ENTAAHttpModule

Matthew Harshbarger
2016/05/18 16:08

Table of Contents

HttpModule

This application uses modified version of the ENTAA HttpModule. The ENTAA HttpModule handles authentication, and usually stores use information in the IIS Cache. Cookies are used to identify the current user and set the Context User.

How it Worked Before

1. **The AuthenticateRequest Event is Received and the Module Checks for a Cookie or Request Token**
    1. If a token is found in the request, it compares it with the token value in the cookie.
        - If the token does not match the cookie or the cookie is empty, the ENTAA Client attempts to authenticate the user.
    2. If a user is authenticated, the module sets a cookie, the token's IIS cache value, and Context User
        - The cookie contains the tokenId and is used to locate and authenticate the user in the IIS Cache for subsequent requests.

How it Works in the New Module

The IIS Cache and Cookie have been eliminated, sessions are used to store the ENTAA information.

1. **The PostAcquireRequestState Event is Received and the Module Checks for a new ENTAA Token in the request, or a session user.**
    1. If a token is found in the request, it compares it with the token value in session.
        - If the token does not match the session or the session token does not exist the ENTAA Client attempts to authenticate the user.
    2. If a user exists in session the module sets session values and the context user.
    3. The application can now easily edit the session contents to add custom privileges or sign the user out.

The ENTAA section of the Web.config has been updated as follows:

Parameter	Changes	Description
appid	Expanded Purpose	Your ENTAA Application ID, used to identify the application to ENTAA, and now also used to identify the application to the session storage database.
host	None	ENTAA service host
return	None	ENTAA return URL
setcontextuser	**New**	true or false - determines if the User Principle object should be set from the session, default value is **true**
sessionident	**New**	A unique identifier for the application. This is different from the appid setting and is used when checking/storing session data to prevent session collision cross site when the sessionstate value is set to keep. **This is required if the sessionstate is set to keep.**. If not specified, the ENTAA APPID will be used instead.
sessionstate	**New**	Used to configure the session retention cross website. This is an **optional setting** and may ignored if you do not want to share non-entaa values from user sessions. • keep - Session data is always retained even if the sessionident does not match. For example, given Website A and Website B, if you want to share non-ENTAA user session data accross sites you would set this to **keep**. If you choose to keep session information, be aware that collisions are possible if Website A uses a session value differently from Website B. It is recommended to leave this value off unless you have a specific business need.

bypass	None	true or false - determines if authentication will be avoided and the use of a mock user data be used - ***ALWAYS SET TO FALSE IN PRODUCTION***
bypass.id	None	user id - the id of the mock user account you want to create - ex: test.user@iowaid, test.user@iowa.gov, etc.
bypass.email	None	email - the email of the mock user account you want to create - ex: test.user@gmail.com, test.user@iowa.gov. etc.
bypass.fname	None	first name - the first name of the mock user account you want to create - ex: test, bob, kate, etc.
bypass.lname	None	last name - the last name of the mock user account you want to create - ex: user, smith, johnson, etc.
bypass.privileges	None	privilege list - a comma separated list of privilege values you want to inject for the bypass user account - these values should match privileges you have or intend to set up in ENTAA via ENTAA admin screens - The values are case sensitive so they must match what you plan to set up in ENTAA - ex: ADMIN, STAFF, fileclerk, etc.

Update To New Module

If you choose to implement the new module here are a few things you may experience and how to address them.

1. The names of the old DLLs are not the same as the new ones you you will need to remove the old DLLs from your project and add the new ones.
    - OLD DLL NAMES: iowa.dasite.entaa.aspnet.dll, iowa.dasite.entaa.client.dll and iowa.dasite.mvcnet.core.dll
    - NEW DLL NAMES: iowa.entaa.client.dll, iowa.entaa.security.dll and iowa.entaa.session.dll
2. Namespace changed for some of the classes so you will need to change the imports.
    - OLD NAMESPACE: ex: iowa.dasite.mvcnet.core.security, iowa.dasite.entaa.aspnet.config, iowa.dasite.entaa.client
    - NEW NAMESPACE: ex: iowa.entaa.security, iowa.entaa.session.config, iowa.entaa.client
3. ActionResult [ANNOTATION] changed for AllowAnonymous security filter
    - OLD [ANNOTATION]: [DASAllowAnonymous]
    - NEW [ANNOTATION]: [iowa.entaa.security.AllowAnonymous] - NOTE: you cannot use [AllowAnonymous] as it conflicts with our security framework
4. Log4Net was removed as a dependency of the new DLLs which required an old version of log4net (v1.2.10.0). The old DLLs did some logging that the new ones will no longer do. If your app already has the old log4net DLL and you just want to keep using it things should continue to be fine. However if you choose to upgrade to a newer version of log4net you may encounter some errors depending on how you implemented log4net.
    -
5.