

Grouper web services - authentication - custom authn

| | | | | | |
|---------------------------|----------------------------------|--------------------------------|---|-------------------------------------|----------------------------------|
| Wiki Home | Download Grouper | Grouper Guides | Community Contributions | Developer Resources | Deployment Guide |
|---------------------------|----------------------------------|--------------------------------|---|-------------------------------------|----------------------------------|

If you want custom authentication (e.g. pass in a token, and decode it), then implement the interface `edu.internet2.middleware.grouper.ws.security.WsCustomAuthentication` and configure your fully qualified classname in the `grouper-ws.properties`. The default is an implementation of this interface as an example: `edu.internet2.middleware.grouper.ws.security.WsGrouperDefaultAuthentication`, which just gets the user from the container: `HttpServletRequest.getUserPrincipal().getName()`

```
/**
 * <pre>
 * implement this interface and provide the class to the classpath and grouper-ws.properties
 * to override the default of HttpServletRequest.getUserPrincipal();
 * for non-Rampart authentication
 *
 * if user is not found, throw a runtime exception. Could be WsInvalidQueryException
 * which is a type of runtime exception (experiment and see what you want the response to
 * look like)
 *
 * </pre>
 */
public interface WsCustomAuthentication {

    /**
     * retrieve the current username (subjectId) from the request object.
     * @param HttpServletRequest
     * @return the logged in username (subjectId)
     * @throws WsInvalidQueryException if there is a problem
     */
    public String retrieveLoggedInSubjectId(HttpServletRequest httpServletRequest)
        throws WsInvalidQueryException;

}
```

Manage users

Manage users in the implemented system

Configure

Note the file locations in the container are listed in the [v2.5 container documentation](#)

| File | Value | Description |
|---|---|--|
| <code>grouper.hibernate.properties</code> | <code>grouper.is.ws.basicAuthn=false</code> | This is the default provided with container, do not overlay |
| <code>web.xml</code> | Should be an empty element | This is the default provided with container, do not overlay |
| <code>server.xml</code> | ajp 8009 connector element: <code>tomcatAuthentication="false"</code> | This is the default provided with container, do not overlay Tomcat is not doing authn so that attribute needs to be false |

| | | |
|-----------------------|---|---|
| grouper-ws.properties | <pre> # to provide custom authentication (instead of the default HttpServletRequest.getUserPrincipal() # for non-Rampart authentication. Class must implement the interface: # edu.internet2.middleware.grouper.ws.security. WsCustomAuthentication MyAuthenticator # class must be fully qualified. e.g. edu.school.whatever. # blank means use default: edu.internet2.middleware.grouper.ws. security.WsGrouperDefaultAuthentication ws.security.non-rampart.authentication.class = edu.school.edu. authn.WhateverImpl </pre> | Overlay the grouper-ws.properties or configure in the database. |
| whateverCustom.jar | copy to: /opt/grouper/grouperWebapp/WEB-INF/libWs | WS jars go in that directory |
| grouper-www.conf | Do not have any authn directives here | This is the default provided with container, do not overlay |