

# 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: 1.0.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 may 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

1 are registered.  
2  
3 Applications typically do not access this type directly; it provides  
4 authentication services for the **System.Net.WebRequest** type.  
5

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

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

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

## Summary

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

## Parameters

Parameter	Description
<i>challenge</i>	A <b>System.String</b> 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 <b>System.Net.WebRequest</b> that received <i>challenge</i> .
<i>credentials</i>	The <b>System.Net.ICredentials</b> associated with <i>request</i> . [Note: The <b>System.Net.WebRequest.Credentials</b> property of <i>request</i> is used to supply this argument.]

## Return Value

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

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

## Description

The **System.Net.AuthenticationManager.Authenticate** method invokes the **System.Net.IAuthenticationModule.Authenticate** 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

1 registered via the **System.Net.AuthenticationManager.Register**  
2 method.

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

7 **Exceptions**

8  
9

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

10  
11  
12

# 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 <b>System.Net.WebRequest</b> containing a request for access to a resource.
<i>credentials</i>	The <b>System.Net.ICredentials</b> 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

1 that support preauthentication allow clients to improve server  
2 efficiency by avoiding extra round trips caused by authentication  
3 challenges.]  
4  
5 [Note: If an authorization module supports preauthentication of  
6 requests, its  
7 **System.Net.IAuthenticationModule.CanPreAuthenticate** property  
8 returns **true**. Note that preauthentication requires that an  
9 authentication of the client has already occurred. The information  
10 obtained from the initial authentication is used to provide the  
11 **System.Net.Authorization.Message** that is sent to the server as an  
12 authentication header in *request*.]

### 13 Exceptions

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

# 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 <b>System.Net.IAuthenticationModule</b> 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
<b>System.ArgumentNullException</b>	<i>authenticationModule</i> is <b>null</b> .

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

```
[ILASM]
.method public hidebysig static void Unregister(class
System.Net.IAuthenticationModule authenticationModule)

[C#]
public static void Unregister(IAuthenticationModule
authenticationModule)
```

## Summary

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

## Parameters

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

## Description

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

## Exceptions

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



# 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 <b>System.String</b> 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
<b>System.ArgumentNullException</b>	<i>authenticationScheme</i> is <b>null</b> .
<b>System.InvalidOperationException</b>	There is no registered module with the <i>authenticationScheme</i> authentication type.

# 1 AuthenticationManager.RegisteredModule 2 s Property

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

## 10 Summary

11 Gets a list of registered authentication modules.

## 12 Property Value

13

14 A **System.Collections.IEnumerator** that provides access to the list  
15 of registered authentication modules.

## 16 Description

17 This property is read-only.

18

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

23