icw
eHEALTH FRAMEWORK

eHealth Framework

How to implement an Access Decision Voter

Imprint

InterComponentWare
Altrottstraße 31
69190 Walldorf
Tel.: +49 (0) 6227 385 0
Fax.: +49 (0) 6227 385 199

Document version: preliminary
Document Language: en (US)
Product Name: eHealth Framework
Product Version: 2.10.3
Last Change: 23.03.2010
Editorial Staff: BAS Technology

Notice

The wording in this document applies equally to women and men. The masculine form was selected to ease the comprehensibility and legibility of the text.

Note:

The current version of this document has a draft status and various chapters are still in review.

The document is collaboratively built with the use of the Darwin-Information-Typing-Architecture (DITA) and has therefore a draft status concerning styles and layout. The necessary adaptations are currently also in a developmental stage.

Contents

1 Overview

Purpose

This howto shows what steps are needed to extend eHF Authorization with a custom access decision voter.

Scope

The howto describes how a decision voter is developed and configured into an assembly.

For more details on the eHealth Framework please refer to the eHealth Framework Reference Documentation on page 3.

2 Implementing an Access Decision Voter

2.1 Implement the voter

This step shows you how to implement an access decision voter, given you already know the functional requirements for your voter.

1. Create a class implementing the `com.icw.ehf.authorization.voter.AccessDecisionVoter` interface
2. Implement the `decide(Permission permission, Principal... principals): AccessDecisionResult` method
   Herein comes all of the access decision logic.
3. Implement the `supports(Permission permission): boolean` method
   This method provides a convenient shortcut to determine if a certain voter has to be invoked at all.

2.2 Implement an access decision engine

To combine the results of several voters you need an access decision engine implementing a certain combination algorithm.

This task shows how to implement an access decision engine, combining several voters using an arbitrary combination algorithm.

1. Create a class extending the abstract class `com.icw.ehf.authorization.voter.AccessDecisionEngine`
   Basically, the AccessDecisionEngine class is a composite AccessDecisionVoter. It implements the `AccessDecisionVoter` interface and holds a list of voters you can arbitrarily combine.
2. Put your combination logic into the `decide(Permission permission, Principal... principals): AccessDecisionResult` method
3. Implement the `supports(Permission permission): boolean` method
   Here you can delegate to the voters in the list.

2.3 Integrate the voter into the application

This task shows you how to integrate your voter, or decision engine for that matter, into an eHF based application, which we will call assembly further on.

1. Define and configure a spring bean for your voter
   This can be done in the assembly itself or in a module. In the latter case, make sure to export the spring bean from the module.
2. If applicable, define and configure a spring bean for your decision engine, injecting all necessary voters
3. Inject your voter/engine into the root access decision engine or replace it
   NOTE: The bean id of the root voter or decision engine, respectively, needs to be `accessDecisionEngine`.

3 References

E

ICW eHealth Framework Reference Documentation
BAS Technology and eHF Committers, InterComponentWare AG (2008-2010)
http://idn.icw-global.com/downloads.html