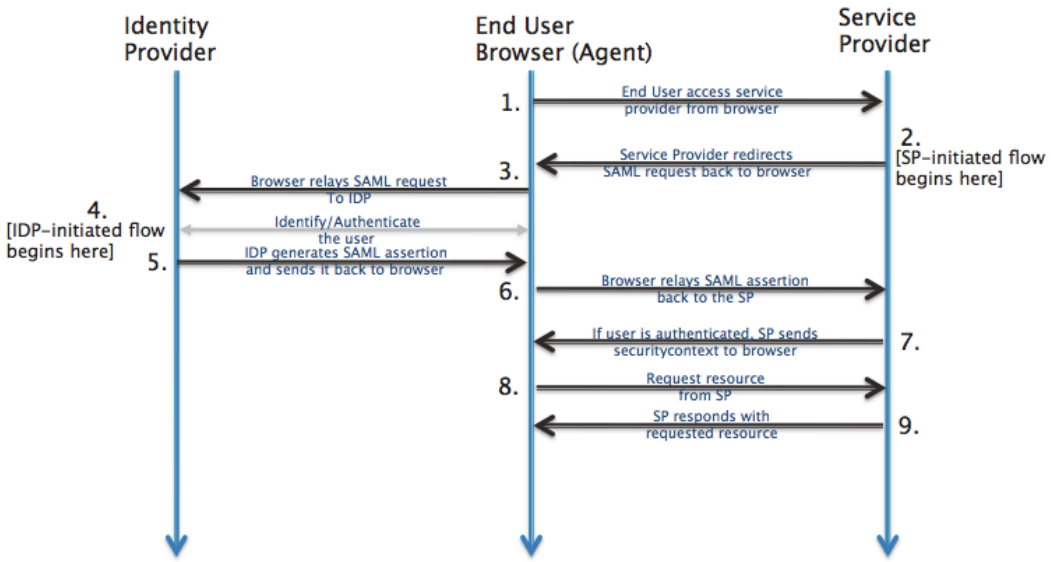
# Activating SAML sample in BA-server

**Note**: Activating SAML sample explains how to enable the SAML logic, **regardless** of which IdP we choose to use as a means to perform user authentication

## 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 | SSOCircle.com , OKTA, … | 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, … |
| 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

# Pre-requisites for SAML authentication provider sample

1. Complete **one of** the following documentations:
    a. “Registering in SSOCircle IdP”
    b. “Registering in OKTA-developer IdP”
    c. “Registering in PingIdentity IdP”
    d. “Registering in Salesforce-developer IdP”
    e. “Registering in MS ADFS 3.0 IdP”

2. Have the following 3 items of information:
    a. The URL for the chosen 3rd party identification provider (IdP)
    b. The absolute path to the chosen’s IdP metadata xml file
    c. The absolute path to Pentaho SP metadata xml file
        i. Next to this document, you should have a “`resources`” folder. Inside it you will find a standard Pentaho SP metadata xml for download.

3. Pentaho-SAML OSGI .kar file
    a. `git clone` https://github.com/pentaho/pentaho-engineering-samples
    b. Navigate to /Samples for Extending Pentaho/Reference Implementations/Security/SAML 2.0/
    c. Maven build the .kar file ( “`maven package`” )
    d. OSGI .kar file built and created in (…)/SAML 2.0/pentaho-saml-assembly/target

# Preparing BA-server for SAML authentication provider sample

**Note 1:** Use BA-Server EE 6.0.0-GA build

**Note 2:** This section includes preparation tasks, i.e. **tasks that only need doing once**

1. Edit pentaho-solutions/system/karaf/etc/custom.properties:
    a. find "`org.springframework.security.context, \`" and replace it with "`org.springframework.security.context; version\="2.0.8.RELEASE", \`"
    b. below the line above add a new one:
    "`org.springframework.security.ui; version\="2.0.8.RELEASE", \`"

    **please ensure** that both lines end with the "`, \`" ( comma, whitespace, backward slash ), as stated above.

2. Start BA-server
3. Next to this document, you should have a "`resources`" folder with 3 files:

    a. pentaho-sp.xml
    b. applicationContext-spring-security-saml.xml
    c. logout.jsp

4. Place the built pentaho-saml.kar file into pentaho-solutions/system/karaf/deploy
    a. check log files to see if all went well; you should see a line stating:
    `Creating configuration from pentaho.saml.cfg`
    b. look into pentaho-solutions/system/karaf/etc
        i. you will notice a `pentaho.saml.cfg` file was created

5. Stop BA-server.
6. Place `applicationContext-spring-security-saml.xml` in pentaho-solutions/system
7. Place `logout.jsp` in tomcat/webapps/pentaho
8. Edit `pentaho.saml.cfg` and update the following 3 keys with the values mentioned in the "Pre-requisites for the SAML authentication provider sample" section:
    a. `saml.idp.url` : The URL for the chosen 3rd party identification provider (IdP)
    b. `saml.idp.metadata.filesystem`: The absolute path to the chosen's IdP metadata xml file
    c. `saml.sp.metadata.filesystem`: The absolute path to Pentaho SP metadata xml file

Example:

```
saml.idp.url=http://idp.ssocircle.com

saml.idp.metadata.filesystem=/users/pteixeira/saml/idp/ssocircl
e-idp-metadata.xml

saml.sp.metadata.filesystem=/users/pteixeira/saml/sp/pentaho-
sp-metadata.xml
```

9. Save and close the file.
10. Done.

## Activating BA-server's SAML authentication sample

1. Stop BA-server.
2. Edit pentaho-solutions/system/pentaho-spring-beans.xml
3. If not there yet, place line

```
<import resource="applicationContext-spring-security-saml.xml" />
```

after all other applicationContext-*.xml lines and before the pentahoObjects.spring.xml one. Example:

```
(…)
<import resource="applicationContext-spring-security-jdbc.xml" />
<import resource="applicationContext-spring-security-saml.xml" />
<import resource="pentahoObjects.spring.xml" />
(…)
```

4. Save and close the file.
5. Edit pentaho-solutions/system/security.properties and change the provider value to "`saml`".
    a. Example: from "`provider= jackrabbit`" to "`provider=saml`"
6. Save and close the file.
7. Done.

## De-activating BA-server's SAML authentication sample

1. Stop BA-server
2. Edit pentaho-solutions/system/pentaho-spring-beans.xml
3. Delete/Comment line

```
<import resource="applicationContext-spring-security-saml.xml" />
```

4. Save and close the file.
5. Edit pentaho-solutions/system/security.properties and change the provider value to something other than "saml".
    a. Example: from "`provider=saml`" to "`provider=jackrabbit`"
6. Save and close the file.
7. Done.

# Q&A

## Q1 | Can we add internationalization support to the logout page ?

Yes. Please do the following steps:

1. Open tomcat/webapps/Pentaho/logout.jsp with an editor of your choice
2. Locate the div with class "`logout-msg-wrapper`"
    a. Its content should be something like "`You have logged out of the User Console.`"
    b. Replace that with:
    `<%=Messages.getInstance().getString("UI.PUC.LOGOUT.HEADER")%>`
3. Locate the button with class "`back-to-login-btn`"
    a. Its content should be something like "`Return to the Login Page`"
    b. Replace that with:
    `<%=Messages.getInstance().getString("UI.PUC.LOGOUT.BUTTON")%>`
4. Save and close the file.
5. Using a tool such as Winrar, Winzip, 7-zip, etc.., open ( do not extract ) /tomcat/webapps/WEB-INF/lib/ pentaho-platform-extensions-6.0-SNAPSHOT.jar
6. Inside it, navigate to `/org/pentaho/platform/web/jsp/messages/`
7. Edit messages.properties and add the following 2 lines:

```
UI.PUC.LOGOUT.HEADER=You have logged out of the User Console.
UI.PUC.LOGOUT.BUTTON=Return to the Login Page
```

8. Redo step 7, this time for any of the other messages_<country>.properties that exist at that location
    a. Add the same keys, but a properly localized message
9. Save and close the files. Save and close the jar
    a. At this point the extraction tool you're using may ask you if you would like to update the jar file. Reply "yes".
10. Restart the server.