Creating a User Authentication for a Java Web Service Using a Certificate

Release 650
Copyright

© Copyright 2004 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.
Icons in Body Text

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>⚠️</td>
<td>Caution</td>
</tr>
<tr>
<td>_example</td>
<td>Example</td>
</tr>
<tr>
<td>_note</td>
<td>Note</td>
</tr>
<tr>
<td>Recommendation</td>
<td>Recommendation</td>
</tr>
<tr>
<td>_syntax</td>
<td>Syntax</td>
</tr>
</tbody>
</table>

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see Help on Help → General Information Classes and Information Classes for Business Information Warehouse on the first page of any version of SAP Library.

Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Example text</em></td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><code>&lt;Example text&gt;</code></td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Keys on the keyboard, for example, F2 or ENTER.</td>
</tr>
</tbody>
</table>
Creating a User Authentication for a Java Web Service Using a Certificate

Task
This tutorial takes you through the development steps required for adding a user authentication using a certificate to an existing Web service. For this purpose, the predefined Web service CreditCheck is used.

For further information about the Web service CreditCheck, see the following tutorial: Creating a Web Service

The Web service CreditCheck offers the simple function of checking the credit-worthiness of a customer using the customer number. If a customer number between 1001 and 1005 is passed, the credit limit of the customer is returned. Otherwise a message informs you that no corresponding customer exists.

Objectives
By the end of this tutorial, you will be able to:

✔ Create a user authentication using a certificate for an existing Web service.

Prerequisites
☐ Your SAP J2EE Engine is configured for the use of SSL.
☐ You have the certificate used by the client for authentication and the certificate of the corresponding CA in the form of crt files.

Systems, Installations, and Authorizations
☐ You have installed the SAP NetWeaver Developer Studio and configured it for use of the SAP J2EE Engine.
☐ You can access the SAP J2EE Engine using the Visual Administrator.

Knowledge
☐ You can create and deploy a Web service based on an EJB. For more information, see Web Services Toolset.

Next Step:
Importing Projects [Page 6]
Importing Projects

The following projects are available for this tutorial:

- EJB project CreditLimitCheck (on which the Web service is based)
- EAR project CreditLimitCheckEAR (contains the Web service CreditCheck)

The projects are available in the file JAVA_SERVER_RAW.zip. To download the file, click here.

Procedure

Importing Projects into the SAP NetWeaver Developer Studio

1. Download the ZIP file and save it to any directory on your local hard disk or directly in the work area of the SAP NetWeaver Developer Studio.
2. Unpack the ZIP file.
3. Start the SAP NetWeaver Developer Studio.
4. Import the projects CreditLimitCheck and CreditLimitCheckEAR and open the J2EE Explorer in the J2EE deployment perspective.

Result

Once you have imported the project into the SAP Developer Studio, the following structure is displayed in the J2EE Explorer.
Next Step:

Adding a Web Service Configuration [Page 7]

Adding a Web Service Configuration

You can use a Web service configuration to specify the runtime properties of a Web service. At least one configuration is assigned to each Web service. When creating a proxy, a logical port is created for each configuration.

You will learn how to create a new Web service configuration - based on an existing Web service - and how to specify the settings for the activation of the user authentication using a certificate.

Prerequisites

☐ The structures of your projects CreditLimitCheck and CreditLimitCheckEar are displayed in the J2EE Explorer.

Procedure

Creating and Configuring a Web Service Configuration

1. To create a new configuration, open the WS Deployment Descriptor Editor by double-clicking the node CreditCheck - Web Service Configurations. In the tree structure that appears, select the entry Web Service Configurations – CreditCheck, and choose Add.

2. The dialog box Web Services appears. Enter the name of the new configuration (STRONG) in the field Configuration Name and choose Finish.

3. Select the node Web Service Configurations – CreditLimitPrivateLocal – STRONG – Security to display the security settings of the new configuration.

4. Select HTTPS as the Transport Protocol and HTTP Authentication as the Authentication Mechanism. Activate the setting X.509 Certificate and save your settings using STRG-S, for example.
Deploying a Modified Web Service

1. In the context menu for the project CreditLimitCheck, choose Build EJB Archive.
2. In the context menu for the project CreditLimitCheckEAR, choose Build Application Archive.
3. In the context menu of the node CreditLimitCheckEAR – CreditLimitCheckEAR.ear, call the function Deploy to j2ee engine and wait for the message in the format hh:mm:ss [nnn] Finished Deployment [more] displayed in the Deploy Output View.

Result

Based on the Web service CreditCheck, you created the new configuration STRONG and made it accessible by deploying the Web service on the J2EE Engine. If the Web service is called using this configuration, a user authentication via a certificate is required from the Web service consumer to be able to use the Web service. HTTPS is used for the data transfer.

The newly created configuration is displayed in the WS Deployment Descriptor Editor under the node Web Service Configurations – CreditCheck.

The overview page of the Web service CreditCheck contains a table with the header Deploy-time features for Binding: STRONGBinding and the name/value pairs TLSType – SSL, AuthenticationMechanism – HTTP, AuthenticationMethod – CertAuth, and SupportsSSO2Authentication – false.

To access the overview page of a Web service, you call the address http://<host>:<port>/wsnavigator in your browser and select the name of the requested Web service in the tree structure Available Web Services.
Next Step:
Configuring the Server for Checking the Client Certificate [Page 9]

![Configuring the Server for Checking the Client Certificate]

Configuring the Server for Checking the Client Certificate

The configuration of the Web service now requires authentication using a client certificate, but you must also make some settings in the J2EE Engine. The J2EE must be configured in such a way that it checks incoming certificates for their trustworthiness. In addition, the client certificate must be assigned to a J2EE user. The Web service client logged on via the certificate then acts as this J2EE user.

Prerequisites
- The SSL provider of the dispatcher was started.

Procedure

Importing the Client Certificate

1. Start the Visual Administrator and log on to the J2EE Engine that contains your Web service.
2. Select the node `<SID> - Server … - Services – Key Storage` (where `<SID>` is the system ID of the used J2EE) and activate the tabs `Runtime` (top) and `Data` (bottom) in the right window.
3. Create a new keystore view by choosing the `Create View` button below the `Views` area. Give the new view a descriptive name - for example, `ClientCertificates` - and confirm your entry. The new entry is displayed in the `Views` area.
4. Select your newly created keystore view.
5. In the `Entry` section (bottom right), choose `Load`. In the window that opens, select the `.crt` file that contains the certificate used by the client for the logon and choose `OK`. The certificate is now imported and is listed in the `Entries` field.

Assigning the Certificate to a J2EE User

1. In the Visual Administrator, select the node `<SID> - Server … - Services – Security Provider` (where `<SID>` is the system ID of the J2EE) and activate the tabs `Runtime` and `User Management` in the right window.
2. Activate the tabs **Users** and **Tree** (bottom). In the tree structure, select the user as which the Web service client is to be logged on to the J2EE engine. The properties of the selected user are displayed on the right.

3. Choose **Add** in the area **Certificates**. The window **Add Certificates** appears.

4. Select your newly created keystore view (for example, ClientCertificates). Select the certificate used by the client for the authentication and choose **OK**. A new entry appears in the area **Certificates** and the certificate is assigned to the user.

**Importing the Certificate of the Corresponding CA**

1. In the Visual Administrator, select the node `<SID> - Server ... - Services – Key Storage` (where `<SID>` is the system ID of the J2EE) and activate the tabs **Runtime** (top) and **Data** (bottom) in the right window.

2. In the area **Views**, select the keystore view **TrustedCAs**.

3. In the **Entry** section (bottom right), choose **Load**. In the window that opens, select the `.crt` file that contains the certificate of the corresponding CA and choose **OK**. The certificate is now imported and is listed in the **Entries** field.

**Adjusting the SSL Configuration**

1. In the Visual Administrator, select the node `<SID> - Server ... - Services – SSL Provider`.

2. In the right window, activate the tab **Runtime** and double-click the used dispatcher in the list.

3. In the field **Configuration**, select the host/port configuration that is to be used for calling the Web service and activate the tab **Client Authentication** in the bottom right window.
4. Choose *Request client certificate*.
5. Choose *Add* (on the right). The window *Available Certificates* appears.
6. Select the CA certificate you imported in the previous section and choose *OK to confirm*. The certificate is now displayed in the area *Trusted Certification Authorities* and is therefore considered trustworthy.

**Result**

By importing the client certificate and assigning it to a specific J2EE user, the client is now able to authenticate itself to the server as this user. Since the certificate of the corresponding CA was imported and added to the trustworthy CAs, the certificate of the client is also accepted.

---

**Exporting the Certificate Used for SSL (Optional)**

When establishing a secure connection to a Web service client, the server sends a certificate for its authentication. Before the certificate can be checked for its trustworthiness by the Web service client, it must be exported from the J2EE Engine of the server for it then to be made available to the consumer of the service.

If the Web service and its client are located on the same SAP J2EE Engine, this step is not necessary.

**Prerequisites**

☐ The SSL provider of the used dispatcher was started.

**Procedure**

**Exporting the Certificate**

1. Start the Visual Administrator and log on to the SAP J2EE Engine that contains your Web service.
2. Select the node *<SID> - Server … - Services – SSL Provider*, where *<SID>* is the system ID of the J2EE Engine.
3. In the right window, activate the *Runtime* tab and double-click the used dispatcher in the list.
4. In the field *Configuration*, select the host/port configuration that is to be used for calling the Web service and activate the tab *Server Identity* in the bottom right window. Make a note of the text that appears in the field *Enabled Credentials*.
5. In the left window, select the node *<SID> - Server … - Services – Key Storage* and activate the tabs *Runtime* (top) and *Data* (bottom) in the right window.
6. In the list *Views*, select the entry *service_ssl* and in the list *Entries*, select the row with the noted key name and the extension `-cert`. 
7. Choose **Export** in the area **Entry** (bottom right) and, in the dialog box that appears, save the certificate as a crt file in your file system.

**Result**

The certificate used by the server for the SSL handshake was exported into a crt file. It can now be used by a Web service client to check the identity of the called server.