonRpcDocumentation\)](#)

Encodes the specified [JsonRpcDocumentation](#) into a HTML string.

[ExportDocumentBytes\(JsonRpcDocumentation\)](#)

Exports the JSON-RPC documentation to a byte array.

Constructor JsonRpcHtmlExport

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

JsonRpcHtmlExport()

```
public JsonRpcHtmlExport()
```

Property ExportMetadata

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

ExportMetadata

Gets or sets an boolean indicating if the documentation metadata should be exported in the HTML.

```
public bool ExportMetadata { get; set; }
```

Property Value

[bool](#)

Property ExportSummary

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

ExportSummary

Gets or sets an boolean indicating if an summary should be exported in the HTML.

```
public bool ExportSummary { get; set; }
```

Property Value

[bool](#)

Property Header

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Header

Gets or sets an optional object to append to the header of the exported HTML.

```
public object? Header { get; set; }
```

Property Value

[object](#)

Property Style

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Style

Gets or sets the CSS styles used in the HTML export.

```
public string? Style { get; set; }
```

Property Value

[string](#)

Method EncodeDocumentationHtml

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

EncodeDocumentationHtml(JsonRpcDocumentation)

Encodes the specified [JsonRpcDocumentation](#) into a HTML string.

```
protected string EncodeDocumentationHtml(JsonRpcDocumentation documentation)
```

Parameters

documentation [JsonRpcDocumentation](#)

The [JsonRpcDocumentation](#) instance.

Returns

[string](#) ↗

Method ExportDocumentBytes

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

ExportDocumentBytes(JsonRpcDocumentation)

Exports the JSON-RPC documentation to a byte array.

```
public byte[] ExportDocumentBytes(JsonRpcDocumentation documentation)
```

Parameters

documentation [JsonRpcDocumentation](#)

The JSON-RPC documentation to export.

Returns

[byte](#) 

A byte array containing the exported documentation.

Class JsonRpcJsonExport

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

Provides an JSON-based [IJsonRpcDocumentationExporter](#).

```
public sealed class JsonRpcJsonExport : IJsonRpcDocumentationExporter
```

Implements

[IJsonRpcDocumentationExporter](#)

Constructors

[JsonRpcJsonExport\(\)](#)

Creates an new [JsonRpcJsonExport](#) instance with default parameters.

[JsonRpcJsonExport\(JsonOptions\)](#)

Creates an new [JsonRpcJsonExport](#) instance with the provided LightJson.JsonOptions instance.

Properties

[JsonOptions](#)

The [JsonOptions](#) instance used to encode the documentation.

Methods

[EncodeDocumentation\(JsonRpcDocumentation\)](#)

Encodes the specified documentation into an LightJson.JsonValue.

[ExportDocumentBytes\(JsonRpcDocumentation\)](#)

Exports the JSON-RPC documentation to a byte array.

Constructor JsonRpcJsonExport

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

JsonRpcJsonExport()

Creates an new [JsonRpcJsonExport](#) instance with default parameters.

```
public JsonRpcJsonExport()
```

JsonRpcJsonExport(JsonOptions)

Creates an new [JsonRpcJsonExport](#) instance with the provided LightJson.JsonOptions instance.

```
public JsonRpcJsonExport(JsonOptions options)
```

Parameters

options JsonOptions

Property JsonOptions

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

JsonOptions

The [JsonOptions](#) instance used to encode the documentation.

```
public JsonOptions JsonOptions { get; set; }
```

Property Value

JsonOptions

Method EncodeDocumentation

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

EncodeDocumentation(JsonRpcDocumentation)

Encodes the specified documentation into an LightJson.JsonValue.

```
public JsonValue EncodeDocumentation(JsonRpcDocumentation documentation)
```

Parameters

documentation [JsonRpcDocumentation](#)

The JSON-RPC documentation to encode.

Returns

JsonValue

Method ExportDocumentBytes

Namespace:[Sisk.JsonRPC.Documentation](#)

Assembly:Sisk.JsonRPC.dll

ExportDocumentBytes(JsonRpcDocumentation)

Exports the JSON-RPC documentation to a byte array.

```
public byte[] ExportDocumentBytes(JsonRpcDocumentation documentation)
```

Parameters

documentation [JsonRpcDocumentation](#)

The JSON-RPC documentation to export.

Returns

[byte](#) 

A byte array containing the exported documentation.

Struct JsonRpcError

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Represents an JSON-RPC error.

```
public readonly struct JsonRpcError
```

Constructors

[JsonRpcError\(\)](#)

Creates an new instance of the [JsonRpcError](#) structure.

[JsonRpcError\(int, string\)](#)

Creates an new instance of the [JsonRpcError](#) structure with given parameters.

[JsonRpcError\(int, string,JsonValue\)](#)

Creates an new instance of the [JsonRpcError](#) structure with given parameters.

Properties

[Code](#)

Gets the JSON-RPC error code.

[Data](#)

Gets the JSON-RPC error additional data.

[Message](#)

Gets the JSON-RPC error message.

Constructor JsonRpcError

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

JsonRpcError()

Creates an new instance of the [JsonRpcError](#) structure.

```
public JsonRpcError()
```

JsonRpcError(int, string)

Creates an new instance of the [JsonRpcError](#) structure with given parameters.

```
public JsonRpcError(int code, string message)
```

Parameters

code [int](#)

The JSON-RPC error code.

message [string](#)

The JSON-RPC error message.

JsonRpcError(int, string, JsonValue)

Creates an new instance of the [JsonRpcError](#) structure with given parameters.

```
public JsonRpcError(int code, string message, JsonValue data)
```

Parameters

code [int](#)

The JSON-RPC error code.

message [string](#)

The JSON-RPC error message.

data [JsonValue](#)

The JSON-RPC error additional data.

Property Code

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Code

Gets the JSON-RPC error code.

```
public int Code { get; }
```

Property Value

[int](#)

Property Data

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Data

Gets the JSON-RPC error additional data.

```
public JsonValue Data { get; }
```

Property Value

JsonValue

Property Message

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Message

Gets the JSON-RPC error message.

```
public string Message { get; }
```

Property Value

[string](#)

Class JsonRpcException

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Represents an error that occur during the JSON-RPC application execution.

```
public class JsonRpcException : Exception, ISerializable
```

Implements

[ISerializable](#) ↗

Constructors

[JsonRpcException\(string\)](#)

Initializes a new instance of the [JsonRpcException](#) class with a specified error message.

[JsonRpcException\(string, int, object?\)](#)

Initializes a new instance of the [JsonRpcException](#) class with a specified error message, error code, and additional data.

Properties

[Code](#)

Gets the error code associated with the JSON-RPC error.

[Data](#)

Gets additional data associated with the error, if any.

Methods

[AsRpcError\(\)](#)

Converts the current [JsonRpcException](#) into a [JsonRpcError](#).

Constructor JsonRpcException

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

JsonRpcException(string)

Initializes a new instance of the [JsonRpcException](#) class with a specified error message.

```
public JsonRpcException(string message)
```

Parameters

message [string](#)

The error message that explains the reason for the exception.

JsonRpcException(string, int, object?)

Initializes a new instance of the [JsonRpcException](#) class with a specified error message, error code, and additional data.

```
public JsonRpcException(string message, int code, object? data)
```

Parameters

message [string](#)

The error message that explains the reason for the exception.

code [int](#)

The error code associated with the JSON-RPC error.

data [object](#)

Additional data associated with the error.

Property Code

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Code

Gets the error code associated with the JSON-RPC error.

```
public int Code { get; }
```

Property Value

[int](#)

Property Data

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Data

Gets additional data associated with the error, if any.

```
public object? Data { get; }
```

Property Value

[object](#)

Method AsRpcError

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

AsRpcError()

Converts the current [JsonRpcException](#) into a [JsonRpcError](#).

```
public JsonRpcError AsRpcError()
```

Returns

[JsonRpcError](#)

A [JsonRpcError](#) representing the error details.

Class JsonRpcHandler

Namespace:[Sisk.JsonRPC](#)

Assembly:[Sisk.JsonRPC.dll](#)

Represents a handler for JSON-RPC requests.

```
public sealed class JsonRpcHandler
```

Constructors

[JsonRpcHandler\(HttpServer\)](#)

Initializes a new instance of the [JsonRpcHandler](#) class.

Properties

[JsonSerializerOptions](#)

Gets the JSON serializer options used for serialization and deserialization.

Methods

Gets the collection of JSON-RPC methods available in this handler.

[Transport](#)

Gets the transport layer used for communication.

Methods

[GetDocumentation\(\)](#)

Gets the documentation for this JSON-RPC handler.

[GetDocumentation\(JsonRpcDocumentationMetadata\)](#)

Gets the documentation for this JSON-RPC handler.

Constructor JsonRpcHandler

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

JsonRpcHandler(HttpServer)

Initializes a new instance of the [JsonRpcHandler](#) class.

```
public JsonRpcHandler(HttpServer parentServer)
```

Parameters

parentServer [HttpServer](#)

Defines the [HttpServer](#) where the JSON-RPC instance will run on.

Property JsonSerializerOptions

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

JsonSerializerOptions

Gets the JSON serializer options used for serialization and deserialization.

```
public JsonOptions JsonSerializerOptions { get; }
```

Property Value

JsonOptions

Property Methods

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Methods

Gets the collection of JSON-RPC methods available in this handler.

```
public JsonRpcMethodCollection Methods { get; }
```

Property Value

[JsonRpcMethodCollection](#)

Property Transport

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Transport

Gets the transport layer used for communication.

```
public JsonRpcTransportLayer Transport { get; }
```

Property Value

[JsonRpcTransportLayer](#)

Method GetDocumentation

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

GetDocumentation()

Gets the documentation for this JSON-RPC handler.

```
public JsonRpcDocumentation GetDocumentation()
```

Returns

[JsonRpcDocumentation](#)

GetDocumentation(JsonRpcDocumentationMetadata)

Gets the documentation for this JSON-RPC handler.

```
public JsonRpcDocumentation GetDocumentation(JsonRpcDocumentationMetadata metadata)
```

Parameters

metadata [JsonRpcDocumentationMetadata](#)

The [JsonRpcDocumentationMetadata](#) to generate in the documentation.

Returns

[JsonRpcDocumentation](#)

Class JsonRpcMethodCollection

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Represents a collection of JSON-RPC methods, allowing for dynamic addition and removal of methods.

```
public sealed class JsonRpcMethodCollection
```

Constructors

[JsonRpcMethodCollection\(\)](#)

Methods

[AddMethod\(string, Delegate\)](#)

Adds a method to the collection with the specified name.

[AddMethodsFromType\(Type\)](#)

Adds methods from the specified type to the collection without prefixing method names.

[AddMethodsFromType\(Type, object?\)](#)

Adds methods from the specified type to the collection without prefixing method names.

[AddMethodsFromType\(Type, object?, bool\)](#)

Adds methods from the specified type to the collection, optionally prefixing method names with the type name.

[AddMethodsFromType<T>\(T, bool\)](#)

Adds methods from the specified type to the collection, optionally prefixing method names with the type name.

[RemoveMethod\(string\)](#)

Removes a method from the collection by its name.

Constructor JsonRpcMethodCollection

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

JsonRpcMethodCollection()

```
public JsonRpcMethodCollection()
```

Method AddMethod

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

AddMethod(string, Delegate)

Adds a method to the collection with the specified name.

```
public void AddMethod(string name, Delegate method)
```

Parameters

name [string](#)

The name of the method to add.

method [Delegate](#)

The delegate representing the method to add.

Method AddMethodsFromType

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

AddMethodsFromType<T>(T, bool)

Adds methods from the specified type to the collection, optionally prefixing method names with the type name.

```
public void AddMethodsFromType<T>(T target, bool prefixTypes = false) where T : notnull
```

Parameters

target T

The target object instance containing the methods.

prefixTypes bool

Indicates whether to prefix method names with the type name.

Type Parameters

T

The type from which to scan and add methods.

AddMethodsFromType(Type, object?, bool)

Adds methods from the specified type to the collection, optionally prefixing method names with the type name.

```
public void AddMethodsFromType(Type type, object? target, bool prefixTypes)
```

Parameters

type Type

The type from which to scan and add methods.

target [object](#)

The target object instance containing the methods.

prefixTypes [bool](#)

Indicates whether to prefix method names with the type name.

AddMethodsFromType(Type, object?)

Adds methods from the specified type to the collection without prefixing method names.

```
public void AddMethodsFromType(Type type, object? target)
```

Parameters

type [Type](#)

The type from which to scan and add methods.

target [object](#)

The target object instance containing the methods.

AddMethodsFromType(Type)

Adds methods from the specified type to the collection without prefixing method names.

```
public void AddMethodsFromType(Type type)
```

Parameters

type [Type](#)

The type from which to scan and add methods.

Method RemoveMethod

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

RemoveMethod(string)

Removes a method from the collection by its name.

```
public void RemoveMethod(string name)
```

Parameters

name [string](#)

The name of the method to remove.

Class JsonRpcRequest

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Represents an JSON-RPC request message.

```
public sealed class JsonRpcRequest
```

Properties

[Id](#)

Gets the ID of the JSON-RPC message.

[Method](#)

Gets the method name of the JSON-RPC message.

[Parameters](#)

Gets the LightJson.JsonValue containing the message parameter values.

[Version](#)

Gets the version used in the JSON-RPC message.

Property Id

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Id

Gets the ID of the JSON-RPC message.

```
public JsonValue Id { get; }
```

Property Value

JsonValue

Property Method

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Method

Gets the method name of the JSON-RPC message.

```
public string Method { get; }
```

Property Value

[string](#)

Property Parameters

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Parameters

Gets the LightJson.JsonValue containing the message parameter values.

```
public JsonValue Parameters { get; }
```

Property Value

JsonValue

Property Version

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Version

Gets the version used in the JSON-RPC message.

```
public string Version { get; }
```

Property Value

[string](#)

Class JsonRpcResponse

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Represents an JSON-RPC response message.

```
public sealed class JsonRpcResponse
```

Properties

[Error](#)

Gets the JSON-RPC response error.

[Id](#)

Gets the JSON-RPC response id.

[Result](#)

Gets the JSON-RPC response result.

[Version](#)

Gets the JSON-RPC response version. This property will always return "2.0".

Methods

[CreateErrorResponse\(JsonValue, JsonRpcError\)](#)

Creates an new error [JsonRpcResponse](#) with given parameters.

[CreateSuccessResponse\(JsonValue, JsonValue\)](#)

Creates an new success [JsonRpcResponse](#) with given parameters.

Property Error

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Error

Gets the JSON-RPC response error.

```
public JsonRpcError? Error { get; }
```

Property Value

[JsonRpcError?](#)

Property Id

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Id

Gets the JSON-RPC response id.

```
public JsonValue Id { get; }
```

Property Value

JsonValue

Property Result

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Result

Gets the JSON-RPC response result.

```
public JsonValue? Result { get; }
```

Property Value

JsonValue?

Property Version

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Version

Gets the JSON-RPC response version. This property will always return "2.0".

```
public string Version { get; }
```

Property Value

[string](#)

Method CreateErrorResponse

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

CreateErrorResponse(JsonValue, JsonRpcError)

Creates an new error [JsonRpcResponse](#) with given parameters.

```
public static JsonRpcResponse CreateErrorResponse(JsonValue id, JsonRpcError error)
```

Parameters

id JsonValue

The JSON-RPC response id.

error [JsonRpcError](#)

The JSON-RPC response error.

Returns

[JsonRpcResponse](#)

Method CreateSuccessResponse

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

CreateSuccessResponse(JsonValue, JsonValue)

Creates an new success [JsonRpcResponse](#) with given parameters.

```
public static JsonRpcResponse CreateSuccessResponse(JsonValue id, JsonValue result)
```

Parameters

id JsonValue

The JSON-RPC response id.

result JsonValue

The JSON-RPC response object.

Returns

[JsonRpcResponse](#)

Class JsonRpcServerConfigurationEventArgs

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Represents the class which contains event data for the JSON-RPC configuration event.

```
public sealed class JsonRpcServerConfigurationEventArgs : EventArgs
```

Properties

Handler

Gets the configuring [JsonRpcHandler](#).

Router

Gets the target [Router](#) which are being configured.

Property Handler

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Handler

Gets the configuring [JsonRpcHandler](#).

```
public JsonRpcHandler Handler { get; }
```

Property Value

[JsonRpcHandler](#)

Property Router

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Router

Gets the target [Router](#) which are being configured.

```
public Router Router { get; }
```

Property Value

[Router](#)

Class JsonRpcServerHandler

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Provides an [HttpServerHandler](#) for configuring the JSON-RPC handler for the HTTP server.

```
public sealed class JsonRpcServerHandler : HttpServerHandler
```

Constructors

[JsonRpcServerHandler\(\)](#)

Creates an new instance of the [JsonRpcServerHandler](#) class.

Methods

[OnServerStarting\(HttpServer\)](#)

Event that is called immediately before starting the [HttpServer](#).

[OnSetupRouter\(Router\)](#)

Event that is called when an [Router](#) is binded to the HTTP server.

Events

[ConfigureAction](#)

Gets or sets the action which will be called in the configuring [Router](#) with this handler.

Constructor JsonRpcServerHandler

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

JsonRpcServerHandler()

Creates an new instance of the [JsonRpcServerHandler](#) class.

```
public JsonRpcServerHandler()
```

Method OnServerStarting

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

OnServerStarting(HttpServer)

Event that is called immediately before starting the [HttpServer](#).

```
protected override void OnServerStarting(HttpServer server)
```

Parameters

server [HttpServer](#)

The HTTP server entity which is starting.

Method OnSetupRouter

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

OnSetupRouter(Router)

Event that is called when an [Router](#) is binded to the HTTP server.

```
protected override void OnSetupRouter(Router router)
```

Parameters

router Router

The router entity which is binded.

Event ConfigureAction

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Gets or sets the action which will be called in the configuring [Router](#) with this handler.

```
public event EventHandler<JsonRpcServerConfigurationEventArgs>? ConfigureAction
```

Returns

[EventHandler](#)<JsonRpcServerConfigurationEventArgs>

Gets or sets the action which will be called in the configuring with this handler.

Class JsonRpcServerHandlerExtensions

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Provides extensions methods for the HTTP server and JSON-RPC handler.

```
public static class JsonRpcServerHandlerExtensions
```

Methods

[UseJsonRPC\(HttpServerHostContextBuilder, EventHandler<JsonRpcServerConfigurationEventArgs>\)](#)

Enables JSON-RPC in this HTTP server.

Method UseJsonRPC

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

**UseJsonRPC(HttpServerHostContextBuilder,
EventHandler<JsonRpcServerConfigurationEventArgs>)**

Enables JSON-RPC in this HTTP server.

```
public static HttpServerHostContextBuilder UseJsonRPC(this HttpServerHostContextBuilder builder,  
EventHandler<JsonRpcServerConfigurationEventArgs> configure)
```

Parameters

builder [HttpServerHostContextBuilder](#)

The self [HttpServerHostContextBuilder](#) for fluent chaining.

configure [EventHandler<JsonRpcServerConfigurationEventArgs>](#)

The event handler callback that is called to configure routes and web methods for the JSON-RPC.

Returns

[HttpServerHostContextBuilder](#)

Class JsonRpcTransportLayer

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

Provides transport layers for handling JSON-RPC communications.

```
public sealed class JsonRpcTransportLayer
```

Properties

[HttpGet](#)

Gets the action to handle HTTP GET requests.

[HttpPost](#)

Gets the action to handle HTTP POST requests.

[WebSocket](#)

Gets the event handler for WebSocket message reception.

Property `HttpGet`

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

`HttpGet`

Gets the action to handle HTTP GET requests.

```
public RouteAction HttpGet { get; }
```

Property Value

[RouteAction](#)

Property HttpPost

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

HttpPost

Gets the action to handle HTTP POST requests.

```
public RouteAction HttpPost { get; }
```

Property Value

[RouteAction](#)

Property WebSocket

Namespace:[Sisk.JsonRPC](#)

Assembly:Sisk.JsonRPC.dll

WebSocket

Gets the event handler for WebSocket message reception.

```
public EventHandler<WebSocketMessage> WebSocket { get; }
```

Property Value

[EventHandler](#)<WebSocketMessage>

Namespace Sisk.ModelContextProtocol

Classes

[McpProvider](#)

Represents a server that hosts a Model Context Protocol server.

[McpProviderExtensions](#)

Provides extension methods for configuring and handling requests with the Model Context Protocol (MCP).

[McpTool](#)

Represents a tool that can be hosted and executed by an MCP server.

[McpToolContext](#)

Provides context for the execution of an MCP tool.

[McpToolResult](#)

Represents the result of executing an MCP tool.

Delegates

[McpToolHandler](#)

Represents a delegate that handles the execution of an MCP tool.

Class McpProvider

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Represents a server that hosts a Model Context Protocol server.

```
public sealed class McpProvider
```

Constructors

[McpProvider\(\)](#)

Creates a new instance of the [McpProvider](#) class.

[McpProvider\(string, string, Version\)](#)

Creates a new instance of the [McpProvider](#) class with the specified server details.

Fields

[PROTOCOL_VERSION](#)

Represents the current supported version of the Model Context Protocol.

Properties

[ServerName](#)

Gets or sets the internal name of the server.

[ServerTitle](#)

Gets or sets the display name of the server.

[ServerVersion](#)

Gets or sets the version of the MCP server.

Tools

Gets or sets the list of MCP tools hosted by this server.

Methods

[HandleRequestAsync\(HttpRequest, CancellationToken\)](#)

Handles an incoming HTTP request for MCP operations asynchronously.

Constructor McpProvider

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

McpProvider()

Creates a new instance of the [McpProvider](#) class.

```
public McpProvider()
```

McpProvider(string, string, Version)

Creates a new instance of the [McpProvider](#) class with the specified server details.

```
public McpProvider(string serverName, string serverTitle, Version serverVersion)
```

Parameters

serverName [string](#)

The name of the server.

serverTitle [string](#)

The title of the server.

serverVersion [Version](#)

The version of the server.

Field PROTOCOL_VERSION

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Represents the current supported version of the Model Context Protocol.

```
public const string PROTOCOL_VERSION = "2025-06-18"
```

Returns

[string](#)

Represents the current supported version of the Model Context Protocol.

Property ServerName

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

ServerName

Gets or sets the internal name of the server.

```
public string? ServerName { get; set; }
```

Property Value

[string](#)

Property ServerTitle

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

ServerTitle

Gets or sets the display name of the server.

```
public string? ServerTitle { get; set; }
```

Property Value

[string](#)

Property ServerVersion

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

ServerVersion

Gets or sets the version of the MCP server.

```
public Version ServerVersion { get; set; }
```

Property Value

[Version](#)

Property Tools

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Tools

Gets or sets the list of MCP tools hosted by this server.

```
public IList<McpTool> Tools { get; set; }
```

Property Value

[IList](#)<[McpTool](#)>

Method HandleRequestAsync

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

HandleRequestAsync([HttpRequest](#), [CancellationToken](#))

Handles an incoming HTTP request for MCP operations asynchronously.

```
public Task<HttpResponse> HandleRequestAsync(HttpRequest request, CancellationToken cancellation  
= default)
```

Parameters

request [HttpRequest](#)

The incoming HTTP request.

cancellation [CancellationToken](#)

A token to observe for cancellation requests.

Returns

[Task](#)<[HttpResponse](#)>

An HTTP response representing the result of the request handling.

Class McpProviderExtensions

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Provides extension methods for configuring and handling requests with the Model Context Protocol (MCP).

```
public static class McpProviderExtensions
```

Methods

[HandleMcpRequestAsync\(HttpRequest, CancellationToken\)](#)

Handles an incoming HTTP request using the configured MCP provider.

[UseMcp\(HttpServerHostContextBuilder, McpProvider\)](#)

Configures the HTTP server host builder to use a specific MCP provider.

[UseMcp\(HttpServerHostContextBuilder, Action<McpProvider>\)](#)

Configures the HTTP server host builder to use an MCP provider built with the provided action.

Method HandleMcpRequestAsync

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

HandleMcpRequestAsync([HttpRequest](#), [CancellationToken](#))

Handles an incoming HTTP request using the configured MCP provider.

```
public static Task<HttpResponse> HandleMcpRequestAsync(this HttpRequest request,  
CancellationToken cancellation = default)
```

Parameters

request [HttpRequest](#)

The HTTP request to handle.

cancellation [CancellationToken](#)

A token to monitor for cancellation requests.

Returns

[Task](#)<[HttpResponse](#)>

A task that represents the asynchronous operation, containing the HTTP response.

Exceptions

[InvalidOperationException](#)

Thrown if the MCP provider has not been configured.

Method UseMcp

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

UseMcp(HttpServerHostContextBuilder, McpProvider)

Configures the HTTP server host builder to use a specific MCP provider.

```
public static HttpServerHostContextBuilder UseMcp(this HttpServerHostContextBuilder builder,  
McpProvider provider)
```

Parameters

builder [HttpServerHostContextBuilder](#)

The HTTP server host builder to configure.

provider [McpProvider](#)

The MCP provider to use.

Returns

[HttpServerHostContextBuilder](#)

The configured HTTP server host builder.

Exceptions

[ArgumentNullException](#) ↴

Thrown if the provider is null.

UseMcp(HttpServerHostContextBuilder, Action<McpProvider>)

Configures the HTTP server host builder to use an MCP provider built with the provided action.

```
public static HttpServerHostContextBuilder UseMcp(this HttpServerHostContextBuilder builder,  
Action<McpProvider> providerBuilder)
```

Parameters

builder [HttpServerHostContextBuilder](#)

The HTTP server host builder to configure.

providerBuilder [Action`1<McpProvider>](#)

An action to configure the MCP provider.

Returns

[HttpServerHostContextBuilder](#)

The configured HTTP server host builder.

Class McpTool

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Represents a tool that can be hosted and executed by an MCP server.

```
public sealed class McpTool
```

Constructors

[McpTool\(string, string, JsonSchema, McpToolHandler, string?\)](#)

Initializes a new instance of the [McpTool](#) class.

Properties

Description

Gets a description of what the tool does.

ExecuteAsync

Gets or sets the handler function that will be executed when the tool is invoked.

Name

Gets the unique name of the tool.

Schema

Gets the JSON schema that defines the expected input arguments for the tool.

Title

Gets the display title of the tool. If null, the [Name](#) will be used.

Constructor McpTool

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

McpTool(string, string, JsonSchema, McpToolHandler, string?)

Initializes a new instance of the [McpTool](#) class.

```
public McpTool(string name, string description, JsonSchema schema, McpToolHandler  
executionHandler, string? title = null)
```

Parameters

name [string](#)

The unique name of the tool.

description [string](#)

A description of what the tool does.

schema [JsonSchema](#)

The JSON schema defining the tool's input arguments.

executionHandler [McpToolHandler](#)

The handler function that will be executed when the tool is invoked.

title [string](#)

The optional display title of the tool.

Property Description

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Description

Gets a description of what the tool does.

```
public string Description { get; }
```

Property Value

[string](#)

Property ExecuteAsync

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

ExecuteAsync

Gets or sets the handler function that will be executed when the tool is invoked.

```
public McpToolHandler ExecuteAsync { get; set; }
```

Property Value

[McpToolHandler](#)

Property Name

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Name

Gets the unique name of the tool.

```
public string Name { get; }
```

Property Value

[string](#)

Property Schema

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Schema

Gets the JSON schema that defines the expected input arguments for the tool.

```
public JsonSchema Schema { get; }
```

Property Value

JsonSchema

Property Title

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Title

Gets the display title of the tool. If null, the [Name](#) will be used.

```
public string? Title { get; }
```

Property Value

[string](#) ↗

Class McpToolContext

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Provides context for the execution of an MCP tool.

```
public sealed class McpToolContext
```

Constructors

[McpToolContext\(\)](#)

Properties

Arguments

Gets the arguments provided for the tool execution as a JSON object.

Cancellation

Gets a token to observe for cancellation requests.

Metadata

Gets any additional metadata associated with the tool execution.

Request

Gets the incoming HTTP request that triggered the tool execution.

Server

Gets the MCP server instance associated with the current context.

ToolName

Gets the name of the tool being executed.

Constructor McpToolContext

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

McpToolContext()

```
public McpToolContext()
```

Property Arguments

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Arguments

Gets the arguments provided for the tool execution as a JSON object.

```
public required JsonObject Arguments { get; init; }
```

Property Value

JsonObject

Property Cancellation

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Cancellation

Gets a token to observe for cancellation requests.

```
public required CancellationToken Cancellation { get; init; }
```

Property Value

[CancellationToken](#) ↗

Property Metadata

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Metadata

Gets any additional metadata associated with the tool execution.

```
public required JsonObject Metadata { get; init; }
```

Property Value

JsonObject

Property Request

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Request

Gets the incoming HTTP request that triggered the tool execution.

```
public required HttpRequest Request { get; init; }
```

Property Value

[HttpRequest](#)

Property Server

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Server

Gets the MCP server instance associated with the current context.

```
public required McpProvider Server { get; init; }
```

Property Value

[McpProvider](#)

Property ToolName

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

ToolName

Gets the name of the tool being executed.

```
public required string ToolName { get; init; }
```

Property Value

[string](#)

Delegate McpToolHandler

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Represents a delegate that handles the execution of an MCP tool.

```
public delegate Task<McpToolResult> McpToolHandler(McpToolContext context)
```

Parameters

context [McpToolContext](#)

The context in which the tool is executed.

Returns

[Task](#)<[McpToolResult](#)>

A task that, when completed, yields the [McpToolResult](#) of the tool execution.

Constructors

[McpToolHandler\(object, nint\)](#)

Methods

[BeginInvoke\(McpToolContext, AsyncCallback, object\)](#)

[EndInvoke\(IAsyncResult\)](#)

[Invoke\(McpToolContext\)](#)

Constructor McpToolHandler

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

McpToolHandler(object, nint)

```
public McpToolHandler(object @object, nint method)
```

Parameters

object object ↗

method nint ↗

Method BeginInvoke

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

BeginInvoke(McpToolContext, AsyncCallback, object)

```
public virtual IAsyncResult BeginInvoke(McpToolContext context, AsyncCallback callback,
object @object)
```

Parameters

context [McpToolContext](#)

callback [AsyncCallback](#)

object [object](#)

Returns

[IAsyncResult](#)

Method EndInvoke

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

EndInvoke(IAsyncResult)

```
public virtual Task<McpToolResult> EndInvoke(IAsyncResult result)
```

Parameters

result [IAsyncResult](#)

Returns

[Task](#)<McpToolResult>

Method Invoke

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Invoke(McpToolContext)

```
public virtual Task<McpToolResult> Invoke(McpToolContext context)
```

Parameters

context [McpToolContext](#)

Returns

[Task](#)<McpToolResult>

Class McpToolResult

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Represents the result of executing an MCP tool.

```
public sealed class McpToolResult
```

Constructors

[McpToolResult\(JsonValue\)](#)

Initializes a new instance of the [McpToolResult](#) class.

Properties

[Result](#)

Gets the JSON representation of the tool result.

Methods

[Combine\(params McpToolResult\[\]\)](#)

Combines multiple [McpToolResult](#) objects into a single result. If the input contains a single result, it is returned directly. Otherwise, all individual results are unpacked and combined into a JSON array.

[CreateAudio\(ReadOnlySpan<byte>, string\)](#)

Creates an audio-based result for an MCP tool.

[CreateImage\(ReadOnlySpan<byte>, string\)](#)

Creates an image-based result for an MCP tool.

`CreateText(string)`

Creates a text-based result for an MCP tool.

Operators

`implicit operator McpToolResult(string)`

Constructor McpToolResult

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

McpToolResult(JsonValue)

Initializes a new instance of the [McpToolResult](#) class.

```
public McpToolResult(JsonValue result)
```

Parameters

result JsonValue

The JSON value representing the tool result.

Property Result

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Result

Gets the JSON representation of the tool result.

```
public JsonValue Result { get; }
```

Property Value

JsonValue

Method Combine

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

Combine(params McpToolResult[])

Combines multiple [McpToolResult](#) objects into a single result. If the input contains a single result, it is returned directly. Otherwise, all individual results are unpacked and combined into a JSON array.

```
public static McpToolResult Combine(params McpToolResult[] results)
```

Parameters

results [McpToolResult](#)[]

A collection of [McpToolResult](#) objects to combine.

Returns

[McpToolResult](#)

A single [McpToolResult](#) containing the combined results.

Method CreateAudio

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

CreateAudio(ReadOnlySpan<byte>, string)

Creates an audio-based result for an MCP tool.

```
public static McpToolResult CreateAudio(ReadOnlySpan<byte> audioBytes, string mimeType  
= "audio/wav")
```

Parameters

audioBytes [ReadOnlySpan<byte>](#)

The byte array representing the audio data.

mimeType [string](#)

The MIME type of the audio (e.g., "audio/wav"). Defaults to "audio/wav".

Returns

[McpToolResult](#)

An [McpToolResult](#) representing an audio result.

Method CreateImage

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

CreateImage(ReadOnlySpan<byte>, string)

Creates an image-based result for an MCP tool.

```
public static McpToolResult CreateImage(ReadOnlySpan<byte> imageBytes, string mimeType  
= "image/png")
```

Parameters

imageBytes [ReadOnlySpan<byte>](#)

The byte array representing the image data.

mimeType [string](#)

The MIME type of the image (e.g., "image/png"). Defaults to "image/png".

Returns

[McpToolResult](#)

An [McpToolResult](#) representing an image result.

Method CreateText

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

CreateText(string)

Creates a text-based result for an MCP tool.

```
public static McpToolResult CreateText(string text)
```

Parameters

text string ↗

The text content of the result.

Returns

[McpToolResult](#)

An [McpToolResult](#) representing a text result.

Operator implicit operator

Namespace:[Sisk.ModelContextProtocol](#)

Assembly:Sisk.ModelContextProtocol.dll

implicit operator McpToolResult(string)

```
public static implicit operator McpToolResult(string text)
```

Parameters

text string ↗

Returns

[McpToolResult](#)

Namespace Sisk.Ssl

Classes

[CertificateUtil](#)

Provides a set of useful functions to issue development certificates for the [SslProxy](#).

[SslProxy](#)

Represents a HTTP/1.1 proxy server that forwards traffic over SSL/HTTPS into an insecure HTTP gateway.

[SslProxyExtensions](#)

Provides extension methods for [SslProxy](#).

[SslProxyServerHandler](#)

Provides event handlers and hooks for [SslProxy](#).

Class CertificateUtil

Namespace:[Sisk.Ssl](#)

Assembly:[Sisk.SslProxy.dll](#)

Provides a set of useful functions to issue development certificates for the [SslProxy](#).

```
public static class CertificateUtil
```

Methods

[CreateDevelopmentCertificate\(params string\[\]\)](#)

Creates a self-signed certificate for the specified DNS names.

[CreateTrustedDevelopmentCertificate\(params string\[\]\)](#)

Creates a self-signed certificate for the specified DNS names and adds them to the local user's certificate store.

Method CreateDevelopmentCertificate

Namespace:[Sisk.Ssl](#)

Assembly:Sisk.SslProxy.dll

CreateDevelopmentCertificate(params string[])

Creates a self-signed certificate for the specified DNS names.

```
public static X509Certificate2 CreateDevelopmentCertificate(params string[] dnsNames)
```

Parameters

dnsNames [string](#) []

The certificate DNS names.

Returns

[X509Certificate2](#)

Method CreateTrustedDevelopmentCertificate

Namespace:[Sisk.Ssl](#)

Assembly:[Sisk.SslProxy.dll](#)

CreateTrustedDevelopmentCertificate(params string[])

Creates a self-signed certificate for the specified DNS names and adds them to the local user's certificate store.

```
public static X509Certificate2 CreateTrustedDevelopmentCertificate(params string[] dnsNames)
```

Parameters

dnsNames [string](#) []

The certificate DNS names.

Returns

[X509Certificate2](#)

Class SslProxy

Namespace:[Sisk.Ssl](#)

Assembly:[Sisk.SslProxy.dll](#)

Represents a HTTP/1.1 proxy server that forwards traffic over SSL/HTTPS into an insecure HTTP gateway.

```
public sealed class SslProxy : IDisposable
```

Implements

[IDisposable](#) ↗

Constructors

[SslProxy\(int, X509Certificate, IPPEndPoint\)](#)

Initializes a new instance of the [SslProxy](#) class.

Properties

[AllowedProtocols](#)

Gets or sets the SSL/HTTPS protocols allowed for connections.

[CheckCertificateRevocation](#)

Gets or sets a value indicating whether to check for certificate revocation.

[ClientCertificateRequired](#)

Gets or sets a value indicating whether client certificates are required for authentication.

[GatewayEndpoint](#)

Gets the proxy endpoint.

[GatewayHostname](#)

Gets or sets a fixed proxy host header value for incoming requests.

[GatewayTimeout](#)

Gets or sets the maximum time that the gateway should take to respond to a connection or message from the proxy.

[ProxyAuthorization](#)

Gets or sets the Proxy-Authorization header value for creating an trusted gateway between the application and the proxy.

[ServerCertificate](#)

Gets the SSL certificate used by the proxy server.

[UseGatewayHttps](#)

Gets or sets whether the [SslProxy](#) should use HTTPS for the gateway connection or plain HTTP.

Methods

[Dispose\(\)](#)

[Start\(\)](#)

Starts the [SslProxy](#) and start routing traffic to the set remote endpoint.

Constructor SslProxy

Namespace:[Sisk.Ssl](#)

Assembly:Sisk.SslProxy.dll

SslProxy(int, X509Certificate, IPEndPoint)

Initializes a new instance of the [SslProxy](#) class.

```
public SslProxy(int sslListeningPort, X509Certificate certificate, IPEndPoint remoteEndpoint)
```

Parameters

sslListeningPort [int](#)

The port number on which the proxy server listens for incoming connections.

certificate [X509Certificate](#)

The SSL/TLS certificate used by the proxy server.

remoteEndpoint [IPEndPoint](#)

The remote endpoint to which the proxy server forwards traffic.

Property AllowedProtocols

Namespace:[Sisk.Ssl](#)

Assembly:[Sisk.SslProxy.dll](#)

AllowedProtocols

Gets or sets the SSL/HTTPS protocols allowed for connections.

```
public SslProtocols AllowedProtocols { get; set; }
```

Property Value

[SslProtocols](#)

Property CheckCertificateRevocation

Namespace:[Sisk.Ssl](#)

Assembly:[Sisk.SslProxy.dll](#)

CheckCertificateRevocation

Gets or sets a value indicating whether to check for certificate revocation.

```
public bool CheckCertificateRevocation { get; set; }
```

Property Value

[bool](#)

Property ClientCertificateRequired

Namespace:[Sisk.Ssl](#)

Assembly:Sisk.SslProxy.dll

ClientCertificateRequired

Gets or sets a value indicating whether client certificates are required for authentication.

```
public bool ClientCertificateRequired { get; set; }
```

Property Value

bool

Property GatewayEndpoint

Namespace:[Sisk.Ssl](#)

Assembly:[Sisk.SslProxy.dll](#)

GatewayEndpoint

Gets the proxy endpoint.

```
public IPPEndPoint GatewayEndpoint { get; }
```

Property Value

[IPEndPoint](#)

Property GatewayHostname

Namespace:[Sisk.Ssl](#)

Assembly:[Sisk.SslProxy.dll](#)

GatewayHostname

Gets or sets an fixed proxy host header value for incoming requests.

```
public string? GatewayHostname { get; set; }
```

Property Value

[string](#)

Property GatewayTimeout

Namespace:[Sisk.Ssl](#)

Assembly:Sisk.SslProxy.dll

GatewayTimeout

Gets or sets the maximum time that the gateway should take to respond to a connection or message from the proxy.

```
public TimeSpan GatewayTimeout { get; set; }
```

Property Value

[TimeSpan](#)

Property ProxyAuthorization

Namespace:[Sisk.Ssl](#)

Assembly:[Sisk.SslProxy.dll](#)

ProxyAuthorization

Gets or sets the Proxy-Authorization header value for creating an trusted gateway between the application and the proxy.

```
public string? ProxyAuthorization { get; set; }
```

Property Value

[string](#)

Property ServerCertificate

Namespace:[Sisk.Ssl](#)

Assembly:Sisk.SslProxy.dll

ServerCertificate

Gets the SSL certificate used by the proxy server.

```
public X509Certificate ServerCertificate { get; }
```

Property Value

[X509Certificate](#)

Property UseGatewayHttps

Namespace:[Sisk.Ssl](#)

Assembly:Sisk.SslProxy.dll

UseGatewayHttps

Gets or sets whether the [SslProxy](#) should use HTTPS for the gateway connection or plain HTTP.

```
public bool UseGatewayHttps { get; set; }
```

Property Value

bool

Method Dispose

Namespace:[Sisk.Ssl](#)

Assembly:Sisk.SslProxy.dll

Dispose()

```
public void Dispose()
```

Method Start

Namespace:[Sisk.Ssl](#)

Assembly:Sisk.SslProxy.dll

Start()

Starts the SslProxy and start routing traffic to the set remote endpoint.

```
public void Start()
```

Class SslProxyExtensions

Namespace:[Sisk.Ssl](#)

Assembly:[Sisk.SslProxy.dll](#)

Provides extension methods for [SslProxy](#).

```
public static class SslProxyExtensions
```

Methods

[UseSsl\(HttpServerHostContextBuilder, short, X509Certificate?, SslProtocols, bool, object?, bool\)](#)

Configures the [HttpServerHostContext](#) to use [SslProxy](#) with the specified parameters.

Method UseSsl

Namespace:[Sisk.Ssl](#)

Assembly:[Sisk.SslProxy.dll](#)

UseSsl([HttpServerHostContextBuilder](#), short, [X509Certificate?](#), [SslProtocols](#), bool, [object?](#), bool)

Configures the [HttpServerHostContext](#) to use [SslProxy](#) with the specified parameters.

```
public static HttpServerHostContextBuilder UseSsl(this HttpServerHostContextBuilder builder,
short sslListeningPort, X509Certificate? certificate = null, SslProtocols allowedProtocols =
SslProtocols.Tls12 | SslProtocols.Tls13, bool clientCertificateRequired = false, object?
proxyAuthorization = null, bool onlyUseIPv4 = false)
```

Parameters

builder [HttpServerHostContextBuilder](#)

The [HttpServerHostContextBuilder](#) instance to configure.

sslListeningPort [short](#)

The port number on which the server will listen for SSL/HTTPS connections.

certificate [X509Certificate](#)

Optional. The SSL/HTTPS certificate to use for encrypting communications.

allowedProtocols [SslProtocols](#)

Optional. The SSL/HTTPS protocols allowed for the connection. Defaults to [Tls12](#) and [Tls13](#).

clientCertificateRequired [bool](#)

Optional. Specifies whether a client certificate is required for authentication. Defaults to `false`.

proxyAuthorization [object](#)

Optional. Specifies the Proxy-Authorization header value for creating an trusted gateway between the application and the proxy.

onlyUseIPv4 [bool](#)

Optional. Specifies whether DNS Resolve may also use IPv6 addresses or should only use IPv4 addresses

Returns

[HttpServerHostContextBuilder](#)

The configured [HttpServerHostContextBuilder](#) instance.

Class SslProxyServerHandler

Namespace:[Sisk.Ssl](#)

Assembly:[Sisk.SslProxy.dll](#)

Provides event handlers and hooks for [SslProxy](#).

```
public sealed class SslProxyServerHandler : HttpServerHandler
```

Constructors

[SslProxyServerHandler\(SslProxy\)](#)

Creates an new [SslProxyServerHandler](#) instance with the specified [SslProxy](#) instance.

Properties

[SecureProxy](#)

Gets the [SslProxy](#) instance used in this server handler.

Constructor SslProxyServerHandler

Namespace:[Sisk.Ssl](#)

Assembly:[Sisk.SslProxy.dll](#)

SslProxyServerHandler(SslProxy)

Creates an new [SslProxyServerHandler](#) instance with the specified [SslProxy](#) instance.

```
public SslProxyServerHandler(SslProxy secureProxy)
```

Parameters

secureProxy [SslProxy](#)

The [SslProxy](#) instance.

Property SecureProxy

Namespace:[Sisk.Ssl](#)

Assembly:Sisk.SslProxy.dll

SecureProxy

Gets the [SslProxy](#) instance used in this server handler.

```
public SslProxy SecureProxy { get; }
```

Property Value

[SslProxy](#)

Namespace TinyComponents

Classes

[HtmlElement](#)

Represents an HTML element for rendering

[HtmlElementExtensions](#)

Provides extension methods for [HtmlElement](#) objects.

[MarkdownText](#)

Represents a Markdown text with settings for rendering.

[NodeAttributeCollection](#)

Represents an dictionary collection of attributes.

[NodeComment](#)

Represents a node comment in a document.

[RenderableFunction](#)

Represents an object which their renderable contents is called by an function.

[RenderableText](#)

Represents an simple renderable text.

[XmlNode](#)

Represents an renderable XML node.

[XmlNodeExtensions](#)

Provides extension methods for [INode](#) objects.

Interfaces

[INode](#)

Represents an renderable node.

Class HtmlElement

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Represents an HTML element for rendering

```
public class HtmlElement : INode
```

Implements

[INode](#)

Extension Methods

```
HtmlElementExtensions.WithClass<THtmlElement>(THtmlElement, params string[]) ,  
HtmlElementExtensions.WithId<THtmlElement>(THtmlElement, string) ,  
HtmlElementExtensionsWithName<THtmlElement>(THtmlElement, string) ,  
HtmlElementExtensions.WithStyle<THtmlElement>(THtmlElement, object) ,  
XmlNodeExtensions.SelfClosed<TXmlNode>(TXmlNode) ,  
XmlNodeExtensions.WithAttribute<TXmlNode>(TXmlNode, string) ,  
XmlNodeExtensions.WithAttribute<TXmlNode>(TXmlNode, string, object?) ,  
XmlNodeExtensions.WithAttributes<TXmlNode>(TXmlNode, object?) ,  
XmlNodeExtensions.WithContent<TXmlNode>(TXmlNode, Action<TXmlNode>) ,  
XmlNodeExtensions.WithContent<TXmlNode>(TXmlNode, object?) ,  
XmlNodeExtensions.WithContent<TXmlNode>(TXmlNode, string?)
```

Constructors

[HtmlElement\(\)](#)

Initializes a new instance of the [HtmlElement](#) class with no container HTML element.

[HtmlElement\(string\)](#)

Initializes a new instance of the [HtmlElement](#) class with the specified tag name.

[HtmlElement\(string, Action<HtmlElement>\)](#)

Initializes a new instance of the [HtmlElement](#) class with the specified tag name.

[HtmlElement\(string, object?\)](#)

Initializes a new instance of the [HtmlElement](#) class with the specified tag name and content.

`HtmlElement(string, string?)`

Initializes a new instance of the `HtmlElement` class with the specified tag name and content.

Properties

`Attributes`

Gets or sets the collection of HTML attributes for the element.

`Children`

Gets or sets the collection of child elements within this element.

`classList`

Gets or sets the list of CSS classes for the HTML element. Initializes with an empty list. Use this to apply CSS class names to the element.

`Id`

Gets or sets the ID attribute of the HTML element. Used to uniquely identify the element within the page.

`Name`

Gets or sets the name attribute of the HTML element. The name is used to reference elements in JavaScript, or to reference form data after a form is submitted.

`SelfClosing`

Gets or sets a value indicating whether the element is self-closing.

`Style`

Gets or sets the CSS style object used to render the style attribute.

`TabIndex`

Gets or sets the tab index of the HTML element.

`TagName`

Gets or sets the tag name of the HTML element (e.g., "div", "span").

`TooltipTitle`

Gets or sets the tooltip text to display for the HTML element.

Methods

`Create(string)`

Creates an [HtmlElement](#) from the specified emmet template.

[Create\(string, Action<HtmlElement>\)](#)

Creates an [HtmlElement](#) from the specified emmet template and configures it using the specified action.

[Create\(string, object?, object?, object?\[\], bool\)](#)

Creates an [HtmlElement](#) from the specified emmet template and adds the specified style, attributes, and children.

[Format\(FormattableString\)](#)

Formats the specified HTML string format, escaping the string interpolation pieces.

[Fragment\(Action<HtmlElement>\)](#)

Creates an fragment [HtmlElement](#) with specified self-action.

[Fragment\(params object?\[\]\)](#)

Creates an fragment [HtmlElement](#) with specified children.

[GetAttributes\(\)](#)

Represents the protected method which gets the attributes to be rendered.

[ToString\(\)](#)

Renders the HTML element into a string with optional pretty formatting.

Operators

[operator +\(HtmlElement, object?\)](#)

Constructor HtmlElement

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

HtmlElement()

Initializes a new instance of the [HtmlElement](#) class with no container HTML element.

```
public HtmlElement()
```

HtmlElement(string)

Initializes a new instance of the [HtmlElement](#) class with the specified tag name.

```
public HtmlElement(string tagName)
```

Parameters

tagName [string](#)

The name of the tag to be used for the HTML element. The tag name will be converted to lowercase.

HtmlElement(string, object?)

Initializes a new instance of the [HtmlElement](#) class with the specified tag name and content.

```
public HtmlElement(string tagName, object? content)
```

Parameters

tagName [string](#)

The name of the tag to be used for the HTML element. The tag name will be converted to lowercase.

content [object](#)

Optional parameter that defines content for the creating HTML tag.

HtmlElement(string, string?)

Initializes a new instance of the [HtmlElement](#) class with the specified tag name and content.

```
public HtmlElement(string tagName, string? content)
```

Parameters

tagName [string](#)

The name of the tag to be used for the HTML element. The tag name will be converted to lowercase.

content [string](#)

Optional parameter that defines content for the creating HTML tag.

HtmlElement(string, Action<HtmlElement>)

Initializes a new instance of the [HtmlElement](#) class with the specified tag name.

```
public HtmlElement(string tagName, Action<HtmlElement> content)
```

Parameters

tagName [string](#)

The name of the tag to be used for the HTML element. The tag name will be converted to lowercase.

content [Action](#)<HtmlElement>

Optional parameter that defines content for the creating HTML tag.

Property Attributes

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Attributes

Gets or sets the collection of HTML attributes for the element.

```
public NodeAttributeCollection Attributes { get; set; }
```

Property Value

[NodeAttributeCollection](#)

Property Children

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Children

Gets or sets the collection of child elements within this element.

```
public ICollection<object?> Children { get; set; }
```

Property Value

[ICollection](#)<[object](#)>

Property ClassList

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

ClassList

Gets or sets the list of CSS classes for the HTML element. Initializes with an empty list. Use this to apply CSS class names to the element.

```
public ICollection<string> ClassList { get; set; }
```

Property Value

[ICollection](#)<[string](#)>

Property Id

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Id

Gets or sets the ID attribute of the HTML element. Used to uniquely identify the element within the page.

```
public string? Id { get; set; }
```

Property Value

[string](#)

Property Name

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Name

Gets or sets the name attribute of the HTML element. The name is used to reference elements in JavaScript, or to reference form data after a form is submitted.

```
public string? Name { get; set; }
```

Property Value

[string](#)

Property SelfClosing

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

SelfClosing

Gets or sets a value indicating whether the element is self-closing.

```
public bool SelfClosing { get; set; }
```

Property Value

[bool](#)

Property Style

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Style

Gets or sets the CSS style object used to render the style attribute.

```
public object? Style { get; set; }
```

Property Value

[object](#)

Property TabIndex

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

TabIndex

Gets or sets the tab index of the HTML element.

```
public int? TabIndex { get; set; }
```

Property Value

[int](#)?

Property TagName

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

TagName

Gets or sets the tag name of the HTML element (e.g., "div", "span").

```
public string TagName { get; set; }
```

Property Value

[string](#)

Property TooltipTitle

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

TooltipTitle

Gets or sets the tooltip text to display for the HTML element.

```
public string? TooltipTitle { get; set; }
```

Property Value

[string](#)

Method Create

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Create(string)

Creates an [HtmlElement](#) from the specified emmet template.

```
public static HtmlElement Create(string emmetString)
```

Parameters

emmetString string ↗

Returns

[HtmlElement](#)

Create(string, object?, object?, object?[], bool)

Creates an [HtmlElement](#) from the specified emmet template and adds the specified style, attributes, and children.

```
public static HtmlElement Create(string emmetString, object? style = null, object? attributes = null, object?[]? children = null, bool selfClosing = false)
```

Parameters

emmetString string ↗

The emmet template string.

style object ↗

An object containing style properties to apply to the element. Can be null.

attributes [object](#)

An object containing attribute names and values to apply to the element. Can be null.

children [object](#)[]

An array of objects to put as children of the creating element. Can be null.

selfClosing [bool](#)

An boolean indicating if the creating element should be self-closed or not.

Returns

[HtmlElement](#)

A new [HtmlElement](#) based on the emmet template with the specified style, attributes, and children.

Create(string, Action<HtmlElement>)

Creates an [HtmlElement](#) from the specified emmet template and configures it using the specified action.

```
public static HtmlElement Create(string emmetString, Action<HtmlElement> self)
```

Parameters

emmetString [string](#)

The emmet template string.

self [Action](#)<HtmlElement>

An action to configure the created [HtmlElement](#).

Returns

[HtmlElement](#)

A new [HtmlElement](#) based on the emmet template and configured using the specified action.

Method Format

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Format(FormattableString)

Formats the specified HTML string format, escaping the string interpolation pieces.

```
public static string Format(FormattableString htmlString)
```

Parameters

htmlString FormattableString ↗

The instance of [FormattableString](#).

Returns

string ↗

Method Fragment

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Fragment(params object?[])

Creates an fragment [HtmlElement](#) with specified children.

```
public static HtmlElement Fragment(params object?[] children)
```

Parameters

children [object](#)[]

An array of objects to put as children of the creating fragment.

Returns

[HtmlElement](#)

Fragment(Action<HtmlElement>)

Creates an fragment [HtmlElement](#) with specified self-action.

```
public static HtmlElement Fragment(Action<HtmlElement> action)
```

Parameters

action [Action](#)<[HtmlElement](#)>

An action that defines content for the creating HTML element.

Returns

[HtmlElement](#)

Method GetAttributes

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

GetAttributes()

Represents the protected method which gets the attributes to be rendered.

```
protected virtual IDictionary<string, object?> GetAttributes()
```

Returns

[IDictionary](#)<[string](#), [object](#)>

Method ToString

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

ToString()

Renders the HTML element into a string with optional pretty formatting.

```
public override string ToString()
```

Returns

[string](#)

The rendered HTML string.

Operator operator +

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

operator +(HtmlElement, object?)

```
public static HtmlElement operator +(HtmlElement a, object? b)
```

Parameters

a [HtmlElement](#)

b [object](#)

Returns

[HtmlElement](#)

Class HtmlElementExtensions

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Provides extension methods for [HtmlElement](#) objects.

```
public static class HtmlElementExtensions
```

Methods

[WithClass<THtmlElement>\(THtmlElement, params string\[\]\)](#)

Specifies the HTML element class list of this [HtmlElement](#).

[WithId<THtmlElement>\(THtmlElement, string\)](#)

Specifies the HTML element ID of this [HtmlElement](#).

[WithName<THtmlElement>\(THtmlElement, string\)](#)

Specifies the HTML element name of this [HtmlElement](#).

[WithStyle<THtmlElement>\(THtmlElement, object\)](#)

Adds css styles through the style attribute on this [HtmlElement](#).

Method WithClass

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

WithClass<THtmlElement>(THtmlElement, params string[])

Specifies the HTML element class list of this [HtmlElement](#).

```
public static THtmlElement WithClass<THtmlElement>(this THtmlElement node, params string[]  
classNames) where THtmlElement : HtmlElement
```

Parameters

node THtmlElement

The current [HtmlElement](#).

classNames [string](#) ↗

One or more classes to add to this [HtmlElement](#).

Returns

THtmlElement

The self [HtmlElement](#) object for fluent chaining.

Type Parameters

THtmlElement

The object type which implements [HtmlElement](#).

Method WithId

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

WithId<THtmlElement>(THtmlElement, string)

Specifies the HTML element ID of this [HtmlElement](#).

```
public static THtmlElement WithId<THtmlElement>(this THtmlElement node, string id) where  
THtmlElement : HtmlElement
```

Parameters

node THtmlElement

The current [HtmlElement](#).

id string ↗

The element ID.

Returns

THtmlElement

The self [HtmlElement](#) object for fluent chaining.

Type Parameters

THtmlElement

The object type which implements [HtmlElement](#).

Method WithName

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

WithName<THtmlElement>(THtmlElement, string)

Specifies the HTML element name of this [HtmlElement](#).

```
public static THtmlElement WithName<THtmlElement>(this THtmlElement node, string name) where  
THtmlElement : HtmlElement
```

Parameters

node THtmlElement

The current [HtmlElement](#).

name [string](#) ↗

The element name.

Returns

THtmlElement

The self [HtmlElement](#) object for fluent chaining.

Type Parameters

THtmlElement

The object type which implements [HtmlElement](#).

Method WithStyle

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

WithType<THtmlElement>(THtmlElement, object)

Adds css styles through the style attribute on this [HtmlElement](#).

```
public static THtmlElement WithType<THtmlElement>(this THtmlElement node, object styleObject)  
where THtmlElement : HtmlElement
```

Parameters

node THtmlElement

The current [HtmlElement](#).

styleObject object

The object which contains CSS properties and values to style the component.

Returns

THtmlElement

The self [HtmlElement](#) object for fluent chaining.

Type Parameters

THtmlElement

The object type which implements [HtmlElement](#).

Interface INode

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Represents an renderable node.

```
public interface INode
```

Extension Methods

```
XmlNodeExtensions.SelfClosed<TXmlNode>(TXmlNode) ,  
XmlNodeExtensions.WithAttribute<TXmlNode>(TXmlNode, string) ,  
XmlNodeExtensions.WithAttribute<TXmlNode>(TXmlNode, string, object?) ,  
XmlNodeExtensions.WithAttributes<TXmlNode>(TXmlNode, object?) ,  
XmlNodeExtensions.WithContent<TXmlNode>(TXmlNode, Action<TXmlNode>) ,  
XmlNodeExtensions.WithContent<TXmlNode>(TXmlNode, object?) ,  
XmlNodeExtensions.WithContent<TXmlNode>(TXmlNode, string?)
```

Properties

Attributes

Gets or sets an collection of attributes for this node.

Children

Gets or sets the collection of child elements (object[]) within this node.

SelfClosing

Gets or sets a value indicating whether this XML node is self-closing.

TagName

Gets or sets the tag name of the XML node.

Property Attributes

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Attributes

Gets or sets an collection of attributes for this node.

```
NodeAttributeCollection Attributes { get; set; }
```

Property Value

[NodeAttributeCollection](#)

Property Children

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Children

Gets or sets the collection of child elements ([object](#)) within this node.

```
ICollection<object?> Children { get; set; }
```

Property Value

[ICollection](#)<[object](#)>

Property SelfClosing

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

SelfClosing

Gets or sets a value indicating whether this XML node is self-closing.

```
bool SelfClosing { get; set; }
```

Property Value

[bool](#)

Property TagName

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

TagName

Gets or sets the tag name of the XML node.

```
string TagName { get; set; }
```

Property Value

string ↗

Class MarkdownText

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Represents a Markdown text with settings for rendering.

```
public sealed class MarkdownText
```

Constructors

[MarkdownText\(\)](#)

Initializes a new instance of the [MarkdownText](#) class with default settings and an empty text.

[MarkdownText\(string\)](#)

Initializes a new instance of the [MarkdownText](#) class with the specified Markdown text and default settings.

Properties

[MarkdownSettings](#)

Gets or sets the Markdown settings used for conversion.

[Text](#)

Gets or sets the Markdown text to be converted.

Methods

[ToString\(\)](#)

Converts the Markdown text to HTML using the specified settings.

Constructor MarkdownText

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

MarkdownText()

Initializes a new instance of the [MarkdownText](#) class with default settings and an empty text.

```
public MarkdownText()
```

MarkdownText(string)

Initializes a new instance of the [MarkdownText](#) class with the specified Markdown text and default settings.

```
public MarkdownText(string markdownText)
```

Parameters

markdownText string ↗

The Markdown text to be converted.

Property MarkdownSettings

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

MarkdownSettings

Gets or sets the Markdown settings used for conversion.

```
public CommonMarkSettings MarkdownSettings { get; set; }
```

Property Value

CommonMarkSettings

See Also

[CommonMarkSettings](#)

Property Text

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Text

Gets or sets the Markdown text to be converted.

```
public string Text { get; set; }
```

Property Value

[string](#)

Method ToString

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

ToString()

Converts the Markdown text to HTML using the specified settings.

```
public override string ToString()
```

Returns

[string](#)

The converted HTML text.

See Also

[CommonMark.CommonMarkConverter.Convert\(System.String,CommonMark.CommonMarkSettings\)](#)

Class NodeAttributeCollection

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Represents an dictionary collection of attributes.

```
public class NodeAttributeCollection : IDictionary<string, object?>,
ICollection<KeyValuePair<string, object?>>, IEnumerable<KeyValuePair<string, object?>>,
IEnumerable
```

Implements

[IDictionary](#)<string, object>, [ICollection](#)<KeyValuePair<string, object>>,
[IEnumerable](#)<KeyValuePair<string, object>>, [IEnumerable](#)

Constructors

[NodeAttributeCollection\(\)](#)

Creates an new instance of the [NodeAttributeCollection](#).

Properties

[Count](#)

[IsReadOnly](#)

[this\[string\]](#)

[Keys](#)

[Values](#)

Methods

Add(KeyValuePair<string, object?>)

Add(string, object?)

Clear()

Contains(KeyValuePair<string, object?>)

ContainsKey(string)

CopyTo(KeyValuePair<string, object?>[], int)

GetEnumerator()

Remove(KeyValuePair<string, object?>)

Remove(string)

TryGetValue(string, out object?)

Constructor NodeAttributeCollection

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

NodeAttributeCollection()

Creates an new instance of the [NodeAttributeCollection](#).

```
public NodeAttributeCollection()
```

Property Count

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Count

```
public int Count { get; }
```

Property Value

[int ↗](#)

Property IsReadOnly

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

IsReadOnly

```
public bool IsReadOnly { get; }
```

Property Value

[bool](#) ↗

Property Keys

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Keys

```
public ICollection<string> Keys { get; }
```

Property Value

[ICollection](#)<[string](#)>

Property Values

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Values

```
public ICollection<object?> Values { get; }
```

Property Value

[ICollection](#)<[object](#)>

Property this

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

this[string]

```
public object? this[string key] { get; set; }
```

Parameters

key [string](#) ↗

Property Value

[object](#) ↗

Method Add

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Add(string, object?)

```
public void Add(string key, object? value)
```

Parameters

key string ↗

value object ↗

Add(KeyValuePair<string, object?>)

```
public void Add(KeyValuePair<string, object?> item)
```

Parameters

item KeyValuePair<string ↗, object ↗>

Method Clear

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Clear()

```
public void Clear()
```

Method Contains

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Contains(KeyValuePair<string, object?>)

```
public bool Contains(KeyValuePair<string, object?> item)
```

Parameters

item [KeyValuePair](#)<string, object>

Returns

[bool](#)

Method ContainsKey

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

ContainsKey(string)

```
public bool ContainsKey(string key)
```

Parameters

key string ↗

Returns

bool ↗

Method CopyTo

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

CopyTo(KeyValuePair<string, object?>[], int)

```
public void CopyTo(KeyValuePair<string, object?>[] array, int arrayIndex)
```

Parameters

array KeyValuePair<string, object?>[]

arrayIndex int

Method GetEnumerator

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

GetEnumerator()

```
public IEnumarator<KeyValuePair<string, object?>> GetEnumerator()
```

Returns

[IEnumarator](#)<KeyValuePair<string, object>>

Method Remove

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Remove(string)

```
public bool Remove(string key)
```

Parameters

key [string](#)

Returns

[bool](#)

Remove(KeyValuePair<string, object?>)

```
public bool Remove(KeyValuePair<string, object?> item)
```

Parameters

item [KeyValuePair](#)<[string](#), [object](#)>

Returns

[bool](#)

Method TryGetValue

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

TryGetValue(string, out object?)

```
public bool TryGetValue(string key, out object? value)
```

Parameters

key [string](#)

value [object](#)

Returns

[bool](#)

Class NodeComment

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Represents a node comment in a document.

```
public sealed class NodeComment
```

Constructors

[NodeComment\(string\)](#)

Initializes a new instance of the [NodeComment](#) class with the specified comment.

Properties

[Comment](#)

Gets or sets the comment text.

Methods

[ToString\(\)](#)

Returns a string representation of the comment in the format of an XML comment.

Constructor NodeComment

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

NodeComment(string)

Initializes a new instance of the [NodeComment](#) class with the specified comment.

```
public NodeComment(string comment)
```

Parameters

comment [string](#)

The comment text.

Property Comment

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Comment

Gets or sets the comment text.

```
public string Comment { get; set; }
```

Property Value

[string](#)

Method ToString

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

ToString()

Returns a string representation of the comment in the format of an XML comment.

```
public override string ToString()
```

Returns

[string](#)

A string representation of the comment.

Class RenderableFunction

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Represents an object which their renderable contents is called by an function.

```
public sealed class RenderableFunction
```

Constructors

[RenderableFunction\(Func<object?>\)](#)

Creates an new [RenderableFunction](#) class with the specified function.

Properties

[Callable](#)

Gets or sets the renderable function.

Methods

[ToString\(\)](#)

Invokes [Callable](#) and returns its result as an string.

Constructor RenderableFunction

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

RenderableFunction(Func<object?>)

Creates an new [RenderableFunction](#) class with the specified function.

```
public RenderableFunction(Func<object?> callable)
```

Parameters

callable [Func<object>](#)

The function which will result the contents to be rendered.

Property Callable

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Callable

Gets or sets the renderable function.

```
public Func<object?> Callable { get; set; }
```

Property Value

[Func<object>](#)

Method ToString

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

ToString()

Invokes [Callable](#) and returns its result as an string.

```
public override string? ToString()
```

Returns

[string](#)↗

Class RenderableText

Namespace:[TinyComponents](#)

Assembly:[TinyComponents.dll](#)

Represents an simple renderable text.

```
public sealed class RenderableText
```

Constructors

[RenderableText\(object?\)](#)

Creates an new instance of [RenderableText](#) with the specified text contents from the object, encoding it as an HTML entity.

[RenderableText\(object?, bool\)](#)

Creates an new instance of [RenderableText](#) with the specified text contents from the object.

Properties

[Contents](#)

Gets or sets the contents which this text will render.

[Escape](#)

Gets or sets whether this text should be XML/HTML encoded or not.

Methods

[Raw\(object?\)](#)

Creates an new instance of [RenderableText](#) with the provided raw, unencoded text.

[SafeRenderSubject\(object?\)](#)

Renders the specified object into an safe HTML content.

[ToString\(\)](#)

Renders this [RenderableText](#) into an string.

Constructor RenderableText

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

RenderableText(object?)

Creates an new instance of [RenderableText](#) with the specified text contents from the object, encoding it as an HTML entity.

```
public RenderableText(object? textContents)
```

Parameters

textContents [object](#)

The object which will be encoded.

RenderableText(object?, bool)

Creates an new instance of [RenderableText](#) with the specified text contents from the object.

```
public RenderableText(object? textContents, bool escape)
```

Parameters

textContents [object](#)

The object which will be encoded.

escape [bool](#)

Determines if the text contents should be HTML encoded or not.

Property Contents

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Contents

Gets or sets the contents which this text will render.

```
public object? Contents { get; set; }
```

Property Value

[object](#)

Property Escape

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Escape

Gets or sets whether this text should be XML/HTML encoded or not.

```
public bool Escape { get; set; }
```

Property Value

bool ↗

Method Raw

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Raw([object?](#))

Creates an new instance of [RenderableText](#) with the provided raw, unencoded text.

```
public static RenderableText Raw(object? contents)
```

Parameters

contents [object](#)

The object which will be converted to text.

Returns

[RenderableText](#)

An [RenderableText](#) with the content value.

Method SafeRenderSubject

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

SafeRenderSubject(object?)

Renders the specified object into an safe HTML content.

```
public static string SafeRenderSubject(object? obj)
```

Parameters

obj [object](#)

The object to be rendered.

Returns

[string](#)

Method ToString

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

ToString()

Renders this [RenderableText](#) into an string.

```
public override string ToString()
```

Returns

[string](#) ↗

The string representation of this [RenderableText](#).

Class XmlNode

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Represents an renderable XML node.

```
public class XmlNode : INode
```

Implements

[INode](#)

Extension Methods

```
XmlNodeExtensions.SelfClosed<TXmlNode>(TXmlNode) ,  
XmlNodeExtensions.WithAttribute<TXmlNode>(TXmlNode, string) ,  
XmlNodeExtensions.WithAttribute<TXmlNode>(TXmlNode, string, object?) ,  
XmlNodeExtensions.WithAttributes<TXmlNode>(TXmlNode, object?) ,  
XmlNodeExtensions.WithContent<TXmlNode>(TXmlNode, Action<TXmlNode>) ,  
XmlNodeExtensions.WithContent<TXmlNode>(TXmlNode, object?) ,  
XmlNodeExtensions.WithContent<TXmlNode>(TXmlNode, string?)
```

Constructors

[XmlNode\(string\)](#)

Initializes a new instance of the [XmlNode](#) class with the specified tag name.

[XmlNode\(string, Action<XmlNode>\)](#)

Initializes a new instance of the [XmlNode](#) class with the specified tag name.

[XmlNode\(string, object?\)](#)

Initializes a new instance of the [XmlNode](#) class with the specified tag name and content.

[XmlNode\(string, string\)](#)

Initializes a new instance of the [XmlNode](#) class with the specified tag name and content.

Properties

[Attributes](#)

Gets or sets the collection of attributes for this node.

[Children](#)

Gets or sets the collection of child elements within this node.

[SelfClosing](#)

Gets or sets a value indicating whether this XML node is self-closing.

[TagName](#)

Gets or sets the tag name of the XML node.

Methods

[ToString\(\)](#)

Renders this [XmlNode](#) into it's XML string representation.

Operators

[operator +\(XmlNode, object?\)](#)

[operator +\(XmlNode, string?\)](#)

[operator +\(XmlNode, XmlNode?\)](#)

Constructor XmlNode

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

XmlNode(string)

Initializes a new instance of the [XmlNode](#) class with the specified tag name.

```
public XmlNode(string tagName)
```

Parameters

tagName [string](#)

The name of the tag to be used for the XML element. The tag name will be converted to lowercase.

XmlNode(string, object?)

Initializes a new instance of the [XmlNode](#) class with the specified tag name and content.

```
public XmlNode(string tagName, object? content)
```

Parameters

tagName [string](#)

The name of the tag to be used for the XML element. The tag name will be converted to lowercase.

content [object](#)

Optional parameter that defines content for the creating XML tag.

XmlNode(string, string)

Initializes a new instance of the [XmlNode](#) class with the specified tag name and content.

```
public XmlNode(string tagName, string content)
```

Parameters

tagName [string](#)

The name of the tag to be used for the XML element. The tag name will be converted to lowercase.

content [string](#)

Optional parameter that defines content for the creating XML tag.

XmlNode(string, Action<XmlNode>)

Initializes a new instance of the [XmlNode](#) class with the specified tag name.

```
public XmlNode(string tagName, Action<XmlNode> content)
```

Parameters

tagName [string](#)

The name of the tag to be used for the XML element. The tag name will be converted to lowercase.

content [Action](#)<[XmlNode](#)>

Optional parameter that defines content for the creating XML tag.

Property Attributes

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Attributes

Gets or sets the collection of attributes for this node.

```
public NodeAttributeCollection Attributes { get; set; }
```

Property Value

[NodeAttributeCollection](#)

Property Children

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Children

Gets or sets the collection of child elements within this node.

```
public ICollection<object?> Children { get; set; }
```

Property Value

[ICollection](#)<[object](#)>

Property SelfClosing

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

SelfClosing

Gets or sets a value indicating whether this XML node is self-closing.

```
public bool SelfClosing { get; set; }
```

Property Value

[bool](#)

Property TagName

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

TagName

Gets or sets the tag name of the XML node.

```
public string TagName { get; set; }
```

Property Value

[string](#)

Method ToString

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

ToString()

Renders this [XmlNode](#) into it's XML string representation.

```
public override string ToString()
```

Returns

[string](#) ↗

Operator operator +

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

operator +(XmlNode, XmlNode?)

```
public static XmlNode operator +(XmlNode a, XmlNode? b)
```

Parameters

a XmlNode

b XmlNode

Returns

XmlNode

operator +(XmlNode, object?)

```
public static XmlNode operator +(XmlNode a, object? b)
```

Parameters

a XmlNode

b object

Returns

XmlNode

operator +(XmlNode, string?)

```
public static XmlNode operator +(XmlNode a, string? b)
```

Parameters

a XmlNode

b string↗

Returns

XmlNode

Class XmlNodeExtensions

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

Provides extension methods for [INode](#) objects.

```
public static class XmlNodeExtensions
```

Methods

[SelfClosed<TXmlNode>\(TXmlNode\)](#)

Specifies that this [INode](#) will be self-closed.

[WithAttribute<TXmlNode>\(TXmlNode, string\)](#)

Specifies an attribute for this [INode](#).

[WithAttribute<TXmlNode>\(TXmlNode, string, object?\)](#)

Specifies an attribute for this [INode](#).

[WithAttributes<TXmlNode>\(TXmlNode, object?\)](#)

Specifies multiple attributes for this [INode](#) from an object.

[WithContent<TXmlNode>\(TXmlNode, Action<TXmlNode>\)](#)

Specifies children contents for this [INode](#).

[WithContent<TXmlNode>\(TXmlNode, object?\)](#)

Specifies children contents for this [INode](#).

[WithContent<TXmlNode>\(TXmlNode, string?\)](#)

Specifies children contents for this [INode](#).

Method SelfClosed

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

SelfClosed<TXmlNode>(TXmlNode)

Specifies that this [INode](#) will be self-closed.

```
public static TXmlNode SelfClosed<TXmlNode>(this TXmlNode node) where TXmlNode : INode
```

Parameters

node TXmlNode

The current [INode](#).

Returns

TXmlNode

The self [INode](#) object for fluent chaining.

Type Parameters

TXmlNode

The object type which implements [INode](#).

Method WithAttribute

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

WithAttribute<TXmlNode>(TXmlNode, string)

Specifies an attribute for this [INode](#).

```
public static TXmlNode WithAttribute<TXmlNode>(this TXmlNode node, string value) where TXmlNode : INode
```

Parameters

node TXmlNode

The current [INode](#).

value string ↗

The attribute name and value.

Returns

TXmlNode

The self [INode](#) object for fluent chaining.

Type Parameters

TXmlNode

The object type which implements [INode](#).

WithAttribute<TXmlNode>(TXmlNode, string, object?)

Specifies an attribute for this [INode](#).

```
public static TXmlNode WithAttribute<TXmlNode>(this TXmlNode node, string name, object? value)
where TXmlNode : INode
```

Parameters

node TXmlNode

The current [INode](#).

name [string](#)

The attribute name.

value [object](#)

The attribute value.

Returns

TXmlNode

The self [INode](#) object for fluent chaining.

Type Parameters

TXmlNode

The object type which implements [INode](#).

Method WithAttributes

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

WithAttributes<TXmlNode>(TXmlNode, object?)

Specifies multiple attributes for this [INode](#) from an object.

```
public static TXmlNode WithAttributes<TXmlNode>(this TXmlNode node, object? attributesObject)
where TXmlNode : INode
```

Parameters

node TXmlNode

The current [INode](#).

attributesObject [object](#)

The object containing attribute names and values.

Returns

TXmlNode

The self [INode](#) object for fluent chaining.

Type Parameters

TXmlNode

The object type which implements [INode](#).

Method WithContent

Namespace:[TinyComponents](#)

Assembly:TinyComponents.dll

WithContent<TXmlNode>(TXmlNode, Action<TXmlNode>)

Specifies children contents for this [INode](#).

```
public static TXmlNode WithContent<TXmlNode>(this TXmlNode node, Action<TXmlNode> selector)  
where TXmlNode : INode
```

Parameters

node TXmlNode

The current [INode](#).

selector [Action](#)<TXmlNode>

The action which will be executed in the self [INode](#).

Returns

TXmlNode

The self [INode](#) object for fluent chaining.

Type Parameters

TXmlNode

The object type which implements [INode](#).

WithContent<TXmlNode>(TXmlNode, object?)

Specifies children contents for this [INode](#).

```
public static TXmlNode WithContent<TXmlNode>(this TXmlNode node, object? contents) where  
TXmlNode : INode
```

Parameters

node `TXmlNode`

The current [INode](#).

contents `object` ↗

The children object.

Returns

`TXmlNode`

The self [INode](#) object for fluent chaining.

Type Parameters

`TXmlNode`

The object type which implements [INode](#).

WithContent<TXmlNode>(TXmlNode, string?)

Specifies children contents for this [INode](#).

```
public static TXmlNode WithContent<TXmlNode>(this TXmlNode node, string? contents) where  
TXmlNode : INode
```

Parameters

node `TXmlNode`

The current [INode](#).

contents `string` ↗

The children object.

Returns

TXmlNode

The self [INode](#) object for fluent chaining.

Type Parameters

TXmlNode

The object type which implements [INode](#).