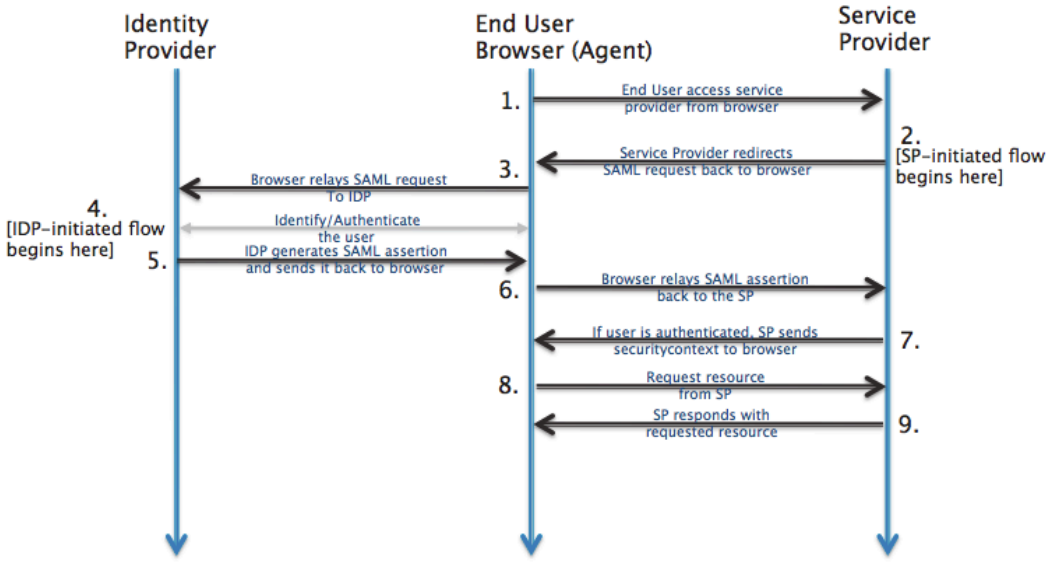
Registering in Okta-developer IdP

SAML: Overview

[1] SAML is mostly used as a web-based authentication mechanism as it relies on the browser being used as an agent that brokers the authentication flow. At high-level, the authentication flow of SAML looks like this:



Identity Provider	developer.okta.com	The 3rd-party entity that takes care of the user's authentication; There are multiple such entities with SAML protocol support, such as OpenSSO, SSOCircle.com, OneLogin.com, Salesforce.com… For this example, we'll be using developer.okta.com
End-user browser agent	Pentaho User	User that accesses BA-server via browser
Service Provider	BA-server	Pentaho BA Server

[1] http://developer.okta.com/docs/guides/saml_guidance.html

Registering in Okta-developer Identification Provider (IdP)

Note: Each user that intends to use BA-server needs to register itself first in Okta-developer

Prerequisites

1. **Have your chosen Service Provider (i.e. Ba-Server) metadata xml file at hand**

   Developer/QA only: if none is created yet, you can leverage on the already existing SP metadata file for SSOCircle.com. For this:
   a. Go to https://pentaho.box.com/s/x0s0hcvs13te25clqo5lenmthu6p9cim
   b. Go to "ssocircle-metadata" folder
   c. Download "pentaho-ssocircle-dev-sp.xml" and rename it to something more identifiable with okta (e.g. "pentaho-okta-dev-sp.xml");

2. **Have a certificate file holding the public signing key at hand**

   Developer/QA only: if none is created yet, we can quickly create one based of the information stored if the service provider's metadata xml file:
   a. Create a new (empty) file with extension .cer (e.g. "pentaho-dev-certificate.cer")
   b. Open it using an editor of your choice (Notepad++, Sublime Text, ..)
   c. Add the following content **as-is** to the file:

```
-----BEGIN CERTIFICATE-----
<certificate-goes-here>
-----END CERTIFICATE-----
```

   d. Open your Service provider (i.e. Ba-Server) metadata xml file using an editor of your choice (Notepad++, Sublime Text, ..)
   e. Locate the `<ds:X509Certificate>` tag
      i. If two such tags exist, we want the one that is inside the `<md:KeyDescriptor use="signing">` tag and not the one inside the `<md:KeyDescriptor use="encryption">`
      ii. Copy its contents **as-is (do not change/edit anything)**
      iii. Paste it in the newly created certificate file, replacing the placeholder `<certificate-goes-here>` with this pasted content
      iv. **Roughly put**, your certificate should look something like this:

```
-----BEGIN CERTIFICATE-----
MIIDUjCCAjqgAwINBgCQYDVQQGEwJGSTEQMA4GA1UE
CBMHVXVzaW1hYTEAoTD1JNNSBTb2Z0d2FyZSBPeTEM
MAoGA1bG8wHhcNMTMwMTAxMTExWhcNMjIxMjMwMTEy
ODAxWjBrMQswMHVVRMA8G1UEBxMGVs7BlLYW/GuHE=
-----END CERTIFICATE-----
```

Registering yourself in developer.okta.com

1. Go to http://developer.okta.com
2. Click “Get a free developer account”
3. Fill in all fields
    a. Be sure to provide a real email, as a confirmation email will be sent to you.
    b. **Important:** The confirmation email will also provide you with the url for **your** Okta assigned hostname; that is the one that we’ll be using from now on.
        i. **Example:** https://dev-128494.oktapreview.com/

4. Once registration is done, login to **your** okta assigned hostname
5. Click “Admin” (top right corner)
6. Click “Add applications” (right-side “Shortcuts” menu, 1rst option)
7. Click “Create New app”
8. In the app name, place “pentaho”
    a. app logo is optional
    b. leave “app visibility” unchanged
9. click “Next”

Field Name	**Field value**
Single sign on URL	http://localhost:8080/pentaho/saml/SSO
Recipient URL and Destination URL checkbox	Checkbox enabled
Audience URI	pentaho
Default Relay State	<empty>
Name ID format	EmailAddress
Application Username	Okta username

10. Click “Show advanced settings”

Field Name	**Field value**
Response	signed
Assertion Signature	signed
Signature algorithm	RSA- SHA256
Digest algorithm	SHA256
Assertion encryption	unencrypted
Enable Single Logout	Checkbox: checked
Single Logout URL	http://localhost:8080/pentaho/saml/SingleLogout
SP Issuer	pentaho
Signature Certificate	Upload the certificate addressed in the “Prerequisites” section, step 2
Authentication Context	PasswordProtectedTransport
Honor Force Auth.	Yes
Saml Issuer ID	http://www.okta.com/${org.externalKey}

11. In the “Attributes Statements” section add 3 fields:

Name	Name format	Value
First Name	Unspecified	user.firstName
Last Name	Unspecified	user.lastName
Email	Unspecified	user.email

12. In the “Group Attributes Statement” add 1 field:
    a. we will be creating and configuring those groups later on (step 22)

Name	Name format	Dropdown	Filter
Pentaho Role	Unspecified	StartsWith	Pentaho:

13. Click the “Preview SAML Assertion” button
    a. **Save this xml metadata information in your local machine**.
    b. **Based on the information we’ve typed so far, this is Okta offering us a auto-generated “Pentaho Service Provider Assertions” that we can add to out metadata xml file**
    c. Open your service provider’s metadata xml file
    d. Scroll all the way down: after the `</md:EntityDescriptor>` tag click enter and paste the xml content Okta has just provided us.
    e. Save and close the file.
14. (Back to the wizard) Click “Next”
15. Check the radio button “I'm an Okta customer adding an internal app”
16. Enable the checkbox “This is an internal app that we have created”
17. Click Finish.
18. Top Menu Bar, select “Applications > Applications”
19. Select app “pentaho”
20. Select the 2nd menu tab “Sign On”
21. On the SAML 2.0 section, click the “Identity Provider Metadata” link
    a. **Save this xml metadata information in your local machine**.
    b. **This is Okta offering us a auto-generated “Okta IdP Metadata” xml.**
    c. Open the IdP metadata xml file and search for the “entityID”; it should hold a key of type http://www.okta.com/${org.externalKey}
        i. Example: `http://www.okta.com/exk58mn1xrPYSE1lA0h7`
    d. **Save this entityID value; later on, we will need to place it in the pentaho.saml.cfg “saml.idp.url” key**
22. (Back to the page) Top Menu bar , select “Directory > Groups”
23. In the “Groups” page, section add 3 groups:

Name	Description
Pentaho:Administrator	Pentaho BA-server’s Administrator Role
Pentaho:Power User	Pentaho BA-server’s Power User Role
Pentaho:Report Author	Pentaho BA-server’s Report Author Role

24. (Back to the page) Top Menu bar, select “Directory > People”
25. Select your user
26. Select the 1st tab , “Applications”
27. Click “Assign Applications”
28. Click “Assign App“ on the “pentaho” application line; click “Save”;
29. Select the 2nd tab, “Groups”
30. In the search bar, start typing “Pentaho”
31. Add **at least** “Pentaho:Administrator” group to your user (you can add others if you choose to do so)
32. Done

Recap

We have:

1. Registered onto developer.okta.com

2. Added an application called “pentaho”, passed the endpoint for it (`http://localhost:8080/pentaho/saml/SSO`) and have configured it to work with SAML 2.0

3. Created Okta “groups” that start with a “Pentaho:” prefix and assigned those to our user (and only those)

4. Configured the SAML response so that the authenticated response (from Okta to Pentaho) also carries a list of attributes, namely those “Pentaho:*” groups

5. Got the idp and sp metadata xml files

6. Got the Okta url to use from the “entityID” value

Q&A

Q1 | Do I need a certificate to sign the authentication requests?

Yes.

For this sample, we are using a certificate provided by spring-security-saml, stored in a .jks (keystore file).

It's already bundled in the saml-authentication-provider sample (jar:/security/keystore.jks).

You can get the original here: https://github.com/spring-projects/spring-security-saml/blob/1.0.1.RELEASE/core/src/test/resources/org/springframework/security/saml/key/keystore.jks

This certificate is used by some IdP's, such as SSOCircle.com and okta.developer.com.

If you plan to connect to some other IdPs, then you must ensure you update the keystore file to include the certificate provided by that Identification Provider.