Trinamix

Enabling SSO on custom ADF application

Blog By - Priya Garg

Enabling SSO on custom ADF application

What is Single Sign On?

Single sign on is an authentication method to access multiple sites using one credentials.

Single sign on worked on the trust relationship created between website using service provider and identity provider. This trust we create upon certificates which get exchanged between service provider and identity provider. This certificate is usually use to sign identity information that's being sent from the identity provider to the service provider so as that the service provider knows it's coming from a trusted source. In SSO, this identity data takes the form of tokens which contain identifying bits of data about the user like user's email address or a user name.

The login flow usually looks like this:

1. A user browses to the website they need access to, aka, the Service Provider.
2. The Service Provider sends a token that contains some information about the user, like their email address, to the Identity Provider, as part of a request to authenticate the user.
3. The Identity Provider first checks to ascertain whether the user has already been authenticated, in which case it will grant the user access to the Service Provider application and skip to step 5.
4. If the user hasn't logged in, they're going to be prompted to try to do so by providing the credentials required by the Identity Provider.
5. Once the Identity Provider validates the credentials provided, it will send a token back to the Service Provider confirming a successful authentication.
6. This token is passed through the user's browser to the Service Provider.
7. The token that's received by the Service Provider is validated consistent with the trust relationship that was set up between the Service Provider and therefore the Identity Provider.
8. The user is granted access to the Service Provider.

Firstly, we'd like to make a decision what is going to be our service provider and identity provider.

There are 3 ways of implementing SSO.

1. Integrating WebLogic hosted application with IDCS where identity provider and service provider both will be IDCS only.
   Below blog we can follow to do configuration on PaaS
   https://blogs.oracle.com/blogbypuneeth/steps-to-configure-saml-20-with-idcs-identity-cloud-service-as-identity-provider-and-jcs-oracle-java-cloud-service-as-service-provider

2. Integrating WebLogic hosted application with SaaS, in this case identity provider will be SaaS and service provider will be IDCS and to implement this we have 2 ways:
    a. Using SAML authentication
        i. Here we create SAML configuration on WebLogic and provide provider meta data file to SaaS to import in their cloud and then SSO will be enabled
        Below blog we can follow to do configuration:
        https://blogs.oracle.com/blogbypuneeth/steps-to-configure-saml-20-with-oracle-saas-as-idp-and-oracle-java-cloud-service-as-sp

    b. Using OAuth and load balancer
        i. Here we will create instance on JCS including load balancer and will create policies on JCS only.
        Need to follow below doc for configuration:

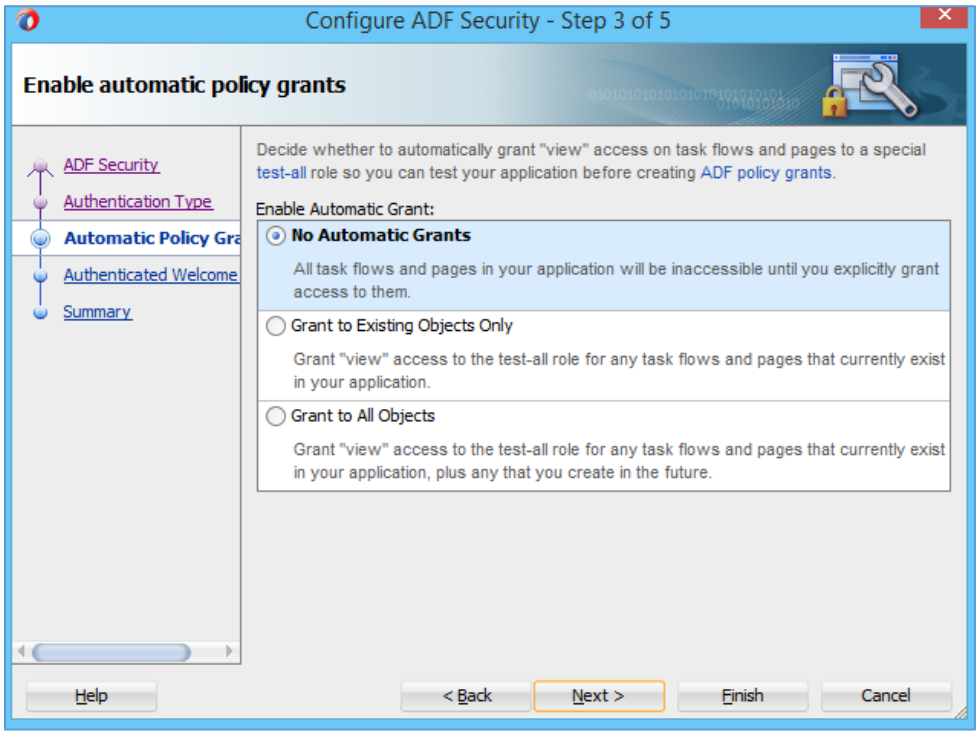
        https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=445968441628726&parent=SrDetailText&sourceId=3-23037909671&id=2434846.1&_afrWindowMode=0&_adf.ctrl-state=bi3g31b43_4

        https://support.oracle.com/epmos/faces/DocumentDisplay?_afrLoop=446092019731924&parent=SrDetailText&sourceId=3-23037909671&id=2637614.1&_afrWindowMode=0&_adf.ctrl-state=bi3g31b43_53

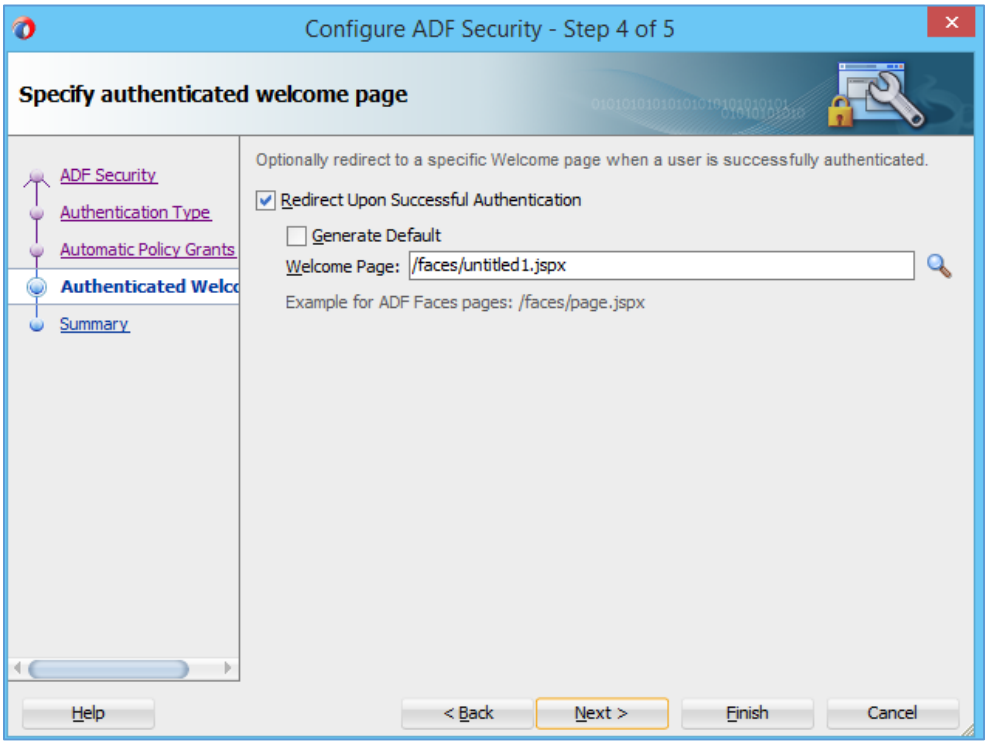
3. Integrating WebLogic hosted application with SaaS, in this case identity provider will be IDCS and service provider will be SaaS.
    https://docs.oracle.com/en/solutions/extend-saas-with-java-cloud-service-apps/enable-oracle-fusion-applications-cloud-service-federation-and-oauth-trust-oracle-identity-cloud-ser1.html#GUID-6CC76095-418B-42E5-A9E6-D3DA3B82FBDB

Below are changes which we need to make in custom ADF application:

Change security type to client based as mentioned below



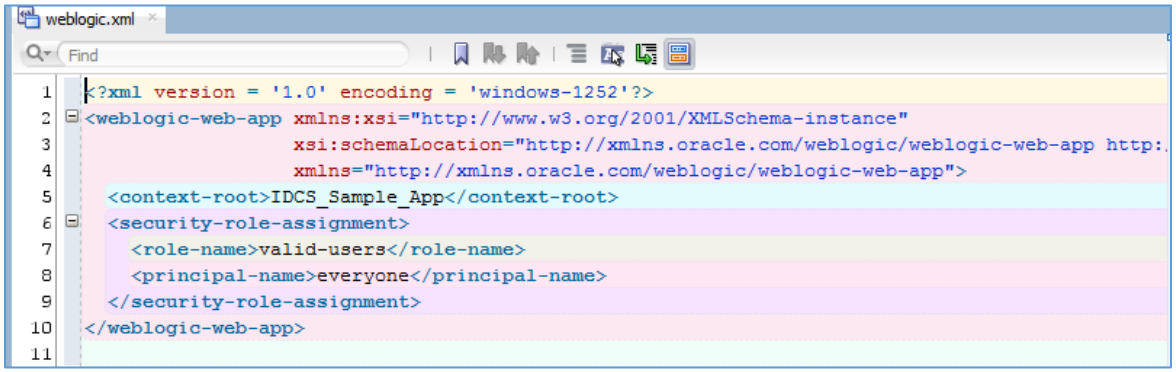
Redirect on desired page



After enabling security, go to web.xml file and check changes there, this code should be there

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>adfAuthentication</web-resource-name>
    <url-pattern>/adfAuthentication</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>valid-users</role-name>
  </auth-constraint>
</security-constraint>
<login-config>
  <auth-method>CLIENT-CERT</auth-method>
</login-config>
<security-role>
  <role-name>valid-users</role-name>
</security-role>
```

Now go to weblogic.xml



Provide context-root name same as whatever you have mentioned in project properties of ViewController.

If you are using load balancer to implement SSO then concat **__protected/** with context-root-name.

Under principal-name, add groups which we have in IDCS.

If there are 3 groups, then there will be 3 principal-name like this

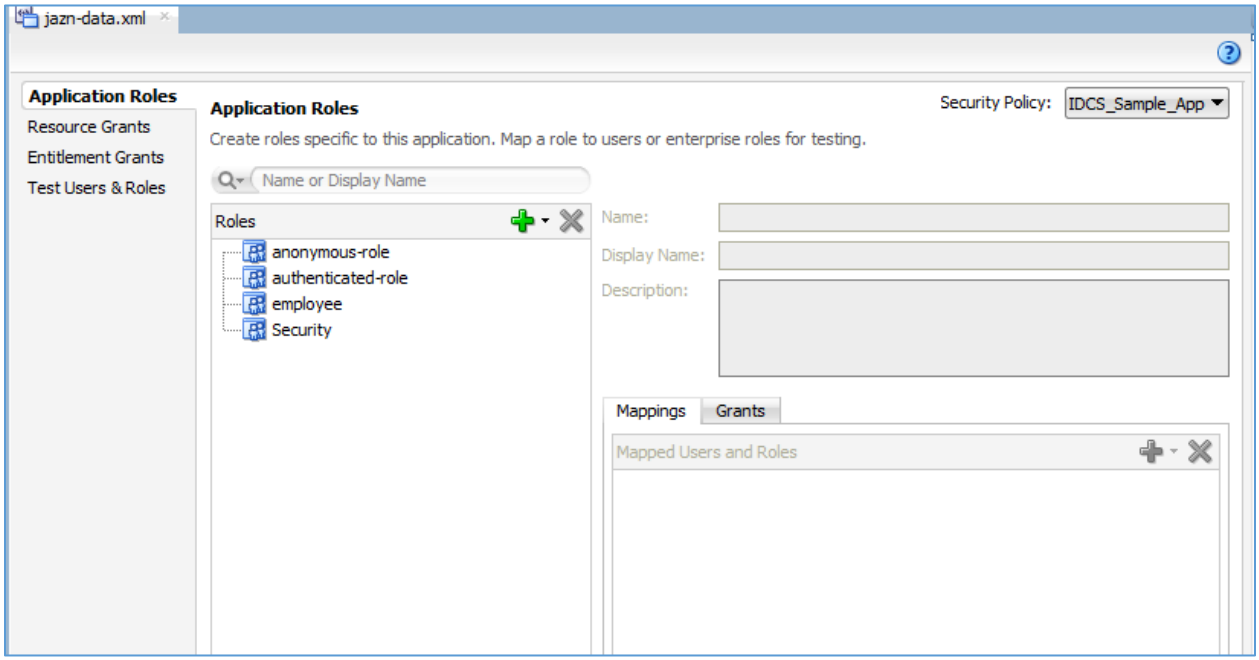
<principal-name>group1</principal-name>

<principal-name>group2</principal-name>

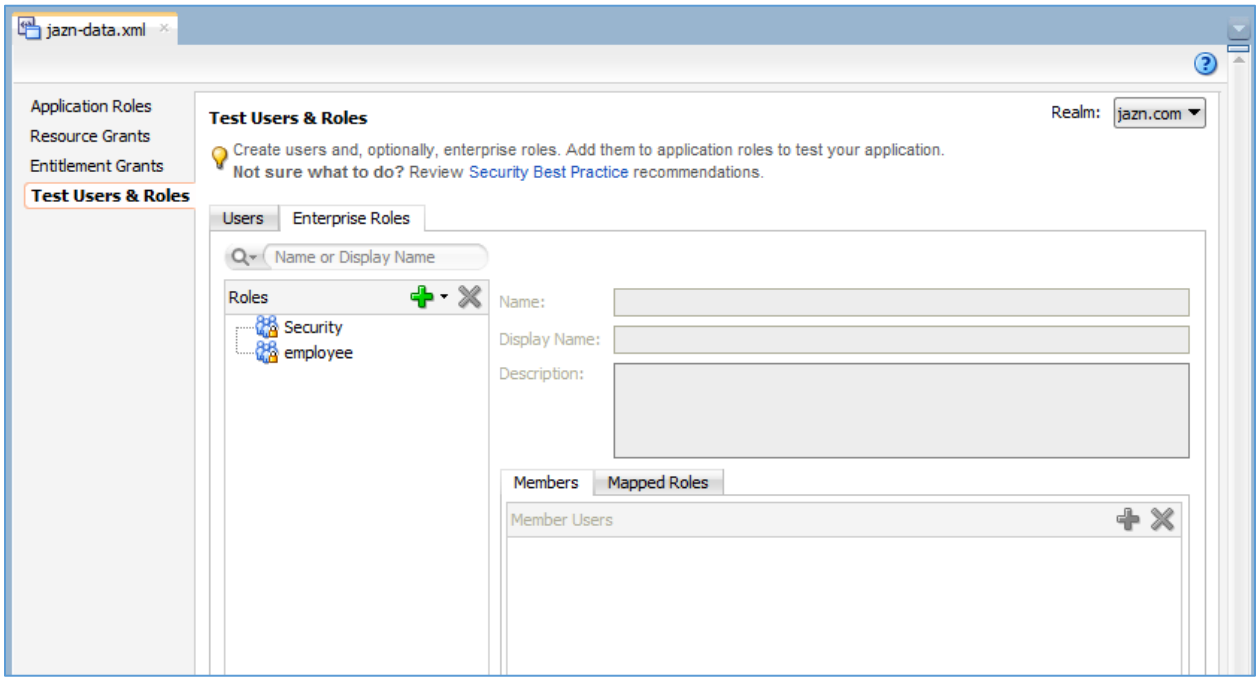
<principal-name>group3</principal-name>

Now go to jazn.xml

Add application role with same name which we have defined in IDCS



And same role defines in enterprise role



Map these roles to application role, after adding make change in code

```
<app-role>
        <name>employee</name>
        <class>oracle.security.jps.service.policystore.ApplicationRole</class>
```

<members>

<member>

<class>weblogic.security.principal.WLSGroupImpl</class>--replace original code with this line of code

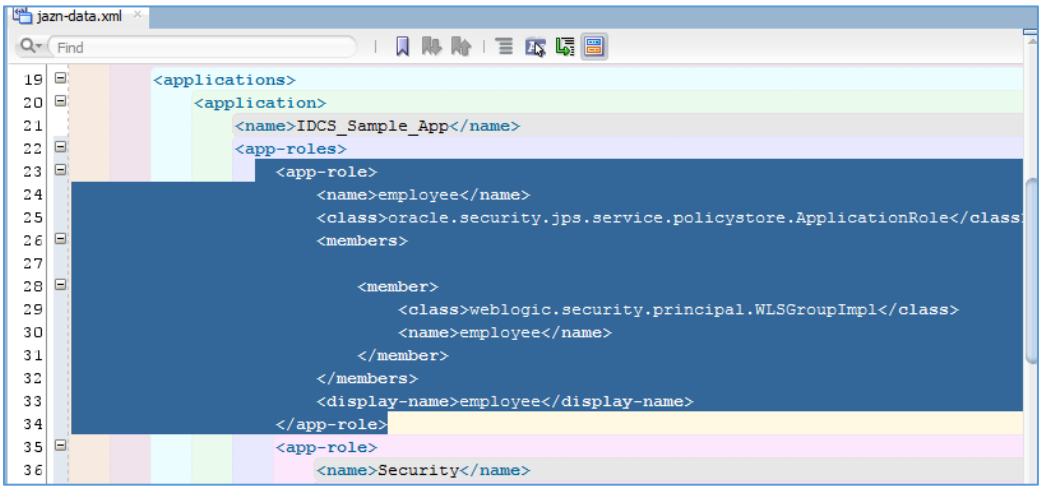
<name>employee</name>

</member>

</members>

<display-name>employee</display-name>

</app-role>



Assign application role to your task flows and webpages as your requirement.

Deploy code and test it.

About Author:-

Priya Garg is working with Trinamix Inc. as a Technical Consultant having good experience in Oracle ADF, Oracle PaaS and also have knowledge on OIC, Oracle visual builder, WebLogic server.