

# System.Net.AuthenticationManager Class

```
[ILAsm]
.class public AuthenticationManager extends System.Object

[C#]
public class AuthenticationManager
```

## Assembly Info:

- *Name:* System
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00]
- *Version:* 2.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

## Summary

Manages the authentication modules called during the client authentication process.

## Inherits From: System.Object

**Library:** Networking

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

## Description

The `System.Net.AuthenticationManager` class manages authentication modules that are responsible for client authentication. [*Note:* The `System.Net.AuthenticationManager` is called by `System.Net.WebRequest` instances to provide information that is sent to servers to authenticate the client. The authentication process might consist of requests to an authentication server separate from the resource server, as well as any other activities required to properly authenticate a client.]

The `System.Net.AuthenticationManager` queries registered authentication modules by calling the `System.Net.IAuthenticationModule.Authenticate` method for each module. The first authentication module that returns a `System.Net.Authorization` instance is used to authenticate the request. An authentication module, which can be any object that implements the `System.Net.IAuthenticationModule` interface, is registered using the `System.Net.AuthenticationManager.Register` method. Authentication modules are called in the order in which they are registered.

Applications typically do not access this type directly; it provides authentication services for the `System.Net.WebRequest` type.

# 1 2 AuthenticationManager.Authenticate(System. 3 String, System.Net.WebRequest, 4 System.Net.ICredentials) Method

```
5 [ILAsm]  
6 .method public hidebysig static class System.Net.Authorization  
7 Authenticate(string challenge, class System.Net.WebRequest request, class  
8 System.Net.ICredentials credentials)  
  
9 [C#]  
10 public static Authorization Authenticate(string challenge, WebRequest  
11 request, ICredentials credentials)
```

## 12 Summary

13 Calls registered authentication modules to find a module that responds to the specified  
14 authentication challenge.

## 15 Parameters

Parameter	Description
<i>challenge</i>	A <code>System.String</code> containing the challenge returned by an Internet host. The content of this string is determined by the authentication protocol(s) used by the server that issued the challenge.
<i>request</i>	The <code>System.Net.WebRequest</code> that received <i>challenge</i> .
<i>credentials</i>	The <code>System.Net.ICredentials</code> associated with <i>request</i> . [Note: The <code>System.Net.WebRequest.Credentials</code> property of <i>request</i> is used to supply this argument.]

## 16 17 Return Value

18 A `System.Net.Authorization` instance containing the response from the authentication  
19 module, or `null` if no authentication module responded to the challenge.

20  
21 Applications do not call this method; it is called by `System.Net.WebRequest` instances.

## 22 Description

23 The `System.Net.AuthenticationManager.Authenticate` method invokes the  
24 `System.Net.IAuthenticationModule.Authenticate` method of each registered  
25 authentication module until one of the modules returns a `System.Net.Authorization`  
26 instance. Authentication modules are called in the order in which they were registered

1 via the `System.Net.AuthenticationManager.Register` method.

2

3 [*Note:* The `System.Net.Authorization.Message` property of the object returned by this  
4 method contains the client's response to the server challenge contained in *challenge*.]

5

6

## 7 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>challenge</i> , <i>request</i> , or <i>credentials</i> is null.

8

9

# AuthenticationManager.PreAuthenticate(System.Net.WebRequest, System.Net.ICredentials) Method

```
[ILAsm]
.method public hidebysig static class System.Net.Authorization
PreAuthenticate(class System.Net.WebRequest request, class
System.Net.ICredentials credentials)

[C#]
public static Authorization PreAuthenticate(WebRequest request,
ICredentials credentials)
```

## Summary

Attempts to obtain a `System.Net.Authorization` instance used to initiate client authentication.

## Parameters

Parameter	Description
<i>request</i>	A <code>System.Net.WebRequest</code> containing a request for access to a resource.
<i>credentials</i>	The <code>System.Net.ICredentials</code> associated with <i>request</i> .

## Return Value

A `System.Net.Authorization` instance if an authentication module can provide authentication information to be sent with *request*; otherwise, `null`. If *credentials* is `null`, this method returns `null`.

## Description

The `System.Net.AuthenticationManager.PreAuthenticate` method invokes the `System.Net.IAuthenticationModule.PreAuthenticate` method of each registered authentication module until one of the modules returns a `System.Net.Authorization` instance. Authentication modules are called in the order in which they were registered via the `System.Net.AuthenticationManager.Register` method.

[*Note:* The `System.Net.Authorization` instance contains the information that will be sent by a client to initiate authentication instead of waiting for the server to request it. Authentication modules that support preauthentication allow clients to improve server efficiency by avoiding extra round trips caused by authentication challenges.]

1  
2 [Note: If an authorization module supports preauthentication of requests, its  
3 `System.Net.IAuthenticationModule.CanPreAuthenticate` property returns `true`.  
4 Note that preauthentication requires that an authentication of the client has already  
5 occurred. The information obtained from the initial authentication is used to provide the  
6 `System.Net.Authorization.Message` that is sent to the server as an authentication  
7 header in *request*.]  
8  
9

## 10 Exceptions

Exception	Condition
<b>System.ArgumentNullException</b>	<i>request</i> is null.

11

12

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  
27  
28  
29  
30  
31

# AuthenticationManager.Register(System.Net.IAuthenticationModule) Method

```
[ILAsm]  
.method public hidebysig static void Register(class  
System.Net.IAuthenticationModule authenticationModule)  
  
[C#]  
public static void Register(IAuthenticationModule authenticationModule)
```

## Summary

Adds an authentication module to the list of registered authentication modules managed by the authentication manager.

## Parameters

Parameter	Description
<i>authenticationModule</i>	The System.Net.IAuthenticationModule to register.

## Description

The System.Net.AuthenticationManager.Register method adds an authentication module to the end of the list of modules managed by the authentication manager. Each registered module is required to have a unique System.Net.IAuthenticationModule.AuthenticationType. If a module with the same System.Net.IAuthenticationModule.AuthenticationType is already registered, this method removes the registered module, and adds *authenticationModule* to the end of the list. [Note: Authentication modules are called in the order in which they were added to the list.]

[Note: To remove a module, call one of the System.Net.AuthenticationManager.Unregister methods.]

## Exceptions

Exception	Condition
System.ArgumentNullException	<i>authenticationModule</i> is null.



1  
2 **AuthenticationManager.Unregister(System.Ne**  
3 **t.IAuthenticationModule) Method**

```
4 [ILAsm]  
5 .method public hidebysig static void Unregister(class  
6 System.Net.IAuthenticationModule authenticationModule)  
  
7 [C#]  
8 public static void Unregister(IAuthenticationModule authenticationModule)
```

9 **Summary**

10 Removes the specified authentication module from the list of registered modules.

11 **Parameters**

Parameter	Description
<i>authenticationModule</i>	The System.Net.IAuthenticationModule module to remove.

12  
13 **Description**

14 [Note: To add an authentication module to the list of registered modules, call the  
15 System.Net.AuthenticationManager.Register method.]  
16  
17

18 **Exceptions**

Exception	Condition
<b>System.ArgumentNullException</b>	<i>authenticationModule</i> is null.
<b>System.InvalidOperationException</b>	<i>authenticationModule</i> is not a registered authentication module.

19

20

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

# AuthenticationManager.Unregister(System.String) Method

```
[ILAsm]  
.method public hidebysig static void Unregister(string  
authenticationScheme)  
  
[C#]  
public static void Unregister(string authenticationScheme)
```

## Summary

Removes the authentication module with the specified authentication type from the list of registered modules.

## Parameters

Parameter	Description
<i>authenticationScheme</i>	A System.String containing the authentication type of the module to remove.

## Description

The *authenticationScheme* is required to match the value returned by the System.Net.IAuthenticationModule.AuthenticationType property of a registered authentication module.

[Note: To add an authentication module to the list of registered modules, call the System.Net.AuthenticationManager.Register method.]

## Exceptions

Exception	Condition
System.ArgumentNullException	<i>authenticationScheme</i> is null.
System.InvalidOperationException	There is no registered module with the <i>authenticationScheme</i> authentication type.

# 1 AuthenticationManager.RegisteredModules 2 Property

```
3 [ILAsm]  
4 .property class System.Collections.IEnumerator RegisteredModules { public  
5 hidebyref static specialname class System.Collections.IEnumerator  
6 get_RegisteredModules() }  
7 [C#]  
8 public static IEnumerable RegisteredModules { get; }
```

## 9 Summary

10 Gets a list of registered authentication modules.

## 11 Property Value

12 A `System.Collections.IEnumerator` that provides access to the list of registered  
13 authentication modules.

## 14 Description

15 This property is read-only.

16  
17 [*Note:* The `System.Net.AuthenticationManager.Register` method adds modules to  
18 the list of registered authentication modules, and the  
19 `System.Net.AuthenticationManager.Unregister` method removes modules from it.]  
20  
21

22