.NET Framework 4 - ASP.NET

**ASP.NET Authorization**

Authorization determines whether an identity should be granted access to a specific resource. In ASP.NET, there are two ways to authorize access to a given resource:

* **File authorization** File authorization is performed by the [FileAuthorizationModule](http://msdn.microsoft.com/en-us/library/system.web.security.fileauthorizationmodule.aspx). It checks the access control list (ACL) of the .aspx or .asmx handler file to determine whether a user should have access to the file. ACL permissions are verified for the user's Windows identity (if Windows authentication is enabled) or for the Windows identity of the ASP.NET process. For more information, see [ASP.NET Impersonation](http://msdn.microsoft.com/en-us/library/xh507fc5.aspx).
* **URL authorization** URL authorization is performed by the [UrlAuthorizationModule](http://msdn.microsoft.com/en-us/library/system.web.security.urlauthorizationmodule.aspx), which maps users and roles to URLs in ASP.NET applications. This module can be used to selectively allow or deny access to arbitrary parts of an application (typically directories) for specific users or roles.

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifUsing URL Authorization

With URL authorization, you explicitly allow or deny access to a particular directory by user name or role. To do so, you create an **authorization** section in the configuration file for that directory. To enable URL authorization, you specify a list of users or roles in the [allow](http://msdn.microsoft.com/en-us/library/acsd09b0.aspx) or [deny](http://msdn.microsoft.com/en-us/library/8aeskccd.aspx) elements of the [authorization](http://msdn.microsoft.com/en-us/library/8d82143t.aspx) section of a configuration file. The permissions established for a directory also apply to its subdirectories, unless configuration files in a subdirectory override them.

The following shows the syntax for the **authorization** section:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl26_ctl00_ctl03_code');" \o "Copy Code)

<authorization>

<[allow|deny] users roles verbs />

</authorization>

The **allow** or **deny** element is required. You must specify either the **users** or the **roles** attribute. Both can be included, but both are not required. The **verbs** attribute is optional.

The **allow** and **deny** elementsgrant and revoke access, respectively. Each element supports the attributes shown in the following table:

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| **users** | Identifies the targeted identities (user accounts) for this element.  Anonymous users are identified using a question mark (?). You can specify all authenticated users using an asterisk (\*). |
| **roles** | Identifies a role (a [RolePrincipal](http://msdn.microsoft.com/en-us/library/system.web.security.roleprincipal.aspx) object) for the current request that is allowed or denied access to the resource. For more information, see [Managing Authorization Using Roles](http://msdn.microsoft.com/en-us/library/9ab2fxh0.aspx). |
| **verbs** | Defines the HTTP verbs to which the action applies, such as **GET**, **HEAD**, and **POST**. The default is "\*", which specifies all verbs. |

The following example grants access to the Kim identity and members of the Admins role, and denies access to the John identity (unless the John identity is included in the Admins role) and to all anonymous users:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl26_ctl00_ctl06_code');" \o "Copy Code)

<authorization>

<allow users="Kim"/>

<allow roles="Admins"/>

<deny users="John"/>

<deny users="?"/>

</authorization>

The following **authorization** section shows how to allow access to the John identity and deny access to all other users:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl26_ctl00_ctl07_code');" \o "Copy Code)

<authorization>

<allow users="John"/>

<deny users="\*"/>

</authorization>

You can specify multiple entities for both the **users** and **roles** attributes by using a comma-separated list, as shown in the following example:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl26_ctl00_ctl08_code');" \o "Copy Code)

<allow users="John, Kim, contoso\Jane"/>

Note that if you specify a domain account name, the name must include both the domain and user name (contoso\Jane).

The following example allows all users to perform an HTTP **GET** for a resource, but allows only the Kim identity to perform a **POST** operation:

[Copy Code](javascript:CopyCode('ctl00_MTCS_main_ctl26_ctl00_ctl09_code');" \o "Copy Code)

<authorization>

<allow verbs="GET" users="\*"/>

<allow verbs="POST" users="Kim"/>

<deny verbs="POST" users="\*"/>

</authorization>

Rules are applied as follows:

* Rules contained in application-level configuration files take precedence over inherited rules. The system determines which rule takes precedence by constructing a merged list of all rules for a URL, with the most recent rules (those nearest in the hierarchy) at the head of the list.
* Given a set of merged rules for an application, ASP.NET starts at the head of the list and checks rules until the first match is found. The default configuration for ASP.NET contains an <allow users="\*"> element, which authorizes all users. (By default, this rule is applied last.) If no other authorization rules match, the request is allowed. If a match is found and the match is a **deny** element, the request is returned with the 401 HTTP status code. If an **allow** element matches, the module allows the request to be processed further.

In a configuration file, you can also create a [location](http://msdn.microsoft.com/en-us/library/b6x6shw7.aspx) element to specify a particular file or directory to which settings in that the **location** element should apply.