

# 1 System.Net.IAuthenticationModule 2 Interface

3  
4

```
5 [ILASM]  
6 .class interface public abstract IAuthenticationModule  
7  
8 [C#]  
9 public interface IAuthenticationModule
```

## 9 Assembly Info:

- 10 • Name: System
- 11 • Public Key: [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- 12 • Version: 1.0.x.x
- 13 • Attributes:
  - 14 ○ CLSCompliantAttribute(true)

## 15 Summary

16

17 Implemented by types that perform client authentication.

18 **Library:** Networking

19

## 20 Description

21 The **System.Net.IAuthenticationModule** interface defines the  
22 properties and methods that types are required to implement to  
23 handle client authentication. Types that implement this interface are  
24 called authentication modules. In addition to implementing this  
25 interface, an authentication module implements an authentication  
26 protocol, such as the Kerberos protocol. The  
27 **System.Net.IAuthenticationModule.AuthenticationType** property  
28 value is a case-insensitive string that typically indicates the protocol  
29 implemented by the module. Each authentication module registered  
30 with the authentication manager is required to have a unique  
31 **System.Net.IAuthenticationModule.AuthenticationType**. The  
32 following string values are reserved for use by modules implementing  
33 the indicated protocols:

AuthenticationType	Protocol
"basic"	Basic as defined by IETF RFC 2617
"digest"	Digest access as defined by IETF RFC 2617
"kerberos"	Kerberos as defined by IETF RFC 1510

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26

[*Note:* Authentication modules are registered with the authentication manager (**System.Net.AuthenticationManager**) by calling the **System.Net.AuthenticationManager.Register** method. When the authentication manager receives an authentication request, registered authentication modules are given the opportunity to handle the authentication in their **System.Net.IAuthenticationModule.Authenticate** method. Similarly, when a client wishes to avoid waiting for the server to request authentication, it can request preauthentication information to send with a request. If the **System.Net.IAuthenticationModule.CanPreAuthenticate** property of a registered module returns **true**, it is among the modules that are given the opportunity to provide the preauthentication information via the **System.Net.IAuthenticationModule.PreAuthenticate** method.

Not all modules receive all authentication and preauthentication requests. The authentication manager searches for an authentication module by invoking the **System.Net.IAuthenticationModule.Authenticate** or **System.Net.IAuthenticationModule.PreAuthenticate** method of each registered module in the order in which it was registered. Once a module returns a **System.Net.Authorization** instance, indicating that it handles the authentication, the authentication manager terminates the search.]

# IAuthenticationModule.Authenticate(System.String, System.Net.WebRequest, System.Net.ICredentials) Method

```
[ILASM]
.method public hidebysig virtual abstract class
System.Net.Authorization Authenticate(string challenge,
class System.Net.WebRequest request, class
System.Net.ICredentials credentials)

[C#]
Authorization Authenticate(string challenge, WebRequest
request, ICredentials credentials)
```

## Summary

Returns an instance of the **System.Net.Authorization** class that provides a response to an authentication challenge.

## Parameters

Parameter	Description
<i>challenge</i>	A <b>System.String</b> containing the authentication challenge sent by the server. The content of this string is determined by the authentication protocol(s) used by the server that issued the challenge.
<i>request</i>	The <b>System.Net.WebRequest</b> instance that received <i>challenge</i> .
<i>credentials</i>	The credentials of the <b>System.Net.WebRequest</b> instance that received <i>challenge</i> .

## Return Value

A **System.Net.Authorization** instance containing the challenge response, or **null** if the challenge cannot be handled.

## Behaviors

If the authentication module can handle *challenge*, this method proceeds with the authentication in accordance with the authentication protocol implemented by the current instance and returns a **System.Net.Authorization** instance containing the challenge response. If the authentication module cannot handle the challenge, this method returns **null**. If the authentication module encounters an error while conducting the authentication process, this method can, but is not required to throw an exception.

## How and When to Override

1            Implement this method to receive and optionally handle requests for  
2            client authentication from the authentication manager.

### 3    **Usage**

4            The **System.Net.AuthenticationManager** class invokes the  
5            **System.Net.IAuthenticationModule.Authenticate** method on  
6            registered authentication modules to allow modules to handle a server  
7            challenge.

8

# 1 IAuthenticationModule.PreAuthenticate(S 2 ystem.Net.WebRequest, 3 System.Net.ICredentials) Method

```
4 [ILASM]  
5 .method public hidebysig virtual abstract class  
6 System.Net.Authorization PreAuthenticate(class  
7 System.Net.WebRequest request, class  
8 System.Net.ICredentials credentials)  
9  
10 [C#]  
11 Authorization PreAuthenticate(WebRequest request,  
ICredentials credentials)
```

## 12 Summary

13 Returns an instance of the **System.Net.Authorization** class  
14 containing client authentication information.

## 15 Parameters

Parameter	Description
<i>request</i>	The <b>System.Net.WebRequest</b> instance associated with the authentication request.
<i>credentials</i>	The credentials associated with the authentication request.

## 18 19 Return Value

21 A **System.Net.Authorization** instance containing client  
22 authentication information to be sent with *request*, or **null** if the  
23 current instance does not support preauthentication.

## 24 Behaviors

25 This method returns an instance of the **System.Net.Authorization**  
26 class containing authentication information to be sent with the  
27 request. This method is required to return **null** if it cannot handle  
28 preauthentication requests from the authentication manager.

## 29 How and When to Override

30 Implement this method if the protocol implemented by the current  
31 instance supports preauthentication.

## 32 Usage

1 This method is used by the authentication manager to handle requests  
2 by clients for authentication information that will be used to preempt a  
3 server's request for authentication. This method is not called unless  
4 the **System.Net.IAuthenticationModule.CanPreAuthenticate**  
5 property returns **true**.  
6  
7

8

# 1 AuthenticationModule.AuthenticationType 2 e Property

```
3 [ILASM]  
4 .property string AuthenticationType { public hidebysig  
5 virtual abstract specialname string  
6 get_AuthenticationType() }  
  
7 [C#]  
8 string AuthenticationType { get; }
```

## 9 Summary

10 Gets the authentication type of the current instance.

## 11 Property Value

12

13 A **System.String** indicating the authentication type of the current  
14 authentication module.

## 15 Description

16 This property is read-only.

## 17 Behaviors

18 The following string values are reserved for use by modules  
19 implementing the indicated protocols:

AuthenticationType	Protocol
"basic"	Basic as defined by IETF RFC 2617
"digest"	Digest access as defined by IETF RFC 2617
"kerberos"	Kerberos as defined by IETF RFC 1510

20

21

## 22 How and When to Override

23 Implement this property to return a string that identifies all instances  
24 of the current type. The string returned by this property typically  
25 indicates the protocol implemented by the current type. For example,  
26 an authentication module that implements the Digest protocol would  
27 typically return "digest" as its **AuthenticationType**.

## 28 Usage

1 The **System.Net.IAuthenticationModule.AuthenticationType**  
2 property must be unique for all registered authentication modules. The  
3 value of this property is used by the  
4 **System.Net.AuthenticationManager.Register** method to  
5 determine if there is already an authentication module registered for  
6 the type. Modules may also be unregistered by passing the value of  
7 this property to the  
8 **System.Net.AuthenticationManager.Unregister** method.  
9

# 1 IAuthenticationModule.CanPreAuthenticate 2 e Property

```
3 [ILASM]  
4 .property bool CanPreAuthenticate { public hideby sig  
5 virtual abstract specialname bool get_CanPreAuthenticate()  
6 }  
7  
8 [C#]  
9 bool CanPreAuthenticate { get; }
```

## 9 Summary

10 Gets a **System.Boolean** value indicating whether the authentication  
11 module supports preauthentication.

## 12 Property Value

13

14 **true** if the authentication module supports preauthentication;  
15 otherwise **false**.

## 16 Description

17 This property is read-only.

## 18 Behaviors

19 The **System.Net.IAuthenticationModule.CanPreAuthenticate**  
20 property is set to **true** to indicate that the authentication module can  
21 respond with a **System.Net.Authorization** instance when the  
22 **System.Net.IAuthenticationModule.PreAuthenticate** method is  
23 called. This property returns **false** if the protocol implemented by the  
24 current authentication module does not support preauthentication.

## 25 Usage

26 This method is used by the authentication manager to determine  
27 which modules receive requests by clients for authentication  
28 information that will be supplied in anticipation of a server's request  
29 for authentication. This method controls whether the  
30 **System.Net.IAuthenticationModule.PreAuthenticate** method can  
31 be called.

32