SecurityContext

Myfaces provides an expression language extension that specializes on the application security. By the help of SecurityContextVariableResolver and the SecurityContextPropertyResolver it's easy to retrieve information from the underlying authentication/authorization mechanism that is used in the application.

Following are the current features

1. **#{securityContext.authType}** : Gives the name of authentication mechanism used like BASIC, FORM or etc. 2. **#{securityContext.remoteUser}** : Returns the name of the current authenticated user 3. **#{securityContext.ifGranted['rolename']}** : If the user is in the role "rolename", returns true or vice versa 4. **#{securityContext.ifAllGranted['rolename1,rolename2']}** : Returns true if user is in all of the roles given in the roles list, vice versa 5. **#{securityContext.ifAnyGranted['rolename1,rolename2']}** : Returns true if user is in any one of the roles given in the roles list, vice versa 6. **#{securityContext.ifNotGranted['rolename1,rolename2']}** : Returns true if user is not in any of the roles given in the roles list, vice versa

SecurityContext is an abstract class that is used when the expressions above are resolved, J2EE container security is used by the default implementation SecurityContextImpl meaning;

ifGranted #{securityContext.ifGranted['rolename']} will yield to [FacesContext].getCurrentInstance().getExternalContext().isUserInRole("rolename").

It's possible to provide your own implementation of the SecurityContext if you're using another mechanism to manage security other than J2EE container like JAAS or ACEGI. In order to plugin your implementation *org.apache.myfaces.SECURITY_CONTEXT* context parameter needs to be configured with the class name of your implementation as the param value.

```
<context-param>
    <param-name>org.apache.myfaces.SECURITY_CONTEXT</param-name>
    <param-value>com.mycompany.MySecurityContextImpl</param-value>
</context-param>
```

Note: User-role Awareness attributes enabledOnUserRole and visibleOnUserRole will be deprecated in future releases.

Portlet SecurityContext

For portlets there is a standardisized way to access these values. The default implementation of SecurityContext `org.apache.myfaces.custom.security.SecurityContextImpl` supports this also.