

Jetty/Tutorial/Realms

From Eclipsepedia

< Jetty | Tutorial

Contents

- 1 Introduction
- 2 Details
 - 2.1 HashLoginService
 - 2.2 JDBCLoginService
- 3 Additional Resources

Introduction

This tutorial describes how to configure security realms to provide authentication and access control for web applications running in Jetty. A realm has a unique name, and is composed of a set of users. Each user has authentication information (for example, a password) and a set of roles associated with him/herself. You can configure one or many different realms depending on your needs.

- Configure a single realm to share common security information across all of your web applications.
- Configure distinct realms to partition your security information webapp by webapp.

Details

A realm—known as a `LoginService`—is available to all web applications on a `Server` instance if you define it in a Jetty configuration file, for example `$JETTY_HOME/etc/jetty.xml`. Here's an example of defining an in-memory type of `LoginService` called the *HashLoginService* (note in Jetty 6 this was called the *HashUserRealm*):

```
<Configure id="Server" class="org.eclipse.jetty.server.Server">

  <Call name="addBean">
    <Arg>
      <New class="org.eclipse.jetty.security.HashLoginService">
        <Set name="name">Test Realm</Set>
        <Set name="config"><SystemProperty name="jetty.home" default="."/>/etc/rea
        <Set name="refreshInterval">0</Set>
      </New>
    </Arg>
  </Call>

</Configure>
```

Alternatively, you can define a `LoginService` for just a single web application in a context file. Here's how to define the same `HashLoginService`, but inside a context xml file instead of a configuration file:

```

<Configure class="org.eclipse.jetty.webapp.WebAppContext">
  <Set name="contextPath">/test</Set>
  <Set name="war"><SystemProperty name="jetty.home" default="."/>/webapps/test</Set>

  <Get name="securityHandler">
    <Set name="loginService">
      <New class="org.eclipse.jetty.security.HashLoginService">
        <Set name="name">Test Realm</Set>
        <Set name="config"><SystemProperty name="jetty.home" default="."/>/etc/r
      </New>
    </Set>
  </Get>
</Configure>

```

Jetty provides a number of different LoginService types from which you can choose.

HashLoginService

This LoginService is a simple realm whose authentication and authorization information is stored in a properties file. Each line in the file contains a username, a password, and zero or more role assignments. The format is:

```
username: password[,rolename ...]
```

where:

- **username** is the user's unique identity;
- **password** is the user's (possibly obfuscated or MD5 encrypted) password;
- **rolename** is the user's role.

For example:

```

admin: CRYPT:adlks..kc.lUg,server-administrator,content-administrator,admin
other: OBF:1xmk1w261u9r1wlclxmQ
guest: guest,read-only

```

You configure the HashLoginService with a name and a reference to the location of the properties file:

```

<Item>
  <New class="org.eclipse.jetty.security.HashLoginService">
    <Set name="name">Test Realm</Set>
    <Set name="config"><SystemProperty name="jetty.home" default="."/>/etc/realm.pro
  </New>
</Item>

```

You can also configure it to check the properties file regularly for changes and reload when changes are detected. The reloadInterval is in seconds:

```

<New class="org.eclipse.jetty.security.HashLoginService">
  <Set name="name">Test Realm</Set>
  <Set name="config"><SystemProperty name="jetty.home" default="."/>/etc/realm.pro
  <Set name="reloadInterval">5</Set>

```

```
<Call name="start"></Call>
</New>
```

JDBCLoginService

In this implementation, authentication and role information is stored in a database accessed via JDBC. A properties file defines the JDBC connection and database table information. Here is an example of a properties file for this realm implementation:

```
jdbcdriver = org.gjt.mm.mysql.Driver
url = jdbc:mysql://localhost/jetty
username = jetty
password = jetty
usertable = users
usertablekey = id
usertableuserfield = username
usertablepasswordfield = pwd
roletable = roles
roletablekey = id
roletablerolefield = role
userroletable = user_roles
userroletableuserkey = user_id
userroletablerolekey = role_id
cachetime = 300
```

The format of the database tables is (pseudo-sql):

```
users
(
  id integer PRIMARY KEY,
  username varchar(100) NOT NULL UNIQUE KEY,
  pwd varchar(50) NOT NULL
);
```

```
user_roles
(
  user_id integer NOT NULL,
  role_id integer NOT NULL,
  UNIQUE KEY (user_id, role_id),
  INDEX(user_id)
);
```

```
roles
(
  id integer PRIMARY KEY,
  role varchar(100) NOT NULL UNIQUE KEY
);
```

Where:

- **users** is a table containing one entry for every user consisting of:
 - **id**—the unique identity of a user

- **user**—the name of the user
- **pwd**—the user's password (possibly obfuscated or MD5 encrypted)
- **user-roles** is a table containing one row for every role granted to a user:
 - **user_id**—the unique identity of the user
 - **role_id**—the role for a user
- **roles** is a a table containing one role for every role in the system:
 - **id**—the unique identifier of a role
 - **role**—a human-readable name for a role

If you want to use obfuscated, MD5 hashed or encrypted passwords the 'pwd' column of the 'users' table must be large enough to hold the obfuscated, hashed or encrypted password text plus the appropriate prefix.

You define a JDBCLoginService with the name of the realm and the location of the properties file describing the database:

```
<New class="org.eclipse.jetty.security.JDBCLoginService">
  <Set name="name">Test JDBC Realm</Set>
  <Set name="config">etc/jdbcRealm.properties</Set>
</New>
```

Additional Resources

See Java Authentication and Authorization Service (JAAS) tutorial for additional information about configuring security realms.

Retrieved from "<http://wiki.eclipse.org/Jetty/Tutorial/Realms>"

Categories: Jetty | Jetty Tutorial

-
- [Home](#)
 - [Privacy Policy](#)
 - [Terms of Use](#)
 - [Copyright Agent](#)
 - [Contact](#)
 - [About Eclipsepedia](#)

Copyright © 2011 The Eclipse Foundation. All Rights Reserved

This page was last modified 21:00, 23 March 2011 by Shirley Boulay. Based on work by Michael Gorovoy and Jan Bartel.

This page has been accessed 14,079 times.