uApprove.jp-2.2.1c のインストールおよび設定方法

目次

- 1. Shibboleth Identity Provider のための uApprove.jp-2.2.1c のインストールおよび設定方法
  - 1.1 はじめに
  - 1.2 前提条件
  - 1.3 インストール
  - 1.4 設定
  - 1.5 実行
  - 1.6 トラブルシューティング
  - 1.7 リファレンス
- 2. Shibboleth Service Providerの設定
  - 2.1 Metadata
- 3 アップデート
  - 3.1 テーブルの追加
  - 3.2 mysql.commandsの更新
  - 3.3 古いjarファイルの削除
  - 3.4 web.xmlの確認

1. Shibboleth Identity Provider のための uApprove.jp-2.2.1c のインストールおよび設定方法

1.1 はじめに

uApprove.jp (uApproveを機能拡張しました)は、 エンドユーザに利用するサービスに送信する属性の表示の選択を提供するアプリケーションです。

本ガイドでは Shibboleth Identity Provider (IdP)のための uApprove.jp のインストールおよび設定方法を説明します。
IdP プラグインおよび uApprove.jp viewer のインストール方法と、 ストレージとしてファイルもしくは SQL データベースを用いた設定方法を示します。

このガイドでは例として以下の設定値を使用します。

- `idp.example.org`
  - Identity Provider の DNS 名
- `/opt/uApprove`
  - uApprove.jp をインストールするディレクトリ
- `/opt/uApprove/conf`
  - uApprove.jp の設定ファイルのディレクトリ
- `/opt/shibboleth-identityprovider-2.x`
  - Shibboleth IdP のインストール時にZIPファイルを展開したディレクトリ (`install.sh` が存在するディレクトリ)
- `/opt/shibboleth-idp`
  - Shibboleth IdP のインストール先
- `${CATALINA_HOME}`
  - Tomcat のインストール先 (例えば、`/usr/java/tomcat`)

1.2 前提条件

uApprove.jp のインストールと設定の前に、以下の前提条件が揃っていることを確認してください。

- Shibboleth Identity Provider **2.3**.
  - 正しく配備した Shibboleth Identity Provider が必要です。
    Java と Tomcat が動作することも含みます。
- SQL Database
  - ストレージとして SQL データベースを使用する場合は SQL サーバが必要です。
    uApprove.jp は既に MySQL 用の設定が行われていますが、他の SQL サーバも使用できます。

1.3 インストール

uApprve.jp をダウンロードして unzip してください:

```
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

ライブラリファイルと設定ファイルをコピーしてください:

```
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

ライブラリファイルと設定ファイルをコピーしてください:

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

1.4 設定

uApprove.jp は二つのソフトウェアから成り立っています。 IdP コンテクストの中で動作する IdP プラグインと IdP コンテクストから独立した uApprove.jp viewer アプリケーションです。
どちらも共通のデータストレージとその操作のためのライブラリを使用します。

1.4.1 uApprove.jp 共通: ストレージ等

ストレージは、主に下記のデータを保管します。

- ユーザ名
- ユーザが同意した Term of use の最終バージョン
- Shibboleth SP にリリースされた暗号化された属性のリスト
- Shibboleth SP の Entity ID

最初にどちらのストレージを使用するか決めなければなりません:

- ファイルベース
    - 少人数の構成で利用することを推奨します。(100 ユーザ未満)
- SQL データベース
    - MySQL データベースを完全にサポートしています。JDBC コネクタを持つデータベースでも動くでしょう。もしかしたら 動かすために高等な設定が必要になるかもしれません。

ファイルベース

ファイルベースストレージの設定は /opt/uApprove/conf/common.properties で行います:

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

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


uApprove-log.xml が置かれるディレクトリが Tomcat から読み書きできることを確認してください。



uApprove-log.xml は uApprove.jp が生成しますので、事前に作成しないでください。

データベース

最初にデータベースとユーザを作成します:

```
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;
```

次に、テーブルを作成します:

```
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;
```

以下のテーブルはuApprove.jp-2.2.1で追加しました。

- *CheckAlways*

以下のテーブルはuApprove.jp-2.2.1cで追加しました。

- *BackChannelAccess*
- *NamesOfApprovedService*

簡単にデータベースの設定が正しいかを確認するためには、下記コマンドを利用できます:

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

データベースを利用する設定は /opt/uApprove/conf/common.properties で行います:

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

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

uApprove.jp は BoneCP ライブラリにより提供される JDBC コネクションプーリングサービスをサポートしています。

コンテナー管理コネクションもアプリケーション管理コネクションも設定可能です。

データベース固有の設定すべては /opt/uApprove/conf/database.properties で行います:

```
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
```

MySQL 以外の SQL サーバを使用する場合は上記の値を使用する SQL サーバに合わせる必要があります。
JDBC ドライバをライブラリフォルダに配置するのを忘れないでください。
必要であれば、SQL コマンドは /opt/uApprove/conf/custom-sql.commands で再定義できます。

共通設定

使用条件のテキストをユーザに表示するためのオプション TermsOfUseManager があります。 TermsOfUseManager を有効にしたい場合は、/opt/uApprove/conf/common.properties で定義します:

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

⚠ 含まれている terms-of-use.xml は空のファイルですので、独自の使用条件のバージョンと本文を記述できます。
実例として SWITCHaai VHO terms of use (doc/terms-of-use-SWITCH.xml) がありますので、参照してください。

IdP プラグインおよび viewer アプリケーションは機密情報を交換するため、 /opt/uApprove/conf/common.properties で定義される共通秘密鍵 (128bit, 16バイト)により暗号化と復号を行います。

```
sharedSecret=QErDXYZEAoS6jooPvdBhQg==
```

簡単に 16 バイトの乱数を生成するためには、以下コマンドを利用できます:

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

1.4.2 IdP プラグインの設定

Shibboleth IdP の調整

IdP プラグインは Shibboleth IdP web アプリケーション /opt/shibboleth-identityprovider-2.x/src/main/webapp/WEB-INF/web.xml で有効にする必要があります:

```
<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>
```

⚠ uApprove.jp の filter および filter-mapping の定義は web.xml の他の filter および filter-mapping より後に記述しなければなりません。

Shibboleth IdP を再デプロイします:

```
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
```

idp.warファイルを${CATALINA_HOME}/webappsにコピーします:

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

SP ブラックリスト

SP ブラックリストはユーザの同意対象から除外するリソースを定義します。
SP ブラックリストはオプションで設定は /opt/uApprove/conf/idp-plugin.properties で行います:

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

sp-blacklist 中の各行ではブラックリスト対象のリソースを表す正規表現の定義を行います:

```
# 例 1: 特定のリソース
https://sp¥.example¥.org/shibboleth

# 例 2: SP に存在するすべてのアプリケーション
https://sp¥.example¥.org/.*

# 例 3: 特定ドメインにあるすべての SP
https://.*¥.example¥.org/.*
```

属性の順序とブラックリストの作成

uApprove.jp は目的のリソースに当該のユーザのために送信されるすべての属性を表示します。
属性リストの順序を定義できます。さらに、必要ならば属性の隠蔽もできます。属性リストの設定はオプションで、 /opt/uApprove/conf/idp-plugin.properties で行います:

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

attribute-list は下記のようになります:

```
# 属性の順序を定義
surname
givenName
postalAddress
...

# 隠す属性
!persistentId
!transientId
```


attribute-list の属性名は /opt/shibboleth-idp/conf/attribute-resolver.xml の 属性 id に従わなければいけません。

その他の設定

データベースを使用する場合は各プロバイダのアクセスに対して記録を残すことが可能で、 これはすべての属性の送信がデータベースに保存されていることを意味します。
IdP プラグイン は monitoring only mode で実行することも可能で、 これはすべてのプロバイダのアクセスを記録しますが、ユーザの同意のためのやりとりは行いません。
viewer web アプリケーションを使用する場合、 IdP プラグインは viewer web アプリケーションのデプロイ先を知っている必要があります。
IdP プラグインに、isPassiveリクエストに関して考慮の必要があることを伝えることができます。
設定は /opt/uApprove/conf/idp-plugin.properties で行います:

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

1.4.3 Viewer 設定

uApprove.jp webapp のための tomcat のデプロイの記述を ${CATALINA_HOME}/conf/Catalina/localhost/uApprove.xml で定義します:

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

設定ディレクトリに従って、viewer アプリケーション の設定を修正します。 変更は/opt/uApprove/viewer-2.2.1c/webapp/WEB-INF/web.xml で行います:

```
<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>
```

必要であれば、ロゴやナビゲーションメニューを変更します。 ロゴは/opt/uApprove/viewer-2.2.1c/webapp/images以下のイメージファイルやURLを使うことができます。

設定は/opt/uApprove/viewer-2.2.1c/webapp/header.jspで行います:

```
<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>
```

uApprove webapp をデプロイします:

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

ローカライズ

viewer web アプリケーションは属性名や説明といった静的なテキストと動的なテキストに対して多言語対応です。
静的なテキストは次の言語が対応しています: 英語(en), ドイツ語(de), フランス語(fr), イタリア語(it), ポルトガル語(pt), 日本語(ja)
ローカライズされた属性名と説明は /opt/shibboleth-idp/conf/attribute-resolver.xml (Shibboleth IdP resolver) から取得します。ローカライズされた属性名や説明は変更や追加ができます:

```
...

<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>
...
```

> SWITCHaai 属性については doc/attribute-descriptions.xml で次の各言語を用意しています: 英語(en), ドイツ語(de), フランス語(fr), イタリア語(it), 日本語(ja)
> 学認の属性については、同じファイルで次の各言語を用意しています: 英語(en), 日本語(ja)

viewer はユーザのブラウザが要求する言語で表示します。その言語が利用できない場合はデフォルトのロケールである *en* を使用します。
特定の言語を強制することもできます。設定は /opt/uApprove/conf/viewer.properties で行います:

```
useLocale=en_US
```

包括的な同意

ユーザが包括的な属性送信同意を与えることができます。これはユーザが二度と uApprove から送信する属性を問われないということです。 ユーザが異なるリソースのアクセスに対してそれぞれ同意を行うようにするためには /opt/uApprove/conf/viewer.properties に より機能を無効にできます:

```
globalConsentPossible=true
```

ログ出力

viewer web アプリケーションは Shibboleth IdP のように LogBack (1.7 リファレンス)を使用します。
logback の設定ファイルは /opt/uApprove/conf/viewer.properties で定義します:

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

ログファイルは logging.xml で設定します。 Tomcat から書き込みできることを確認してください:

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

Tomcat のフロントエンドとしての Apache

Apache HTTPd を tomcat のフロントエンドに使用している場合は、viewer web アプリケーション にプロキシーする設定を /etc/httpd/conf.d/ssl.conf で行います:

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

Apache HTTPd を再起動してください:

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

1.4.4 Shibboleth IdP の profile handler

uApprove.jpはAttribute QueryのSAML応答メッセージの属性のリストをユーザーの同意内容に応じて生成します。

ネームスペースの定義

プロファイルハンドラファイル(例えば、/opt/shibboleth-idp/conf/handler.xml)内に このプラグインのためにネームスペース宣言を追加する必要があります。以下のように行います:

1. <ProfileHandlerGroup> ルート要素のxmlns:xsi 属性の前に xmlns:uajpph="http://www.gakunin.jp/ns/uapprove-jp/profile-handler" 属性を追加
2. xsi:schemaLocation 属性の値のリストの最後に下記を追加
   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">
...
```

Attribute Queryプロファイルハンドラの変更

Attribute Queryプロファイルハンドラを変更する必要があります。以下のように行います:

1. xsi:type 属性の値を ph:SAML1AttributeQuery から uajpph:SAML1AttributeQueryUApprove に変更
2. xsi:type 属性の値を ph:SAML2AttributeQuery から uajpph:SAML2AttributeQueryUApprove に変更

```
...
    <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 の attribute filter

uApprove.jpはSPのメタデータ内の <RequestedAttribute> 要素 (2. Shibboleth Service Providerの設定を参照ください)と IdP の属性フィルターポリシーファイル内の<AttributeFilterPolicy> 要素を 理解することで、属性が必須であるかオプショナルであるかを決定します。

ネームスペースの定義

属性フィルターポリシーファイル(例えば、/opt/shibboleth-idp/conf/attribute-filter.xml)内に このプラグインのためにネームスペース宣言を追加する必要があります。以下のように行います:

1. <AttributeFilterPolicyGroup> ルート要素のxmlns:xsi 属性の前に xmlns:uajpmf="http://www.gakunin.jp/ns/uapprove-jp/afp/mf" 属性を追加
2. xsi:schemaLocation 属性の値のリストの最後に下記を追加
   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 以降はネームスペースの URI が変更になりました。

- uApprove.jp-2.2.1
  http://www.gakunin.jp/ns/uapprove-jp
- uApprove.jp-2.2.1a 以降
  http://www.gakunin.jp/ns/uapprove-jp/afp/mf

ルールの定義

uApprove.jpはPolicy RequirementルールとAttribute RulesのPermit/Deny Valueルールを拡張します:

- Policy Requirementルール
  - このルールは<afp:PolicyRequirementRule xsi:type="uajpmf:AttributeUapprove"> 要素によって定義されます。 そして、SPのメタデータ内に<AttributeConsumingService> 要素が存在する場合に適用されます。
- Attribute RuleのPermit/Deny Value Rule
  - このルールは下記のオプショナルな属性を持つ <afp:PermitValueRule xsi:type="uajpmf:AttributeUapprove"/> 要素と <afp:DenyValueRule xsi:type="uajpmf:AttributeUapprove"/> 要素両方によって定義されます。
    - **isApproved** (オプショナル)
      - ユーザによってSPへ送信を許可された属性を示すブーリアン値です。
        デフォルト値: true
        しかし、もし SPのメタデータ内の<AttributeConsumingService> 要素内の <RequestedAtrribute> 要素で isRequired="true" となっている属性は、必須属性として扱われ、常にSPに送信されます。
    - **requestedOnly** (オプショナル)
      - SPのメタデータで定義された属性のみを表示するか否かを示すブーリアン値です。
        デフォルト値: false

xsi:type="uajpmf:AttributeUapprove" を持たない許可された属性は必須属性として扱われます。

uajpmf:AttributeUapprove Matchファンクションを用いたPermit Valueルールの例を示します:

```
<!--    ================================================================================
case 1: メタデータ内に AttributeConsumingService 要素を持つSPにマッチします。
        eduPersonPrincipalName 属性はオプショナル属性で、ユーザは送信するか否かを選択できます。
        eduPersonAffiliation 属性は必須属性で、常に送信されます。
        ================================================================================ -->
<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: メタデータ内に AttributeConsumingService 要素を持たないSPにマッチします。
        eduPersonPrincipalName 属性 および eduPersonAffiliation 属性共に必須属性です。
        ================================================================================ -->
<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: 全てのSPにマッチします。
        eduPersonPrincipalName 属性 および eduPersonAffiliation 属性はオプショナル属性です。
        ================================================================================ -->
<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 設定

> *list-approvals*アプリケーションを使用する場合は設定しないでください。

オプションである *reset-approvals* は以下の二通りの方法で操作可能です:

- In-flow モード

    - ユーザがログイン手続きにおいてリセット可能であることを意味します。 たとえばログインフォームのチェックボックスを使用するといった方法です。
- Standalone モード
    - JSP などを直接呼ぶ操作です。

In-flow モード

In-Flow の *reset-approvals* アプリケーションを有効にするには、Shibboleth の UsernamePassword ログインフォーム ${CATALINA_HOME}/webapps/idp/login.jsp にチェックボックスの追加で対応可能です。

> in-flow の *reset-approvals* アプリケーションの設定は提供されている JAAS UsernamePassword ログインハンドラのためのものです。 他のログインハンドラ(例えば RemoteUser)を使用している場合は GET パラメータが IdP プラグインに送信されることを確かめてください。

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

Standalone モード

Standalone モードのためには、JSP はコンテナ管理による認証、もしくは REMOTE_USER を提供する Shibboleth Service Provider のどちらかで JSP が保護されている必要があります。

> Shibboleth Service Provider で JSP を保護する設定方法については次節の 1.4.7 List-approvals 設定 をご参照ください。IdP から uid を送信する設定についても記載されています。

standalone JSP は下記のように呼ばれます。この際パラメータ `standalone-next-url` が設定される必要があります。

> *https://idp.example.org/uApprove/reset-approvals.jsp?standalone-next-url=http://go.here.org/after/reset*

1.4.7 List-approvals 設定

> *reset-approvals*アプリケーションを使用する場合は設定しないでください。 以前のバージョンで*reset-approvals*アプリケーションを使用するように設定している場合は、その設定を削除してから以下の設定を行ってください。

オプションである *list-approvals* はStandalone モードで使用します。 REMOTE_USER を提供するShibboleth Service Provider で JSP が保護されている必要があります。

SPの設定

- IdPサーバにShibboleth SPをインストールしてください。IdPとメタデータ交換し、SPでのシボレス認証時にDSを経由せず直接IdPに遷移するように設定してください。
- /etc/shibboleth/shibboleth2.xmlで uid を REMOTE_USERに設定します。

```
<!-- 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">
```

- /etc/shibboleth/attribute-map.xmlに IdP から uid を受取る設定を追加します。

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

IdPの設定

- /opt/shibboleth-idp/conf/attribute-resolver.xmlにuidを設定します。
- /opt/shibboleth-idp/conf/attribute-filter.xmlでuidのみを送出するように設定します。

JSPの設定

- /etc/httpd/conf.d/ssl.conf で以下のように設定します。

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

下記のように実行します。

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

1.5 実行

Tomcat を再起動してください:

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

もしも Tomcat マネージャ(http://idp.example.org:8080/manager/html/)がインストールされている場合、web アプリケーションは Tomcat とは独立に容易に再起動できます。

ログをチェックして、IdP と uApprove web アプリケーションの起動が成功しているかどうか確認してください。
もし成功している場合はあなたの Identity Provider を用いて適当な Service Provider へのアクセスを試みてください。

1.6 トラブルシューティング

1.6.1 ログ出力

IdP プラグインは Shibboleth IdP web アプリケーション中で動作するため IdP logger を使用します。IdP logger の設定は /opt/shibboleth-idp/conf/logging.xml で行います:

```
...
<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>
...
```

uApprove.jp viewer のログは 1.4.3 で説明したように設定します。log level を DEBUG に変更します。

1.6.2 Tomcat, Jasper JSP compiling for 1.5 target

uApprove.jp は JDK1.5 を対象に提供されています。あなたの Tomcat がバージョン 1.5 以上の JDK で実行していたとしても Jasper エンジンは実行中の JSP を別ターゲット(例 1.3)向けにコンパイルしている可能性があります。 以下の ${CATALINA_HOME}/conf/web.xml の設定は JSP のコンパイルターゲットを指定するものです:

```
<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 リファレンス

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

2. Shibboleth Service Providerの設定

2.1 Metadata

SPは <SPSSODescriptor> 要素内の <AttributeConsumingService> 要素内の <RequestedAttribute> 要素を用いて SP が必須とする、もしくは希望する属性を定義できます。 <RequestedAttribute> 要素は下記の属性を持ちます:

- **isRequired** (optional)
  - SPによって必須である、もしくは希望する属性を示すブーリアン値。
    デフォルト値: false (希望する)。

2.1.1 属性の説明

<RequestedAttribute>要素に以下の属性を記述して、属性の説明文を定義できます。定義した説明文は、uApprove.jp の viewer アプリケーションによって表示されます。

- **uajpmd:description** (optional)
  - 属性の説明文。

<uajpmd:RequestedAttributeExtension> 要素を用いると、複数の言語で説明文を定義することができます。 <uajpmd:RequestedAttributeExtension> 要素の定義は uajpmd:description属性の定義に優先します。 <uajpmd:RequestedAttributeExtension> 要素は <SPSSODescriptor> 要素の子要素 <Extensions> に記述します。

<uajpmd:RequestedAttributeExtension> 要素には以下の属性が必要です。

- **uajpmd:FriendlyName**
  - 関連づけたい <RequestedAttribute> 要素のFriendlyName 属性の値と同じ値を記述します。

説明文は <uajpmd:Description> 要素に記述します。<uajpmd:RequestedAttributeExtension> 要素は一つ以上の <uajpmd:Description> 要素をもつことができます。

<uajpmd:Description> 要素には以下の属性が必要です。

- **uajpmd:FriendlyName**
  - 関連づけたい <RequestedAttribute> 要素のFriendlyName 属性の値と同じ値を記述します。

<AttributeConsumingService> 要素の例を下記に示します:

```
<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>
```

3 アップデート

2.2.1bから2.2.1cへのアップデートについては、基本的には1.3 インストールおよび1.4 設定に記述しているように Shibboleth IdP および Viewer を再デプロイしていただきますが、特に注意いただきたい手順を以下に挙げます。

3.1 テーブルの追加

2.2.1cでは以下のテーブルを追加する必要があります。

- BackChannelAccess
  バックチャネルアクセスの記録用
- NamesOfApprovedService
  同意済みSPのサービス名

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

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 BackChannelAccess ENGINE=MyISAM;
ALTER TABLE NamesOfApprovedService ENGINE=MyISAM;
```

3.2 mysql.commandsの更新

SQLを定義したmysql.commandsファイルを2.2.1c付属のファイルに差し換えます。

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

3.3 古いjarファイルの削除

以下のディレクトリに2.2.1bのjarファイル(common-2.2.1b.jar, idp-plugin-2.2.1b.jar)が残っている場合は、Shibboleth IdP および Viewer をデプロイする前に削除します。

- /opt/shibboleth-identityprovider-2.x/lib/
- $CATALINA_HOME/work/Catalina/localhost/idp/WEB-INF/lib/
- $CATALINA_HOME/work/Catalina/localhost/uApprove/WEB-INF/lib/

3.4 web.xmlの確認

- /opt/shibboleth-identityprovider-2.x/src/main/webapp/WEB-INF/web.xml

に追加した設定が 1.4.2 IdP プラグインの設定 の通りになっているか確認してください。
特にuApprove.jpの以前のバージョンでは以下の要素が無かったようですので、この部分に特にご注意ください。

```
    <dispatcher>REQUEST</dispatcher>
    <dispatcher>FORWARD</dispatcher>
```