

# RemoteUserInternalAuthnConfiguration

Current File(s): *conf/authn/remoteuser-internal-authn-config.xml*

Format: Native Spring

- [Overview](#)
- [General Configuration](#)
- [Reference](#)
  - [Beans](#)
- [Notes](#)

## Overview

The authn/RemoteUserInternal login flow relies on whatever container-based mechanism you have available (HTTP BASIC auth, LDAP, Kerberos, other SSO systems, etc.). It is particularly friendly to non-browser profiles such as [ECP](#). By default, this flow is configured without support for advanced authentication controls like passive or forced authentication.

The difference between this flow and the [RemoteUser](#) flow is that this flow doesn't redirect to a protected path; rather, the path of the requested profile flow has to be protected, which will trigger as soon as the client makes its first request. This is primarily suited to the use of basic-authentication and non-browser clients, though of course this will depend on the exact mechanism involved. Using a second SSO mechanism is likely to be incompatible with non-browser clients and the flow descriptor in *authn/general-authn.xml* should be adjusted to reflect this.

The main disadvantage of using this flow for browser use cases is that it will perform the request for authentication without having a chance to determine if the request will succeed, which may be undesirable from a usability perspective.

## General Configuration

Use *authn/remoteuser-internal-authn-config.xml* to configure this flow. Various beans are defined to control how the user identity is extracted from the HTTP request, various transforms to perform on the resulting name prior to final evaluation, and rules for evaluating the name, such as whitelisting, blacklisting, or a matching expression. Simple echoing of the extracted REMOTE\_USER value requires no changes. See the reference below for a complete list.

## Reference

### Beans

Bean ID	Type	Default	Function
shibboleth.authn.RemoteUser.checkRemoteUser	Boolean	true	Whether to check REMOTE_USER for a username
shibboleth.authn.RemoteUser.checkAttributes	List<String>		A list of servlet request attributes to check for a username
shibboleth.authn.RemoteUser.checkHeaders	List<String>		A list of request headers to check for a username
shibboleth.authn.RemoteUser.Lowercase	Boolean	false	Whether to lowercase the username
shibboleth.authn.RemoteUser.Uppercase	Boolean	false	Whether to uppercase the username
shibboleth.authn.RemoteUser.Trim	Boolean	true	Whether to trim leading and trailing whitespace from the username
shibboleth.authn.RemoteUser.Transforms	List<Pair<String, String>>		Pairs of regular expressions and replacement expressions to apply to the username
shibboleth.authn.RemoteUser.whitelistedUsernames	List<String>		A list of usernames to accept (blocking all others)
shibboleth.authn.RemoteUser.blacklistedUsernames	List<String>		A list of usernames to reject (accepting all others)
shibboleth.authn.RemoteUser.matchExpression	java.util.regex.Pattern		A regular expression that must match the username
shibboleth.authn.RemoteUser.resultCachingPredicate	Predicate< <a href="#">ProfileRequestContext</a> >		An optional bean that can be defined to control whether to preserve the authentication result in an IdP session
shibboleth.authn.RemoteUser.addDefaultPrincipals <sup>3.2</sup>	Boolean	true	Whether to add the content of the <code>supportedPrincipals</code> property of the underlying flow descriptor to the resulting Subject

## Notes

TBD