Web Single Sign-On with SAML 2.0

This tutorial demonstrates the usage of SAML 2.0 in different Web SSO Scenarios. We will use Oracle Weblogic Server 12.1.3 as the technical platform.

1 Contents

2 Introduction

While SAML is already widely used in the industry, the configuration within Weblogic Server is complex and in most companies not part of the regular routine. We want to have look at a simple SAML example that was published in an article by Vikrant Sawant in 2007. http://www.oracle.com/au/products/database/sso-with-saml-099684.html This former example demonstrates a Web SSO scenario using SAML 1.1 in Weblogic Server 9.2.
We want to upgrade this example, using SAML 2.0 in Weblogic Server 12.1.3.

This is a tutorial in which we will walk through all the necessary steps to setup and run the SAML 2.0 example. This includes the installation and configuration of weblogic server, creation of two weblogic server domains, installation of the test applications and configuration of the identity provider and service provider domains. To provide a comprehensive overview, the separate tutorial steps are summarized in mind map diagrams. The tutorial comprises a Service Provider initiated flow and an Identity Provider initiated flow, which both will be demonstrated during the testing steps.
As an addition, the tutorial demonstrates the usage of the weblogic feature “virtual user”.

The tutorial was developed and tested on a windows 7 machine. A zip package containing all necessary files is provided at the tutorial website. This also includes a text file with a set of windows commands to help setting up the domains and user configurations. We expect the tutorial to run also on Linux or any other platform supported by weblogic server, although this was not tested.

3 The Web SSO Tutorial

3.1 Tutorial Files

The following files are located at the website www.andreaswittmann.de:

www.andreaswittmann.de/weblogic-corner/saml2_sso/SAML2_Web_SSO_Tutorial.pdf
www.andreaswittmann.de/weblogic-corner/saml2_sso/SAML_SSO.zip

The zip archive contains the following files:

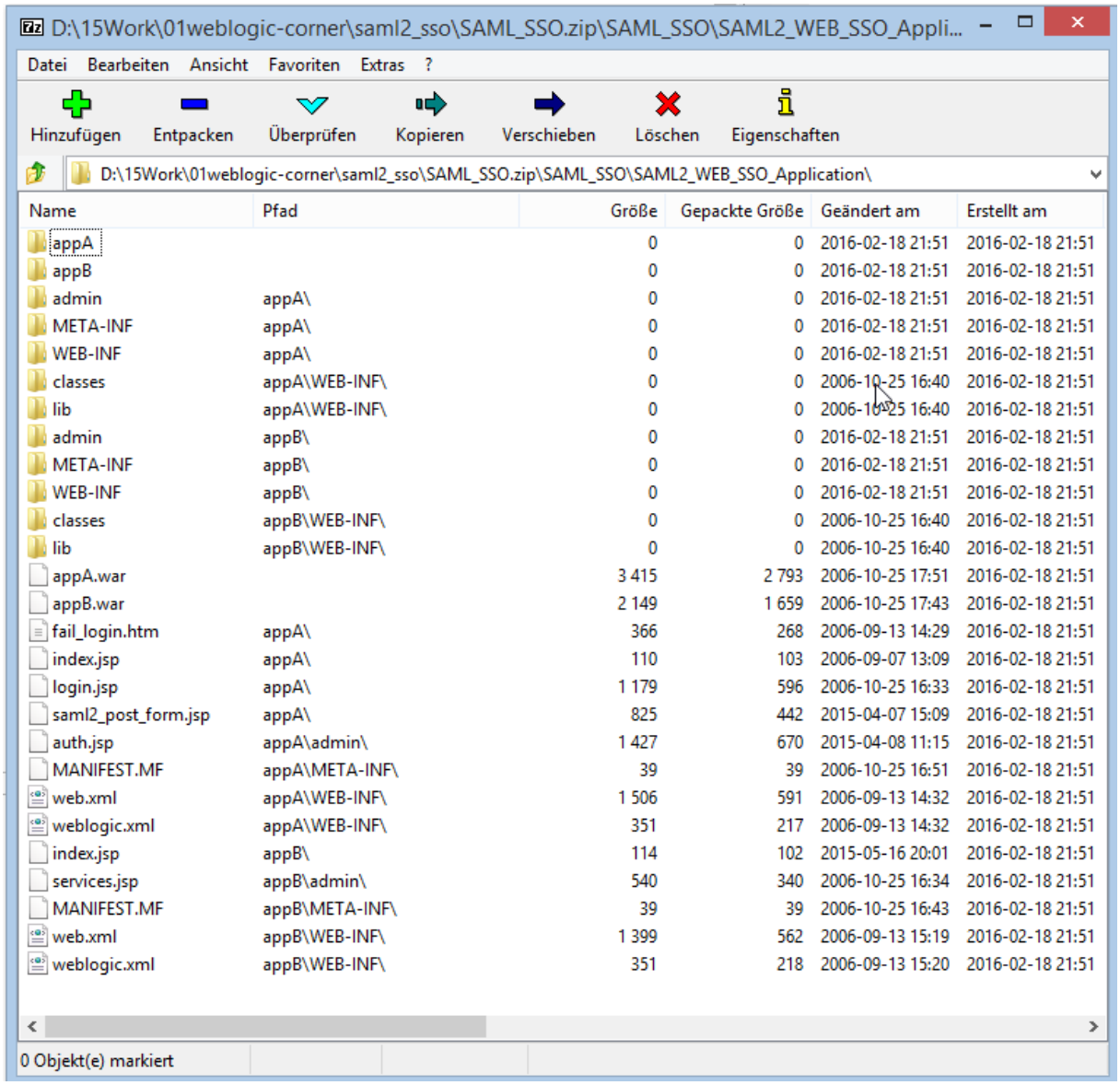


Figure 1. Contents of the archive SAML_SSO.zip

3.2 Documentation Links

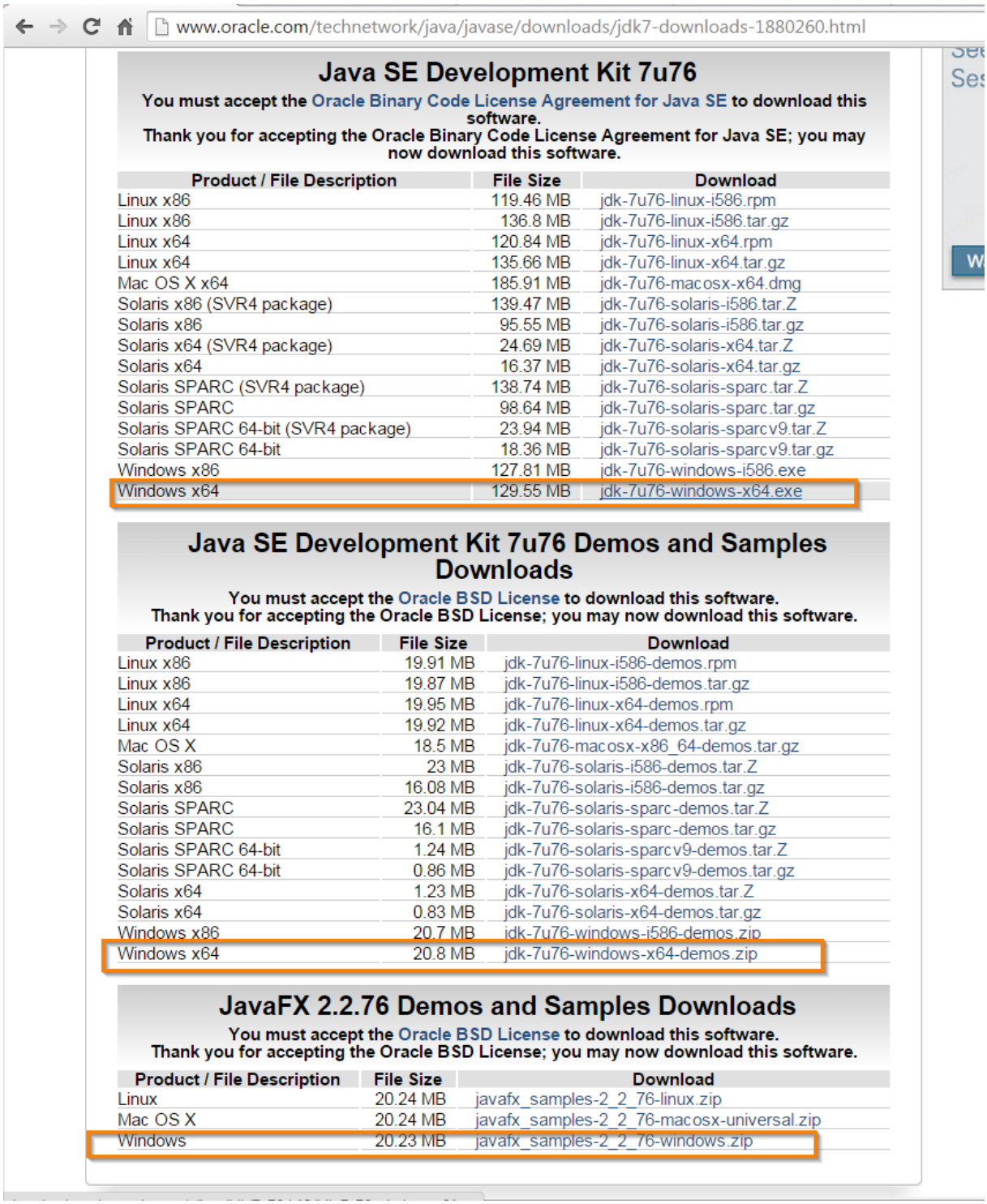
We summarize some documentations links in the following table.

OASIS SAML Home Page	https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security
OASIS SAML Executive Overview (PDF)	http://www.oasis-open.org/committees/download.php/13525/sstc-saml-exec-overview-2.0-cd-01-2col.pdf
OASIS SAML Technical Overview (PDF)	http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf
SAML 2.0,ein Tutorium – Teil 1:	kain_keller_JS_05_0_Annotated.pdf (Article in

Theorie aus XML Spectrum (www.javaspectrum.de)	German)
Configuring SAML 2.0 Services in Oracle® Fusion Middleware Administering Security for Oracle WebLogic Server	http://docs.oracle.com/middleware/1213/wls/SECMG/saml20.htm#SECMG318

3.3 Installation of Java

We install the latest Java 7 JDK from the download link
http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html



www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

Java SE Development Kit 7u76

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	119.46 MB	jdk-7u76-linux-i586.rpm
Linux x86	136.8 MB	jdk-7u76-linux-i586.tar.gz
Linux x64	120.84 MB	jdk-7u76-linux-x64.rpm
Linux x64	135.66 MB	jdk-7u76-linux-x64.tar.gz
Mac OS X x64	185.91 MB	jdk-7u76-macosx-x64.dmg
Solaris x86 (SVR4 package)	139.47 MB	jdk-7u76-solaris-i586.tar.Z
Solaris x86	95.55 MB	jdk-7u76-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.69 MB	jdk-7u76-solaris-x64.tar.Z
Solaris x64	16.37 MB	jdk-7u76-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	138.74 MB	jdk-7u76-solaris-sparc.tar.Z
Solaris SPARC	98.64 MB	jdk-7u76-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.94 MB	jdk-7u76-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.36 MB	jdk-7u76-solaris-sparcv9.tar.gz
Windows x86	127.81 MB	jdk-7u76-windows-i586.exe
Windows x64	129.55 MB	jdk-7u76-windows-x64.exe

Java SE Development Kit 7u76 Demos and Samples Downloads

You must accept the Oracle BSD License to download this software.
Thank you for accepting the Oracle BSD License; you may now download this software.

Product / File Description	File Size	Download
Linux x86	19.91 MB	jdk-7u76-linux-i586-demos.rpm
Linux x86	19.87 MB	jdk-7u76-linux-i586-demos.tar.gz
Linux x64	19.95 MB	jdk-7u76-linux-x64-demos.rpm
Linux x64	19.92 MB	jdk-7u76-linux-x64-demos.tar.gz
Mac OS X	18.5 MB	jdk-7u76-macosx-x86_64-demos.tar.gz
Solaris x86	23 MB	jdk-7u76-solaris-i586-demos.tar.Z
Solaris x86	16.08 MB	jdk-7u76-solaris-i586-demos.tar.gz
Solaris SPARC	23.04 MB	jdk-7u76-solaris-sparc-demos.tar.Z
Solaris SPARC	16.1 MB	jdk-7u76-solaris-sparc-demos.tar.gz
Solaris SPARC 64-bit	1.24 MB	jdk-7u76-solaris-sparcv9-demos.tar.Z
Solaris SPARC 64-bit	0.86 MB	jdk-7u76-solaris-sparcv9-demos.tar.gz
Solaris x64	1.23 MB	jdk-7u76-solaris-x64-demos.tar.Z
Solaris x64	0.83 MB	jdk-7u76-solaris-x64-demos.tar.gz
Windows x86	20.7 MB	jdk-7u76-windows-i586-demos.zip
Windows x64	20.8 MB	jdk-7u76-windows-x64-demos.zip

JavaFX 2.2.76 Demos and Samples Downloads

You must accept the Oracle BSD License to download this software.
Thank you for accepting the Oracle BSD License; you may now download this software.

Product / File Description	File Size	Download
Linux	20.24 MB	javafx_samples-2_2_76-linux.zip
Mac OS X	20.24 MB	javafx_samples-2_2_76-macosx-universal.zip
Windows	20.23 MB	javafx_samples-2_2_76-windows.zip

Figure 2. Java 7 Downloads with Demos and Samples.

We run the installer.

Datei: jdk-7u76-windows-x64.exe
CRC-32: a549a6a7
MD4: b2fbe1a78ca30c96a6e14554488f6b20
MD5: 02365745a4a68a44d6b6f5130a4ad4da
SHA-1: fa316d3c290172632a0a19afdf70e0361410ff54

The installation needs administration privileges.
We install to D:\10Oracle\02Java\jdk1.7.0_76
The documentation can be found http://docs.oracle.com/javase/7/docs/ here.

3.4 Installation of Weblogic Server 12.1.3

Reviewing the certification matrix in Ecxel from this link:
http://www.oracle.com/technetwork/middleware/fusion-middleware/documentation/fmw-1213certmatrix-2226694.xls

Oracle Fusion Middleware 12c (12.1.3.0.0) Certification Matrix

Oracle Fusion Middleware 12c (12.1.3.0.0) Products:
1. Oracle WebLogic Server
2. Oracle Coherence
3. Oracle HTTP Server
4. Oracle Application Development Framework
5. Oracle Fusion Middleware Infrastructure which includes:
   a. Oracle WebLogic and Oracle Fusion Middleware security infrastructure
6. Oracle TopLink
7. Oracle MapViewer
8. Oracle Data Integrator
9. Oracle Enterprise Data Quality
10. Oracle GoldenGate Veridata (Server, C Agent, Java Agent)
11. Oracle Golden Gate Monitor(Server, Java Agent)
12. Oracle Business Process Management
13. SOA Product Line - Oracle Service Bus, Oracle BPEL Process Manager, Oracle Event Processing, Oracle Enterprise Repository, Oracle SOA Suite (Oracle BPEL Process Manager, Oracle Service Bus, Oracle Business Activity Monitoring, Oracle B2B, Oracle Enterprise Scheduler, Oracle Event Processing, Oracle Application Integration Architecture Foundation Pack, Oracle Business Rules, Oracle Mediator, Oracle Human Workflow), Oracle Managed File Transfer and Oracle API Manager.

ALL includes the following products: Fusion Middleware 12c Products, with the Exception of:
Oracle Data Integrator (Agent, Studio, Console), Oracle Data Service Integrator (Server, IDE, Console)
Oracle GoldenGate Veridata (Server, C Agent, Java Agent), Oracle Golden Gate Monitor (Server, Java Agent).

For complete Fusion Middleware, OBI EE, and EPM product certifications , refer to http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html.
For information on supported Virtualization and Partitioning Technologies, refer to http://www.oracle.com/technetwork/middleware/ias/oracleas-supported-virtualization-089265.html.
For Interoperability and Compatibility Information, refer to http://www.oracle.com/pls/topic/lookup?ctx=fmw121300&id=INTOP310
For information on Oracle JDeveloper supported configurations, refer to http://www.oracle.com/technetwork/developer-tools/jdev/documentation/121300-cert-2164864.html
For information on Adapter certifications, refer to http://www.oracle.com/technetwork/middleware/adapters/overview/adapters-12c-certmatrix-2328619.xls

Product Offering	RELEASE	PROCESSOR	OSVERSION	OS Update Type	OS Update Level	OS 3264	ORA CLEAPP	JDKVENDOR	JDKVERSION	JDK3264BIT	EXCEPTIONSADDITIONALINFO
Oracle WebLogic Server Oracle Data Integrator Agent Oracle GoldenGate Veridata Java Agent Oracle GoldenGate Monitor Java Agent	FMW 12.1.3.0.0	Microsoft Windows x64 (64-bit)	7	Service Pack	1	64	64	Oracle JDK	1.7.0_51+	64	This 64 bit operating system is supported for single user only in a development environment.

*OS Update Level	The version listed specifies the minimum update level / service pack / technology level certified. For example, 6 means that 6 and higher is certified.
*JDKVERSION	A plus sign (+) after the fourth digit in the version number indicates that this and all higher versions of the JRE/JINIT/JDK extensions are certified. For example, 1.7.0_55+ means that 1.7.0_55 and any higher 1.7.0_xx versions are certified.

Figure 3. Certification Matrix showing the certified release for Windows 7, 64 bits.

From the OTN we download the zip distribution and the supplement zip from this download link: http://www.oracle.com/technetwork/middleware/weblogic/downloads/wls-main-097127.html

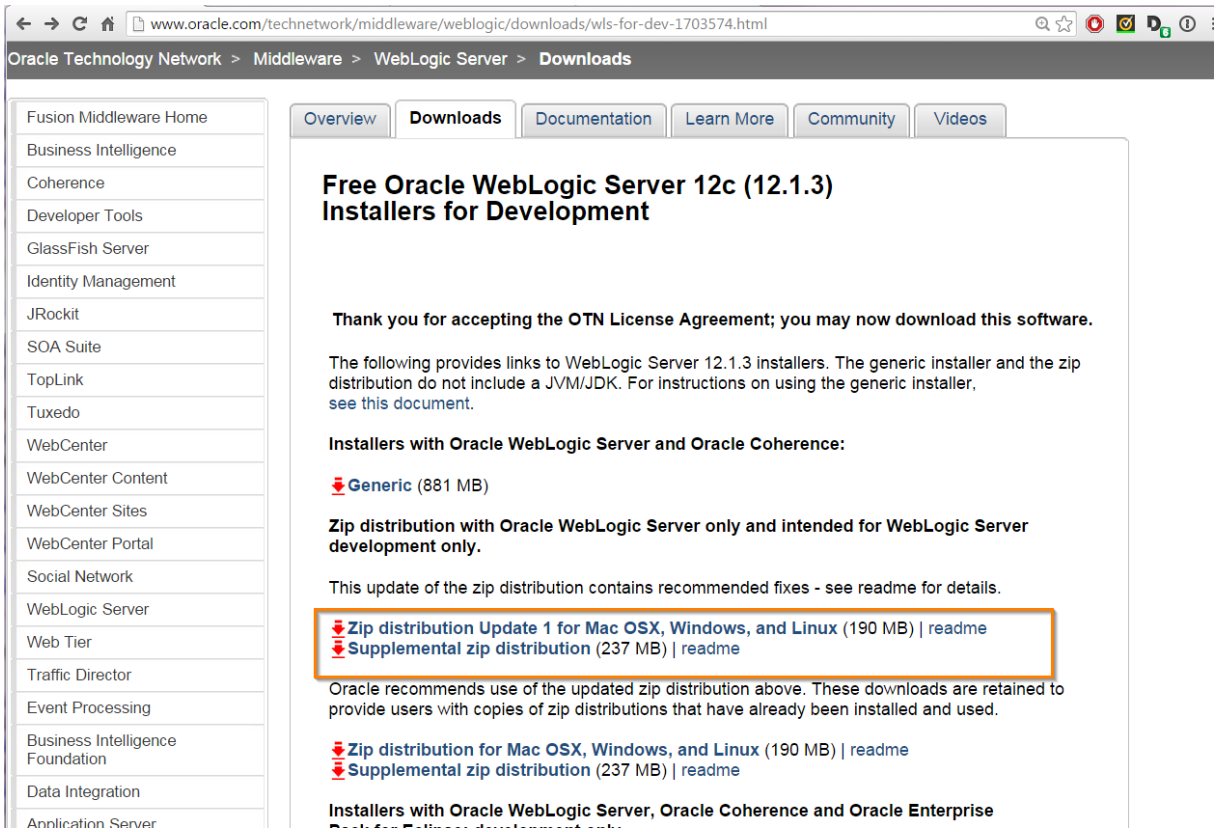


Figure 4. OTN Download Page for Weblogic Server.

We follow the instruction of the Readme.

Frist we unzip the distribution to a location which will become the new middleware home for this installation.
The windows file explorer fails to unzip 2 files from the package because filenames are too long.
We use the jar tool to unzip instead and it works fine.
Running the installation:

```
D:\>cd D:\10Oracle\06WLS12\wls12130
D:\10Oracle\06WLS12\wls12130>set MW_HOME=D:\10Oracle\06WLS12\wls12130
D:\10Oracle\06WLS12\wls12130>set JAVA_HOME=D:\10Oracle\02Java\jdk1.7.0_76
D:\10Oracle\06WLS12\wls12130>configure.cmd

## Creating a new Domain:

D:\10Oracle\06WLS12\domains\mydomain>%JAVA_HOME%\bin\java.exe %JAVA_OPTIONS% -Xmx1024m -XX:MaxPermSize=256m weblogic.Server
## User=weblogic
## Password=welcome1

## Starting new domain (in new shell)
D:
cd D:\10Oracle\06WLS12\domains\mydomain
set MW_HOME=D:\10Oracle\06WLS12\wls12130
set JAVA_HOME=D:\10Oracle\02Java\jdk1.7.0_76
startWebLogic.cmd
```

The newly created domain can be found at http://localhost:7001/console

Running the installation of the supplement package:

```
D:\10Oracle\06WLS1213>set JAVA_HOME=D:\10Oracle\02Java\jdk1.7.0_76
D:\10Oracle\06WLS1213>%JAVA_HOME%\bin\jar  -xvf wls1213_devzip_supplemental_update1.zip
```

Error Message:

```
D:\10Oracle\06WLS12\wls12130>run_samples.cmd
"Setting up proper ACLs for D:\10Oracle\06WLS12\wls12130 ... (operation takes awhile)"
Zuordnungen von Kontennamen und Sicherheitskennungen wurden nicht durchgeführt.
Username:weblogic
```

```
password: Die Version von D:\10Oracle\06WLS12\wls12130\mask.com ist nicht mit der ausgeführten Windows-Version kompatibel.
Öffnen Sie die Systeminformationen des Computers, um zu überprüfen,
 ob eine x86-(32 Bit)- oder eine x64-(64 Bit)-Version des Programms erforderlich ist, und wenden Sie sich anschließend an den
Herausgeber der Software.

Re-enter password:Die Version von D:\10Oracle\06WLS12\wls12130\mask.com ist nicht mit der ausgeführten Windows-Version
kompatibel. Öffnen Sie die Systeminformationen des Computers, um zu übe
rprüfen, ob eine x86-(32 Bit)- oder eine x64-(64 Bit)-Version des Programms erforderlich ist, und wenden Sie sich
anschließend an den Herausgeber der Software.
"pwdset" kann syntaktisch an dieser Stelle nicht verarbeitet werden.
```

There are conflicts with Windows 64 bit versions.
Solutions: We supply username and password on the command line, thus mask.com is not called.
We ignore the ACL Settings, instead we start a command shell as administrator.

```
D:\10Oracle\06WLS12\wls12130>run_samples.cmd  weblogic welcome1
```

To start the example server:

```
D:
cd D:\10Oracle\06WLS12\wls12130\wlserver\samples\domains\wl_server
set MW_HOME=D:\10Oracle\06WLS12\wls12130
set JAVA_HOME=D:\10Oracle\02Java\jdk1.7.0_76
startWebLogic.cmd
```

It will be available at http://192.168.56.1:7001/index.jsp
Ok.

3.5 Example Overview

In this tutorial we want to demonstrate two message flows which stem directly from the “OASIS SAML Technical Overview” document. The first case is the “SP-Initiated SSO with Redirect and POST Binding”. We copy the relevant image from the document and overlay it with the servers and components which will form this example.

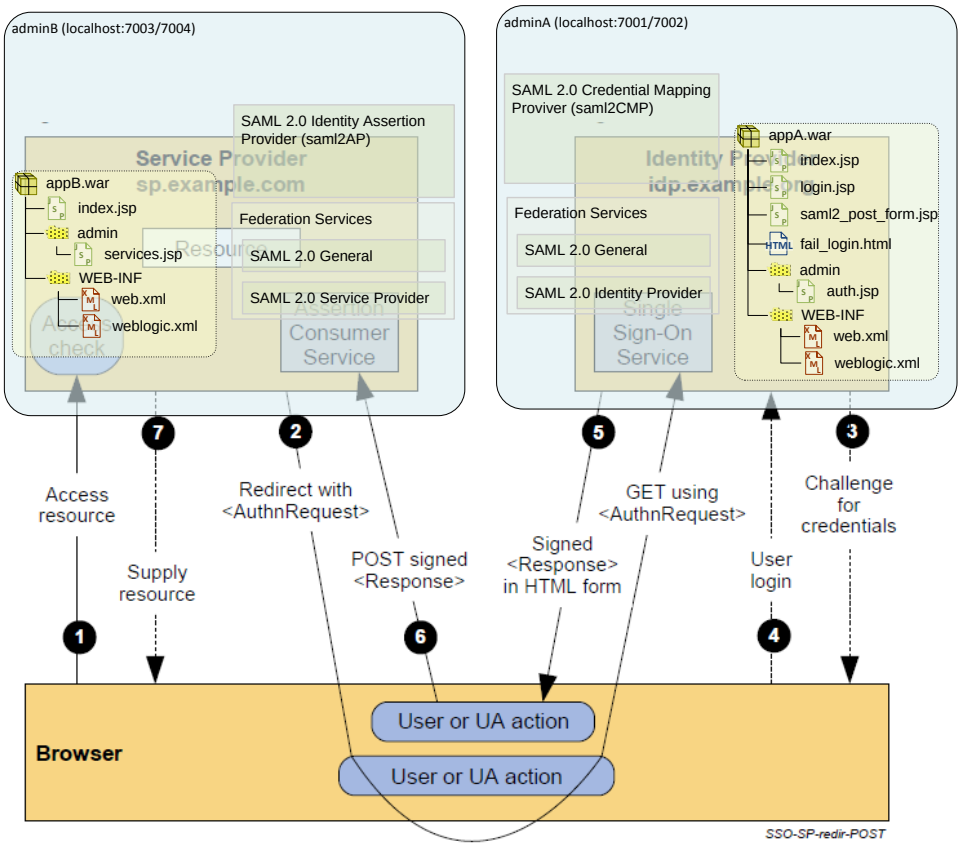


Figure 12: SP-Initiated SSO with Redirect and POST Bindings

Figure 5. OASIS Message Flow picture with overlaid tutorial components.

The Service Provider will be realized by the WLS server adminB, the saml2AP Identity Assertion Provider together with the Federation Services provide the Assertion Consumer Service. The resource, which is accessed, is provided by the services.jsp.
The Identity Provider will be configured as adminA. Here we use a SAML 2.0 Credential Mapping Provider together with the Federation Services to provide the Single Sign-On Service. The Login module will be provided by the login.jsp.

With a very similar picture, we present the IdP-Initiated SSO case.

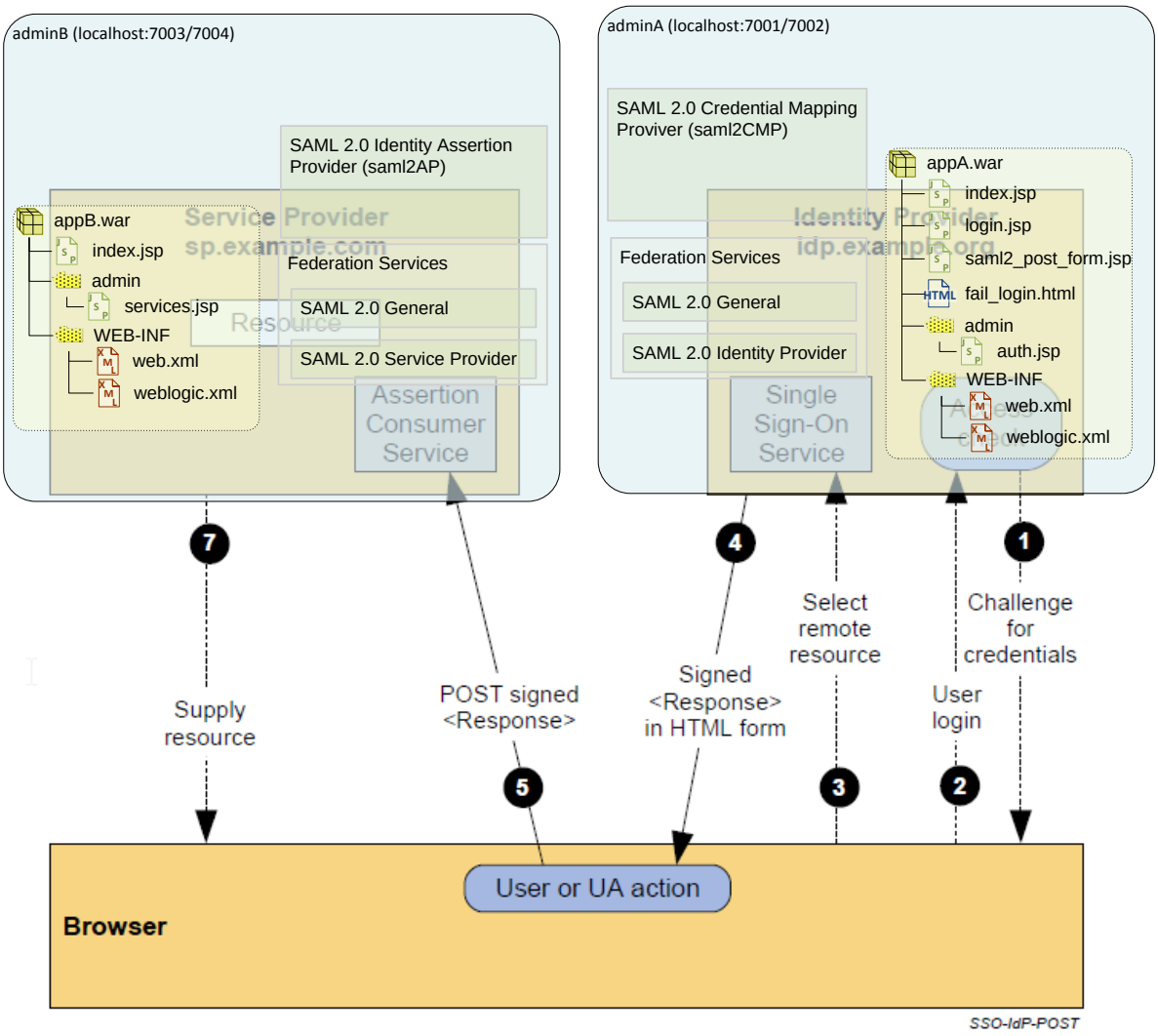


Figure 14: IdP-Initiated SSO with POST Binding

Figure 6. OASIS Message Flow picture with overlaid tutorial components for IdP-Initiated Message Flow.

The components are the same as in the previous picture; however the message flow is different. The page that offers the remote resource in step 3 is provided by admin/auth.jsp. The POST form for step 4 and 5 is provided by saml2_post_form.jsp.

We will build up two WLS domains, each consisting only of a single Admin Server. We will configure the Federation Services between these domains as depicted in the following overview.

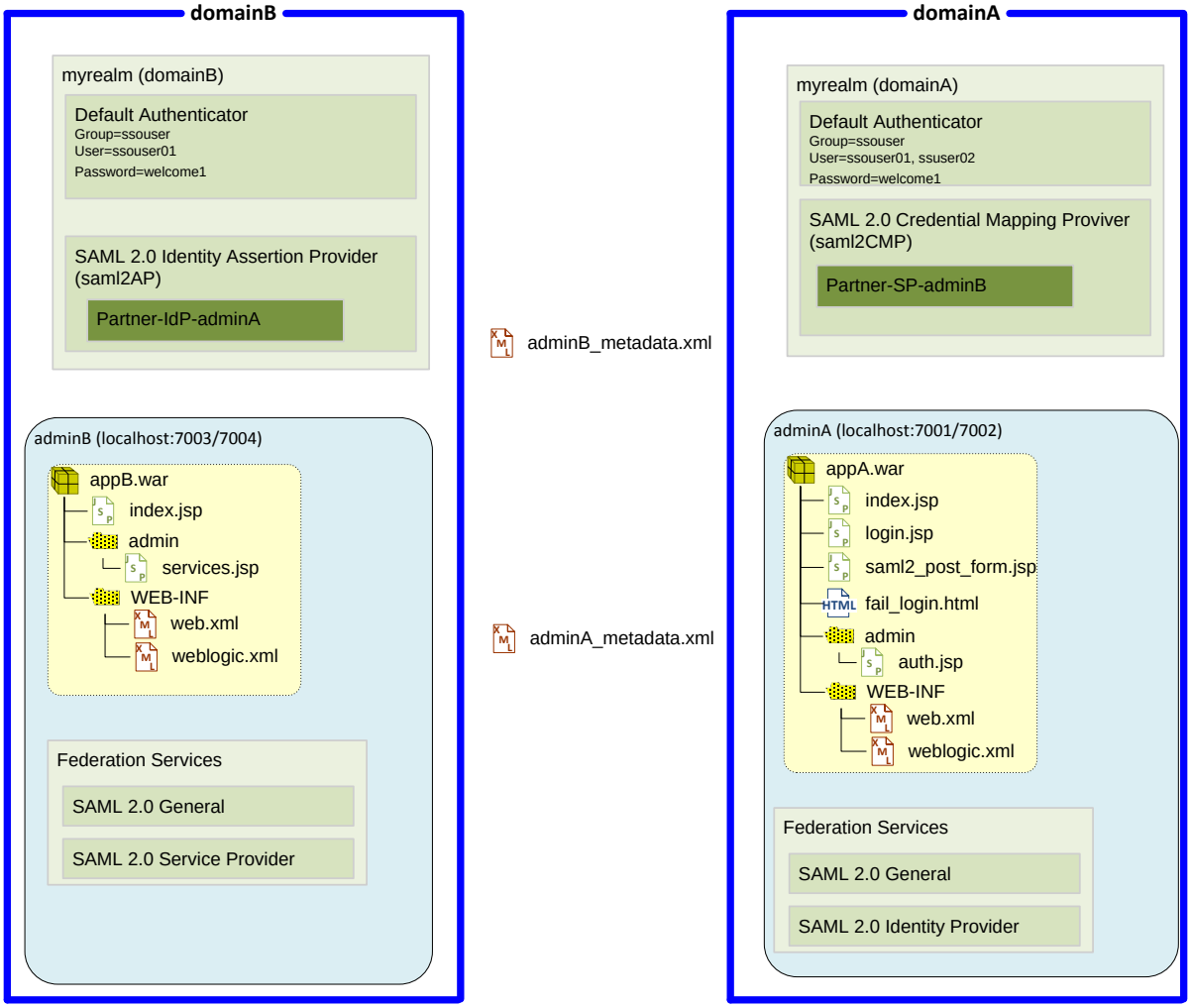


Figure 7. Domain configuration for this example with Federation Services.

We use adminB and adminA in domainB and domainA respectively. adminB will host appB which represents the service provided by the SAML Service Provider. adminA will host appA, which contains a login page and a service selection page. The security realms are also shown, together with the relevant users, groups and security providers.

The following diagram proposes a configuration sequence and depicts configuration details.

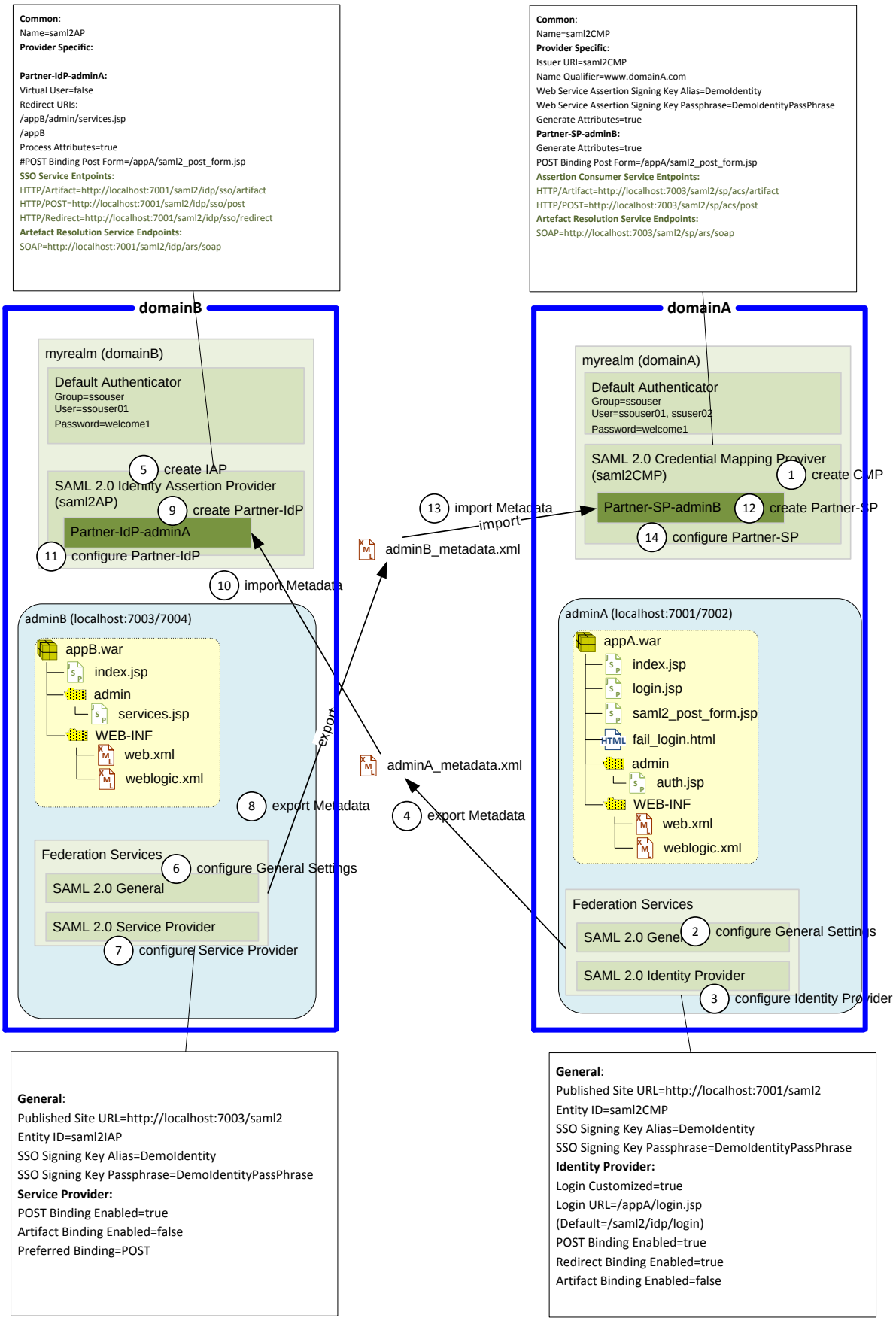


Figure 8. Proposed configuration sequence for the example.

The numbers in the circles propose a configuration sequence which is not mandatory but recommended to complete this task efficiently. The configuration steps are explained on more detail in the Chapter 3.6 and the section numbers map to this sequence.

The activities of the whole tutorial are split into four parts. We depict a summary in the following mind map.

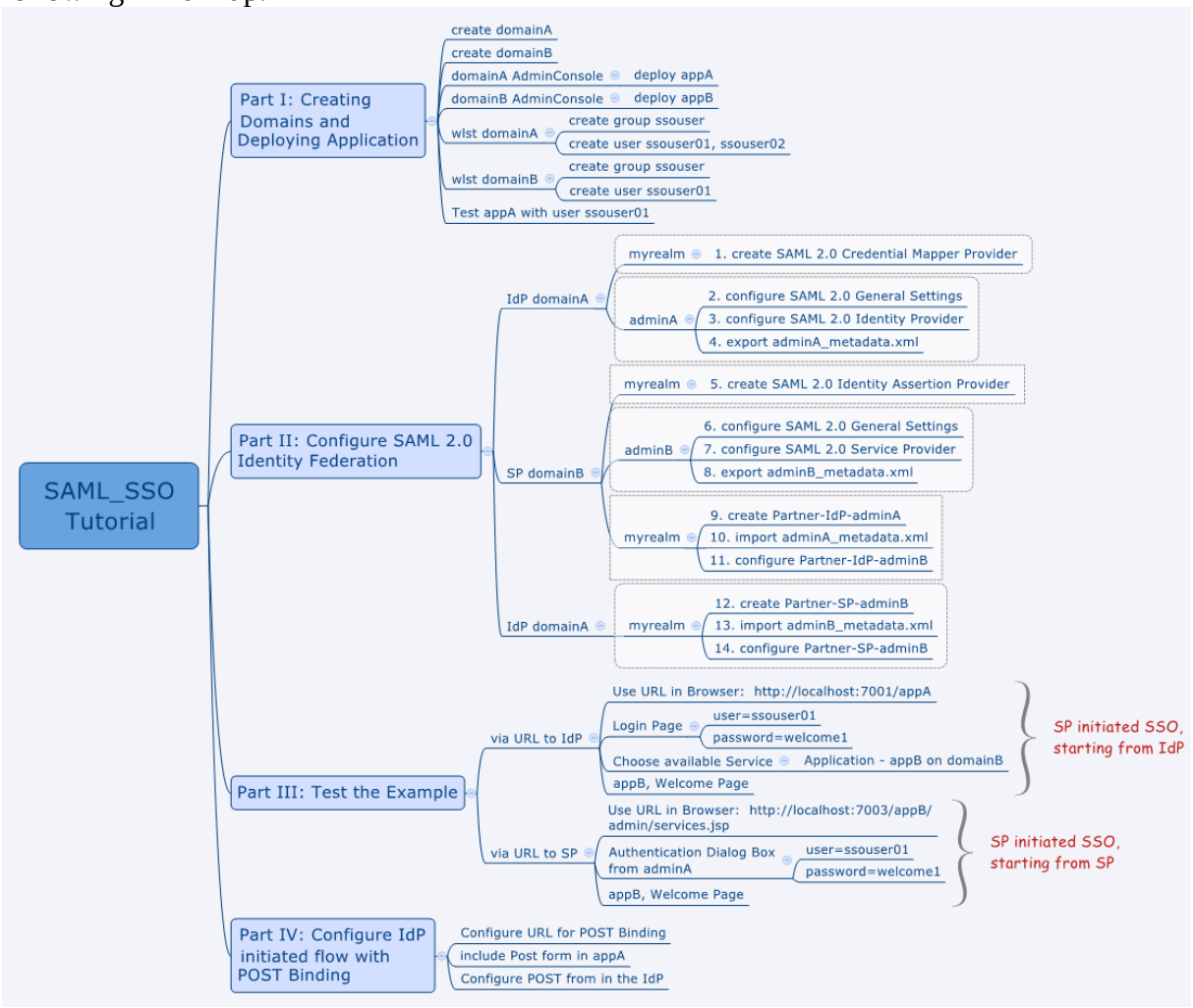


Figure 9. Mind map summarizing the parts of this tutorial.

3.6 Creating Domains and Deploying Applications

In this step we configure two domains and deploy the sample application.
The commands to setup the domains are contained in the file
${EXAMPLE_HOME}\SAML_SSO\SAML_SSO.TXT.
We provide an overview mind map of the configuration steps.

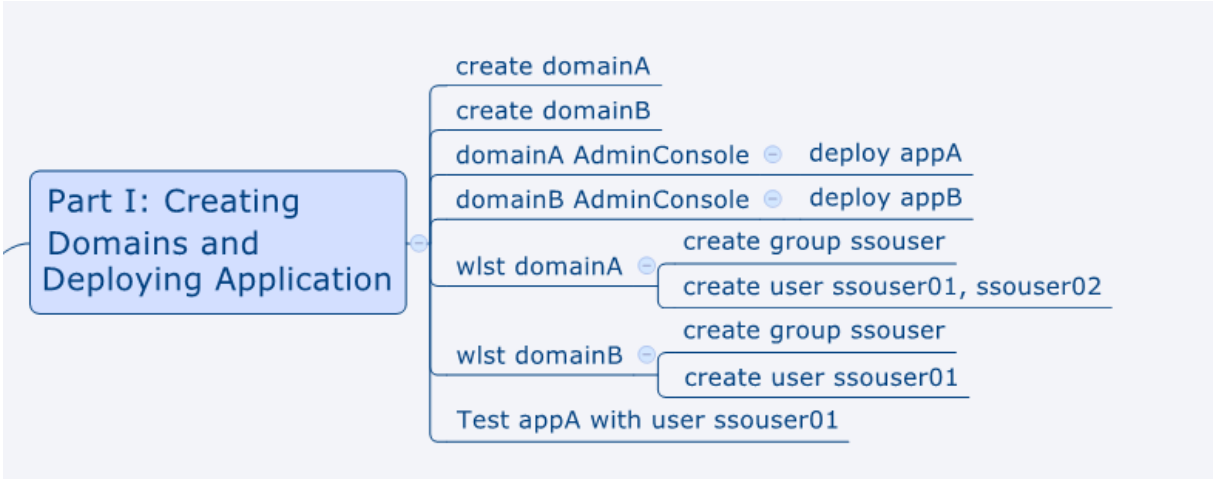


Figure 10. Overview Mind Map for setting up the example domains.

We deploy appA and appB using the admin console. We create the users and groups using the wlst commands in the file SAML_SSO.TXT.

Application Folder	${EXAMPLE_HOME}\SAML_SSO\SAML_SSO.TXT.
domainA	http://localhost:7001/console
appA	http://localhost:7001/appA
User/password	ssouser/welcome1
domainB	http://localhost:7003/console
appB	http://localhost:7003/appB
appB	http://localhost:7003/appB/admin/services.jsp

3.7 Configuring SAML

In following steps we want to configure this SAML example. Since it is easy to get lost during the manual configuration process, we provide an overview mind map.

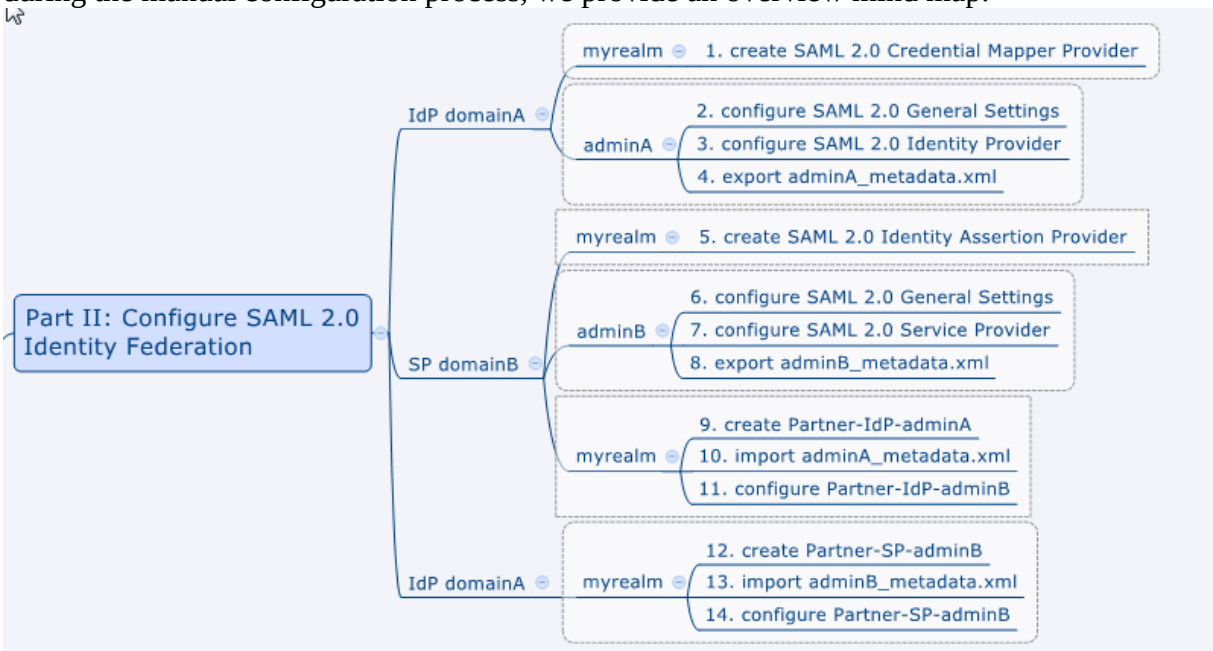


Figure 11. Mind Map Overview of the SAML Configuration Process.

The configuration begins in domainA, which will be configured as Identity Provider. We need to configure a Credential Mapping Provider in the security realm. In the server settings of adminA we need to configure the “Federation Services”. In order to conclude the configuration of domainA we need to import the metadata file of the Service Provider which will be produced during the SAML configuration in domainB. Therefore we continue with the configuration of domainB. After that, we change back to domainA and complete the configuration here.
The individual steps from the mind map are explained in detail in the following sub sections.

3.7.1 create SAML 2.0 Credential Mapper Provider

We start with the IdP in domain.
We enable the SSL Port. We use the Demo Certificates
Configure a new credential mapping provider.
(Security Realms->myrealm->Providers->new->SAML 2.0 Credential Mapping Provider)

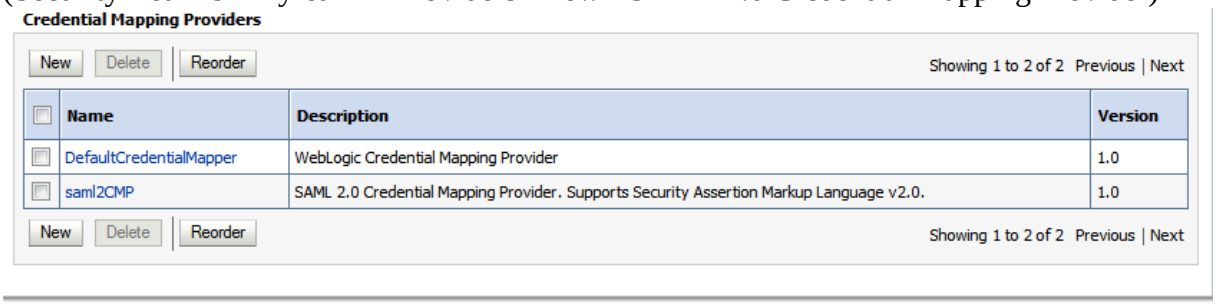


Figure 12. Creating a new SAML 2.0 Credential Mapping Provider for domain.

We configure the newly created provider.

For signing we use the DemoIdentity/DemoIdentityPassPhrase.

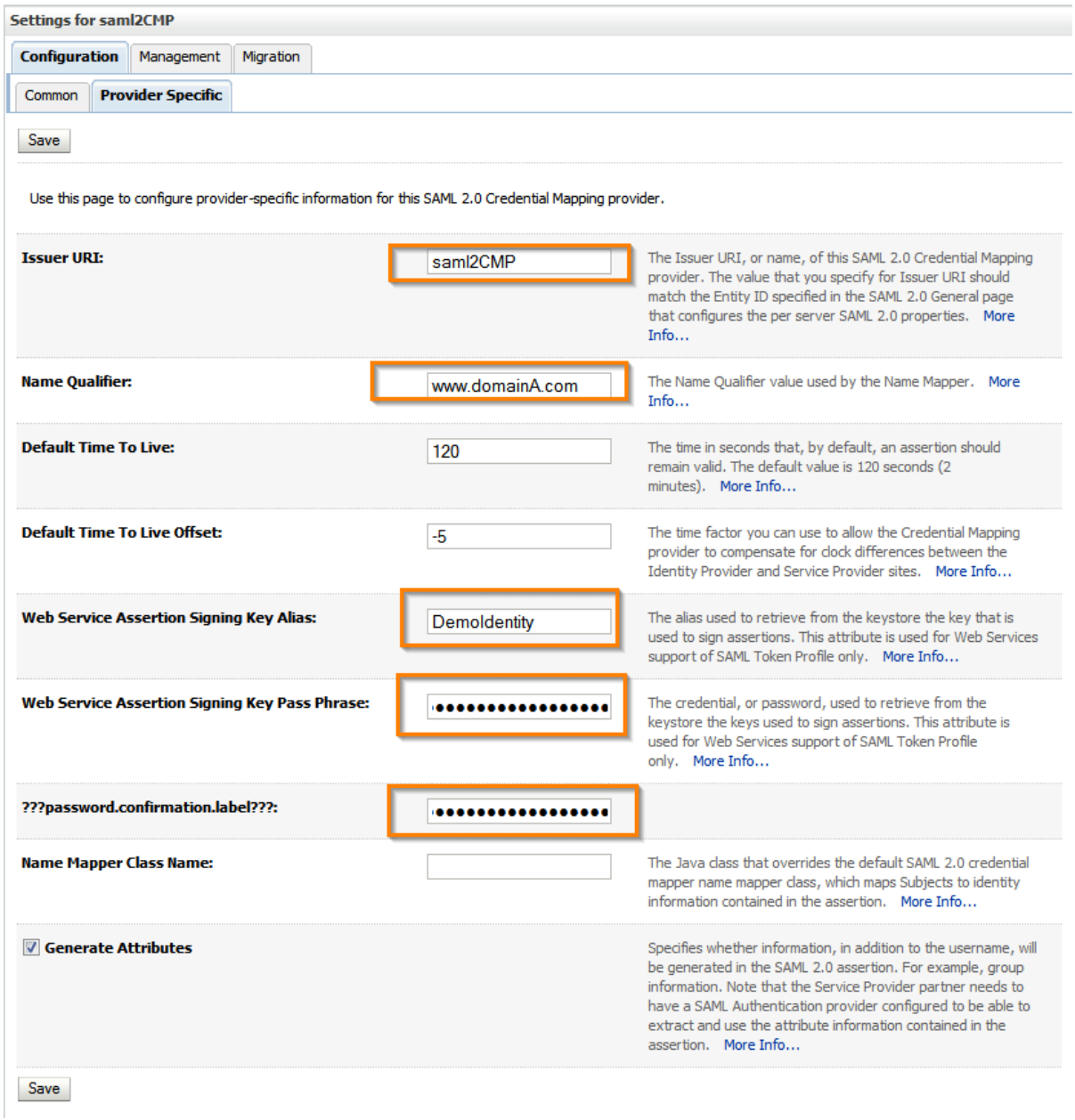


Figure 13. Provider specific configuration of the SAML Credential Mapper.

We need to restart the Admin Server.

3.7.2 configure SAML 2.0 General Settings

Now we create the SAML Metadata or the server specific SAML 2 profile.

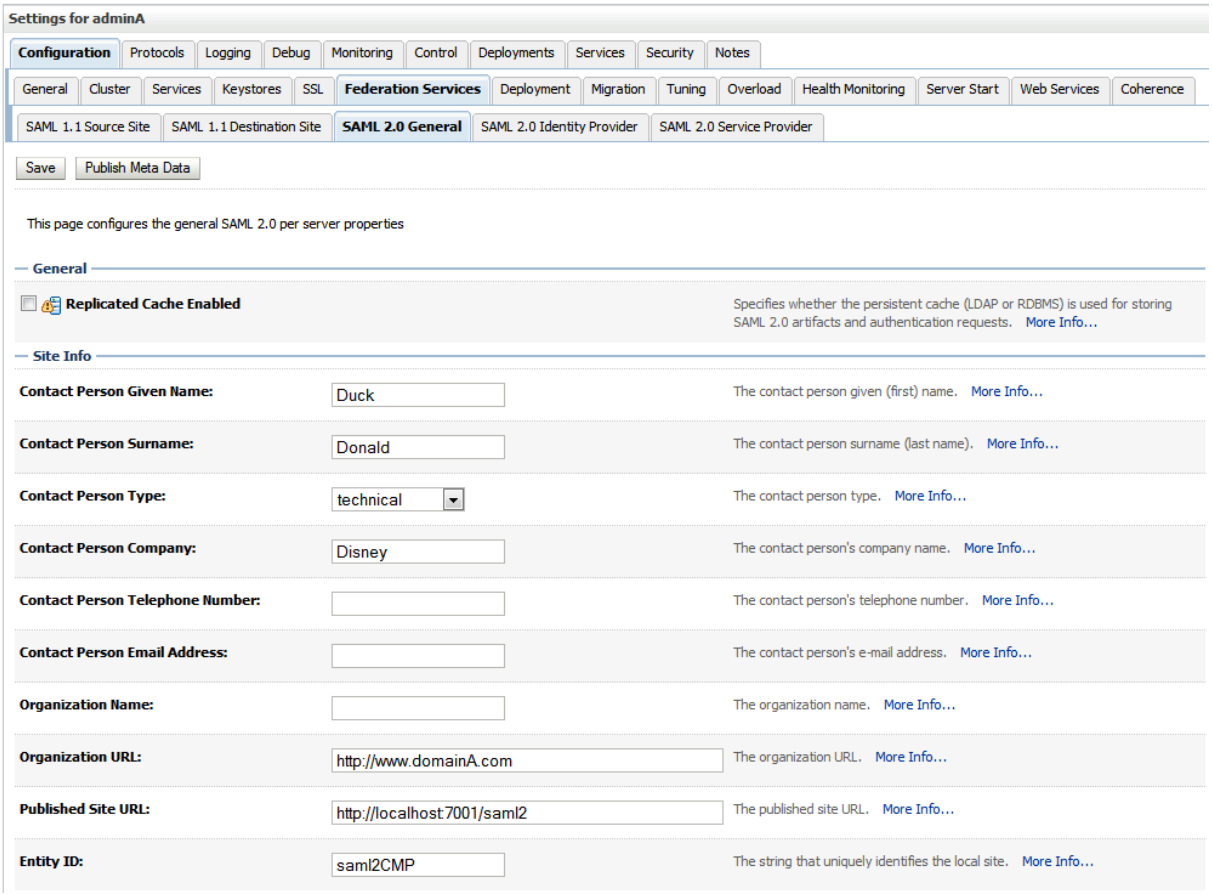


Figure 14. SAML 2.0 configuration of general per server settings.

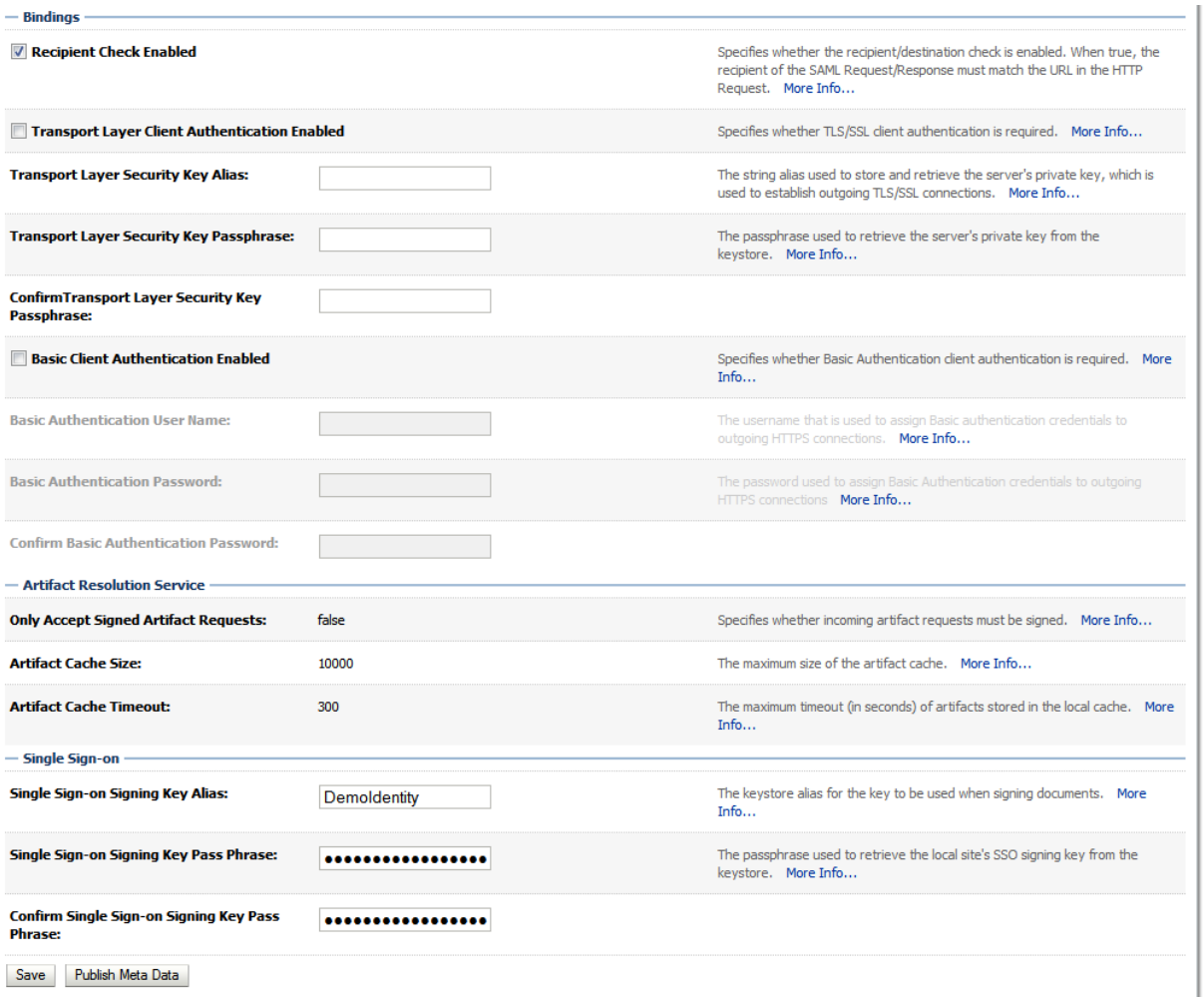


Figure 15. SAML 2.0 configuration of general per server settings. (Part 2)

3.7.3 configure SAML 2.0 Identity Provider

Now we change to the “SAML 2.0 Identity Provider” tab and configure the IdP. Settings of this tab will also go into the xml file containing the metadata.
We need to choose the preferred binding “Redirect”. Otherwise the Artifact Binding will be chosen from WLS.
The preferred binding will be used by the SP in domainB when sending the authentication request to the IdP (Step 2 of the SP-Initiated Flow). This information is transferred to domainB, when the metadata file is exchanged, i.e. when the metadata_adminA.xml is imported to the “Partner-IdP-adminA”.

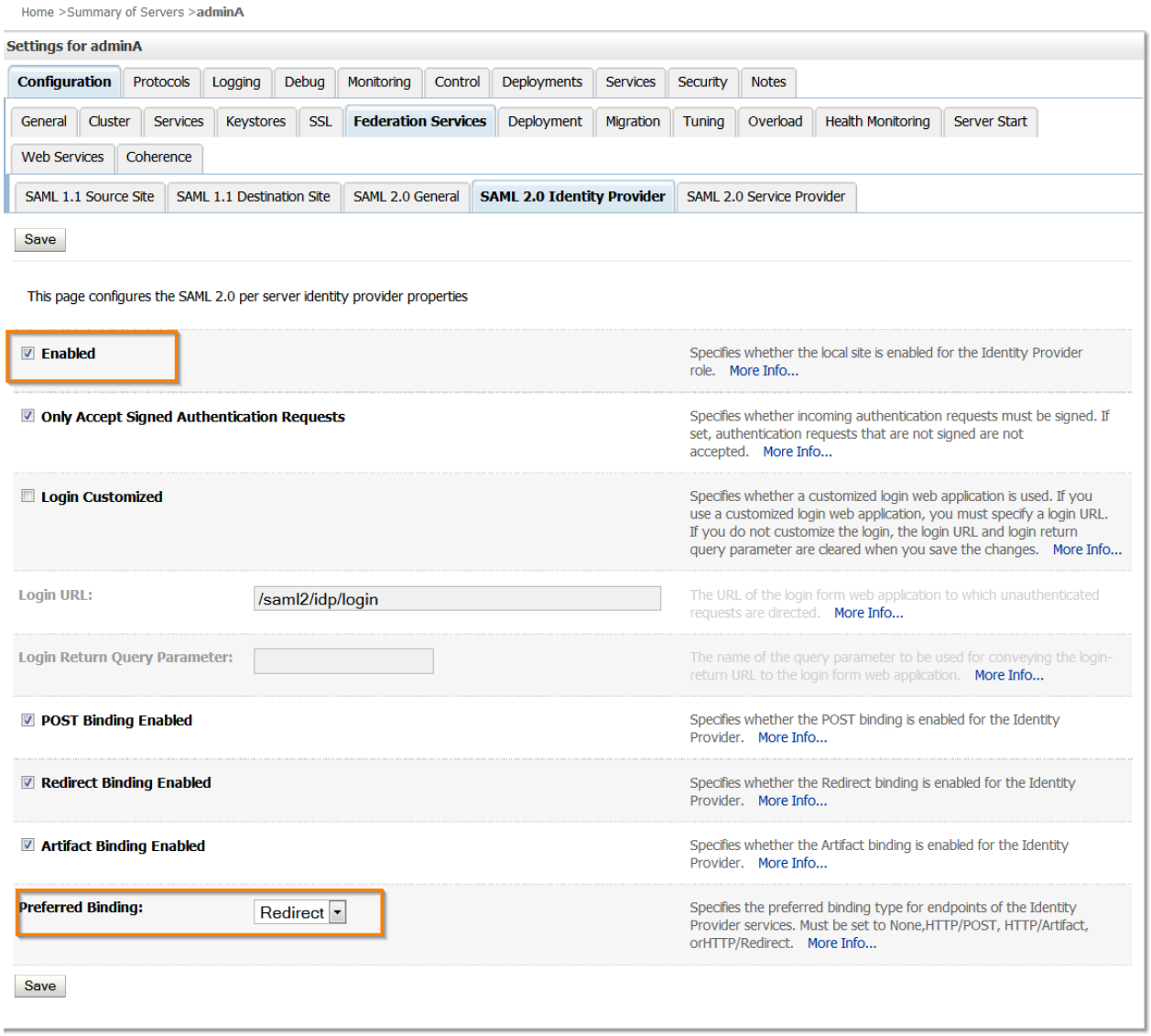


Figure 16. IdP configuration.

3.7.4 export adminA_metadata.xml

We change back to the “SAML 2.0 General” tab and publish the Metadata to the XML file: D:\10Oracle\06WLS12\domains\domainA\admin_metadata.xml

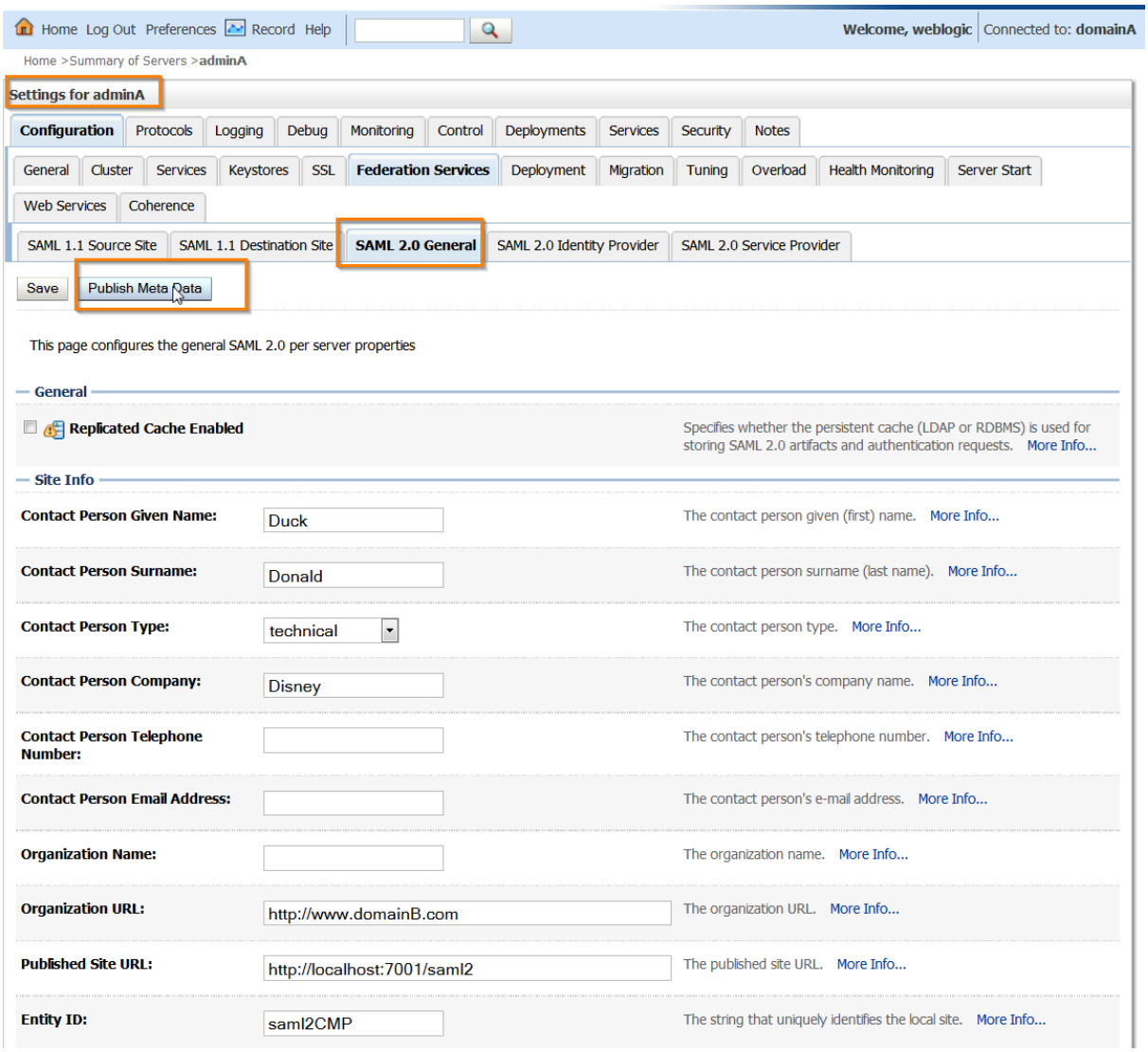


Figure 17. Publishing the Metadata File of the IdP.

3.7.5 create SAML 2.0 Identity Assertion Provider

We create a new SAML2 Authentication Provider in the security realm “myrealm”.

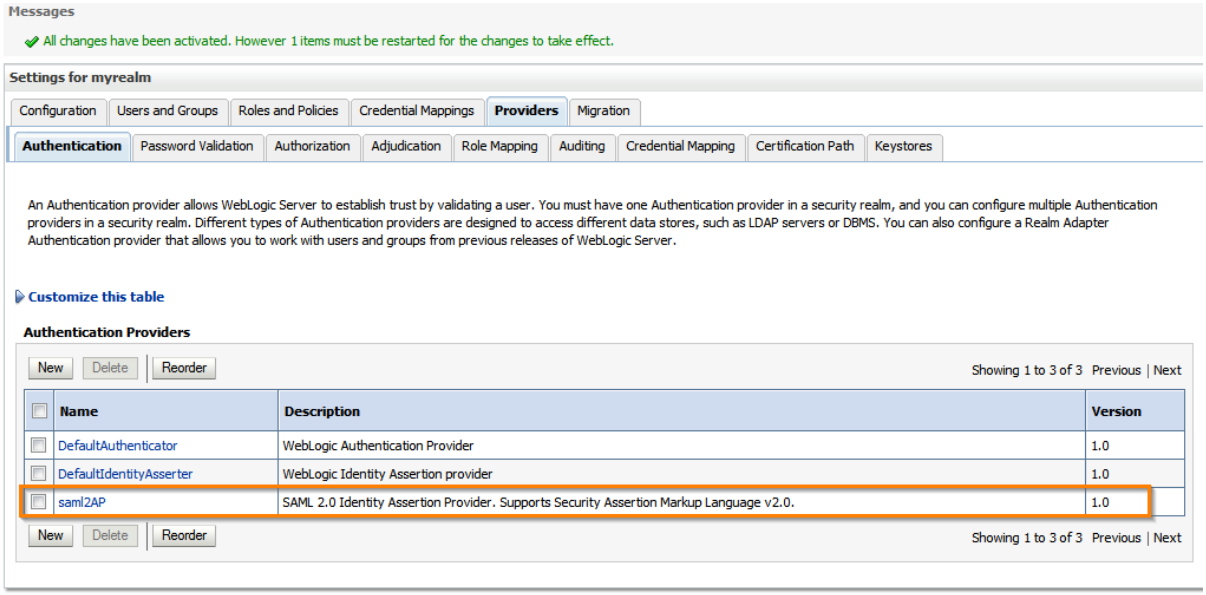
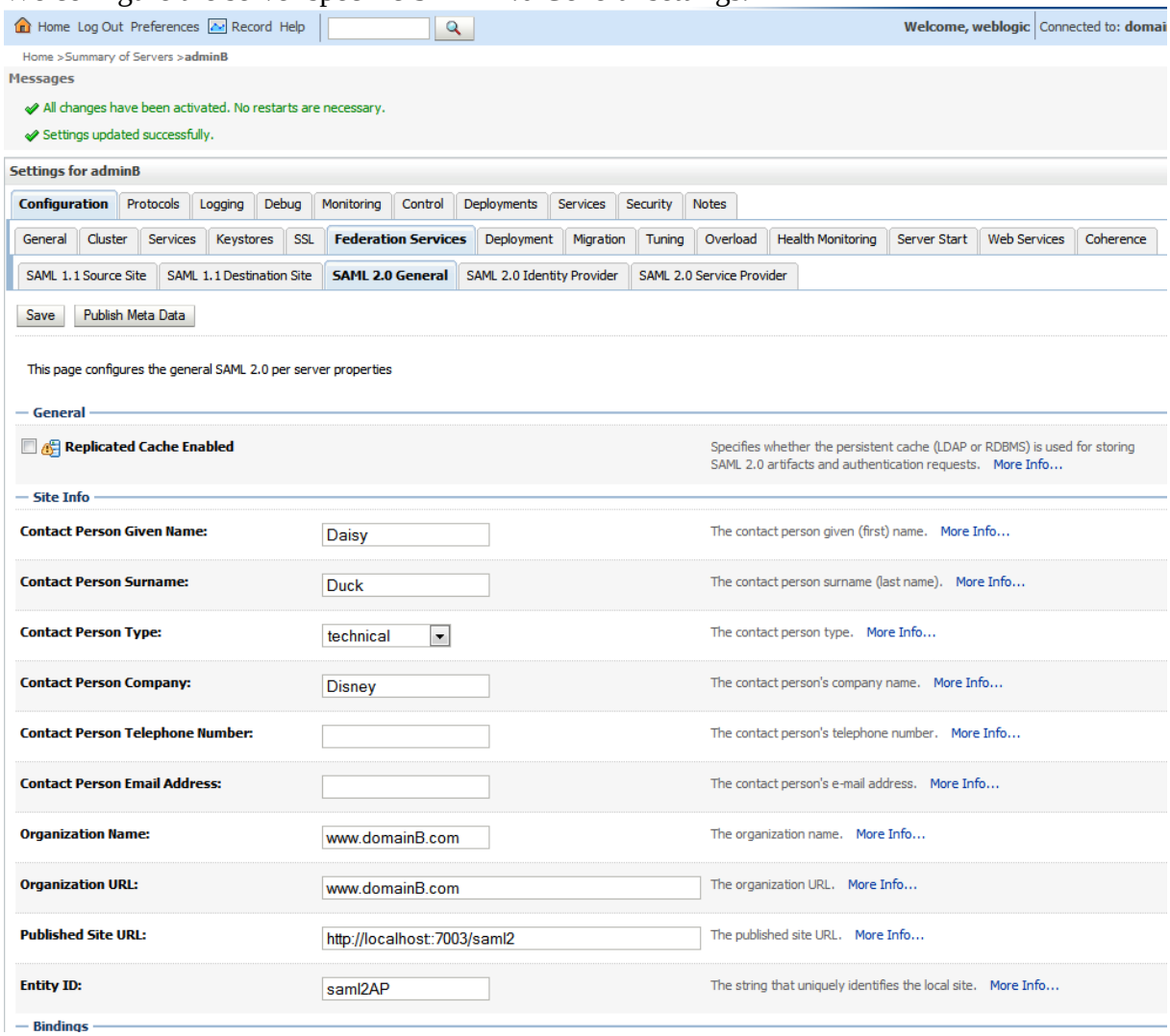


Figure 18. Creating a SAML 2.0 Authentication Provider in domainB.

We restart the server.

3.7.6 configure SAML 2.0 General Settings

We configure the server specific SAML2.0 General settings.



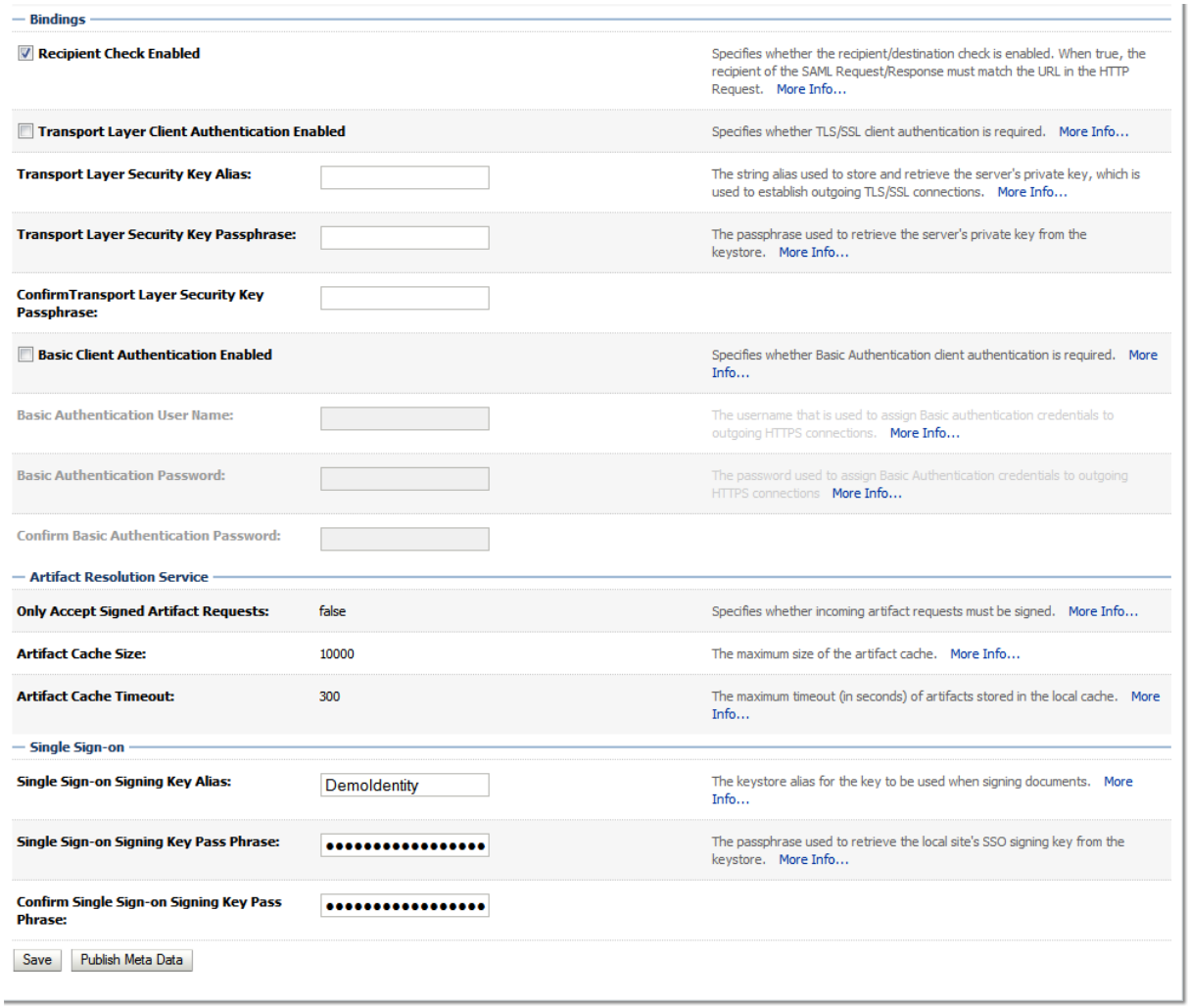


Figure 19. Configuration of the SAML2 General Settings in server adminB.

For signing we use the DemoIdentity/DemoIdentityPassPhrase.

3.7.7 configure SAML 2.0 Service Provider

We change the „SAML 2.0 Service Provider“ Page of adminB. We choose “POST” as preferred Binding. This will influence how the SingleSignOn Service in domainA, or more specific the “Partner-SP-adminB” in the “SAML 2.0 Credential Mapping Provider”, communicates the SAML Assertion to the Service Provider.
There are two options. If we choose POST, the Assertion will be place into an HTML Form and send via POST to the Assertion Consumer Service (samlAP) of domainB.
If we don’t choose anything or choose “Artifact” the IdP will sent a signed artifact via HTTP redirect.
These values will be comunicated to the domainA during import of the “adminB_metadata.xml” file.

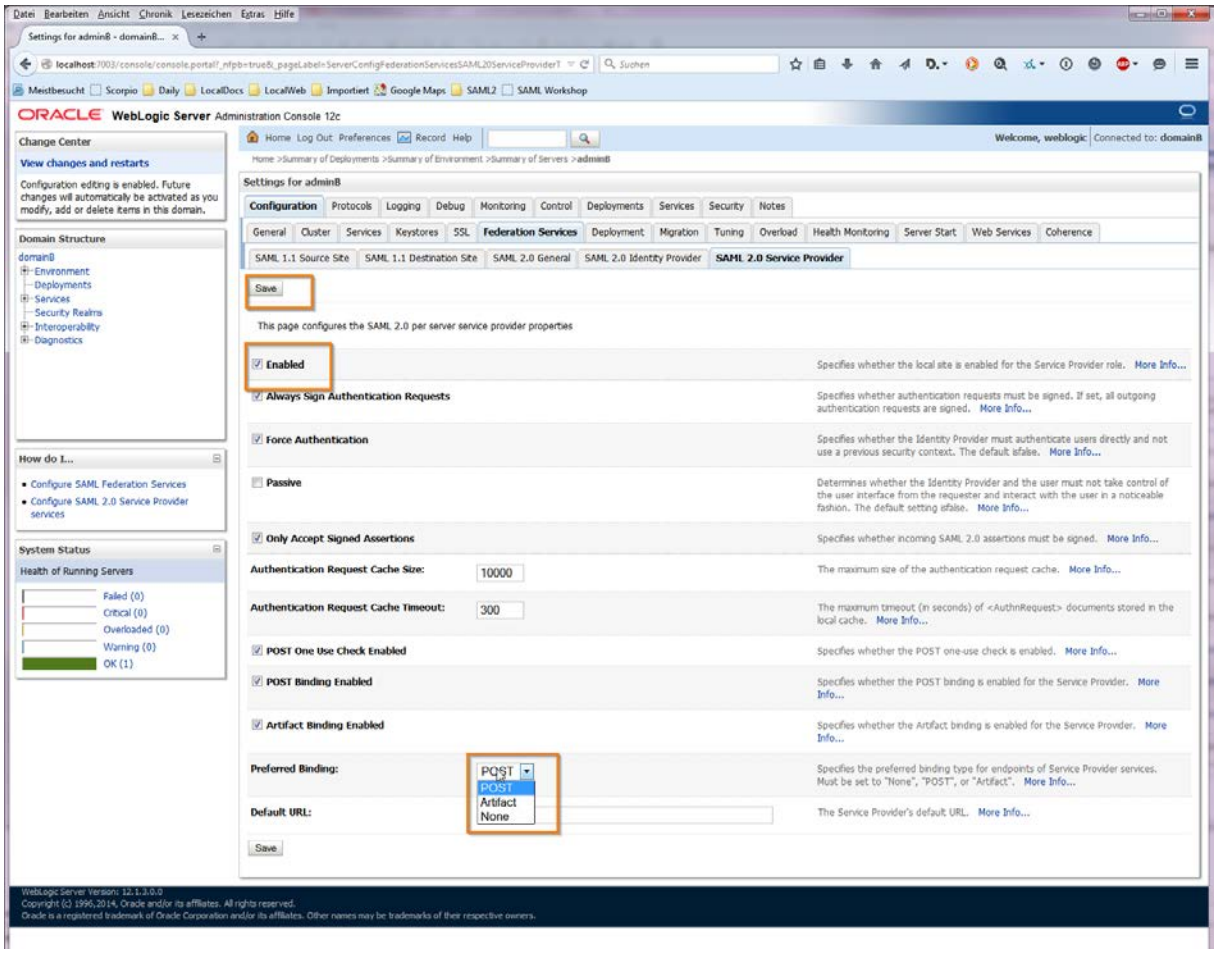


Figure 20. Configuration of the SAML 2.0 Service Provider Settings in adminB.

3.7.8 export adminB_metadata.xml

We change back to the “SAML 2.0 General” Tab and publish the metadata to the file: D:\10Oracle\06WLS12\domains\domainB\adminB_metadata.xml.

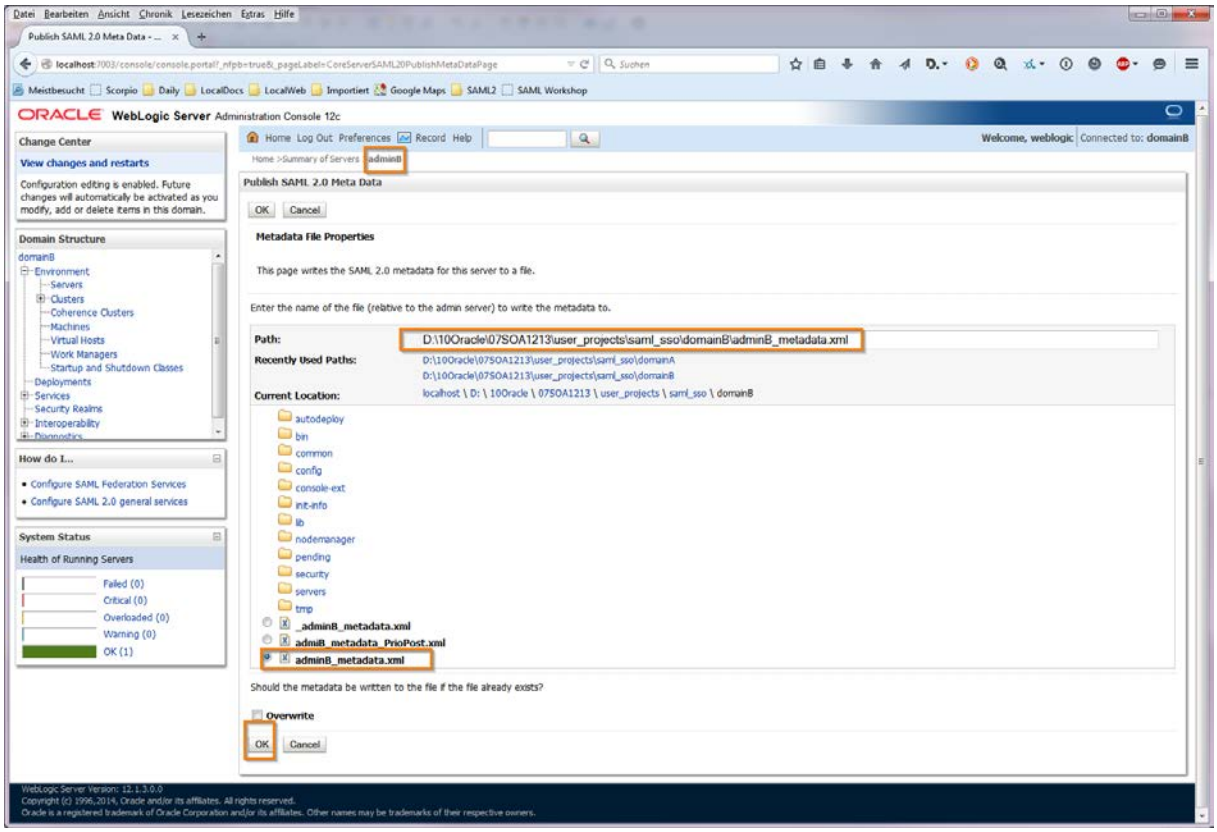


Figure 21. Publishing Metadata for SAML 2.0 Federation Services of adminB

3.7.9 create Partner-Idp-adminA

Now we create a new SSO Identity Provider Partner in the security realm of domainB.

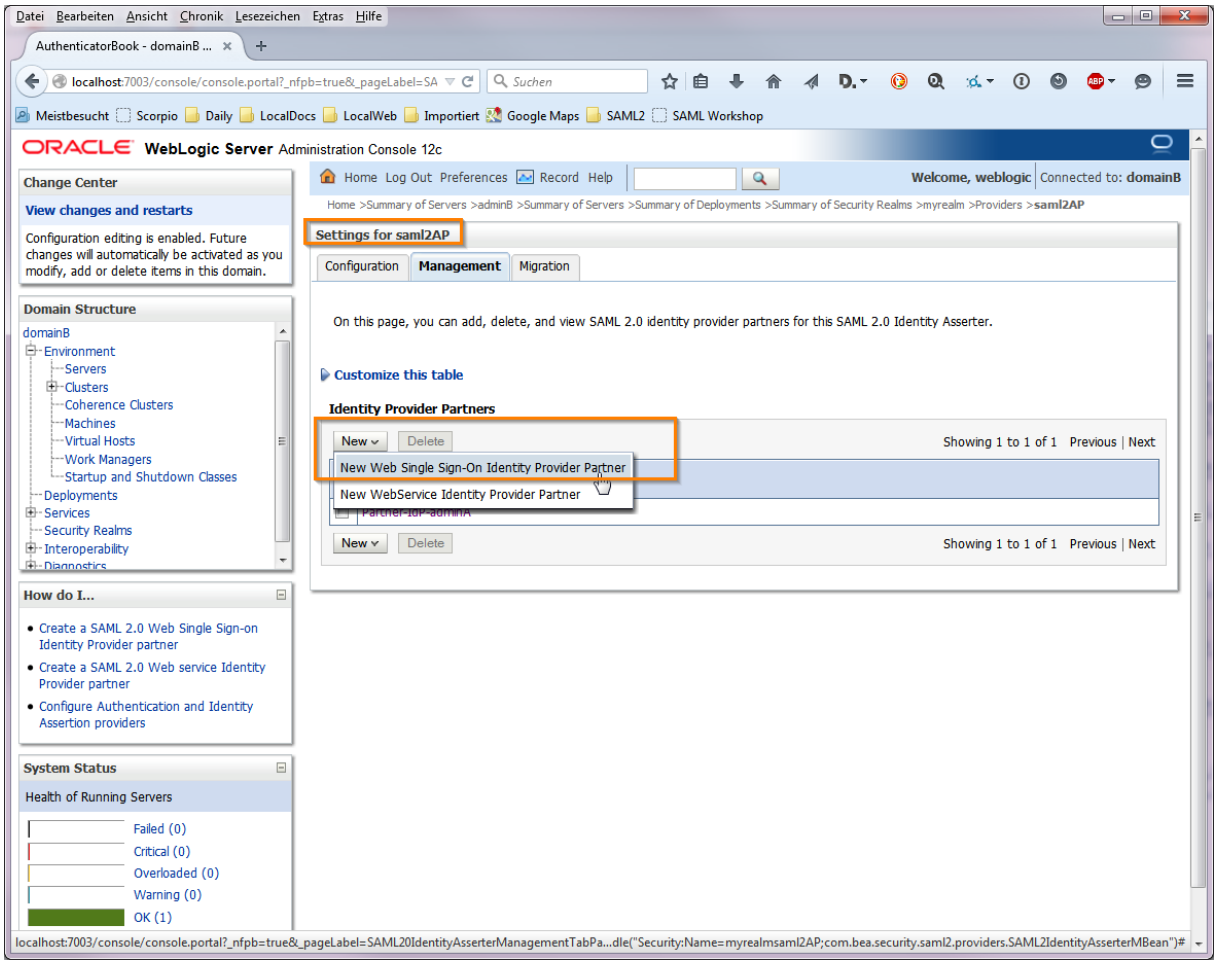


Figure 22. Creating a new SSO IdP Partner Configuration at the "SAML 2.0 Authentication Provider" in domainB

This

3.7.10 import adminA_metadata.xml

We import the metadata file from domainA, which is the IdP Partner.

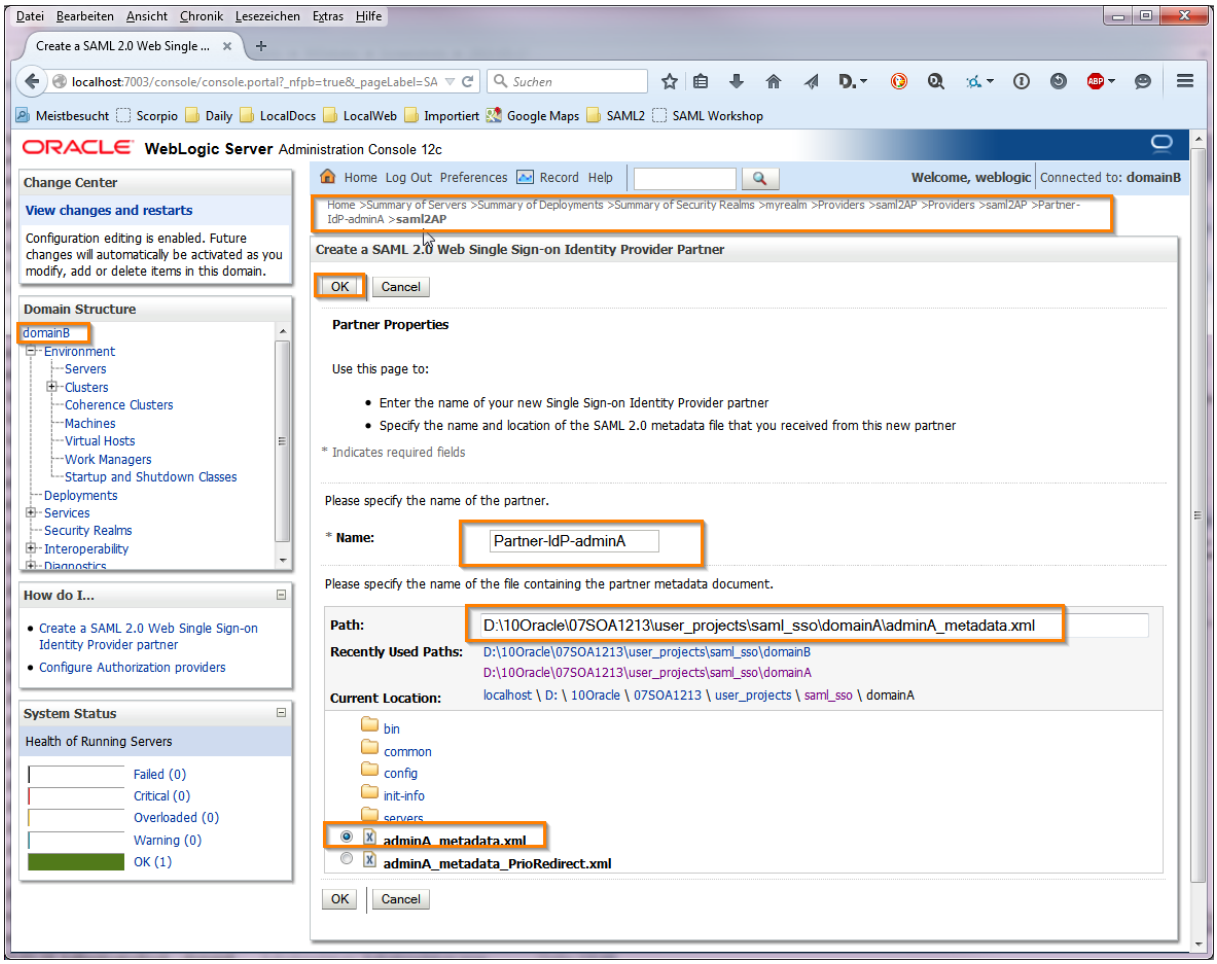


Figure 23. Creating the IsP Partner for the Service Provider domainB.

3.7.11 configure Partner-IdP-adminB

And we enable the newly created Partner site and add redirect URIs for Service Provider initiated SSO.
Redirect URIs:
/appB/admin/services.jsp
/appB

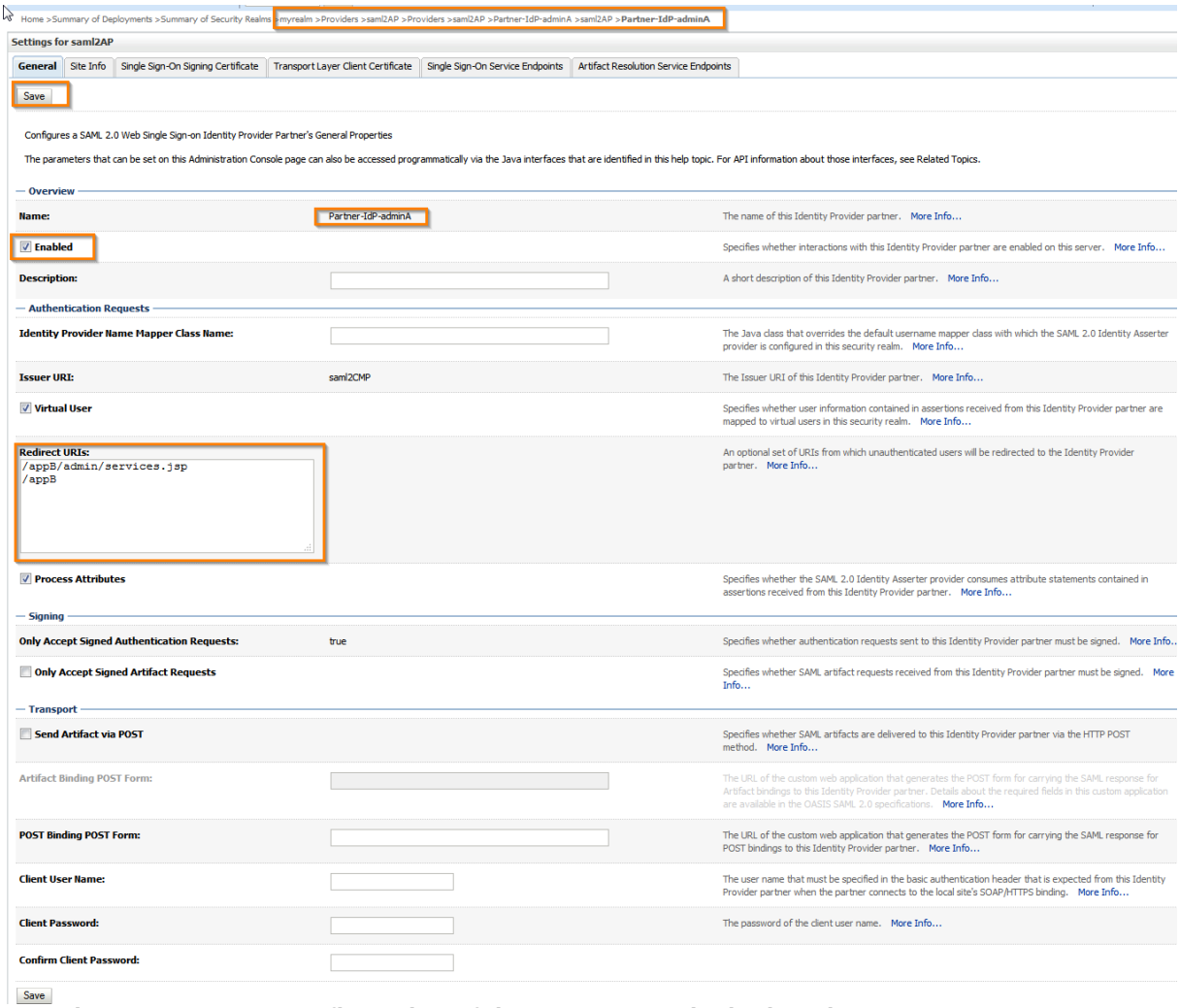


Figure 24. Configuration of the IsP Partner site in domainB.

This concludes the configuration in domainB.

3.7.12 create Partner-SP-adminB

We change to the Admin Server of domain to the “SAML 2.0 Credential Mapping Provider”. We want to create a Service Provider partner.

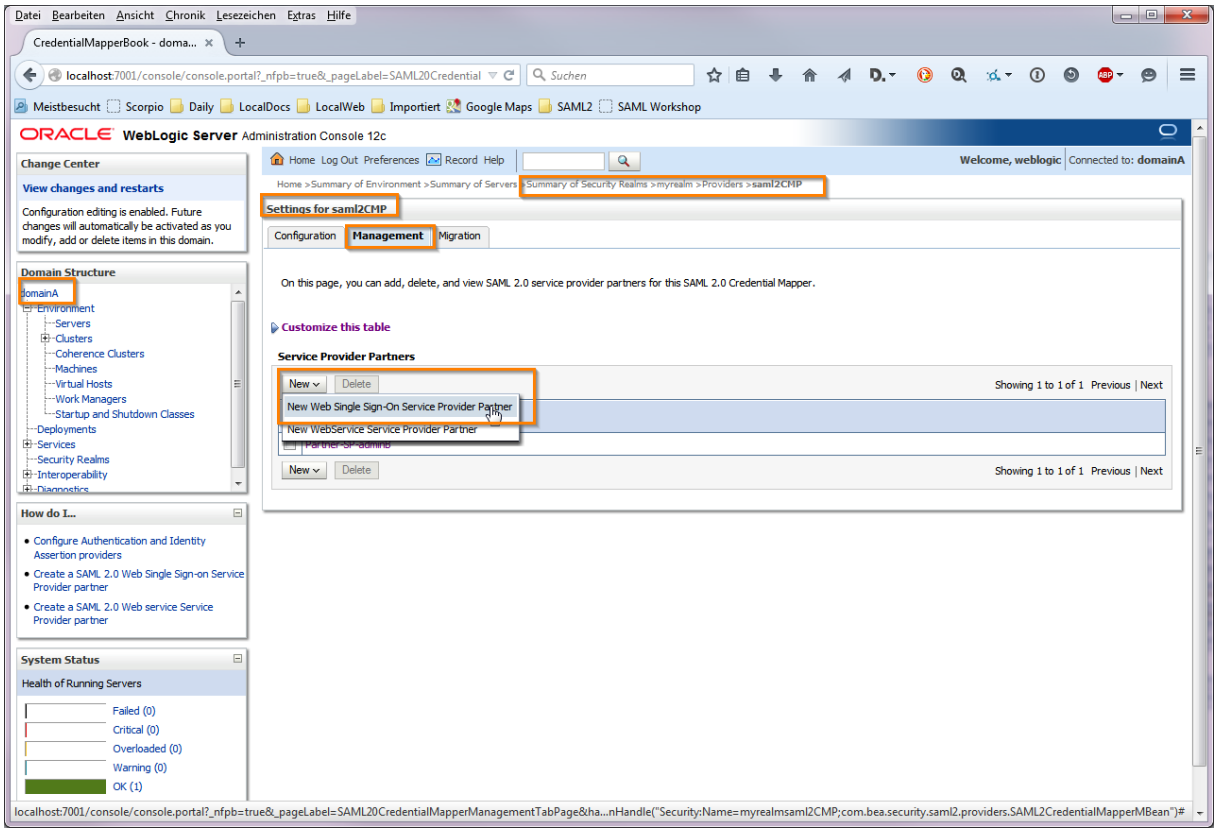


Figure 25. Creating a new Service Provider Partner in domainA.

3.7.13 Import adminB_metadata.xml

We import the metadata file from domainB.

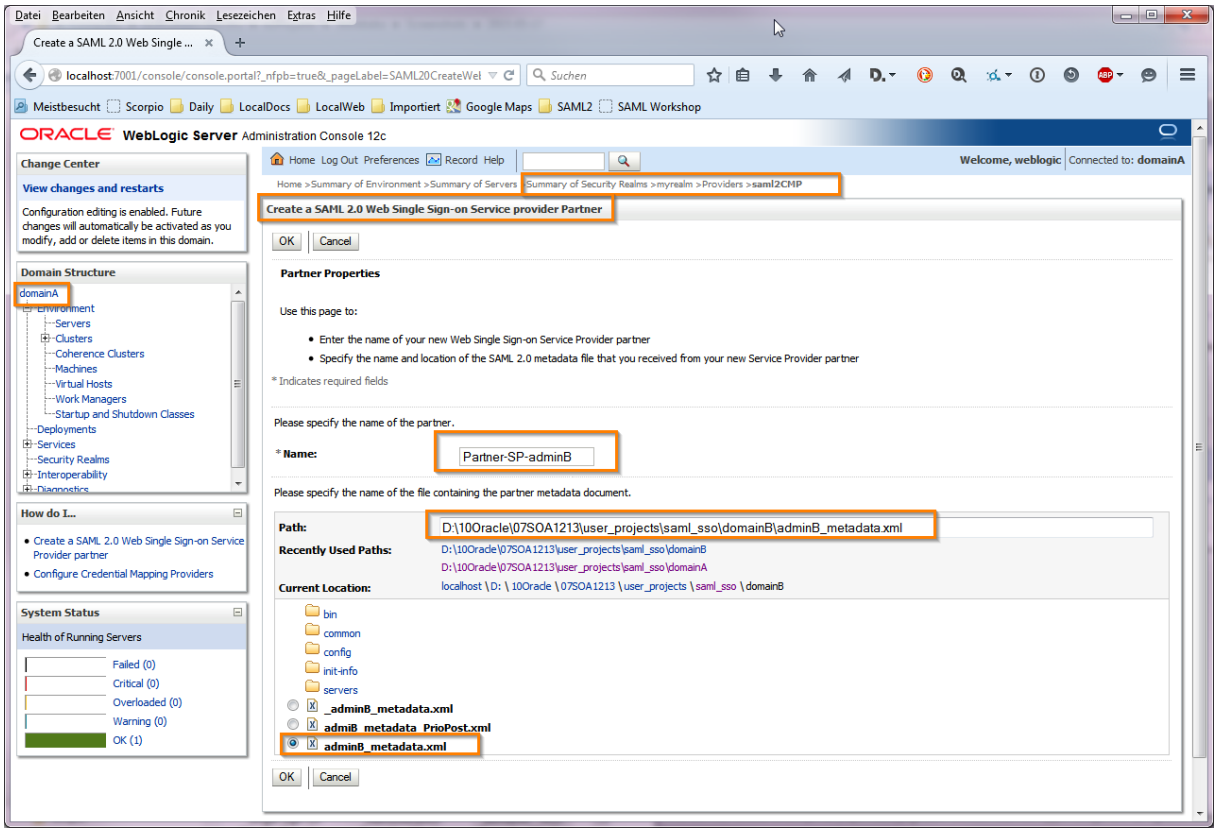


Figure 26. Importing the metadata from domainB.

3.7.14 configure Partner-SP-adminB

And we enable domainB as SAML partner service provider.

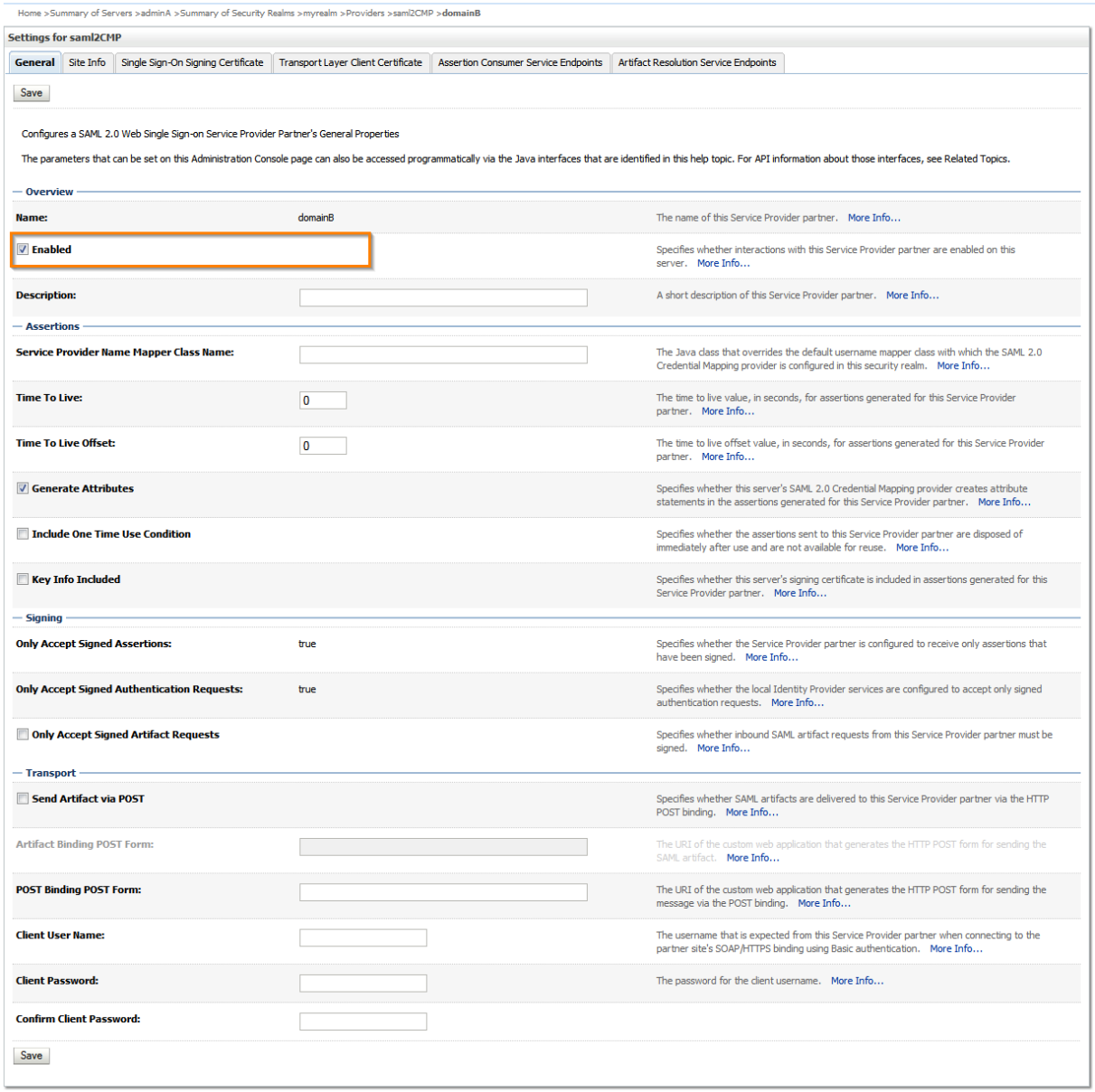


Figure 27. Enable the Service Provider Partner Configuration from the Metadata import.

That's all for the basic configuration of this SAML example.

3.8 Testing the example

We want to test the example via two URLs as indicated in the following mind map.

Figure 28. Testing SP initiated SSO

3.8.1 Testing via URL to IdP

First we go to appA and get redirected to the login page. We provide the credentials.

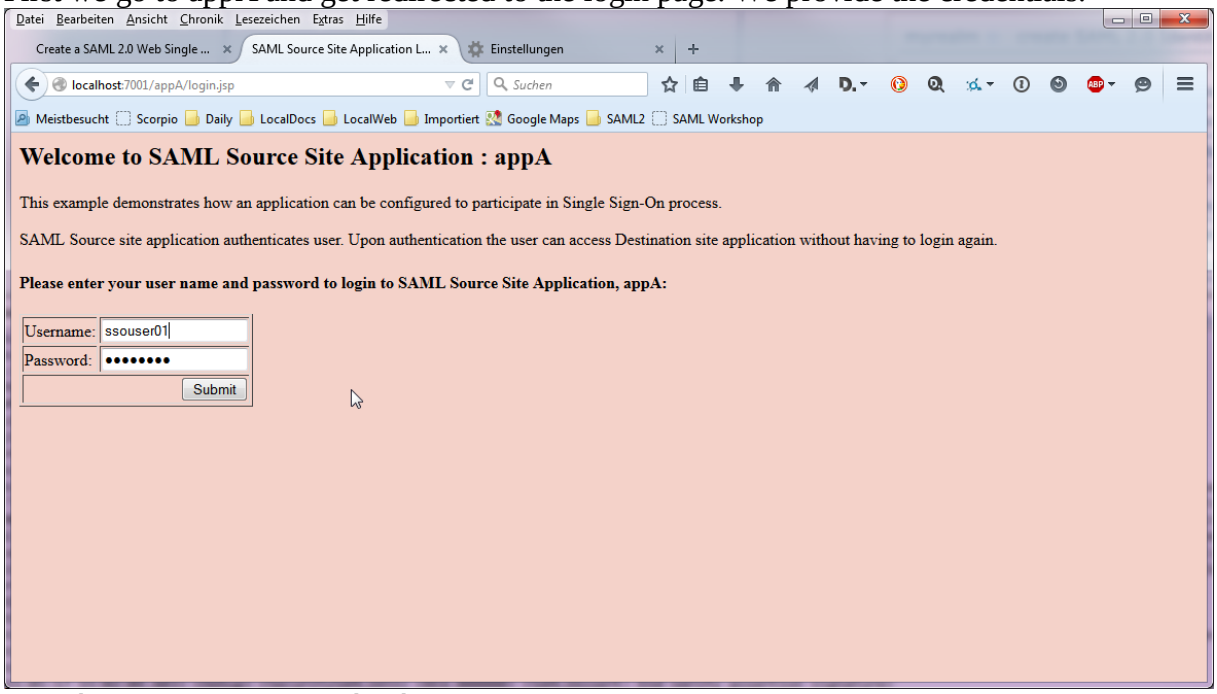


Figure 29. Logging in to appA.

We are logged in. Next we choose a service on appB.

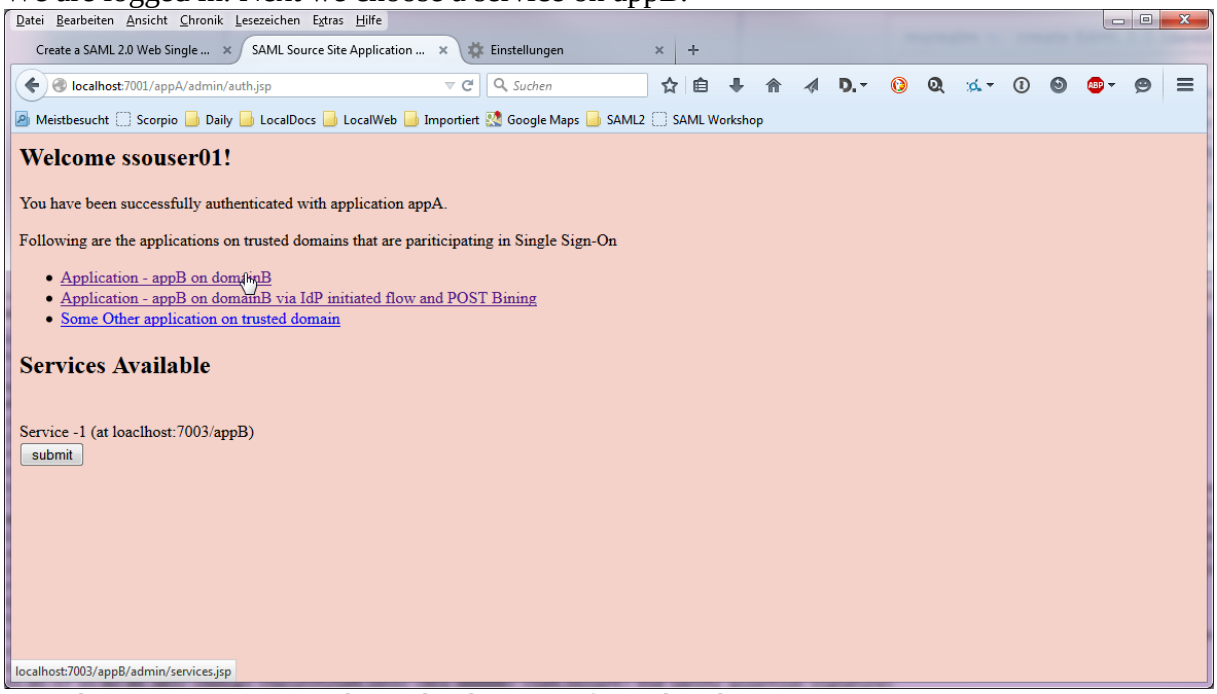


Figure 30. Service Selection Page from the IdP.

We get redirected to appB and are already logged in.

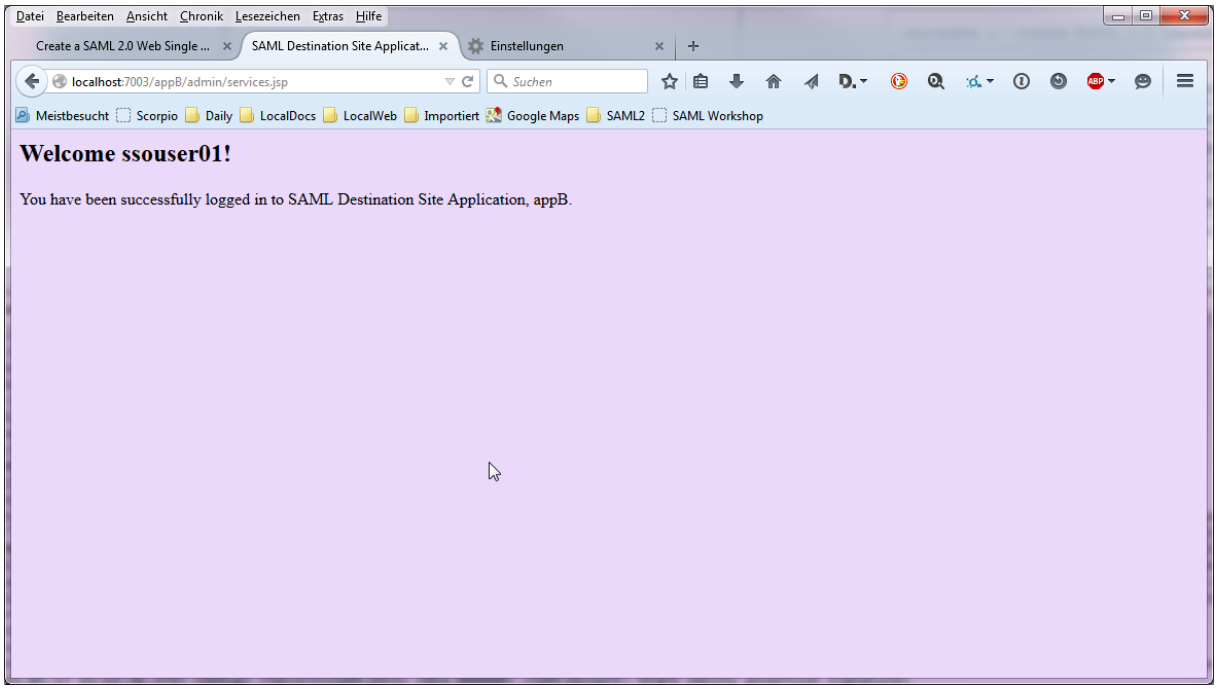


Figure 31. Service Page of appB.

While this looks like an IdP initiated flow, it is actually a SP initiated flow, however starting on the IdP. If we analyze the log files, we will discover that there is no SAML Assertion attached to the HTTP Request for appB. Instead the Assertion Consumer Service from domainB intercepts the call, builds a SAML AuthnRequest, sends it to domainA and receives the SAML Assertion in turn. It verifies the SAML Assertion and forwards to the appB Service page.

3.8.2 Testing via URL to SP

Now we want to go to the appB directly with an unauthenticated request.
In Firefox we delete the history and all cookies first.

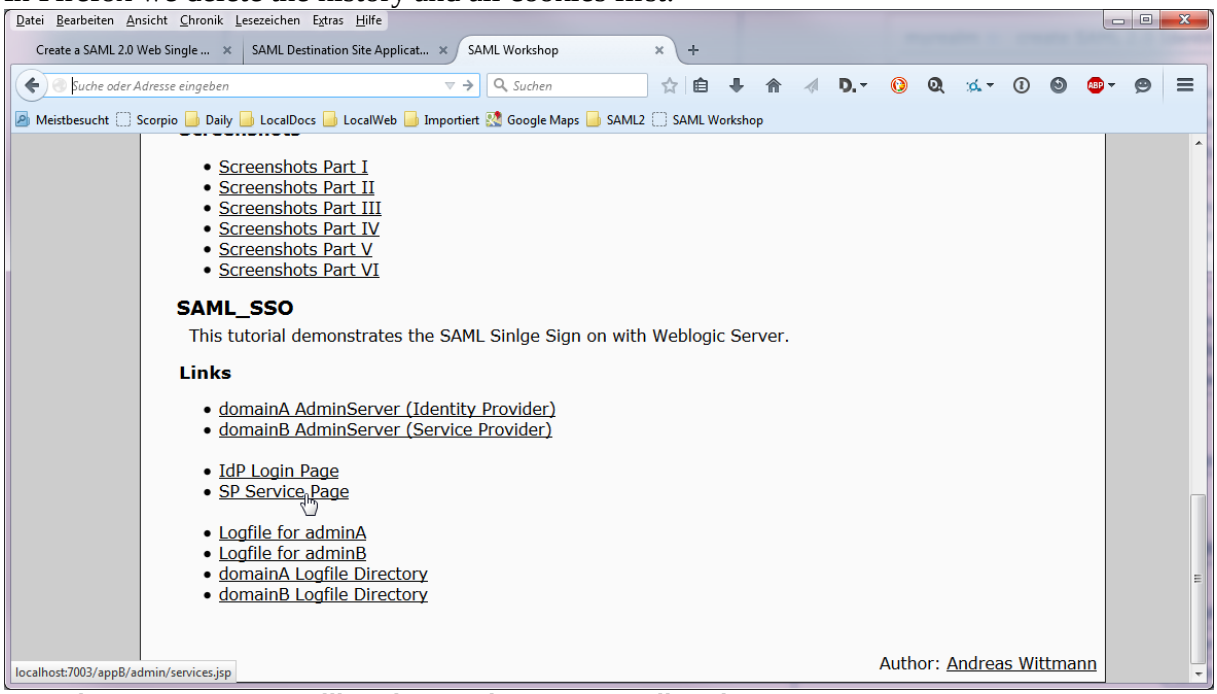


Figure 32. Calling the Service on appB directly.

We are challenged with a Login dialog from adminA.

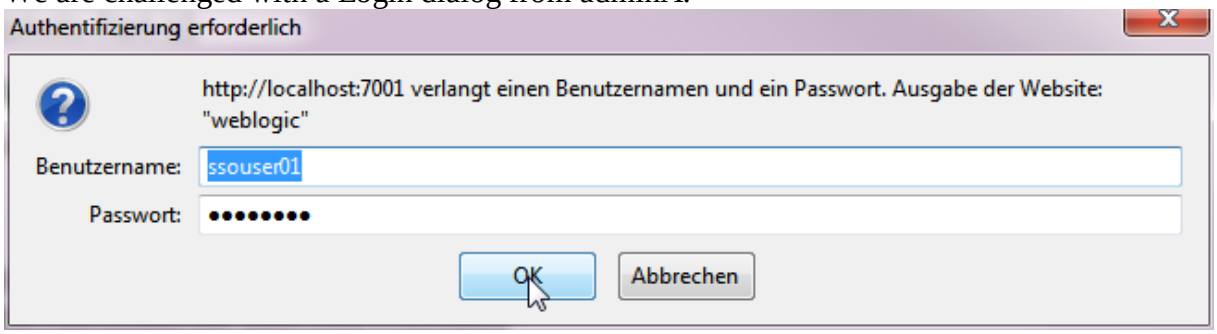


Figure 33. Login Dialog from adminA

After providing user and password we are redirected to the service page of appB and are already logged in.

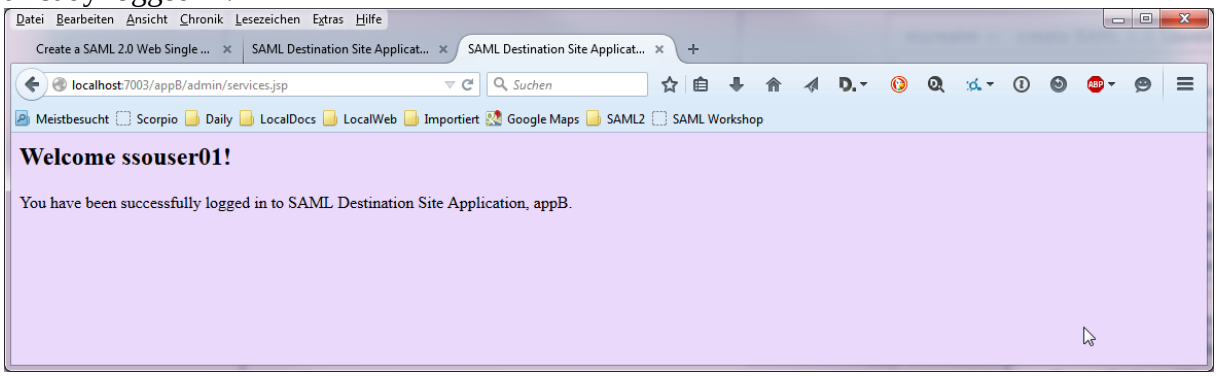


Figure 34. Service page of appB in the SP initiated flow.

Further configuration is needed to specify the login.jsp as login page instead of the standard Authentication Dialog box. We leave this exercise to the interested readers.

3.9 Setting Debug Flags for the Example

For debugging we set the these Properties in the files
D:\10Oracle\06WLS12\domains\domainB\bin\setDomainEnv.cmd
D:\10Oracle\06WLS12\domains\domainA\bin\setDomainEnv.cmd

```
set EXTRA_JAVA_PROPERTIES=-Dweblogic.debug.DebugSecuritySAMLAtn=true -Dweblogic.debug.DebugSecuritySAMLLib=true -
Dweblogic.debug.DebugSecuritySAML2Service=true -Dweblogic.debug.DebugSecuritySAML2CredMap=true -
Dweblogic.debug.DebugSecuritySAML2Atn=true
```

We also want to see milliseconds in the logfiles.
We navigate to Domain->Logging->Advanced and set
Date Format Pattern = yyyy-MM-dd' 'HH:mm:ss.S
We do this for both domains and both servers.

3.10 Configuring IdP initiated flow with POST Binding.

This Blog explains how to configure the IdP initiated flow.
http://fusionsecurity.blogspot.de/2012/06/before-i-forget-it-howto-saml-20-idp.html

Here are the steps to configure this within this example:

3.10.1 Configure an additional end user URL.

In appA/admin/auth.jsp we add an additional URL that points to IdP and uses the target Service URL in the parameter.
It is of the form:
http://<idp-server>:<port>/saml2/idp/sso/initiator?SPName=<SP-Partner-Name>&RequestURL=<target-application-url>

In our case we use:
http://localhost:7001/saml2/idp/sso/initiator?SPName=domainB&RequestURL=http://localhost:7003/appB/admin/services.jsp

3.10.2 Configure the POST Binding POST Form

Within appA we need a jsp that contains a POST form, which posts the SAML Assertion to the Service Provider.
We include sam2_post_form.jsp in appA and redeploy App.
The post form is contained in the file /appA/saml2_post_form.jsp:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<%@ page contentType="text/html;charset=windows-1252"%>
<html>
<head>
</head>
<%
String samlResponse = (String) request.getAttribute("com.bea.security.saml2.samlContent");
String relayState = (String) request.getAttribute("com.bea.security.saml2.relayState");
System.out.println("samp2_post_form.jsp: the samlResponse is: " + samlResponse + ".");
System.out.println("samp2_post_form.jsp: the relayState is: " + relayState + ".");
%>
<body onLoad="document.forms[0].submit();">
<FORM METHOD="POST" ACTION="http://localhost:7003/saml2/sp/acs/post">
<INPUT TYPE="HIDDEN" NAME="RelayState" VALUE="<%=relayState%>"/>
<INPUT TYPE="HIDDEN" NAME="SAMLResponse" VALUE="<%=samlResponse%>">
</FORM>
</body>
</html>
```

In the IdP-Partner configuration we specify the post form. We use the /appA/saml2_post_form.jsp

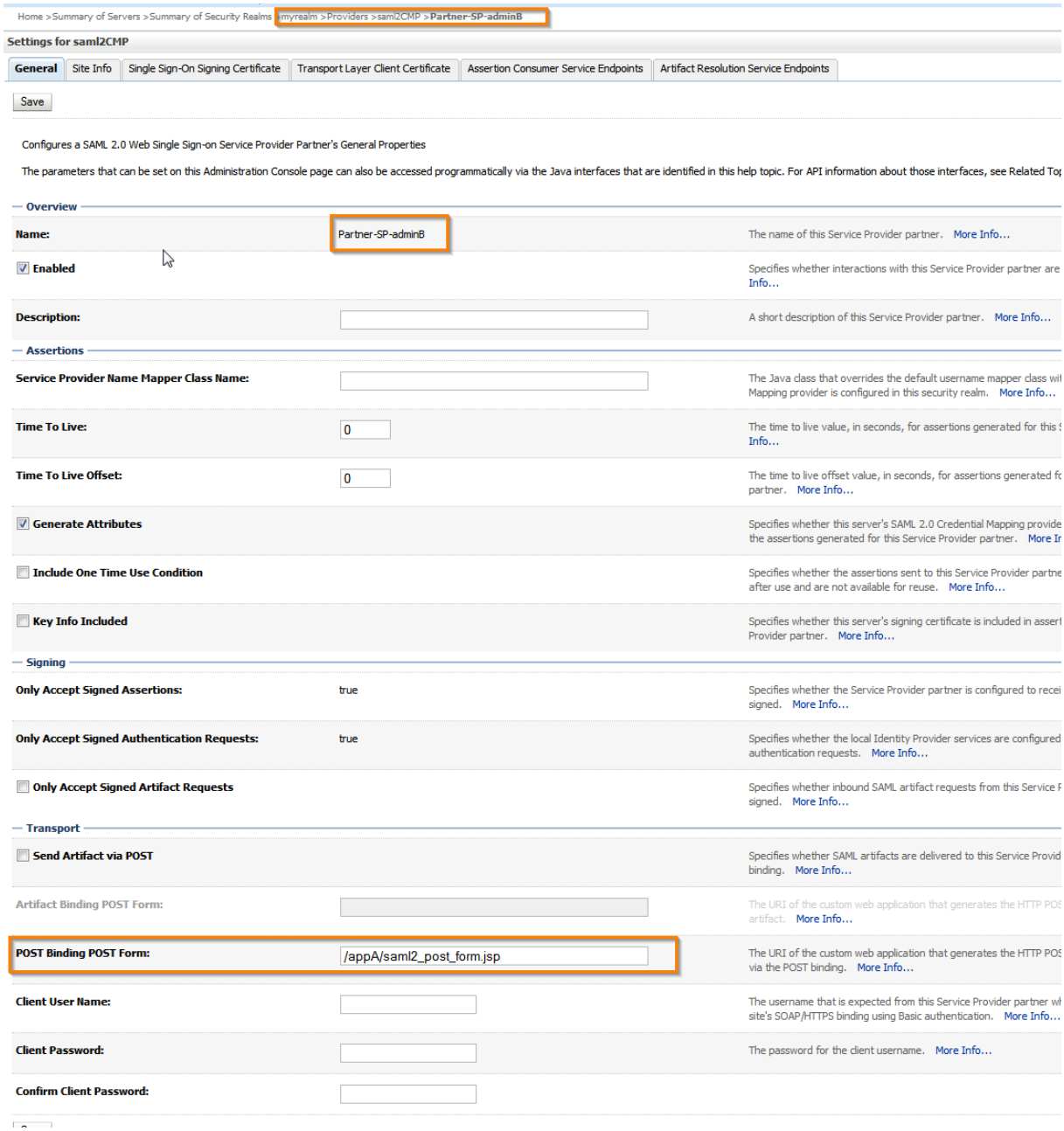


Figure 35. Setting the POST Form on the IdP site for the POST Binding.

For testing, we login at appA with ssouser01 and choose the link for the POST Binding.

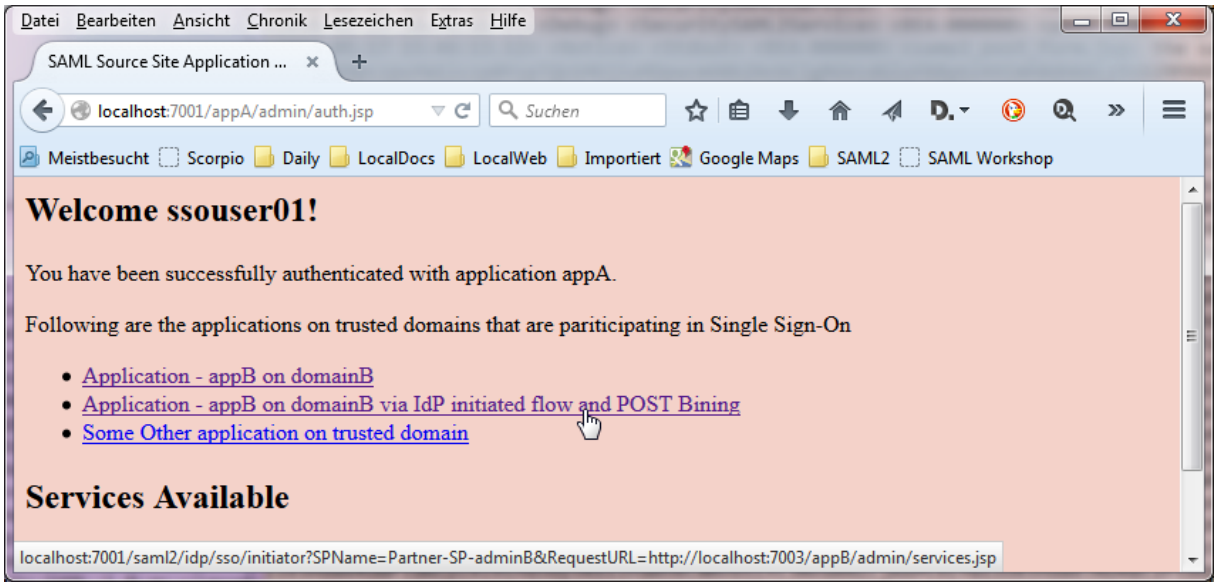


Figure 36. Choosing the IdP initiated flow from appA.

We are directly transferred to the service page of domainB. This time the request is processed by the "Assertion Consumer Service" of domainB. It contains the SAML Assertion, as the following excerpt from the adminB.log demonstrates.

```
<2015-05-17 15:48:39.114> <Debug> <SecuritySAML2Service> <BEA-000000> <SAML2Servlet: Processing request on URI
'/saml2/sp/acs/post'>
<2015-05-17 15:48:39.114> <Debug> <SecuritySAML2Service> <BEA-000000> <getServiceTypeFromURI(): request URI is
'/saml2/sp/acs/post'>
<2015-05-17 15:48:39.115> <Debug> <SecuritySAML2Service> <BEA-000000> <getServiceTypeFromURI(): service URI is
'/sp/acs/post'>
<2015-05-17 15:48:39.115> <Debug> <SecuritySAML2Service> <BEA-000000> <getServiceTypeFromURI(): returning service type 'ACS'>
<2015-05-17 15:48:39.116> <Debug> <SecuritySAML2Service> <BEA-000000> <Assertion consumer service: processing>
<2015-05-17 15:48:39.116> <Debug> <SecuritySAML2Service> <BEA-000000> <get SAMLResponse from http
request:PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz48c2FtbHA6UmVzcG9uc2UgeG1sbnM6c2Ft
…
<2015-05-17 15:48:39.123> <Debug> <SecuritySAML2Service> <BEA-000000> <BASE64 decoded saml message:<?xml version="1.0"
encoding="UTF-8"?><samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:
2.0:protocol" Destination="http://localhost:7003/saml2/sp/acs/post" ID="_0xab777ac5ef829a8140572b099b538d07"
IssueInstant="2015-05-17T13:48:39.030Z" Version="2.0"><saml:Issuer xmlns:saml="ur
n:oasis:names:tc:SAML:2.0:assertion">saml2CMP</saml:Issuer><ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#_0xab777ac5ef829a8140572b099b538d07">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#WithComments"><ec:InclusiveNamespaces
xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="ds saml samlp xs xsi"/></
ds:Transform>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
<ds:DigestValue>SDY0Hn76mvZIuCoQF45+/LB5l0KumkmBIGZ63F82I/s=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
VMQpiMFhDI34XJz8C8dLVRKYP9cWaxyQJdBrzT1pDVSxFhd2q7PITSGIMOshdP2HiB2dsmdYCvHJ
F+F0YKTc+8ljSABfVbZJfwdZBIMJKU8wvPz3uGWKAz7JeKZ431ZEhwaXxG85BNgkdYMq8T0nmu7R
U9YQQYGL5tT/QOQZbmRosIjWZJYe+/Kc4BOqDjPxkXfd5EvHiUI7KleZiYAlxAQBjxkF0C2oGo5k
Kjl5eJgBIX2qjt9v2Qzakc1hq0uR6frcTt3vycTxrxmXSwhyfetejtXGmKMFTy9ykTsdM3f6SEZP
ADXNViuA4yQ+QqzCW8s7DBp95tynv7H82ZLayQ==
</ds:SignatureValue>
</ds:Signature><samlp:Status><samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/></samlp:Status><saml:Assertion
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="_0x97
4db452305af2bf491b72eff5bd736a" IssueInstant="2015-05-17T13:48:39.014Z"
Version="2.0"><saml:Issuer>saml2CMP</saml:Issuer><ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#_0x974db452305af2bf491b72eff5bd736a">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#WithComments"><ec:InclusiveNamespaces
xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="ds saml xs"/></ds:Transfo
rm>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
<ds:DigestValue>627CMXnG8VWRb+STsiWzuYavZZLMLQV4xGBj3H3pQH8=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
hD7p4jdOUNvMSa4WWpy7XZm57OVTAqURmvEaOrvOkoTLJKPhlAoVo1lFznpeJXi75FudVJrMLQbl
5eQtE6QfB20gryzZM82TVjnrA0oTvpU5od86iVX3kZuVagE5QphoxTcCLwD1ntZvvPJaM8qVVdaJ
6E61L/8Pv1OkQFuBk3eFn8UvLYP3Nd3nITLH9ID9STrHaRovxDwFjs2yGIAN/Y7LIIs8aem96iil
71jGJpf1ipHysDx0Lw2Ud4Wnc4atXbDdWdWu1GEzJP7Xcix635vSNGV5G07fUz7Srtt5hV+KCYAv
yGyKuJiSL0CE3e9p8rt+/5jkVKG/SFTvcac+nQ==
</ds:SignatureValue>
```

```
</ds:Signature><saml:Subject><saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified"
NameQualifier="www.domainA.com">ssouser01</saml:NameID><saml:SubjectConfirmation Meth
od="urn:oasis:names:tc:SAML:2.0:cm:bearer"><saml:SubjectConfirmationData NotOnOrAfter="2015-05-17T13:50:34.014Z"
Recipient="http://localhost:7003/saml2/sp/acs/post"/></saml:SubjectConfirmati
on></saml:Subject><saml:Conditions NotBefore="2015-05-17T13:48:34.014Z" NotOnOrAfter="2015-05-
17T13:50:34.014Z"><saml:AudienceRestriction><saml:Audience>saml2AP</saml:Audience></saml:Audienc
eRestriction></saml:Conditions><saml:AuthnStatement AuthnInstant="2015-05-
17T13:48:39.014Z"><saml:AuthnContext><saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified</
saml:AuthnContextClassRef></saml:AuthnContext></saml:AuthnStatement><saml:AttributeStatement><saml:Attribute Name="Groups"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"><sam
l:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">ssouser</saml:AttributeValue></saml:Attribute></saml:A
ttributeStatement></saml:Assertion></samlp:Response>>
<2015-05-17 15:48:39.153> <Debug> <SecuritySAML2Service> <BEA-000000> <<samlp:Response> is signed.>
```

The user is contained in the NameID element and the group is contained in the AttributeStatement element.
This concludes the IdP initiated POST Binding example.

3.11 Configuring Virtual User:

This is explained in the blog post form Biemond: http://biemond.blogspot.de/2011/09/virtual-users-with-saml-in-weblogic.html
We can use virtual users at the SP side, if we need users that are authenticated at the IdP but do not exists in the security realm of domainB. These foreign users are created as virtual users by an extra SAML Authenticator, i.e. this Authenticator populates the subject with principals from the SAML assertion (user and goups).
We need to configure an extra SAML Authenticator in domainA.

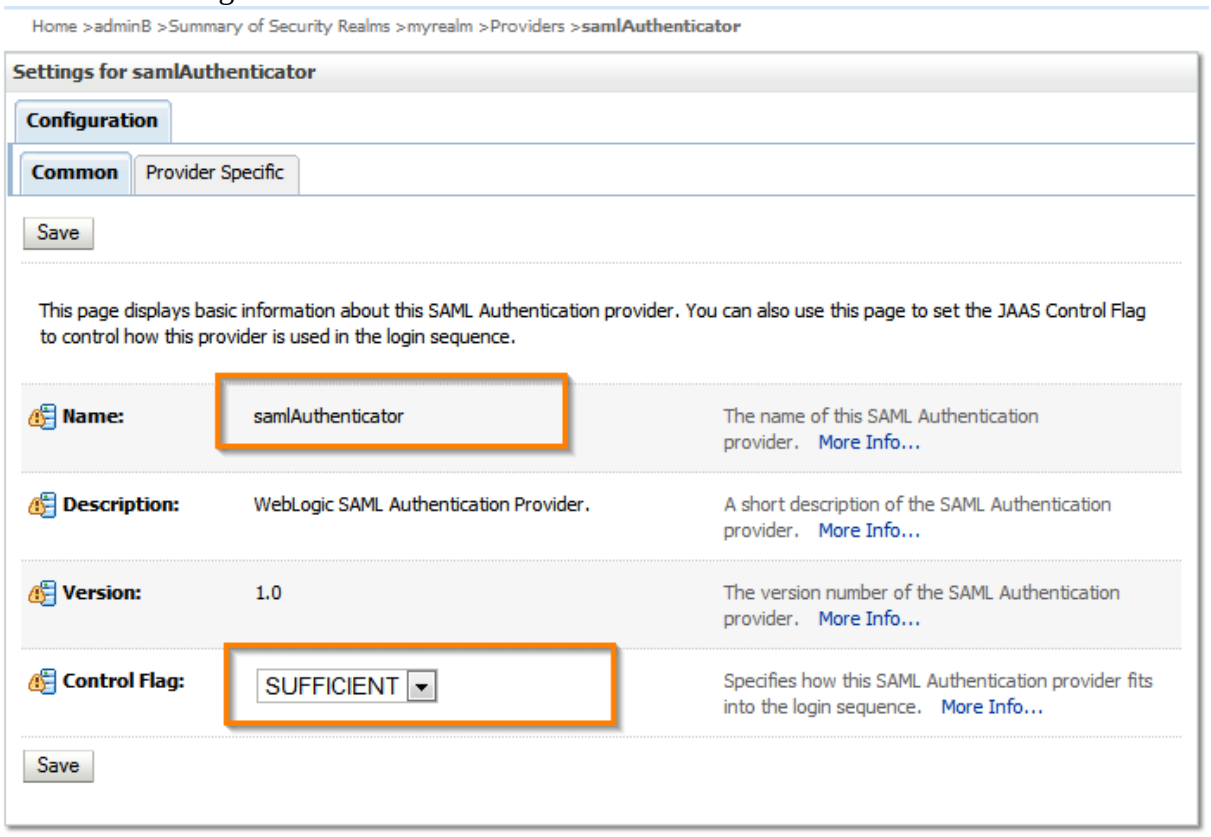


Figure 37. Configuring a SAMLAuthenticator to use virtual users.

We can test this with the user ssouser02 which is present in domainA but not in domainB.

3.12 Setting the Binding Sequence

The SAML Bindings which is chosen by the Federation Services is determined by the sequence as the Bindings appear in the Metadata file. We configure this in step 3. configure SAML 2.0 Identity Provider explained in Chatper 3.6.3 and in step 7. configure SAML 2.0 Service Provider as explained in Chapter 3.6.7. just before exporting the metadata files.
In the Admin Server Console we can prefer a binding and set a default. This is visible in the resulting metadata files as shown below.
adminA_metadata.xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" entityID="saml2CMP">
<md:IDPSSODescriptor WantAuthnRequestsSigned="true" protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<md:KeyDescriptor use="signing">
<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:X509Data>
<ds:X509Certificate>
MIIEYzCCA0ugAwIBAgIJALjMqBVFTXfCMA0GCSqGSIb3DQEBCwUAMHgxCzAJBgNVBAYTAlVTMRAw
DgYDVQQIDAdNeVN0YXRlMQ8wDQYDVQQHDAZNeVRvd24xFzAVBgNVBAoMDk15T3JnYW5pemF0aW9u
MRkwFwYDVQQLDBBGT1IgVEVTVElORyBPTkxZMRIwEAYDVQQDDAlDZXJ0R2VuQ0EwHhcNMTUwNTE1
MTEzNTA0WhcNMzAwNTE2MTEzNTA0WjB2MQswCQYDVQQGEwJVUzEQMA4GA1UECAwHTXlTdGF0ZTEP
MA0GA1UEBwwGTXlUb3duMRcwFQYDVQQKDA5NeU9yZ2FuaXphdGlvbjEZMBcGA1UECwwQRk9SIFRF
U1RJTkcgT05MWTEQMA4GA1UEAwwHU2NvcnBpbzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBAKII7ALqZHOSLcPVU+YrROyncMDHX4VzgEgTxs2zNbVqT8aDLFHrN8SZuhvPnexrH7/eGsES
Qf0P+dJeV8wZyC33lKt/ThcB8nL+59toXNFm65X1+57D5/UAcqIuuQ8VYmI7kjJYdnOGLs/YkkzF
YzvYdkIrLNnyxL3MBfm0SoOqxWcZ5mmtiD6e4r4/l9GpvzarI5+E4WuFzOF1AG6epfU57BZV11tX
/RSMhq3pV/1rzFrEWRxcEm1qFjLNXjY6fO38P6ds5H/KdAbIYPUCGApOK4p7mdK0JU9MNZX/7YRv
EcDSUChZa8dghZPm+j9yLn/ajpmyrin8utRBjDh9KP8CAwEAAaOB8TCB7jAJBgNVHRMEAjAAMA4G
A1UdDwEB/wQEAwID+DAdBgNVHQ4EFgQUinVu9WkkGzWa13lAMx2EIdkx+BIwgbEGA1UdIwSBqTCB
poAUNDj9RdiAz8fS6N8d+KE5sBGIAGqhfKR6MHgxCzAJBgNVBAYTAlVTMRAwDgYDVQQIDAdNeVN0
YXRlMQ8wDQYDVQQHDAZNeVRvd24xFzAVBgNVBAoMDk15T3JnYW5pemF0aW9uMRkwFwYDVQQLDBBG
T1IgVEVTVElORyBPTkxZMRIwEAYDVQQDDAlDZXJ0R2VuQ0GCEEAESIbEQe87ZDqAZkCa/KAwDQYJ
KoZIhvcNAQELBQADggEBACzGpfZWVToZ8fnVXe6LxHB3dAo27o6tt659e4l2nSj3tYmANIC6N+EP
LhrvBTz354oWgZGT/EO4OKNGbHuwfEe5B5O9yZrTJKyUyTSSyxW7Aw5VSHNgQHudNDa9Y0K0Bl3A
M8msNRA2MCjLtey59ENlsOpKTvW1C/1ZzhHF7bgc4W5yOeW+yODXvQgtE22bAHm1L0FBekToZaSa
/wyNEtSBmhUZGQuUoqhufJtX9m/BDomV9+ZvW2da6DZnzqcelLVbtKWe3t3Kd8TAPnIzY9hzy4Tx
euUdASdif2bxjxiix5yD2V2STiQpr0dNy/JJJurOzkEFg47rMoOMt2e0cZE=
</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</md:KeyDescriptor>
<md:ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="http://localhost:7001/saml2/idp/ars/soap" index="0" isDefault="true"/>
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"
Location="http://localhost:7001/saml2/idp/sso/artifact"/>
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="http://localhost:7001/saml2/idp/sso/post"/>
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="http://localhost:7001/saml2/idp/sso/redirect"/>
</md:IDPSSODescriptor>
<md:Organization>
<md:OrganizationName xml:lang="de"/>
<md:OrganizationDisplayName xml:lang="de"/>
<md:OrganizationURL xml:lang="de">http://www.domainB.com</md:OrganizationURL>
</md:Organization>
<md:ContactPerson contactType="technical">
<md:Company>Disney</md:Company>
<md:GivenName>Duck</md:GivenName>
<md:SurName>Donald</md:SurName>
<md:EmailAddress/>
<md:TelephoneNumber/>
</md:ContactPerson>
</md:EntityDescriptor>
```

Here the artifact binding will be chose as the preferred binding for the SingleSignOnServices.
In adminB_metadata.xml the format is more explicit, containing an index number and a default flag.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" entityID="saml2AP">
<md:SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="true"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<md:KeyDescriptor use="signing">
<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:X509Data>
<ds:X509Certificate>
MIIEYzCCA0ugAwIBAgIJAJWD1sRBdhskMA0GCSqGSIb3DQEBCwUAMHgxCzAJBgNVBAYTAlVTMRAw
DgYDVQQIDAdNeVN0YXRlMQ8wDQYDVQQHDAZNeVRvd24xFzAVBgNVBAoMDk15T3JnYW5pemF0aW9u
MRkwFwYDVQQLDBBGT1IgVEVTVElORyBPTkxZMRIwEAYDVQQDDAlDZXJ0R2VuQ0EwHhcNMTUwNTE1
MTEzNTUyWhcNMzAwNTE2MTEzNTUyWjB2MQswCQYDVQQGEwJVUzEQMA4GA1UECAwHTXlTdGF0ZTEP
MA0GA1UEBwwGTXlUb3duMRcwFQYDVQQKDA5NeU9yZ2FuaXphdGlvbjEZMBcGA1UECwwQRk9SIFRF
U1RJTkcgT05MWTEQMA4GA1UEAwwHU2NvcnBpbzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBALf4SFSNVpV8dK6+pK+W5pq0YgKA9bMeTPfIJMdHnuhLIMWudqDnE+gVGUA1KLqoxCO+dlEj
fEfJZ9BzM1sesaNt4K8QDNzUFac5XOdpSI1iI2SUTB1LYVGWHw9E2EFmWEU74f9P/vLyil1ykf7f
VXonWGxZpY2Xvetfni7V/j1dbiMW46DErJ+O4+R3zOMsBRRBLa9/8thpPlQJLppqO9bzeRt+utuB
EnstGW63h5ndSqZtWwhurg5HQ3lZQNsqWe92kEHhFjyzG2sG27UTKzwrOb18m/3i9e1SbychrgVS
7QtwdsPpIMvc43X3PoyQPvwV9b9CEQuel9uZkpky+OkCAwEAAaOB8TCB7jAJBgNVHRMEAjAAMA4G
A1UdDwEB/wQEAwID+DAdBgNVHQ4EFgQUV03RZqkgbAtLe0aRtD034qbvZFAwgbEGA1UdIwSBqTCB
poAUNDj9RdiAz8fS6N8d+KE5sBGIAGqhfKR6MHgxCzAJBgNVBAYTAlVTMRAwDgYDVQQIDAdNeVN0
YXRlMQ8wDQYDVQQHDAZNeVRvd24xFzAVBgNVBAoMDk15T3JnYW5pemF0aW9uMRkwFwYDVQQLDBBG
T1IgVEVTVElORyBPTkxZMRIwEAYDVQQDDAlDZXJ0R2VuQ0GCEEAESIbEQe87ZDqAZkCa/KAwDQYJ
KoZIhvcNAQELBQADggEBADd/OJ4vuhG5IjvOy5Haf44/Y/ajUVowqiL/Cx5HwL27K6vO3fC7Xxha
```

```
80o2ZwtTYinsVopDsgaKu/34+tkoYMwHpbcxsCj6vSuDRi7+3gTsak1e/DQgupfUDrm6JPVPliid
P6wehJteDKQ3gg4jtbnM23qzvsIYNP0D4eVdCfcXNg9WwKkOeY9RvKST/7Us0Ske1ffJZRvtHrmi
qn6sz7KyUjLTAmPm3wHke2rQJl5MGyhBXehKPlEpud+VGxAA81tuGylYAdAIvrvZ+/4OdChblHsx
eYKKoTVl4wygSv26JkilpYFyOk/qyRBIWCmPtNUKP91mMJNCqtA+VLthMJE=
</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</md:KeyDescriptor>
<md:ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="http://localhost:7003/saml2/sp/ars/soap" index="0" isDefault="true"/>
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="http://localhost:7003/saml2/sp/acs/post" index="0" isDefault="true"/>
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"
Location="http://localhost:7003/saml2/sp/acs/artifact" index="1"/>
</md:SPSSODescriptor>
<md:Organization>
<md:OrganizationName xml:lang="de">www.domainB.com</md:OrganizationName>
<md:OrganizationDisplayName xml:lang="de">www.domainB.com</md:OrganizationDisplayName>
<md:OrganizationURL xml:lang="de">http://www.domainB.com</md:OrganizationURL>
</md:Organization>
<md:ContactPerson contactType="technical">
<md:Company>Disney</md:Company>
<md:GivenName>Daisy</md:GivenName>
<md:SurName>Duck</md:SurName>
<md:EmailAddress/>
<md:TelephoneNumber/>
</md:ContactPerson>
</md:EntityDescriptor>
```

Here the HTTP-POST Binding will be chosen for the AssertionConsumerService.
If we want to change this behavior for a Partner-SP or Partner-IdP, we can modify the xml files directly and re-import the modified metadata files.
Alternatively we could change the settings in the configurations of the Identity Provider and Service Provider (step 3 and step 7). But then we have to export and import the files again, just changing the settings will have no effect.

3.13 SAML 2.0 Examples in Blog Posts.

The following is a list of some blogs that demonstrate different aspects of SAML configuration in WLS.

SAML 2.0 Example from Biemond, based on earlier blog post, using ssl demo certs and saml metadata file.
http://biemond.blogspot.de/2009/09/sso-with-weblogic-1031-and-saml2.html
http://biemond.blogspot.de/2009/05/sso-with-weblogic-103-and-saml.html

Steps to configure SAML 2 on Weblogic Server 10.3.0, using pointbase for the rdbms security realm.
https://blogbypuneeth.wordpress.com/2011/01/15/steps-to-configure-saml-2-on-weblogic-server-10-3-0/

Configure WSO2 Identity Server SAML2 IDP with Oracle Weblogic as Service Provider
Example of integration between WSO2 and WLS
http://tanyamadurapperuma.blogspot.de/2013/09/configure-wso2-identity-server-saml2.html

3.14 Conclusion

This tutorial comprises a comprehensive description of a web single-sign-on scenario using SAML 2.0 in weblogic server. It demonstrates all steps necessary to install, configure and run a demo application. The whole tutorial is split into four parts. In the first part we walked through the installation of weblogic server on a windows machine and the creation of two domains. We also installed the sample applications in this part. In part two we looked at the SAML configuration in weblogic server, using the Administration Console. We introduced a recommended configuration sequence which comprises 14 steps and was illustrated by

diagrams and mind maps. While this sequence is not mandatory, it structures the manual configuration process in an efficient manner and can serve as a template for configuring real world SAML scenarios.
In the third part we demonstrated two test cases for the service provider initiated flow scenario. In the fourth part we extended the example to include an identity provider initiated flow scenario and demonstrated an advanced weblogic feature called virtual user.
The tutorial concludes with some configuration and debugging tips and a brief overview of other blog post covering similar subjects.