Design Guide - Authorization Delegates

Overview

In order to enforce policies, your authorization delegate must figure out what to do when asked for certain permissions. In some cases more than one permission is requested of the authorization delegate in order to complete the overall ModeShape operation. In addition, some Fedora operations, such as Container creation, involve several ModeShape operations. All of the ModeShape operations that compose a Fedora API call are performed in a single session with one save point and will succeed or fail together.

Permission checks often rely on the retrieval of metadata about existing nodes.

Repository changes, such as new role assignments, that are being made in an unsaved session are not effective for permission checks, which do not have access to unsaved sessions.

ModeShape Actions

action	context path	notes
add_node	path to the new node	
set_property	path to the property	
read	path to the node	
remove	path to the node	
remove_child_nodes	path to parent node	

Cascading Permissions and Removing Sub-Trees

An attempt to remove a node will trigger a call with the remove action on the subject node and call with the remove_child_nodes action on the parent node. Both must return true for the operation to proceed. If your authorization delegate needs to enforce deletes in a cascading fashion, as when using access roles, then the "remove" action must include the permissions check of remove on descendant nodes. (See AbstractRoleBasedAuthorizationDelegate for an example)

Roles-Aware Authorization Delegates

There is a convenient abstract class for those implementing policy enforcement points that need to be aware of access roles. Extending the AbstractRolesAuthorizationDelegate class can mean having to write only a single method.