Installation and configuration of uApprove.jp-2.2.1c

Table of contents

1. Installation and configuration of uApprove.jp-2.2.1c for Shibboleth Identity Provider

1.1 Introduction

uApprove.jp (which is an extension of uApprove) is an application which allows an end user to view and to select the attribute set which will be sent to the visiting resource.

This guide describes the installation & configuration of the uApprove.jp for Shibboleth Identity Provider (IdP).
It covers the installation the IdP plugin as well as uApprove.jp viewer application and their configuration with a flat file based or SQL based (Database) storage.

The example values used in this guide are:

- `idp.example.org`
  - The DNS name of the Identity Provider
- `/opt/uApprove`
  - The directory, where the uApprove is installed
- `/opt/uApprove/conf`
  - The directory, where the uApprove config is located
- `/opt/shibboleth-identityprovider-2.x`
  - The Shibboleth IdP installation directory (where `install.sh` lives)
- `/opt/shibboleth-idp`
  - The directory, where the Shibboleth IdP is installed
- `${CATALINA_HOME}`
  - The directory, where Tomcat is installed (i.e. `/usr/java/tomcat`).

1.2 Prerequisites

Before start install and configure uApprove.jp, assure that the following prerequisites are met:

- Shibboleth Identity Provider **2.3**.
  - Shibboleth Identity Provider has to be deployed properly.
    That implied that Java and Tomcat are working.
- SQL Database
  - If a SQL Database as storage is used, A SQL Server has to be prepared.
    Shipped uApprove.jp is already configured for MySQL, but another SQL Server can be adapted.

1.3 Installation

Download and unzip the uApprove.jp package:

```
wget https://meatwiki.nii.ac.jp/confluence/download/attachments/13501031/uApprove.jp-2.2.1c-bin.zip?api=v2
unzip uApprove.jp-2.2.1c-bin.zip -d /opt/
ln -s /opt/uApprove.jp-2.2.1c /opt/uApprove
```

1.3.1 IdP-Plugin

Copy libraries and configuration:

```
cd /opt/uApprove
mkdir conf logs war
unzip idp-plugin-2.2.1c-bin.zip
cp idp-plugin-2.2.1c/conf-template/* conf/
cp idp-plugin-2.2.1c/lib/* /opt/shibboleth-identityprovider-2.x/lib/
```

1.3.2 Viewer

Copy libraries and configuration:

```
cd /opt/uApprove
unzip viewer-2.2.1c-bin.zip
cp viewer-2.2.1c/conf-template/* conf/
```

1.4 Configuration

uApprove.jp consist of two pieces of software, the IdP plugin which runs within the IdP context and the uApprove.jp viewer application which is separated.
Both of them use a common data storage and the according libraries to operate with it.

1.4.1 uApprove.jp common: Storage etc.

The storage holds mainly the following data:

- Username
- Last version of the term of use that be accepted
- List of encrypted attributes that were release to a Shibboleth SP
- Entity ID of Shibboleth SP

First you have to decide which storage implementation should be used:

- **File based**
  - This is only recommended for small installation (< 100 users)
- **SQL Database**
  - MySQL database is fully supported, but any SQL Database which has a JDBC connector should work. May be advanced configuration is needed.

File based

A file based storage have to be setup in /opt/uApprove/conf/common.properties:

```
#storageType=database
#databaseConfig=/opt/uApprove/conf/database.properties

storageType=file
flatFile = /opt/uApprove/data/uApprove-log.xml
```


Assure that the directory in which `uApprove-log.xml` is stored is read- and writable by Tomcat.

Don't create `uApprove-log.xml` beforehand because uApprove.jp generates it.

Database

First, a databases and a user have to be created:

```
mysql -u root -p
mysql>
CREATE DATABASE uApprove;
CREATE USER 'uApprove'@'localhost' IDENTIFIED BY 'uApprove';
GRANT USAGE ON *.* TO 'uApprove'@'localhost';
GRANT SELECT , INSERT , UPDATE , DELETE ON `uApprove`.* TO 'uApprove'@'localhost';
ALTER DATABASE uApprove DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Second, generate the table structures:

```
mysql -u root -p
mysql>
use uApprove;

create table ArpUser (
  idxArpUser int unsigned auto_increment primary key,
  auUserName varchar(255) not null,
  auLastTermsVersion varchar(255),
  auFirstAccess timestamp,
  auLastAccess timestamp
);
create index idxUserName on ArpUser (auUserName );

create table ShibProvider (
  idxShibProvider int unsigned auto_increment primary key,
  spProviderName varchar(255)
);

insert into ShibProvider (idxShibProvider) values (1);
create index idxProvidername on ShibProvider (spProviderName);

create table AttrReleaseApproval (
  idxAttrReleaseApproval int unsigned auto_increment primary key,
  araIdxArpUser int unsigned references ArpUser ( idxArpUser ),
  araIdxShibProvider int unsigned references ShibProvider( idxShibProvider ),
  araTimeStamp timestamp not null,
  araTermsVersion varchar(255),
  araAttributes text(2048)
);

create table ProviderAccess (
  idxProviderAccess int unsigned auto_increment primary key,
  paIdxArpUser int unsigned references ArpUser( idxArpUser ),
  paIdxShibProvider int unsigned references ShibProvider( idxShibProvider ),
  paAttributesSent text,
  paTermsVersion varchar(255),
  paIdxAttrReleaseApproval int unsigned references AttrReleaseApproval ( idxAttrReleaseApproval ),
  paShibHandle varchar(255),
  paTimeStamp timestamp not null
);

create table CheckAlways (
  idxCheckAlways int unsigned auto_increment primary key,
  caIdxArpUser int unsigned references ArpUser ( idxArpUser ),
  caIdxShibProvider int unsigned references ShibProvider ( idxShibProvider ),
  caTimeStamp timestamp not null
);

create table BackChannelAccess (
  idxBackChannelAccess int unsigned auto_increment primary key,
  bcaTimeStamp timestamp not null,
  bcaIdxAttrReleaseApproval int unsigned not null references AttrReleaseApproval ( idxAttrReleaseApproval ),
  bcaLastAccess timestamp,
  bcaAccessCount int unsigned
);

create table NamesOfApprovedService (
  nasIdxAttrReleaseApproval int unsigned not null references AttrReleaseApproval(idxAttrReleaseApproval),
  nasServiceNames text NOT NULL
);

ALTER TABLE ArpUser ENGINE=MyISAM;
ALTER TABLE AttrReleaseApproval ENGINE=MyISAM;
ALTER TABLE CheckAlways ENGINE=MyISAM;
ALTER TABLE ProviderAccess ENGINE=MyISAM;
ALTER TABLE ShibProvider ENGINE=MyISAM;
ALTER TABLE BackChannelAccess ENGINE=MyISAM;
ALTER TABLE NamesOfApprovedService ENGINE=MyISAM;
```

The following table added in uApprove.jp-2.2.1:

- *CheckAlways*

The following tables added in uApprove.jp-2.2.1c:

- *BackChannelAccess*
- *NamesOfApprovedService*

For easy checking whether the setting of the database is right, you can use the following command:

```
echo 'SHOW TABLES' | mysql -u uApprove -p -h localhost uApprove
Enter password:
ArpUser
AttrReleaseApproval
BackChannelAccess
CheckAlways
NamesOfApprovedService
ProviderAccess
ShibProvider
```

A Database setup is configured in /opt/uApprove/conf/common.properties:

```
storageType=database
databaseConfig=/opt/uApprove/conf/database.properties

#storageType=file
#flatFile=/opt/uApprove/data/uApprove-log.xml
```

uApprove.jp supports JDBC connection pooling provided by the BoneCP library.
It is possible to configure container managed connections as well as application managed connection pooling.

All database specific configuration is done in /opt/uApprove/conf/database.properties:

```
sqlCommands=/opt/uApprove/conf/mysql.commands

# first option to use jndi and container managed connections
# resourceName=jdbc/mypool

# second option to use application managed connection pooling
# this is provided by the bonecp library http://jolbox.com/
# these are the required parameters
driver=com.mysql.jdbc.Driver
url=jdbc:mysql://localhost:3306/uApprove
user=uApprove
password=uApprove

# optional parameters for bonecp
#
# connectionTestStatement=SELECT 1
# minConnectionsPerPartition=1
# maxConnectionsPerPartition=5
# partitionCount=2
```

If another SQL server than MySQL is used, the values above has to be adjusted.
Don't forget to put according JDBC driver into the library folder.
If it is necessary, the SQL commands can be re-defined in `/opt/uApprove/conf/custom-sql.commands`

Other common configuration

The TermsOfUseManager, which shows a Terms of use text to the user is optional. If the TermsOfUseManager should be active, define the terms in /opt/uApprove/conf/common.properties:

```
termsOfUse=/opt/uApprove/conf/terms-of-use.xml
```


The shipped `terms-of-use.xml` is an empty one, which can be filled by custom terms of use version and text.
If you want see a real world example, take a look at the SWITCHaai VHO terms of use (`doc/terms-of-use-SWITCH.xml`).

Cause the IdP plugin and the viewer application exchange some confidential information, this is encrypted and decrypted by a shared secret (128 bit, 16 bytes) in /opt/uApprove/conf/common.properties:

```
sharedSecret=QErDXYZEAoS6jooPvdBhQg==
```

For easy creating a random value, containing 16 bytes, you can use the following command:

```
openssl rand -base64 16 2>/dev/null
```

1.4.2 IdP plugin configuration

Shibboleth IdP adjustments

The IdP plugin has to be enabled within Shibboleth IdP web application /opt/shibboleth-identityprovider-2.x/src/main/webapp/WEB-INF/web.xml:

```
<web-app>
  ...

  <filter>
    <filter-name>uApprove.jp IdP plugin</filter-name>
    <filter-class>ch.SWITCH.aai.uApprove.idpplugin.Plugin</filter-class>
    <init-param>
      <param-name>Config</param-name>
      <param-value>
        /opt/uApprove/conf/idp-plugin.properties;
        /opt/uApprove/conf/common.properties;
      </param-value>
    </init-param>
  </filter>

  <filter-mapping>
    <filter-name>uApprove.jp IdP plugin</filter-name>
    <url-pattern>/*</url-pattern>
    <dispatcher>REQUEST</dispatcher>
    <dispatcher>FORWARD</dispatcher>
  </filter-mapping>

</web-app>
```


The `filter` and `filter-mapping` about uApprove.jp must be defined after than other `filter` and `filter-mapping` in `web.xml`.

Redeploy Shibboleth IdP:

```
cd /opt/shibboleth-identityprovider-2.x/
sh install.sh
Buildfile: src/installer/resources/build.xml

install:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Be sure you have read the installation/upgrade instructions on the Shibboleth website before proceeding.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Where should the Shibboleth Identity Provider software be installed? [/opt/shibboleth-idp]

The directory '/opt//opt/shibboleth-idp' already exists.  Would you like to overwrite this Shibboleth configuration? (yes, [no])
no
Updating property file: /opt/shibboleth-identityprovider-2.x/src/installer/resources/install.properties
Copying 61 files to /opt/shibboleth-idp/lib
Copying 5 files to /opt/shibboleth-idp/lib/endorsed
Copying 1 file to /opt/shibboleth-identityprovider-2.x/src/installer
Building war: /opt/shibboleth-identityprovider-2.x/src/installer/idp.war
Copying 1 file to /opt/shibboleth-idp/war
Deleting: /opt/shibboleth-identityprovider-2.x/src/installer/web.xml
Deleting: /opt/shibboleth-identityprovider-2.x/src/installer/idp.war

BUILD SUCCESSFUL
Total time: 17 seconds
```

copy idp.war into ${CATALINA_HOME}/webapps:

```
cp /opt/shibboleth-idp/war/idp.war ${CATALINA_HOME}/webapps/
```

SP blacklist

The SP blacklist defines Resources which are excluded from the user consent.
The SP blacklist is optional and can be configured by /opt/uApprove/conf/idp-plugin.properties:

```
spBlacklist=/opt/uApprove/conf/sp-blacklist
```

In sp-blacklist, each line defines a regular expression pattern for black listed resource identifier:

```
# Example 1: specific resource
https://sp¥.example¥.org/shibboleth

# Example 2: all applications within a Service Provider
https://sp¥.example¥.org/.*

# Example 3: all Service Provider within a specific domain
https://.*¥.example¥.org/.*
```

Attributes ordering and blacklisting

uApprove.jp shows all attributes which will be released for the current user to the target resource.

It is possible to define an order of the attribute listing and if necessary, attributes can also be hidden. The configuration of the attribute list is optional done by /opt/uApprove/conf/idp-plugin.properties:

```
attributeList=/opt/uApprove/conf/attribute-list
```

An attribute-list can be look like:

```
# Defined attribute order
surname
givenName
postalAddress
...

# attributes to hide
!persistentId
!transientId
```


The attribute name in `attribute-list` has to comply with the attribute id in `/opt/shibboleth-idp/conf/attribute-resolver.xml`

Other configuration

If the database storage is used, it is possible to log each provider access, which means every attribute release is stored in the database.
It is also possible, that the IdP plugin runs in monitoring only mode, which logs every provider access, but do not interact with the user (for user consent)
If the viewer web application is used, the IdP plugin has to know, where it is deployed.
it is possible to tell the IdP plugin, that it should take care about *isPassive* requests.
The configuration is done in /opt/uApprove/conf/idp-plugin.properties:

```
logProviderAccess=false
monitoringOnly=false
uApproveViewer=https://idp.example.org/uApprove/Controller
isPassiveSupport=false
```

1.4.3 Viewer configuration

Define a tomcat deployment descriptor for the uApprove.jp webapp in ${CATALINA_HOME}/conf/Catalina/localhost/uApprove.xml:

```
<Context docBase="/opt/uApprove/war/uApprove.war"
         privileged="true"
         antiResourceLocking="false"
         antiJARLocking="false"
         unpackWAR="false" />
```

Adjust the viewer application according you configuration directory in /opt/uApprove/viewer-2.2.1c/webapp/WEB-INF/web.xml:

```
<web-app>
  ...
  <context-param>
    <param-name>Config</param-name>
    <param-value>
        /opt/uApprove/conf/viewer.properties;
        /opt/uApprove/conf/common.properties;
    </param-value>
  </context-param>
  ...
<web-app>
```

It is possible to change the logo and the navigation menu. The logo can be used image file in /opt/uApprove/viewer-2.2.1c/webapp/images and /or URL.
The configuration are done in /opt/uApprove/viewer-2.2.1c/webapp/header.jsp:

```
<body class="switchaai">
<div class="box-aai" style="width: 650px;">
  <img src="images/GakuNin_logo.png" alt="GakuNin-logo" class="switchaai" height="32" width="162">
  <br>
  <span class="switchaai"><a href="http://www.gakunin.jp/" class="switchaai">About GakuNin</a></span>
```

Deploy the uApprove.jp webapp:

```
cd /opt/uApprove/viewer-2.2.1c/
ant deploy -Duapprove.deployment=/opt/uApprove/war
```

Localization

The viewer web application is multi lingual for the static text and dynamic text, like attribute names and descriptions.
For the static text, the following languages are supported: en, de, fr, it, pt, ja.
For the attribute names and descriptions, the localized content is taken from /opt/shibboleth-idp/conf/attribute-resolver.xml (Shibboleth IdP resolver). It's possible to adjust or extent it:

```
...

<resolver:AttributeDefinition id="postalAddress" xsi:type="Simple" xmlns="urn:mace:shibboleth:2.0:resolver:ad" sourceAttributeID="
postalAddress">
  <resolver:Dependency ref="myLDAP" />
  <resolver:DisplayName xml:lang="en">Business postal address</resolver:DisplayName>
  <resolver:DisplayName xml:lang="de">Geschäftsadresse</resolver:DisplayName>
  <resolver:DisplayName xml:lang="fr">Adresse Professionnelle</resolver:DisplayName>
  <resolver:DisplayName xml:lang="it">Indirizzo professionale</resolver:DisplayName>
  <resolver:DisplayName xml:lang="ja">所属組織住所</resolver:DisplayName>
  <resolver:DisplayDescription xml:lang="en">Business postal address: Campus or office address</resolver:DisplayDescription>
  <resolver:DisplayDescription xml:lang="de">Adresse am Arbeitsplatz</resolver:DisplayDescription>
  <resolver:DisplayDescription xml:lang="fr">Adresse de l'institut, de l'universite</resolver:DisplayDescription>
  <resolver:DisplayDescription xml:lang="it">Indirizzo professionale: Indirizzo dell'istituto o dell'ufficio</resolver:
DisplayDescription>
  <resolver:DisplayDescription xml:lang="ja">所属組織(大学、会社など)の住所</resolver:DisplayDescription>
  <resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
                             name="urn:mace:dir:attribute-def:postalAddress" />
  <resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
                             name="urn:oid:2.5.4.16" friendlyName="postalAddress" />
</resolver:AttributeDefinition>
...
```


For the SWITCHaai attributes, there is a prepared `doc/attribute-descriptions.xml` for the languages: en, de, fr, it, ja.
For the GakuNin attributes, there is a prepared the same file for the languages: en, ja.

The viewer uses the language according to the users browsers request, if that language is not available, the default locale *en* will be taken.
It is possible to enforce a specific language. This can be defined optional in /opt/uApprove/conf/viewer.properties:

```
useLocale=en_US
```

Global consent

It is possible that a user can give global attribute release consent, which mean, that she/he is never be asked again by uApprove.jp. If it is required, that a user has to give consent to each different resource access, global consent has to be disabled in /opt/uApprove/conf/viewer.properties:

```
globalConsentPossible=true
```

Logging

The viewer web application uses LogBack (see 1.7 References) like the Shibboleth IdP.
The logback configuration file is defined in /opt/uApprove/conf/viewer.properties:

```
loggingConfig=/opt/uApprove/conf/logging.xml
```

Adjust `logging.xml` for the log location, assure that the file is writable by Tomcat:

```
<configuration>
  ...
  <appender class="ch.qos.logback.core.FileAppender" name="RootFileAppender">
    <file>/opt/uApprove/logs/uApprove.log</file>
    ...
  </appender>
  ...
</configuration>
```

Apache in front of Tomcat

If you are using Apache HTTPd in front of your Tomcat setup, proxy the viewer web application in /etc/httpd/conf.d/ssl.conf :

```
<VirtualHost idp.example.org:443>
  ...
  <Location /uApprove>
    Allow from all
    ProxyPass ajp://localhost:8009/uApprove
  </Location>
  ...
</VirtualHost>
```

Restart Apache HTTPd:

```
/sbin/service httpd restart
```

1.4.4 Shibboleth IdP's profile handler

uApprove.jp generates the list of attributes depending on the user consent in the Attribute Query SAML Response message.

Define the Namespace

In you profile handler file (e.g. /opt/shibboleth-idp/conf/handler.xml) you'll need to add the namespace declaration for this plugin. To do this:

1. Add the attribute `xmlns:uajpph="http://www.gakunin.jp/ns/uapprove-jp/profile-handler"` before the `xmlns:xsi` attribute on the root `<ProfileHandlerGroup>` element.
2. Add the following at the end of the whitespace delimited list of values for the `xsi:schemaLocation` attribute: `http://www.gakunin.jp/ns/uapprove-jp/profile-handler classpath:/schema/shibboleth-2.0-idp-profile-handler-uapprovejp.xsd`

```
...
<ph:ProfileHandlerGroup
        xmlns:ph="urn:mace:shibboleth:2.0:idp:profile-handler"
        xmlns:uajpph="http://www.gakunin.jp/ns/uapprove-jp/profile-handler"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:mace:shibboleth:2.0:idp:profile-handler classpath:/schema/shibboleth-2.0-idp-profile-handler.xsd
                            http://www.gakunin.jp/ns/uapprove-jp/profile-handler classpath:/schema/shibboleth-2.0-idp-profile-handler-
uapprovejp.xsd">
...
```

Modify Attribute Query Profile Handler

You'll change the Attribute Query profile handlers. To do this:

1. change the value of `xsi:type` from `ph:SAML1AttributeQuery` to `uajpph:SAML1AttributeQueryUApprove`
2. change the value of `xsi:type` from `ph:SAML2AttributeQuery` to `uajpph:SAML2AttributeQueryUApprove`

```
...
    <ph:ProfileHandler xsi:type="uajpph:SAML1AttributeQueryUApprove" inboundBinding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-
binding"
                       outboundBindingEnumeration="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding">
        <ph:RequestPath>/SAML1/SOAP/AttributeQuery</ph:RequestPath>
    </ph:ProfileHandler>
...
    <ph:ProfileHandler xsi:type="uajpph:SAML2AttributeQueryUApprove" inboundBinding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
                       outboundBindingEnumeration="urn:oasis:names:tc:SAML:2.0:bindings:SOAP">
        <ph:RequestPath>/SAML2/SOAP/AttributeQuery</ph:RequestPath>
    </ph:ProfileHandler>
...
```

1.4.5 Shibboleth IdP's attribute filter

uApprove.jp determines whether an attribute is mandatory or optional by recognizing <RequestedAttribute> elements in metadata of an SP (see 2. Configuration of Shibboleth Service Provider) and <AttributeFilterPolicy> elements in attribute filter policy file of the IdP.

Define the Namespace

In you attribute filter policy file (e.g. /opt/shibboleth-idp/conf/attribute-filter.xml) you'll need to add the namespace declaration for this plugin. To do this:

1. Add the attribute xmlns:uajpmf="http://www.gakunin.jp/ns/uapprove-jp/afp/mf" before the xmlns:xsi attribute on the root <AttributeFilterPolicyGroup> element.
2. Add the following at the end of the whitespace delimited list of values for the xsi:schemaLocation attribute: http://www.gakunin.jp/ns/uapprove-jp/afp/mf classpath:/schema/shibboleth-2.0-afp-mf-uapprovejp.xsd

```
...
<afp:AttributeFilterPolicyGroup id="ShibbolethFilterPolicy"
        xmlns:afp="urn:mace:shibboleth:2.0:afp"
        xmlns:basic="urn:mace:shibboleth:2.0:afp:mf:basic"
        xmlns:saml="urn:mace:shibboleth:2.0:afp:mf:saml"
        xmlns:uajpmf="http://www.gakunin.jp/ns/uapprove-jp/afp/mf"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:mace:shibboleth:2.0:afp classpath:/schema/shibboleth-2.0-afp.xsd
                            urn:mace:shibboleth:2.0:afp:mf:basic classpath:/schema/shibboleth-2.0-afp-mf-basic.xsd
                            urn:mace:shibboleth:2.0:afp:mf:saml classpath:/schema/shibboleth-2.0-afp-mf-saml.xsd
                            http://www.gakunin.jp/ns/uapprove-jp/afp/mf classpath:/schema/shibboleth-2.0-afp-mf-uapprovejp.xsd
">
```

uApprove.jp-2.2.1a or later, the namespace URI has been changed.

- uApprove.jp-2.2.1
  http://www.gakunin.jp/ns/uapprove-jp
- uApprove.jp-2.2.1a (or later)
  http://www.gakunin.jp/ns/uapprove-jp/afp/mf

Define the Rule

uApprove.jp enhances Policy Requirement Rule and Permit/Deny Value Rule in Attribute Rules:

- Policy Requirement Rule
  - This rule is defined by <afp:PolicyRequirementRule xsi:type="uajpmf:AttributeUapprove"> element and is applied when <AttributeConsumingService> elements exist in metadata of an SP.
- Permit/Deny Value Rule in Attribute Rule
  - This rule is defined by both <afp:PermitValueRule xsi:type="uajpmf:AttributeUapprove"/> elements and <afp:DenyValueRule xsi:type="uajpmf:AttributeUapprove"/> elements with the following optional attributes:
  - **isApproved** (optional)

        - Boolean flag indicated that this attribute is allowed to release to an SP by an end user,
          Default value: true.
          However, If the `isRequired` attribute of `<RequestedAtrribute>` elements within `<AttributeConsumingService>` elements in metadata of SP is true, this attribute is the mandatory attribute and is always released to an SP.
    - **requestedOnly** (optional)
        - Boolean flag indicated that only this attribute which is defined in metadata of an SP should be displayed,
          Default value: false

The permitted attributes without `xsi:type="uajpmf:AttributeUapprove"` are handled as the mandatory attributes.

Example Permit Value Rule using the uajpmf:AttributeUapprove Match Function:

```
<!--    ============================================================================
case 1: match SPs that has some AttributeConsumingService elements in metadata.
        The eduPersonPrincipalName attribute is the optional attribute and the end user
        can select whether to release. The eduPersonAffiliation attribute is the mandatory
        attribute and is always released.
        ============================================================================ -->
<afp:AttributeFilterPolicy id="PolicyforSPwithAttributeConsumingService">
    <afp:PolicyRequirementRule xsi:type="uajpmf:AttributeUapprove" />

    <afp:AttributeRule attributeID="eduPersonPrincipalName">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeUapprove" />
    </afp:AttributeRule>

    <afp:AttributeRule attributeID="eduPersonAffiliation">
        <afp:PermitValueRule xsi:type="basic:ANY" />
    </afp:AttributeRule>

    ...
</afp:AttributeFilterPolicy>

<!--    ============================================================================
case 2: match SPs that doesn't have any AttributeConsumingService elements in metadata.
        The eduPersonPrincipalName attribute and the eduPersonAffiliation attribute are
        the mandatory attributes.
        ============================================================================ -->
<afp:AttributeFilterPolicy id="PolicyforSPwithoutAttributeConsumingService">
    <afp:PolicyRequirementRule xsi:type="basic:NOT">
        <basic:Rule xsi:type="uajpmf:AttributeUapprove"/>
    </afp:PolicyRequirementRule>

    <afp:AttributeRule attributeID="eduPersonPrincipalName">
        <afp:PermitValueRule xsi:type="basic:ANY" />
    </afp:AttributeRule>

    <afp:AttributeRule attributeID="eduPersonAffiliation">
        <afp:PermitValueRule xsi:type="basic:ANY" />
    </afp:AttributeRule>

    ...
</afp:AttributeFilterPolicy>

<!--    ============================================================================
case 3: match all SPs.
        The eduPersonPrincipalName attribute and the eduPersonAffiliation attribute are
        the optional attributes.
        ============================================================================ -->
<afp:AttributeFilterPolicy id="PolicyforAnyone">
    <afp:PolicyRequirementRule xsi:type="basic:ANY" />

    <afp:AttributeRule attributeID="eduPersonPrincipalName">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeUapprove" />
    </afp:AttributeRule>

    <afp:AttributeRule attributeID="eduPersonAffiliation">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeUapprove" />
    </afp:AttributeRule>
```

```
    ...
</afp:AttributeFilterPolicy>
```

1.4.6 Reset-approvals configuration

Do not use this configuration if you use *list-approvals* application.

The optional *reset-approvals* can be operated in two ways:

- In-flow mode
    - The In-flow operation means, that a user can reset his setting during the login process. As example this is done by a checkbox on the login form.
- Standalone mode
    - The standalone operation can is as JSP which can be called directly.

In-flow mode

For enabling the In-Flow *reset-approvals* application, the Shibboleth UsernamePassword login form ${CATALINA_HOME}/webapps/idp/login.jsp can be adapted by adding a checkbox:

Followed, the In-Flow *reset-settings* application configuration is for the shipped JAAS UsernamePassword login handler. If you are using another login handler (i.e. RemoteUser) you have to assure, that the GET parameter is transmitted to the IdP plugin.

```
...
<form ...>
  <table>
  ...
    <tr>
      <td colspan="2">
        <input type="checkbox" name="resetuserconsent" value="true" />
        Reset my attribute release approvals
      </td>
    </tr>
  </table>
</form>
...
```

Standalone mode

For the standalone mode, the JSP has to be protected either by container managed authentication or Shibboleth Service Provider, that the `REMOTE_USER` is provided.

The standalone JSP can be called like `https://idp.example.org/uApprove/reset-approvals.jsp?standalone-next-url=http://go.here.org/after/reset`. The parameter `standalone-next-url` has to be set.

1.4.7 List-approvals configuration

Do not use this configuration if you use *reset-approvals* application. Please delete the settings in the previous version.

For the standalone mode, the JSP has to be protected either by Shibboleth Service Provider, that the `REMOTE_USER` is provided.

SP configuration

- Install and setup Shibboleth SP on IdP host. The metadata exchange with the IdP, and configure the SP to transition to IdP directly IdP without through the DS when execute Shibboleth authentication.
- Add uid to REMOTE_USER in /etc/shibboleth/shibboleth2.xml.

```
<!-- The ApplicationDefaults element is where most of Shibboleth's SAML bits are defined. -->
<ApplicationDefaults entityID="https://idp.example.jp/shibboleth"
                     REMOTE_USER="uid eppn persistent-id targeted-id">
```

- Add attribute definition of uid in /etc/shibboleth/attribute-map.xml.

```
<Attribute name="urn:mace:dir:attribute-def:uid" id="uid"/>
<Attribute name="urn:oid:0.9.2342.19200300.100.1.1" id="uid"/>
```

IdP configuration

- Define uid in /opt/shibboleth-idp/conf/attribute-resolver.xml.
- Define Policy Requirement Rule to allow only uid to /opt/shibboleth-idp/conf/attribute-filter.xml.

JSP configuratrion

- Add the following definition to /etc/httpd/conf.d/ssl.conf.

```
...
<LocationMatch /uApprove/list-approvals.jsp>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  require valid-user
</LocationMatch>
...
```

The JSP can be called like:

> *https://idp.example.org/uApprove/list-approvals.jsp*

1.5 Run

Restart Tomcat:

```
/sbin/service tomcat6 stop && sleep 10 && /sbin/service tomcat6 start
```


If the Tomcat manager (http://idp.example.org:8080/manager/html/) is installed, web applications can easily be restarted separate.

Check the logs, if the startup of the IdP and uApprove.jp web application was successful.
If it was successful, try to access any Service Provider using your Identity Provider.

1.6 Troubleshooting

1.6.1 Logging

Cause the IdP plugin runs within the Shibboleth IdP web application, it uses the IdP logger. The IdP logger is configured by /opt/shibboleth-idp /conf/logging.xml:

```
...
<logger name="ch.SWITCH.aai" level="DEBUG">
 <appender-ref ref="IDP_PROCESS"/>
</logger>
<logger name="jp.gakunin.shibboleth" level="DEBUG">
 <appender-ref ref="IDP_PROCESS"/>
</logger>
...
```

The logging of uApprove.jp viewer application is configured as described. Adjust the log level to DEBUG.

1.6.2 Tomcat, Jasper JSP compiling for 1.5 target

uApprove.jp is shipped for a JDK 1.5 target. However your Tomcat setup runs with a JDK 1.5, it could be that the Jasper engine compiles the JSP's on the fly for another target (e.g. 1.3). The following configuration in ${CATALINA_HOME}/conf/web.xml specifies for which target the JSP's has to be compiled:

```
<servlet>
  ...
  <init-param>
    <param-name>compilerSourceVM</param-name>
    <param-value>1.5</param-value>
  </init-param>
  <init-param>
    <param-name>compilerTargetVM</param-name>
    <param-value>1.5</param-value>
  </init-param>
  ...
</servlet>
```

1.7 References

- Install Shibboleth 2.3 Identity Provider, Tomcat and Apache
- LogBack manual

2. Configuration of Shibboleth Service Provider

2.1 Metadata

An SP can define the attributes that the SP requires or desires using <RequestedAttribute> elements within <AttributeConsumingService> elements within <SPSSODescriptor> element. The <RequestedAttribute> element have the following attributes:

- **isRequired** (optional)
    - Boolean flag indicated that this attribute is required or desired by the SP,
      Default value: false.

2.1.1 Description of attribute

Can be described the description of attribute by adding a following attribute to the <RequestedAttribute> element. The described description is displayed in Attribute Selection Page by the viewer application of uApprove.jp.

- **uajpmd:description** (optional)
  - The description of this attribute.

Using a <uajpmd:RequestedAttributeExtension> element can provide the description by multiple language. The descriptions defined by <uajpmd:RequestedAttributeExtension> element takes precedence over the descrived one by the uajpmd:description attribuite. The <uajpmd:RequestedAttributeExtension> element must be contained within the <Extensions> element that is a chiled of <SPSSODescriptor> element.

The <uajpmd:RequestedAttributeExtension> element must have the following attibutes:

- **uajpmd:FriendlyName**
  - Specify the same value as the FriendlyName atttibute of the <RequestedAttribute> element to associate this element.

The description of attribute is described in the <uajpmd:Description> element. The <uajpmd:RequestedAttributeExtension> element can be contain one or more <uajpmd:Description> elements.

The <uajpmd:Description> element must have the following attibutes:

- **xml:lang**
  - The language used in the description.

Example:

```
<md:EntitiesDescriptor Name="uapprovejp-dev-metadata.xml"
                    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
                    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                    xmlns:shibmd="urn:mace:shibboleth:metadata:1.0"
                    xmlns:uajpmd="http://www.gakunin.jp/ns/uapprove-jp/metadata"
                    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  ...
  <md:EntityDescriptor entityID="...">
    <md:SPSSODescriptor>
      ...

      <md:Extensions>
        ...
        <RequestedAttributeExtension
          xmlns="http://www.gakunin.jp/ns/uapprove-jp/metadata"
          FriendlyName="displayName">
          <Description xml:lang="en">Our SP uses the displayName attribute in order to display your name to our web page</Description>
          <Description xml:lang="ja">SPはウェブページに名前を表示するためにdisplayName属性を使用します</Description>
        </RequestedAttributeExtension>
        ...
      </md:Extensions>
      ...

      <md:AttributeConsumingService index="1">
          <md:ServiceName xml:lang="en">Sample Service</md:ServiceName>
          <md:ServiceDescription xml:lang="en">
              An example service that requires a human-readable identifier and optional name and e-mail address.
          </md:ServiceDescription>

        <md:RequestedAttribute FriendlyName="eduPersonPrincipalName"
                          Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
                          NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                          isRequired="true"/>
        <md:RequestedAttribute FriendlyName="mail"
                          Name="urn:oid:0.9.2342.19200300.100.1.3"
                          NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                          uajpmd:description="Our SP uses the mail attribute in order to fill the registration form with your mail
address."/>
        <md:RequestedAttribute FriendlyName="displayName"
                          Name="urn:oid:2.16.840.1.113730.3.1.241"
                          NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
      </md:AttributeConsumingService>
      ...
  </md:SPSSODescriptor>
```