



ADUCID Integration Manual Tomcat

Version 3.1.0.RC3

Release date

June 17, 2016

Table of Contents

1. Purpose of this document	3
2. Prerequisites	3
3. Architecture and integration	3
4. Adapter installation	4
4.1. Preparing Tomcat	4
4.1.1. Tomcat 5.5	5
4.1.2. Tomcat 6.0	5
4.1.3. Tomcat 7.0	5
4.2. Configuring server.xml	5
4.2.1. Description of filter (valve) attributes	5
4.2.2. Configuration using ADUCID® database only	6
4.2.3. Configuration using ADUCID® database and a defined realm	7
4.3. Configuring web.xml	8
5. Attributes available to the application	9
6. Logging settings	9
7. Hello World application	9
7.1. Prerequisites	9
7.2. Installation	9
7.3. Troubleshooting	10
8. Appendix A	10
9. Appendix B	11

1. Purpose of this document

This document serves as an integration manual for incorporating ADUCID® technology into the Tomcat web container.

2. Prerequisites

When reading this document, the following basic knowledge is assumed:

- Tomcat Installation Guide 5, 6 or 7
- aducid-architecture.pdf
- Knowledge of web technologies, programming and integration of web applications

3. Architecture and integration

In order to simplify the integration of ADUCID® technology with Tomcat web container, a Tomcat filter, known as Tomcat Valve, was created that allows only authenticated and authorized requests to pass through to the protected section.

From deployment perspective, it represents the following logic model:

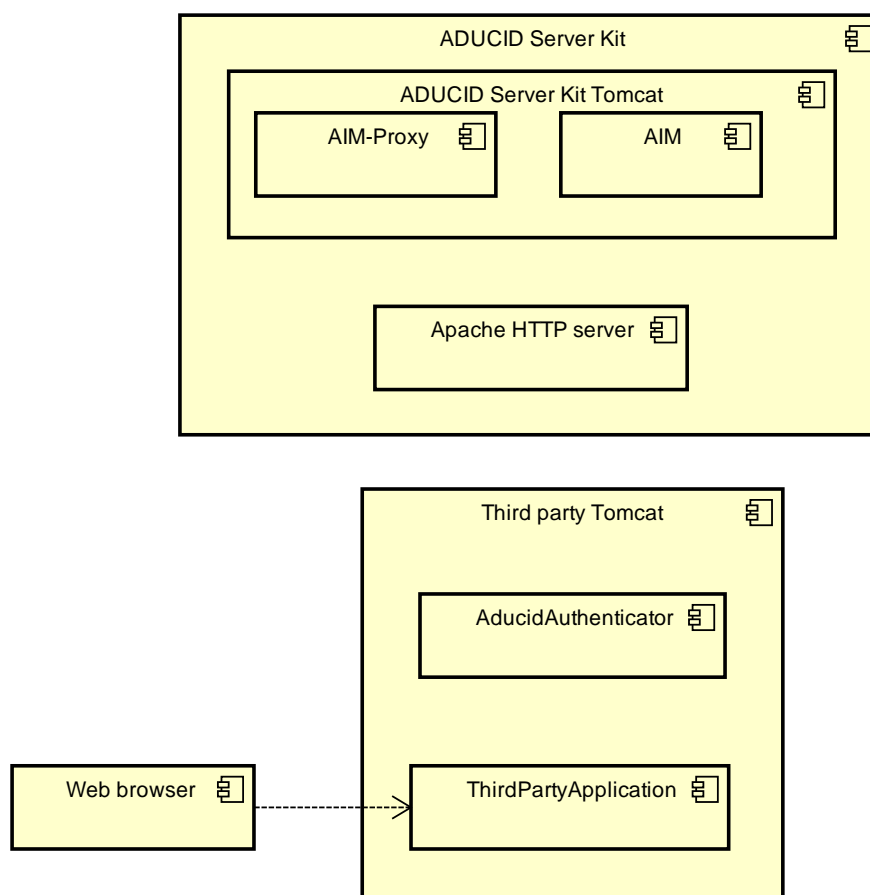


Figure 3-1 Logic deployment diagram

AducidAuthenticator is the key component in this model. It must be configured in the Tomcat web server that runs the application that you want to secure with ADUCID® technology. This component is responsible for:

- 1/ authenticating users against AIM server
- 2/ finding Principal for the authenticated user

The AducidAuthenticator component functions in the following way (to simplify the first use, the AIM-proxy component is used to handle authentication process):

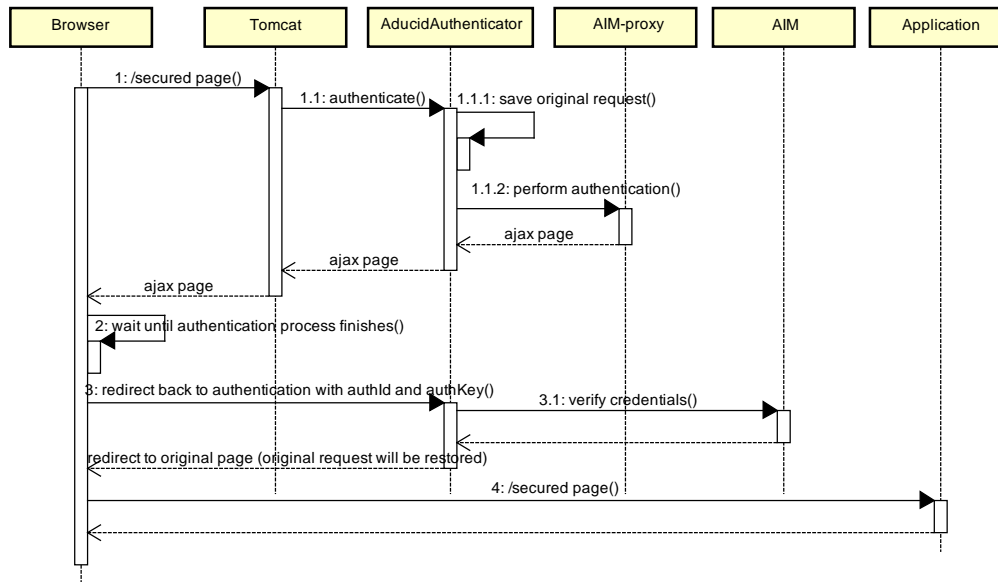


Figure 3-2 Unauthenticated access to the protected section

1. User accesses a secure page within the application.
2. Tomcat detects that the page is protected with container authentication and delegates the processing to AducidAuthenticator.
3. If the request has not been authenticated yet, AducidAuthenticator stores the original request in the session and delegates the start of authentication process to the AIM-proxy component using redirect.
4. Once the authentication process has finished, the result is returned along with credentials to the authentication filter.
5. Authentication filter verifies the credentials against the AIM's authentication server.
6. If the authentication succeeds, a redirect to the original page is performed.
7. The original request is restored.
8. The application processes the request with verified credentials and with Principal assigned.

From the application's perspective, only authenticated and authorized requests are passed to the protected section.

4. Adapter installation

Adapter is available for Tomcat 5.5, Tomcat 6.0 and Tomcat 7.0, and need Java 1.5 or later to run. The installation consists of two steps: Configuring ADUCID® technology in Tomcat and modifying the web.xml web application descriptor.

4.1. Preparing Tomcat

This section describes installation of ADUCID® technology to Tomcat web container.

4.1.1. Tomcat 5.5

To install the adapter, following modifications are necessary:

1. Unpack aducid-tomcat-v5-[version]-lib.zip and copy the libraries to the \$TOMCAT_HOME/server/lib folder.
2. In the \$TOMCAT_HOME/conf/server.xml file, add the com.aducid.tomcat.AducidAuthenticatorV5 filter to the context that you want to secure with ADUCID® technology. For individual attributes that can be configured for the filter, see section 4.2.

4.1.2. Tomcat 6.0

1. Unzip aducid-tomcat-v5-[version]-lib.zip and copy the libraries to the \$TOMCAT_HOME/lib folder.
2. In the \$TOMCAT_HOME/conf/server.xml file, add the com.aducid.tomcat.AducidAuthenticatorV5 filter to the context that you want to secure with ADUCID® technology. For individual attributes that can be configured for the filter, see section 4.2.

4.1.3. Tomcat 7.0

1. Unzip aducid-tomcat-v7-[version]-lib.zip and copy the libraries to the \$TOMCAT_HOME/lib folder.
2. In the \$TOMCAT_HOME/conf/server.xml file, add the com.aducid.tomcat.AducidAuthenticatorV7 filter to the context that you want to secure with ADUCID® technology. For individual attributes that can be configured for the filter, see section 4.2.

4.2. Configuring server.xml

This section provides a list of attributes for the adapter and two different configuration scenarios.

4.2.1. Description of filter (valve) attributes

The following provides an overview of attributes that can be set for the filter. For sample configurations, see subsections below.

Attribute name	Type of value	Description
className	Constant: com.aducid.tomcat.AducidAuthenticatorV5 for Tomcat 5.5 and 6 or com.aducid.tomcat.AducidAuthenticatorV7 for Tomcat 7	(mandatory) Java class of the adapter
cache	Boolean	(optional) Indicates whether the Principal should be cached for further requests, or a re-authentication is to be performed for each request. If re-authentication is enforced (cache=false), it is performed with original credentials only for the duration of the AIM session. Once the AIM session expires, another login into PEIG is enforced. Default = true
loginPage	Relative/absolute URL	(mandatory, if qrLoginPage is not set) URL of the page to which redirect is performed for the authentication to start. For easier integration, you can use the AIM-proxy component, or create your own authentication page.
qrLoginPage	Relative/absolute URL	(mandatory, if loginPage is not set) URL of the page to which redirect is performed for the authentication to start over QR code. See ADUCID Web Platform Javadoc documentation (class ControllerBase) to learn, how to prepare QR login page.

Attribute name	Type of value	Description
errorPage	Relative/absolute URL	(mandatory) A page within the application to which the user is forwarded in the event of authentication error.
aimUrl	Relative/absolute URL	(mandatory) URL of the endpoint where the R4 interface is available (URL of the AIM server).
collectionId	String	(mandatory) Identifier of the attribute set that stores user attributes from which the username and potential role are derived. Collections are assigned by the AIM administrator.
userAttribute	String	(mandatory) The name of the attribute in ADUCID® database that represents username.
roleAttribute	String	(mandatory, if defaultRole or useRealm are not set) The name of the attribute in ADUCID® database that represents role. This attribute is ignored if role lookup is used in the defined realm (useRealm=true). If the realm is not used (useRealm=false), this attribute is optional if defaultRole is defined.
defaultRole	String	(mandatory, if roleAttribute or useRealm are not set) The fixed role value. Use this attribute when the role is not defined in the ADUCID® database and the role has a constant value for all users. This attribute is ignored if the role lookup is used in the defined realm (useRealm=true). If the realm is not used (useRealm=false), this attribute is ignored if the roleAttribute is defined.
useRealm	Boolean	(mandatory) Indicates whether Principal creations should be delegated to the realm defined in Tomcat. Principal is looked up based on username found in the ADUCID® database. Default = false
logoutUrl	String	(mandatory) A regular expression URL pattern to log out an authentication session. If the current URL matches this pattern, the log out is called immediately.
logoutPage	Relative/absolute URL	(mandatory) A page within the application to which the user is forwarded in the event of an authentication log out.
crossContextPath	String	(optional) A path to the different application, where errorPage and logoutPage URLs are available. The path is added to URLs automatically.
verifyLF	Boolean	(optional) Indicates whether local factor should be verified. Default = false
proofingStatus	Constant: approved or notApproved	(optional) Indicate whether to check proofing status. If status is set to "approved" value, user must be ADUCID® approved. If status is set to "notApproved" value, user must not be ADUCID® approved. If not set, user's proofing status is ignored.

Table 4-1 Filter attributes

4.2.2. Configuration using ADUCID® database only

In this scenario, username and roles are retrieved from ADUCID® database. For successful configuration, you need to have a set of user attributes defined that contains an attribute representing a username and attribute representing a list of roles. These attributes, along with the defined attribute set's name, are provided by the ADUCID® administrator.

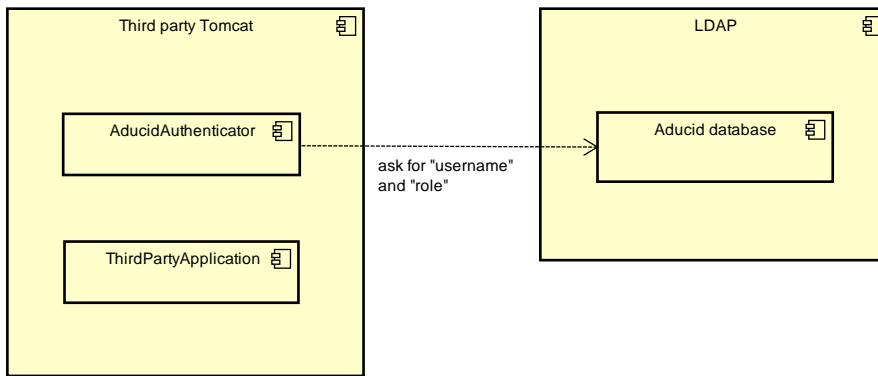


Figure 4-1 Using ADUCID® database only scheme

The following xml snippet represents an AducidAuthenticator configuration for a given context (with the name “aducid-tomcat-webapp”).

```

<Context
  docBase="tomcat_webapp"
  path="/aducid-tomcat-webapp">
  <Valve
    className="com.aducid.tomcat.AducidAuthenticatorV7"
    aimUrl="http://orangebox.example.com/AIM/services/R4"
    loginPage="http://orangebox.example.com/AIM-proxy/process"
    logoutUrl="/web/Logout"
    logoutPage="/Logout.jsp"
    errorPage="/error.jsp"
    collectionId="ADUCID_USER"
    userAttribute="LoginName"
    roleAttribute="aducidRole"
    useRealm="false"
    cache="true"
    verifyLF="true"
    proofingStatus="approved" />
  </Context>
  
```

In this scenario, the key attributes are collectionId, userAttribute and roleAttribute. They are assigned by the AIM server administrator. In this scenario, the useRealm must be set to false.

4.2.3. Configuration using ADUCID® database and a defined realm

In this scenario, only username is retrieved from the ADUCID® database. Roles are then retrieved from the realm defined in the Tomcat server configuration using the username provided. For successful configuration, you need to a set of user attributes defined that contains an attribute representing a username. This attribute, along with the defined attribute set's name, are provided by the ADUCID® administrator.

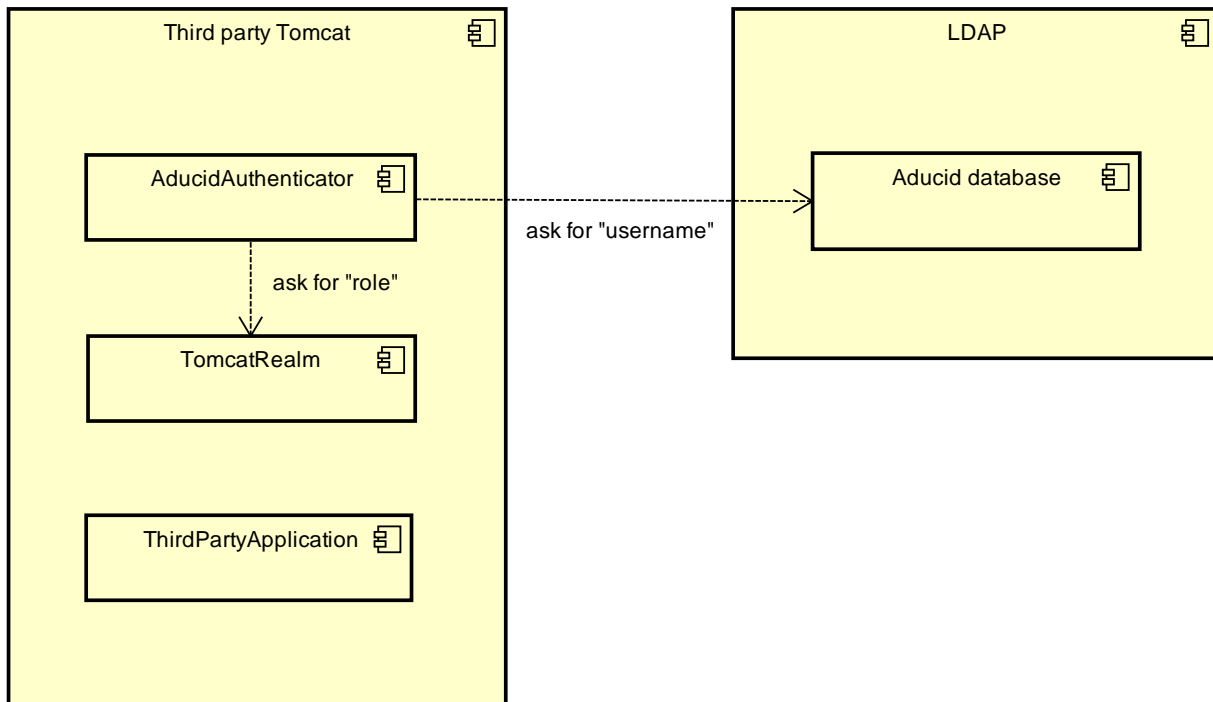


Figure 4-2 Using ADUCID® database and Tomcat realm scheme

The following xml snippet represents an AducidAuthenticator configuration for a given context (with the name “aducid-tomcat-webapp”).

```
<Context
  docBase="tomcat_webapp"
  path="/aducid-tomcat-webapp">
  <Valve
    className="com.aducid.tomcat.AducidAuthenticatorV7"
    aimUrl="http://orangebox.example.com/AIM/services/R4"
    loginPage="http://orangebox.example.com/AIM-proxy/process"
    logoutUrl="/web/logout"
    logoutPage="/Logout.jsp"
    errorPage="/error.jsp"
    collectionId="ADUCID_USER"
    userAttribute="LoginName"
    useRealm="true"
    cache="true"
    verifyLF="true"
    proofingStatus="approved" />
  </Context>
```

In this scenario, the key attributes are collectionId and userAttribute. They are assigned by the AIM server administrator. In this scenario, the useRealm must be set to true.

4.3. Configuring web.xml

Modifying web.xml is simple: you need to change current authentication method to the ADUCID constant (see Appendix A). Next, you need to add a special /authCheck URL, where the credentials for web-resource-collection are verified.

5. Attributes available to the application

When the authentication process has finished, the following attributes propagated as `HttpServletRequest` attributes are available to the application.

Attribute name	Type of value	Description
exception	java.lang.Throwable	Exception in the event that the authentication was not successful
originalContextPath	java.lang.String	Context path of application where exception occurred. Useful for error page buttons like "Try again".

Table 5-1 Attributes available to the application

The adapter also creates a `Principal` object that is available through `request.getUserPrincipal()` and whose form depends on the value of the `useRealm` attribute. If that value is `useRealm=true`, the `Principal` returned was generated by the realm set in `server.xml`. For `useRealm=false` value, the `com.aducid.tomcat.pojo.DefaultAducidUser[V5/V7]` object is returned.

6. Logging settings

Because the adapter is being implemented at the Tomcat web server level, the `log4j.properties` file must be placed in the `/common/classes` directory of the installation for Tomcat 5.5, and in the `/lib` directory of the installation for Tomcat 6.0 and Tomcat 7.0. For an example of the `log4j.properties` file, see Appendix B.

7. Hello World application

The installation package contains a sample web application "Hello World" that serves as an example of ADUCID® technology integration. The installation DVD contains source files of the sample application.

7.1. Prerequisites

Preliminary condition must be fulfilled to be sure that sample application is fully working:

- PEIG must be running (Client's side of ADUCID®)
- There must be valid ADUCID® approved identity with personal factor on ADUCID Server Kit

7.2. Installation

To install the web application, follow these steps:

- 1/ Ask the user administrator to add a `loginName` attribute, `jsmith` value for example, to your profile. Potentially also ask role administrator for the role, demo application works with `RegAdmin` and `UserAdmin` roles, on figure 7-1 `jsmith` has `RegAdmin` role only.
- 2/ Configure Aducid Authenticator as per instructions in section 4.2.
- 3/ Rename supplied sample web application "Hello World" from `aducid-tomcat-webapp-[version].war` to `aducid-tomcat-webapp.war`.
- 4/ Load the sample web application into Tomcat, specifically to `$TOMCAT_HOME/webapps` folder
- 5/ Run Tomcat, so that the installed sample web application unpacks.
- 6/ Stop Tomcat.
- 7/ Set application configuration parameters in the file `$TOMCAT_HOME/webapps/aducid-tomcat-webapp/WEB-INF/classes/config.properties`:
 - a/ `appName`—the application name shown on the log out page
 - b/ `loginPage`—link to the log in page, used from the error and log out page

c/ refreshAfterErrorPage—link to a page where the user can go if an error occurs
8/ Run Tomcat.

After installation, sample application is accessible on address:

```
http://[ip_address_of_your_container]:[port_of_your_container]/aducid-tomcat-webapp
```

For example on address:

```
http://10.20.29.189:8080/aducid-tomcat-webapp
```

In case of installing on Linux OS, consider also iptables settings.

The web application you have just installed contains two JSP pages.

/unsecured.jsp - this page is freely accessible.

/web/secured.jsp - this page is protected by the “RegAdmin” and “UserAdmin” roles.

If both the application and Tomcat are configured properly, the /web/secured.jsp page will be accessible. Accessibility of individual sections depends on the role assigned.

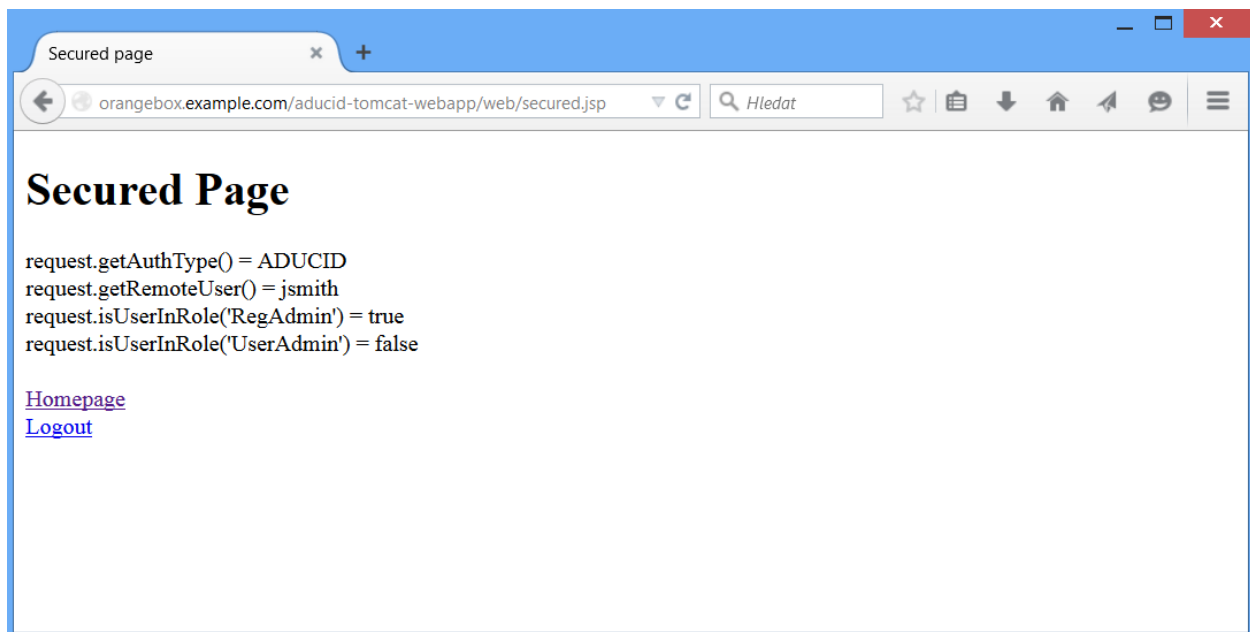


Figure 7-1 Protected section of the sample application

7.3. Troubleshooting

If the result of authentication process is not according figure 7-1 check following:

- PEIG is active – indicator is green
- There is valid ADUCID® approved identity with personal factor on ADUCID Server Kit

8. Appendix A

The appendix shows the relevant part of the web.xml descriptor.

```
<security-constraint>  
  <web-resource-collection>
```

```
<web-resource-name>secured resources</web-resource-name>
<url-pattern>/web/*</url-pattern>
<url-pattern>/authCheck</url-pattern>
</web-resource-collection>
<auth-constraint>
  <role-name>RegAdmin</role-name>
  <role-name>UserAdmin</role-name>
</auth-constraint>
</security-constraint>
<login-config>
  <auth-method>ADUCID</auth-method>
</login-config>
<security-role>
  <role-name>RegAdmin</role-name>
</security-role>
<security-role>
  <role-name>UserAdmin</role-name>
</security-role>
```

9. Appendix B

Example of log4j.properties settings

```
log4j.rootCategory=INFO, CONSOLE
log4j.logger.com.aducid.tomcat=DEBUG
log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern=[%p] [%c] %m%n
```