

APS Web Tools

User Guide

Version: 1.0.0

Author: Tommy Svensson

Copyright © 2012 Natusoft AB

Table of Contents

1 APSWebTools	1
1.1 APIs	1

1 APSWebTools

This is not an OSGi bundle! This is a plain jar containing utilities for web applications. Specifically APS administration web applications. This jar has to be included in each web application that wants to use it.

Among other things it provides support for being part of the APS administration web login (APSAdminWebLoginHandler). Since the APS administration web is built using Vaadin it has Vaadin support classes. APSVaadinOSGiApplication is a base class used by all APS administration webs.

1.1 APIs

The following are the APIs for a few selected classes. The complete javadoc for this library can be found at <http://apidoc.natusoft.se/APSWebTools/>.

public class **APSAdminWebLoginHandler** extends APSLoginHandler implements APSLoginHandler.HandlerInfo [se.natusoft.osgi.aps.tools.web] {

This is a login handler to use by any admin web registering with the *APSAdminWeb* to validate that there is a valid login available.

public APSAdminWebLoginHandler(BundleContext context)

Creates a new *APSAdminWebLoginHandler*.

Parameters

context - The bundle context.

public void setSessionIdFromRequestCookie(HttpServletRequest request)

Sets the session id from a cookie in the specified request.

Parameters

request - The request to get the session id cookie from.

public void setSessionIdFromRequestCookie(CookieTool.CookieReader cookieReader)

Sets the session id from a cookie in the specified request.

Parameters

cookieReader - The cookie reader to get the session id cookie from.

public void saveSessionIdOnResponse(HttpServletResponse response)

Saves the current session id on the specified response.

Parameters

response - The response to save the session id cookie on.

public void saveSessionIdOnResponse(CookieTool.CookieWriter cookieWriter)

Saves the current session id on the specified response.

Parameters

cookieWriter - The cookie writer to save the session id cookie on.

```
}
```

```
public class APSLLoginHandler implements LoginHandler [se.natusoft.osgi.aps.tools.web] {
```

This class validates if there is a valid logged in user and also provides a simple login if no valid logged in user exists.

This utility makes use of APSAuthService to login auth and APSSessionService for session handling. Trackers for these services are created internally which requires the shutdown() method to be called when no longer used to cleanup.

The bundle needs to import the following packages for this class to work:

```
se.natusoft.osgi.aps.api.auth.user;version="[0.9,2)",
se.natusoft.osgi.aps.api.misc.session;version="[0.9,2)"
```

protected void setHandlerInfo(HandlerInfo handlerInfo)

Sets the handler info when not provided in constructor.

Parameters

handlerInfo - The handler info to set.

public void shutdown()

Since this class internally creates and starts service trackers this method needs to be called on shutdown to cleanup!

public String getLoggedInUser()

This returns the currently logged in user or null if none are logged in.

public boolean hasValidLogin()

Returns true if this handler sits on a valid login.

public boolean login(String userId, String pw)

Logs in with a userid and a password.

This method does not use or modify any internal state of this object! It only uses the APSAuthService that this object sits on. This allows code sitting on an instance of this class to use this method for validating a user without having to setup its own service tracker for the *APSAuthService* when this object is already available due to the code also being an *APSAdminWeb* member. It is basically a convenience.

Returns

true if successfully logged in, false otherwise.

Parameters

userId - The id of the user to login.

pw - The password of the user to login.

public boolean login(String userId, String pw, String requiredRole)

Logs in with a userid and a password, and a required role.

This method does not use or modify any internal state of this object! It only uses the APSAuthService that this object sits on. This allows code sitting on an instance of this class to use this method for validating a user without having to setup its own service tracker for the *APSAuthService* when this object is already available due to the code also being an *APSAdminWeb* member. It is basically a convenience.

Returns

a valid User object on success or null on failure.

Parameters

userId - The id of the user to login.

pw - The password of the user to login.

requiredRole - If non null the user is required to have this role for a successful login. If it doesn't null will

```
public static interface HandlerInfo    [se.natusoft.osgi.aps.tools.web] {
```

Config values for the login handler.

String getSessionId()

Returns

An id to an APSSessionService session.

void setSessionId(String sessionId)

Sets a new session id.

Parameters

sessionId - The session id to set.

String getSessionName()*Returns*

The name of the session data containing the logged in user if any.

String getRequiredRole()*Returns*

The required role of the user for it to be considered logged in.

}

public interface **LoginHandler** [se.natusoft.osgi.aps.tools.web] {

This is a simple API for doing a login.

public boolean isValidLogin()

Returns true if this handler sits on a valid login.

boolean login(String userId, String pw)

Logs in with a userid and a password.

Returns

true if successfully logged in, false otherwise.

Parameters

userId - The id of the user to login.

pw - The password of the user to login.

public void shutdown()

If the handler creates service trackers or other things that needs to be shutdown when no longer used this method needs to be called when the handler is no longer needed.

}
