IBM InfoSphere Information Services Director
Version 8 Release 7

Handbuch zum Publizieren sicherer Services

SC12-4672-00

IBM InfoSphere Information Services Director
Version 8 Release 7

Handbuch zum Publizieren sicherer Services

SC12-4672-00

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen und Marken" auf Seite 55 gelesen werden.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM InfoSphere Information Services Director Version 8 Release 7,
IBM Form SC19-3484-00
herausgegeben von International Business Machines Corporation, USA

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
TSC Germany
Kst. 2877
Oktober 2011

Inhaltsverzeichnis

Sichere Services publizieren

Nachdem Sie einen Service mithilfe von IBM® InfoSphere Information Services Director entworfen und implementiert haben, schützen Sie diesen Service über die IBM InfoSphere Information Server-Konsole oder mithilfe eines Assembliertools.

Vorbereitende Schritte

- In diesen Themen wird vorausgesetzt, dass Sie IBM InfoSphere Information Server und InfoSphere Information Services Director 8.5 mit IBM WebSphere Application Server, Version 6.1 oder 7.0, mit den unterstützten entsprechenden Fixpackstufen verwenden. Die unterstützte Fixpackstufe für Ihre installierte Version von WebSphere Application Server finden Sie auf der Seite mit den Systemanforderungen für IBM InfoSphere Information Server: http://www.ibm.com/software/data/infosphere/info-server/overview/requirements.html

Informationen zu diesem Vorgang

- In diesen Tasks wird davon ausgegangen, dass Sie mit Web-Services, SOAP über HTTP oder SOAP über JMS und anderen SOA-Konzepten (Service-Oriented Architecture) vertraut sind und Erfahrung haben mit InfoSphere Information Services Director und dem Entwerfen, Implementieren und Testen von Services mithilfe von InfoSphere Information Server.
- In den Beispielen, die das Schützen eines in diesen Tasks implementierten Service veranschaulichen, wird die SOAP-über-HTTP-Bindung verwendet.
- In vielen der beschriebenen Tasks werden die Implementierungsdeskriptordateien von Services modifiziert. Zum Bearbeiten dieser verschiedenen Deskriptordateien können Sie ein Assembliertool wie IBM Rational Application Developer oder IBM WebSphere Application Server Toolkit verwenden. Für die Tasks in diesen Themen werden Rational Application Developer 7.0 und IBM WebSphere Application Server Toolkit 6.0 verwendet. In den meisten dieser Prozeduren wird das Assembliertool Rational Application Developer Version 7.5.4 verwendet.
  - Rational Application Developer und Rational Software Architect nutzen eine gemeinsame Codebasis, sodass Rational Application Developer und Rational Software Architect hiernach gegeneinander ausgetauscht werden können. Rational Software Architect ist eine Obermenge von Rational Application Developer und anderen IBM Rational-Angeboten (z. B. Rational Web Developer und Rational Software Modeler).
  - Unter den verschiedenen verfügbaren IBM Assembliertools bieten Rational Software Architect bzw. Rational Application Developer eine bessere Unterstützung für die Web-Services-Sicherheit, insbesondere durch die assistentengestützte Automatisierung der meisten Tasks und durch Abstraktionen der Sicherheitsdetails in den Implementierungsdeskriptoren. Abgesehen von den Assistenten sind die Benutzerschnittstellen und die Editoren in Rational Application Developer 7.0 und IBM WebSphere Application Server Toolkit 6.0 nahezu identisch.

Web-Service-Sicherheit - Übersicht

Web-Service-Sicherheit ist ein Branchenstandard, der beschreibt, wie nutzdatenspezifische Sicherheit auf SOAP-basierte Standard-Web-Services angewendet werden kann.

Anmerkung: Im Allgemeinen bezieht sich der Begriff "Web-Services" auf SOAP-formatierte Nachrichten, die über HTTP übermittelt werden. In diesem Thema geht es jedoch um eine Reihe von Standards, die als WS-Security oder Web-Services-Sicherheit bezeichnet werden. Die WS-Security-Authentifizierung von Benutzernamenstokens ist beispielsweise eine der vielen Sicherheitstechnologien, die zu den WS-Security-Standards gehören. Einige Sicherheitsmechanismen sind nicht Bestandteil der WS-Security, wie z. B. die HTTP-Basisauthentifizierung, die J2EE-Autorisierung und die SSL-Verschlüsselung.

Die Web-Service-Sicherheit stützt sich auf andere akzeptierte Standards, wie z. B. die digitale XML-Signatur und die XML-Verschlüsselung, und nutzt mathematische Algorithmen, Techniken und vorhandene Softwareressourcen, die schon lange für die Kommunikation und die Datensicherheit im Internet und für ältere sichere Datenaustauschverfahren, wie z. B. EDI-Transaktionen (Electronic Data Interchange), verwendet werden. Die Web-Service-Sicherheit ist erforderlich, weil HTTPS mit SSL für komplexe Anwendungen nicht ausreicht. Insbesondere gilt dies für Anwendungen, die längere Prozessabläufe mit unterschiedlichen Behandlungsroutinen (Anwendungen) erfordern, die an einem kompletten Web-Service-Datenaustausch oder einer kompletten Web-Service-Aktivität teilnehmen.

Zur Veranschaulichung wird häufig ein Service verwendet, der eine komplette Kaufanforderung umfasst. Er kann Folgendes enthalten: bestimmte Kreditkartendetails für die Fakturierungsabteilung, Kontenaktualisierungsinformationen für den Kundenservice und einfache Produktnummerninformationen für die Durchführung einer Bestellung und Lieferung der Ware. In einer verteilten Serviceumgebung arbeiten unterschiedliche Anwendungen im Unternehmen zusammen, um den vollständigen Service bereitzustellen. So kann es beispielsweise kritisch sein, dass die Bestandsanwendung Zugriff auf Produktbestellungsdetails hat, um festzustellen, ob sich das gewünschte Produkt im Bestand befindet. Genauso kritisch aber ist, dass kein Zugriff auf detaillierte Konteninformationen oder Kreditkartennummern besteht. Es ist wichtig, dass die an Web-Service-Interaktionen beteiligten Parteien die jeweils andere Partei prüfen und so sicherstellen können, dass niemand den Inhalt der Nachricht während der Übertragung geändert hat.

Authentifizierung, Integrität und Vertraulichkeit sind die zentralen Sicherheitsmerkmale der Web-Service-Sicherheit. Weitere Sicherheitsaspekte, wie z. B. der fälschungssichere Herkunftsnachweis, können aus den Kernfunktionen im Framework der Web-Service-Sicherheit abgeleitet werden.

Sicherheit im Allgemeinen und die Web-Service-Sicherheit im Besonderen finden zunehmend Beachtung in Diskussionen zur serviceorientierten Architektur (SOA) und zum Thema 'Informationen als Service'. Der Web-Service-Sicherheitsstandard wurde von Anbietern und Kunden bislang nur langsam angenommen. Dennoch wurde die erste Stufe der Web-Service-Sicherheitsstandards bereits vor mehreren Jahren bestätigt, und das Interesse nimmt zu. Zu den Gründen für die langsame Annahme gehören die Komplexität, Bedenken hinsichtlich der Leistung, anbieterseitig fehlende Unterstützung durch Tools, nicht ausgereifte Standards (oder zumindest entsprechende Befürchtungen) sowie die schlichte Tatsache, dass die Web-Service-Sicherheit für viele Web-Service-Implementierungen als nicht erforderlich betrachtet wird. HTTPS mit SSL war bisher für die meisten Punkt-zu-Punkt-Web-

Service-Implementierungen akzeptabel und komplexes Multihopping, bei dem die Web-Service-Sicherheit unerlässlich ist, wird in den meisten Unternehmen nach wie vor als technische Neuerung mit hohem Fehlerpotenzial angesehen. Außerdem befinden sich viele Web-Service-Produktionsimplementierungen hinter einer Firewall oder stellen keine sensiblen Informationen bereit, sodass in diesen Fällen kein Bedarf an Web-Service-Sicherheit besteht.

Da die serviceorientierte Architektur jetzt häufiger Verwendung findet (zumindest in der Architektur von Unternehmen und in Planungsdiskussionen) und Web-Services häufiger für komplexe Mulithoppinglösungen vorgeschrieben werden, in denen viele Parteien auf die Nutzdaten einer Nachricht zugreifen, wird der Web-Service-Sicherheit mehr und mehr Beachtung geschenkt. Folglich sind die Web-Service-Sicherheit und die Fähigkeit von IBM InfoSphere Information Services Director, diese Sicherheit zu unterstützen, ins Bewusstsein von Architekten und Entscheidungsträgern innerhalb und außerhalb von IBM gerückt.

Web-Service-Sicherheit ist nicht einfach. Ihre Implementierung erfordert umfachreiche Planung und Disziplin sowie Kooperation zwischen Client- und Serverimplementierungsteams. Dabei müssen verschiedene Aspekte sehr sorgfältig durchdacht werden, wie z. B. die folgenden:

- Welche Arten der Web-Service-Sicherheit sind erforderlich und gewünscht? (Authentifizierung, Unterzeichnen, Verschlüsselung usw.)
- Welche Richtungen sollen angestrebt werden? (Client zu Host? Host zurück zu Client? Beide Richtungen für alle Arten der Web-Service-Sicherheit? Weiterleitung und Vermittlerautorisierungen?)
- Welche Abschnitte und Unterabschnitte einer Nachricht sollen berücksichtigt werden? Welche Technologien sind betroffen (z. B. X.509)?

Aus technologischer Sicht ist der größte Teil der von der Web-Service-Sicherheit definierten Laufzeittechnologie bereits vorhanden. Verfahren für das digitale Unterzeichnen, die sich in der Branche bewährt haben, Verschlüsselungsalgorithmen mit privaten Schlüsseln und die Integration in verschiedenen Protokollen werden bereits in der Web-Service-Sicherheit eingesetzt. Neu hingegen sind Syntax und Regeln für die Definition der Web-Service-Sicherheit für Requester- und Provider-Engines sowie die spezifischen Dateinamen, Erweiterungen und angepassten Eigenschaften jeder Anbieterimplementierung.

Sicherheit mithilfe der IBM InfoSphere Information Server-Konsole hinzufügen

Mithilfe von IBM InfoSphere Information Services Director in der IBM InfoSphere Information Server-Konsole können Sie die HTTP-Basisauthentifizierung, die Authentifizierung über WS-Security-Benutzernamenstokens oder Datenschutz über SSL-Verschlüsselung hinzufügen.

Informationen zu diesem Vorgang

Sie können Ihren Services über die IBM InfoSphere Information Server-Konsole oder mithilfe eines Assembliertools Sicherheit hinzufügen. Die Verwendung der IBM InfoSphere Information Server-Konsole wird bevorzugt, da die Konsole einige Tasks automatisiert.

HTTP-Basisauthentifizierung mithilfe der IBM InfoSphere Information Server-Konsole hinzufügen

Führen Sie diese Task aus, um Ihrem Service die HTTP-Basisauthentifizierung hinzuzufügen.

Informationen zu diesem Vorgang

Für diese Task wird vorausgesetzt, dass eine SOAP-über-HTTP-Verbindung zum Service besteht. Für andere Bindungen, wie z. B. REST-, REST 2.0-, RSS- oder EJB-Bindungen, ist die Konfiguration ähnlich.

Vorgehensweise

1. Wählen Sie im Arbeitsbereich **Informationsserviceanwendung** eine Anwendung aus und klicken Sie auf **Öffnen**.
2. Klicken Sie auf **Bearbeiten**.
3. Klicken Sie doppelt auf den Service, den Sie mit der HTTP-Basisauthentifizierung schützen wollen.
4. Wählen Sie **Authentifizierung erforderlich** in der Sicherheitsanzeige des Übersichtsteilfensters aus.
5. Klicken Sie auf das Servicebindungsteilfenster.
6. Wählen Sie **HTTP-Basis** im Menü **Authentifizierungsunterstützung** aus.
7. Klicken Sie auf **Anwendung speichern**.
8. Klicken Sie auf **Anwendung schließen**.

Nächste Schritte

Sie müssen den Service erneut implementieren, damit diese Änderungen wirksam werden.

Wenn Sie einem Service mithilfe der IBM InfoSphere Information Server-Konsole die HTTP-Basisauthentifizierung hinzufügen, fügt IBM InfoSphere Information Services Director eine Standardberechtigungsregel hinzu, die es nur Benutzern mit der Rolle eines InfoSphere Information Services Director-Konsumenten, -Administrators oder -Katalogmanagers erlaubt, die Operationen innerhalb dieser Services aufzurufen.

Authentifizierung über Benutzernamenstokens mithilfe der IBM InfoSphere Information Server-Konsole hinzufügen

Führen Sie diese Task aus, um Ihrem Service die Authentifizierung über Benutzernamenstokens hinzuzufügen.

Informationen zu diesem Vorgang

Für diese Task wird vorausgesetzt, dass eine SOAP-über-HTTP-Verbindung zum Service besteht.

Vorgehensweise

1. Wählen Sie im Arbeitsbereich **Informationsserviceanwendung** eine Anwendung aus und klicken Sie auf **Öffnen**.
2. Klicken Sie auf **Bearbeiten**.
3. Klicken Sie doppelt auf den Service, den Sie mit der Authentifizierung über Benutzernamenstokens schützen wollen.
4. Wählen Sie **Authentifizierung erforderlich** in der Sicherheitsanzeige des Übersichtsteilfensters aus.
5. Klicken Sie auf das Servicebindungsteilfenster.
6. Wählen Sie **Token des Benutzernamens für die WS-Sicherheit** im Menü **Authentifizierungsunterstützung** aus.
7. Klicken Sie auf **Anwendung speichern**.
8. Klicken Sie auf **Anwendung schließen**.

Nächste Schritte

Sie müssen den Service erneut implementieren, damit diese Änderungen wirksam werden.

Wenn Sie einem Service mithilfe der IBM InfoSphere Information Server-Konsole die Authentifizierung hinzufügen, fügt IBM InfoSphere Information Services Director eine Standardberechtigungsregel hinzu, die es nur Benutzern mit der Rolle eines InfoSphere Information Services Director-Konsumenten, -Administrators oder -Katalogmanagers erlaubt, die Operationen innerhalb dieser Services aufzurufen.

Datenschutz über SSL-Verschlüsselung mithilfe der IBM InfoSphere Information Server-Konsole hinzufügen

Führen Sie diese Task aus, um Ihrem Service Datenschutz über SSL-Verschlüsselung hinzuzufügen.

Informationen zu diesem Vorgang

Für diese Task wird vorausgesetzt, dass eine SOAP-über-HTTP-Verbindung zum Service besteht. Für andere Bindungen, wie z. B. Text über HTTP, REST, REST 2.0 oder RSS, ist die Konfiguration ähnlich.

Vorgehensweise

1. Wählen Sie im Arbeitsbereich **Informationsserviceanwendung** eine Anwendung aus und klicken Sie auf **Öffnen**.
2. Klicken Sie auf **Bearbeiten**.
3. Klicken Sie doppelt auf den Service, den Sie mit Datenschutz über SSL-Verschlüsselung sichern wollen.
4. Wählen Sie **Vertraulichkeit erforderlich** in der Sicherheitsanzeige des Übersichtsteilfensters aus. Wenn Sie auf das Serviceteilfenster **Bindungen** klicken, sehen Sie, dass HTTPS zum Aufrufen dieses Service erforderlich ist.
5. Klicken Sie auf **Anwendung speichern**.
6. Klicken Sie auf **Anwendung schließen**.

Nächste Schritte

Ihr Anwendungsserver muss für HTTPS und SSL konfiguriert sein, um einen mit SSL verschlüsselten Service zu verwenden.

Sicherheit mithilfe eines Assembliertools hinzufügen

Mithilfe eines Assembliertools können Sie Authentifizierung, Autorisierung, Datenschutz und Datenintegrität hinzufügen.

Informationen zu diesem Vorgang

Sie können Ihren Services über die IBM InfoSphere Information Server-Konsole oder mithilfe eines Assembliertools Sicherheit hinzufügen. Mit der IBM InfoSphere Information Server-Konsole werden einige Tasks automatisiert, die Sie bei Verwendung eines Assembliertools manuell ausführen müssen. Dagegen können Sie einen Service mithilfe eines Assembliertools mit Verfahren schützen, die über die IBM InfoSphere Information Server-Konsole nicht möglich sind.

Bindungsstruktur - Übersicht

Wenn Sie die Sicherheitsfunktionen ohne Zuhilfenahme einer Anwendung wie z. B. eines Assembliertools aktivieren wollen, müssen Sie die zugrunde liegende interne Bindungsstruktur verstehen, um die entsprechenden Einstellungen modifizieren zu können.

Dieser Abschnitt bietet eine Übersicht über die interne Struktur zweier gängiger Bindungen: SOAP über HTTP und SOAP über JMS.

SOAP über JMS

SOAP über JMS ist ein Bindungstyp, mit dem Sie eine Verbindung zu einem Service herstellen können. Lesen Sie dieses Thema, um die Basisstruktur einer SOAP-über-JMS-Bindung zu verstehen. Dieses Wissen ist nützlich, wenn Sie Sicherheitseinstellungen ohne Zuhilfenahme einer Anwendung wie z. B. eines Assembliertools aktivieren wollen.

Aufgabe der SOAP-über-JMS-Bindung ist es, für SOAP-Anforderungen, die über eine JMS-Warteschlange oder ein JMS-Thema an den Service gesendet werden, ein Unmarshaling durchzuführen, diese Anforderungen als EJB-Aufrufe an die IBM InfoSphere Information Server-Serviceimplementierung umzuleiten und für die Antwort, die in einer SOAP-Nachricht an die JMS-Warteschlange oder das JMS-Thema zurückgesendet wird, ein Marshaling durchzuführen.

Folgende Artefakte bilden eine SOAP-über-JMS-Bindung:
- Eine WSDL-Datei, die den Service beschreibt.
- Eine anwendungsserverspezifische Router-MDB (Message-Driven Bean - nachrichtengesteuerte Bean), die für über JMS eingehende SOAP-Anforderungen empfangsbereit ist.
- Eine fassadenstatusunabhängige Sitzungs-EJB, die Anforderungen an die tatsächliche InfoSphere Information Server-Serviceimplementierung umleitet, die die Geschäftslogik enthält, die ein Client aufrufen will.
- Eine Gruppe von Serialisierungsmethoden- und Deserialisierungsmethodenklassen, die SOAP-Nachrichten entsprechend den JAX-RPC-Spezifikationen Java-Objekten zuordnen (beispielsweise ESDL-Typen zu Java-Typen).

Diese Artefakte werden wie folgt gebündelt:

- Eine MDB-JAR-Datei (soapjrouter.jar), die die Router-MDB und ihre Implementierungsdeskriptoren enthält.
- Eine EJB-JAR-Datei (soapjbinding.jar), die die Fassaden-EJB, die zugehörigen Implementierungsdeskriptoren, die WSDL-Datei sowie die Serialisierungs- und Deserialisierungsmethodenklassen enthält.

Nachdem IBM InfoSphere Information Services Director vor der Implementierung eine EAR-Datei (Enterprise Archive) des Service erstellt hat, erstellt und paketiert InfoSphere Information Server bei der Serviceimplementierung die MDB- und EJB-Module in der Serviceimplementierungs-EAR-Datei (siehe folgendes Diagramm).

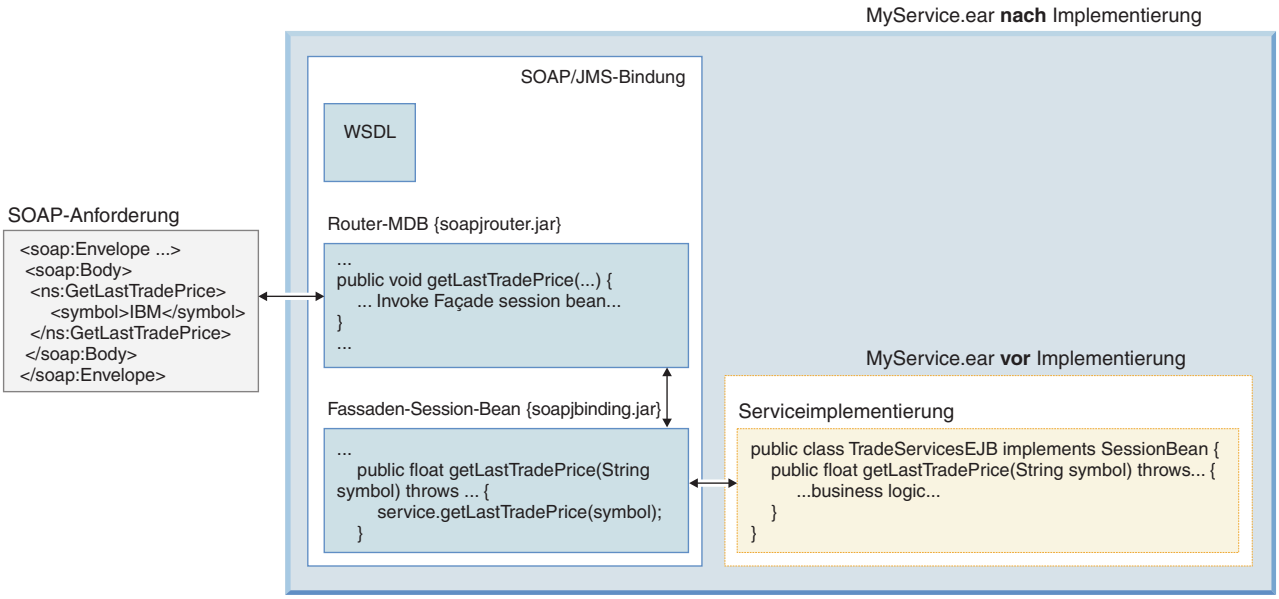

Abbildung 1. Implementierter Service mit SOAP-über-JMS-Bindung

SOAP über HTTP

SOAP über HTTP ist ein Bindungstyp, mit dem Sie eine Verbindung zu einem Service herstellen können. Lesen Sie dieses Thema, um die Basisstruktur einer SOAP-über-HTTP-Bindung zu verstehen. Dieses Wissen ist nützlich, wenn Sie Sicherheitseinstellungen ohne Zuhilfenahme einer Anwendung wie z. B. eines Assembliertools aktivieren wollen.

Die SOAP-über-HTTP-Bindung kann als eine serverseitige Proxy-Komponente betrachtet werden, die sich zwischen Clients und Services befindet. Ihre Aufgabe ist es, für SOAP-Anforderungen, die über HTTP an den Service gesendet werden, ein Unmarshaling durchzuführen, diese Anforderungen als EJB-Aufrufe an die IBM InfoSphere Information Server-Serviceimplementierung umzuleiten und für die Antwort, die in einer SOAP-Nachricht über HTTP an den Client zurückgesendet wird, ein Marshaling durchzuführen.

Folgende Artefakte bilden eine SOAP-über-HTTP-Bindung:

- Eine WSDL-Datei, die den Service beschreibt.
- Ein anwendungsserverspezifisches Router-Servlet, das für über HTTP eingehende SOAP-Anforderungen empfangsbereit ist.
- Eine fassadenstatusunabhängige Sitzungs-EJB, die Anforderungen an die tatsächliche Serviceimplementierung umleitet, die die Geschäftslogik enthält, die ein Client aufrufen will.
- Eine Gruppe von Serialisierungsmethoden- und Deserialisierungsmethodenklassen, die SOAP-Nachrichten entsprechend den JAX-RPC-Spezifikationen Java-Objekten zuordnen (beispielsweise ESDL-Typen zu Java-Typen).

Diese Artefakte werden wie folgt gebündelt:

- Ein Webmodul (soaprouter.war), das das Router-Servlet und seine Implementierungsdeskriptoren enthält.
- Eine EJB-JAR-Datei (soapbinding.jar), die die Fassaden-EJB, die zugehörigen Implementierungsdeskriptoren, die WSDL-Datei sowie die Serialisierungs- und Deserialisierungsmethodenklassen enthält.

Nachdem IBM InfoSphere Information Services Director vor der Implementierung eine EAR-Datei (Enterprise Archive) des Service erstellt hat, erstellt und paketiert InfoSphere Information Server bei der Serviceimplementierung die Web- und EJB-Module in der Serviceimplementierungs-EAR-Datei (siehe folgendes Diagramm).

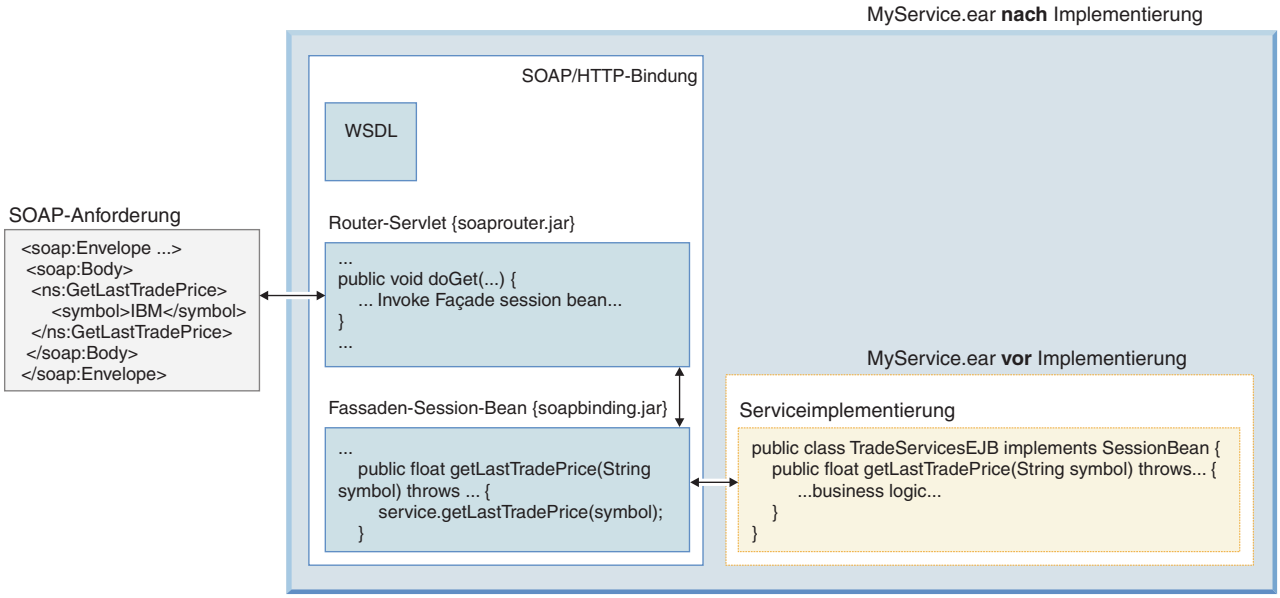

Abbildung 2. Implementierter Service mit SOAP-über-HTTP-Bindung

Assembliertool mit IBM InfoSphere Information Server konfigurieren

Zum Publizieren gesicherter Services konfigurieren Sie ein Assembliertool, mit dem Sie die Deskriptordateien für Servicebindungen modifizieren können.

Informationen zu diesem Vorgang

IBM InfoSphere Information Server Version 8.5 enthält IBM WebSphere Application Server 6.1 oder 7.0, was sich in der Konfiguration des Assembliertools widerspiegeln muss. (Unterstützte Fixpackstufen von WebSphere Application Server finden Sie auf der Seite mit den Systemanforderungen für IBM InfoSphere Information Server: http://www.ibm.com/software/data/infosphere/info-server/overview/requirements.html.) Genauer gesagt, müssen die Laufzeitbibliotheken von WebSphere Application Server 6.0 oder 7.0 für das Assembliertool verfügbar gemacht werden. Das erspart Ihnen eine Reihe von Fehlern und Warnungen beim Importieren der EAR-Datei Ihrer InfoSphere Information Server-Anwendung in das Assembliertool. Außerdem vermeiden Sie Fehler bei der Ausführung, wenn Sie beabsichtigen, Ihren Service mithilfe eines von Ihrem Assembliertool generierten Proxy-Testclients zu testen.

Die folgende Prozedur beschreibt, wie Sie IBM Rational Application Developer mit IBM WebSphere Application Server Version 6.1 oder Version 7.0 konfigurieren. Die Schritte zum Konfigurieren von IBM WebSphere Application Server Toolkit sind identisch.

Vorgehensweise

1. Klicken Sie in Rational Application Developer auf **Fenster** > **Benutzervorgaben**.
2. Wählen Sie **Server** > **Installierte Laufzeiten** in der Liste der Benutzervorgaben aus.
3. Klicken Sie im Fenster **Installierte Serverlaufzeitumgebungen** auf **Hinzufügen**, um die Laufzeitumgebungen von InfoSphere Information Server hinzuzufügen.

   Im Fenster werden IBM WebSphere Application Server-Laufzeiten von Version 6.1 bis Version 7.0 aufgelistet. Bei einigen Laufzeitbibliotheken kann es sich um Stubs handeln, andere enthalten möglicherweise den vollständigen Implementierungscode. Dies ist abhängig von der Auswahl, die Sie während des Installationsprozesses von Rational Application Developer vorgenommen haben.
4. Wählen Sie **WebSphere Application Server v6.1** oder **WebSphere Application Server v7.0** aus dem Ordner **IBM** im Fenster **Neue Serverlaufzeit** aus und klicken Sie auf **Weiter**.
5. Geben Sie `Information Server WAS v6.1` oder `Information Server WAS v7.0` und das Installationsverzeichnis (z. B. `C:\IBM\WebSphere\AppServer`) für diese Laufzeit in das Feld **Name** ein und klicken Sie auf **Fertig stellen**.
6. Klicken Sie auf **OK**, um das Fenster **Benutzervorgaben** zu verlassen.

Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren

Bevor Sie einen Service mithilfe der IBM WebSphere Application Server-Konsole stoppen und exportieren können, müssen Sie die im Service enthaltene Geschäftseinheitslogik inaktivieren (z. B. einen IBM InfoSphere DataStage-Job). Sie können den Service mithilfe der IBM InfoSphere Information Server-Konsole inaktivieren.

Vorbereitende Schritte

- Erstellen Sie einen Service.

Vorgehensweise

1. Melden Sie sich an der IBM InfoSphere Information Server-Konsole an und öffnen Sie das Projekt, das den zu inaktivierenden Service enthält.
2. Wählen Sie **Betrieb > Implementierte Informationsserviceanwendungen** aus.
3. Wählen Sie im Teilfenster **Sicht** die Anwendung aus, die den zu inaktivierenden Service enthält.
4. Klicken Sie auf **Bearbeiten**.
5. Klicken Sie unten im Teilfenster **Sicht** auf **Inaktivieren** und wählen Sie **Inaktivieren** aus.
6. Klicken Sie auf **Speichern** und **Schließen**.

Service mithilfe der WebSphere Application Server-Konsole exportieren

Bevor Sie Sicherheitsfunktionen für einen Service aktivieren können, müssen Sie die Anwendung, die den Service enthält, exportieren, um eine EAR-Datei (Enterprise Archive) zu erstellen. Die EAR-Datei enthält die Deskriptordateien, die Sie modifizieren müssen, um Sicherheitsfunktionen wie z. B. die Authentifizierung zu aktivieren.

Vorbereitende Schritte

- Service erstellen
- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" beschrieben inaktivieren

Vorgehensweise

1. Wählen Sie in WebSphere Application Server Version 6.1 **Anwendungen > Unternehmensanwendungen** aus.

   Wählen Sie in WebSphere Application Server Version 7.0 **Anwendungen** > **Anwendungstypen** > **WebSphere-Unternehmensanwendungen** aus.
2. Wählen Sie die Anwendung aus, die den zu exportierenden Service enthält, und klicken Sie auf **Stopp**.
3. Wählen Sie die Anwendung erneut aus und klicken Sie auf **Exportieren**.
4. Klicken Sie im Fenster **EAR-Dateien von Anwendungen exportieren** auf den Namen der Anwendung.
5. Speichern Sie die Datei in einem Verzeichnis auf Ihrem System.
6. Klicken Sie im Fenster **EAR-Dateien von Anwendungen exportieren** auf **Zurück**, um zum Fenster **Unternehmensanwendungen** zurückzukehren.

EAR-Datei eines nicht gesicherten Service in ein Assembliertool importieren

Nachdem Sie den nicht gesicherten Service inaktiviert und als EAR-Datei (Enterprise Archive) exportiert haben, können Sie diese EAR-Datei mithilfe eines Assembliertools importieren und Sicherheitsfunktionen, wie z. B. Authentifizierung, Autorisierung oder Vertraulichkeit, für einen Service aktivieren.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren" auf Seite 9 beschrieben konfigurieren
- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren

Informationen zu diesem Vorgang

Diese Prozedur zeigt, wie eine Service-EAR-Datei in IBM Rational Application Developer importiert wird. Die Schritte sind ähnlich, wenn Sie IBM WebSphere Application Server Toolkit zum Importieren der Service-EAR-Datei verwenden.

Vorgehensweise

1. Klicken Sie auf der Registerkarte **Projektexplorer** mit der rechten Maustaste und wählen Sie **Importieren** > **EAR-Datei** aus.
2. Klicken Sie im Fenster **Importieren** auf **Durchsuchen**, um die Speicherposition der EAR-Datei auszuwählen.
3. Wählen Sie einen Projektnamen aus.
4. Wählen Sie die Ziellaufzeitumgebung aus, die Sie während der Konfiguration des Assembliertools angegeben haben. Wenn Sie die Task zum Konfigurieren von Rational Application Developer mithilfe von InfoSphere Information Server ausgeführt haben, lautet die korrekte Laufzeitumgebung **Information Server WAS v6.1** oder **Information Server WAS v7.0**.

   Wichtig: Wenn Sie jetzt eine andere Laufzeitumgebung auswählen als während der Konfiguration des Assembliertools, wird die EAR-Datei möglicherweise nicht korrekt importiert.
5. Klicken Sie auf **Weiter**, um das nächste Fenster für **Importieren** aufzurufen.
6. Klicken Sie auf **Weiter**, um das Fenster für JAR-Projekte des EAR-Modules und des EAR-Dienstprogramms aufzurufen.
7. Klicken Sie auf **Fertig stellen**.

Ergebnisse

Die EAR-Datei wird in das Assembliertool importiert. Jetzt können Sie die EAR-Datei und die internen Module, wie z. B. soapbinding, auf der Registerkarte **Projektexplorer** anzeigen.

Anmerkung: Sie können die folgende Warnung ignorieren, falls Sie auf der Registerkarte **Fehler** angezeigt wird: **CHKJ2874W: Migrieren Sie die Standarddatenquellenbindung dieses EJB-Moduls auf eine Standardbindung für CMP-Verbindungsfactory**.

Service schützen

Es gibt vier Methoden, mit denen Sie einen Service schützen können.

Autorisierung mithilfe von J2EE-Integritätsbedingungen für die Sicherheit hinzufügen

Fügen Sie die Autorisierung als eine Möglichkeit zum Schutz eines Service hinzu.

Informationen zu diesem Vorgang

Sie können die Autorisierung hinzufügen, indem Sie die Berechtigungen für das EJB-Verfahren im EJB-Implementierungsdeskriptor definieren. Sie können den Implementierungsdeskriptor in der XML-Datei bearbeiten oder ein Assembliertool wie IBM WebSphere Application Server Toolkit verwenden.

In dem durch die J2EE-Spezifikationen definierten Sicherheitsmodell definieren Sie die Berechtigungen für das EJB-Verfahren deklarativ im Implementierungsdeskriptor und nicht etwa programmgesteuert in der Serviceimplementierung.

Sie können den Zugriff auf einen Service beispielsweise auf die Benutzer beschränken, denen die Rolle **Suite Administrator** in IBM InfoSphere Information Server zugewiesen wurde. In diesem Fall bearbeiten Sie den Implementierungsdeskriptor für die Service-Session-Bean `ejb-jar.xml`, um Sicherheitsrollen und Berechtigungen für Sicherheitsverfahren zu definieren.

J2EE-Integritätsbedingung für die Sicherheit ohne Verwendung eines Assembliertools hinzufügen:

Sie fügen einem Service, wie im Sicherheitsmodell der J2EE-Spezifikation definiert, durch Bearbeiten des EJB-Implementierungsdeskriptors Berechtigungen hinzu. Sie können den Implementierungsdeskriptor in der XML-Datei bearbeiten, wie in dieser Task beschrieben.

Vorbereitende Schritte

- Service erstellen
- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren

Vorgehensweise

1. Entpacken Sie den Inhalt der Datei *Anwendungsname*`.ear` in einem temporären Arbeitsverzeichnis.

2. Bearbeiten Sie den Implementierungsdeskriptor für die Service-Session-Bean *Servicename*.jar#META-INF/ejb-jar.xml und fügen Sie unter dem Element **<assembly-descriptor>** die folgenden Sicherheitselemente hinzu.

   security-role
   Dieses Element ist erforderlich und kann mehrmals angegeben werden. Deklariert Sicherheitsrollen, die in dieser EJB definierte Methoden aufrufen dürfen. Wenn Sie eine Rolle angeben, die von den **<method-permission>**-Elementen verwendet wird, müssen Sie sicherstellen, dass die Rolle im Element **<security-role>** deklariert wird. Das Element **<method-permission>** definiert genau, welche Rollen welche EJB-Methode aufrufen können.

   description
   Dieses Element ist ein untergeordnetes Element des Elements **<security-role>** und ist optional. Eine Textbeschreibung dieser Sicherheitsrolle.

   role-name
   Dieses Element ist ein untergeordnetes Element des Elements **<security-role>** und ist erforderlich. Ein Rollenname, dem einige Berechtigungen für den EJB-Zugriff erteilt wurden.

   method-permission
   Dieses Element ist erforderlich und kann mehrmals angegeben werden. Definiert, welche Rollen welche EJB-Methoden aufrufen können.

   role-name
   Dieses Element ist ein untergeordnetes Element des Elements **<method-permission>**. Dieses Element ist optional und kann mehrmals angegeben werden. Ein Rollenname kann die durch die Elemente **<method>** definierten Methoden aufrufen. Kann nicht mit einem Element **<unchecked>** kombiniert werden.

   unchecked
   Dieses Element ist ein untergeordnetes Element des Elements **<method-permission>** und ist optional. Gibt an, dass für den Aufruf der Methoden, die über die **<method>**-Elemente definiert wurden, kein Zugriffsberechtigung erforderlich ist. Kann nicht mit einem Element **<role-name>** kombiniert werden.

   method Dieses Element ist ein untergeordnetes Element des Elements **<method-permission>**. Dieses Element ist erforderlich und kann mehrmals angegeben werden. Definiert eine EJB-Methode, für die die Zugriffsberechtigung **<role-name>** oder **<unchecked>** gilt.

   ejb-name
   Dieses Element ist ein untergeordnetes Element des Unterelements **<method>** und ist erforderlich. Der Name der EJB, in der diese Methode definiert ist.

   method-name
   Dieses Element ist ein untergeordnetes Element des Unterelements **<method>** und ist erforderlich. Der Name der EJB-Methode.

   Anmerkung: Erforderliche Elemente müssen im Kontext der Aktivierung der Autorisierung betrachtet werden. So wird ein erforderliches Element zwar für das ejb-jar.xml-XML-Schema möglicherweise nicht benötigt, dafür aber für die Aktivierung der Autorisierung.

3. Paketieren Sie den bearbeiteten Implementierungsdeskriptor zusammen mit den übrigen unveränderten Dateien wieder in der Datei *Anwendungsname*.ear.

Beispiel

Das folgende Beispiel zeigt eine SOAP-über-HTTP-Bindung für die Dateien web.xml und ejb-jar.xml, wobei die HTTP-Basisauthentifizierung (ohne SSL-Verschlüsselung) für alle Benutzer aktiviert, die den Rollen SuiteUser und SuiteAdmin zugeordnet sind:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ejb-jar PUBLIC "-//Sun Microsystems, Inc.
//DTD Enterprise JavaBeans 2.0//EN"
"http://java.sun.com/dtd/ejb-jar_2_0.dtd">
<ejb-jar id="ejb-jar_ID">
<description/>
    <display-name>MyApp</display-name>
    <enterprise-beans>
      <session id="MyService">
         <description/>
         <display-name>MyService</display-name>
         <ejb-name>MyService</ejb-name>
         <home>com.ibm.isd.MyApp.MyService.server.MyServiceHome
         </home>

<remote>com.ibm.isd.MyApp.MyService.server.MyServiceRemote
</remote>
         <ejb-class>com.ibm.isd.MyApp.MyService.server.impl.
         MyServiceBean</ejb-class>
         <session-type>Stateless</session-type>
         <transaction-type>Container</transaction-type>
         <resource-ref id="ResRef_MyService_Ref">
           <res-ref-name>jca/AgentConnectionFactory
           </res-ref-name>
           <res-type>com.ascential.asb.agent.ra.ConnectionFactory
           </res-type>
           <res-auth>Application</res-auth>
           <res-sharing-scope>Unshareable</res-sharing-scope>
         </resource-ref>
      </session>
    </enterprise-beans>
    <assembly-descriptor>
      <security-role>
        <role-name>SuiteUser</role-name>
      </security-role>
      <method-permission>
        <role-name>SuiteUser</role-name>
        <method>
          <ejb-name>MyService</ejb-name>
          <method-name>doNothing</method-name>
          <method-params>
            <method-param>int</method-param>
          </method-params>
        </method>
      </method-permission>
    </assembly-descriptor>
</ejb-jar>
```

J2EE-Integritätsbedingung für die Sicherheit mithilfe eines Assembliertools hinzufügen:

Sie fügen einem Service, wie im Sicherheitsmodell der J2EE-Spezifikation definiert, durch Bearbeiten des EJB-Implementierungsdeskriptors Berechtigungen hinzu. Sie können den Implementierungsdeskriptor in der XML-Datei bearbeiten oder ein As-

sembliertool verwenden. Diese Task dokumentiert die Assembliertoolmethode über IBM WebSphere Application Server Server Toolkit.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren" auf Seite 9 beschrieben konfigurieren
- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren
- Service-EAR-Datei wie in „EAR-Datei eines nicht gesicherten Service in ein Assembliertool importieren" auf Seite 11 beschrieben in ein Assembliertool importieren, wenn die Sicherheitsfeatures mithilfe des Tools aktiviert werden sollen

Vorgehensweise

1. Erweitern Sie **EJB-Projekte** > **svcs** auf der Registerkarte **Projektexplorer** und klicken Sie doppelt auf die Implementierungsdeskriptordatei, um sie zu bearbeiten.
2. Klicken Sie auf **Assembly**.
3. Geben Sie im Abschnitt **Sicherheitsrollen** den Namen der hinzuzufügenden Rolle (beispielsweise SuiteUser) in das Feld **Name** ein und klicken Sie auf **Hinzufügen**.
4. Klicken Sie im Bereich **Methodenberechtigungen** auf **Hinzufügen**.
5. Wählen Sie die hinzuzufügende Sicherheitsrolle im Fenster **Methodenberechtigung hinzufügen** aus und klicken Sie auf **Fertig stellen**.
6. Klicken Sie auf **Speichern**, um Ihre Änderungen zu speichern.

Authentifizierungsverfahren hinzufügen

Fügen Sie die Authentifizierung als eine Möglichkeit zum Schutz eines Service hinzu.

Informationen zu diesem Vorgang

Die folgenden Authentifizierungsverfahren werden in dieser Task behandelt:

- Nicht signierte Sicherheitstokens
- Signierte Sicherheitstokens
- HTTP-Basisauthentifizierung

Nicht signierte Sicherheitstokens mithilfe des Assistenten für die WS-Security hinzufügen:

Eine Möglichkeit zur Sicherung eines Service ist die Verwendung eines Sicherheitstokens zur Authentifizierung ankommender Serviceanforderungen. In der Regel verweist ein Sicherheitstoken ohne Vorzeichen auf ein Benutzernamenstoken.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren" auf Seite 9 beschrieben konfigurieren

- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren
- Service-EAR-Datei wie in „EAR-Datei eines nicht gesicherten Service in ein Assembliertool importieren" auf Seite 11 beschrieben in ein Assembliertool importieren, wenn die Sicherheitsfeatures mithilfe des Tools aktiviert werden sollen

Informationen zu diesem Vorgang

Sie können diese Task mithilfe des Assistenten für die WS-Security oder über den Web-Services-Editor ausführen. Der Web-Services-Editor erfordert mehr manuelle Operationen. Gehen Sie wie folgt vor, um Ihrem Service mithilfe des Assistenten für die WS-Security ein Benutzernamenstoken hinzuzufügen.

Vorgehensweise

1. Änderungen der **Java EE-Perspektive**.
2. Erweitern Sie **soapbinding** auf der Registerkarte **Projektexplorer** für den Service, den Sie importiert haben und erweitern Sie anschließend **Services**.
3. Klicken Sie mit der rechten Maustaste auf den Service und wählen Sie **Gesicherter Web-Service** > **Eigenständiges Sicherheitstoken hinzufügen** aus.
4. Wählen Sie den Tokentyp **Benutzernamenstoken** im Fenster **Assistent für die WS-Security - Eigenständiges Sicherheitstoken auf der Serviceseite** aus.
5. Wählen Sie den JAAS-Konfigurationsnamen **system.wssecurity.UsernameToken** aus. Dieser Name gibt an, dass die ankommenden Anmeldeanforderungen von dem Stack der Anmeldemodule verarbeitet werden, die in der JAAS-Anmeldekonfiguration `system.wssecurity.UsernameToken` definiert wurden.
6. Klicken Sie auf **Fertig stellen**.
7. Erweitern Sie auf **soapbinding** > **Services** auf der Registerkarte **Projektexplorer** und klicken Sie mit der rechten Maustaste auf Ihren Service, um zu prüfen, ob die Serviceimplementierungsdeskriptoren ordnungsgemäß aktualisiert wurden. Wählen Sie **Anzeigen** aus und klicken Sie dann auf **Implementierungsdeskriptor**. Der Web-Services-Editor wird geöffnet.
8. Klicken Sie auf die Registerkarte **Erweiterungen**, um zu prüfen, ob über **Konfigurationsdetails zum Anforderungskonsumentenservice** > **Erforderliches Sicherheitstoken** ein Benutzernamenstoken erstellt wurde.
9. Klicken Sie auf die Registerkarte **Bindungen**, um zu prüfen, ob über **Konfigurationsdetails zur Anforderungskonsumentenbindung** > **Tokenkonsument** ein Benutzernamenstoken erstellt wurde.

Nicht signierte Sicherheitstokens mithilfe des Web-Services-Editors hinzufügen:

Eine Möglichkeit zur Sicherung eines Service ist die Verwendung eines Sicherheitstokens zur Authentifizierung ankommender Serviceanforderungen. In der Regel verweist ein Sicherheitstoken ohne Vorzeichen auf ein Benutzernamenstoken.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren" auf Seite 9 beschrieben konfigurieren

- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren
- Service-EAR-Datei wie in „EAR-Datei eines nicht gesicherten Service in ein Assembliertool importieren" auf Seite 11 beschrieben in ein Assembliertool importieren, wenn die Sicherheitsfeatures mithilfe des Tools aktiviert werden sollen

Informationen zu diesem Vorgang

Sie können diese Task mithilfe des Assistenten für die WS-Security oder über den Web-Services-Editor ausführen. Der Web-Services-Editor erfordert mehr manuelle Operationen. Diese Prozedur beschreibt, wie Sie Ihrem Service mithilfe des Web-Services-Editors einen Benutzernamenstoken hinzufügen.

Vorgehensweise

1. Erweitern Sie **soapbinding** > **ejbModule** > **META-INF** auf der Registerkarte **Projektexplorer** in Rational Application Developer und klicken Sie doppelt auf die Deskriptordatei **webservice.xml**.
2. Klicken Sie auf die Registerkarte **Erweiterungen** des Web-Services-Editors.
3. Erweitern Sie **Erweiterung für Web-Service-Beschreibung** und wählen Sie die Beschreibung für Ihren Service aus.
4. Erweitern Sie **Portkomponentenbindung** und wählen Sie die Bindung für Ihren Service aus.
5. Erweitern Sie **Konfigurationsdetails zum Anforderungskonsumentenservice > Erforderliches Sicherheitstoken**.
6. Klicken Sie auf **Hinzufügen**.
7. Wählen Sie im Fenster **Erforderliches Sicherheitstoken** den Tokentyp **Benutzernamenstoken** aus und füllen Sie die übrigen Felder aus.
8. Klicken Sie auf **OK**. Das Benutzernamenstoken wird unter **Konfigurationsdetails zum Anforderungskonsumentenservice** > **Erforderliches Sicherheitstoken** angezeigt.
9. Erweitern Sie **Tokenkonsument** auf der Registerkarte **Bindungskonfiguration**.
10. Klicken Sie auf **Hinzufügen**.
11. Füllen Sie im Fenster **Tokenkonsument** die Felder für den Name, die Klasse, das Sicherheitstoken, den Wertetyp und die URI aus. Stellen Sie sicher, dass Sie das neu erstellte Sicherheitstoken in der Dropdown-Liste **Sicherheitstoken** auswählen. Stellen Sie außerdem sicher, dass Sie das Kontrollkästchen **jaas.config verwenden** auswählen, und geben Sie `system.wssecurity.UsernameToken` im Feld **jaas.config-Name** an. Für die übrigen Felder können Sie die Standardwerte verwenden.
12. Klicken Sie auf **OK**. Der Tokenkonsument wird über **Konfigurationsdetails zur Anforderungskonsumentenbindung > Tokenkonsument** angezeigt.

Signierte Sicherheitstokens mithilfe des Web-Services-Editors hinzufügen:

Eine Möglichkeit zur Sicherung eines Service ist die Verwendung eines signierten Sicherheitstokens zur Authentifizierung ankommender Serviceanforderungen.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren" auf Seite 9 beschrieben konfigurieren
- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren
- Service-EAR-Datei wie in „EAR-Datei eines nicht gesicherten Service in ein Assembliertool importieren" auf Seite 11 beschrieben in ein Assembliertool importieren, wenn die Sicherheitsfeatures mithilfe des Tools aktiviert werden sollen

Informationen zu diesem Vorgang

Signierte Sicherheitstokens sind beispielsweise Kerberos-Ticket-Tokens, X.509-Zertifikattokens oder LTPA-Tokens (Lightweight Third Party Authentication). In dieser Prozedur wird IBM Rational Application verwendet, um die Definition eines X.509-Zertifikattokens als Authentifizierungsverfahren für ankommende Serviceanforderungen zu veranschaulichen.

Vorgehensweise

1. Erweitern Sie **soapbinding** > **ejbModule** > **META-INF** auf der Registerkarte **Projektexplorer** in Rational Application Developer und klicken Sie doppelt auf die Deskriptordatei `webservice.xml`.
2. Klicken Sie auf die Registerkarte **Erweiterungen** des Web-Services-Editors.
3. Erweitern Sie **Erweiterung für Web-Service-Beschreibung** und wählen Sie die Beschreibung für Ihren Service aus.
4. Erweitern Sie **Portkomponentenbindung** und wählen Sie die Bindung für Ihren Service aus.
5. Erweitern Sie **Konfigurationsdetails zum Anforderungskonsumentenservice > Erforderliches Sicherheitstoken**.
6. Klicken Sie auf **Hinzufügen**.
7. Wählen Sie im Fenster **Erforderliches Sicherheitstoken** den Tokentyp für X509-Zertifikattokens aus und füllen Sie die übrigen Felder aus.
8. Klicken Sie auf **OK**. Das X509-Zertifikattoken wird unter **Konfigurationsdetails zum Anforderungskonsumentenservice** > **Erforderliches Sicherheitstoken** angezeigt.
9. Erweitern Sie **Tokenkonsument** auf der Registerkarte **Bindungskonfiguration**.
10. Klicken Sie auf **Hinzufügen**.
11. Füllen Sie im Fenster **Tokenkonsument** die Felder für den Name, die Klasse, das Sicherheitstoken, den Wertetyp und die URI aus. Für die übrigen Felder können die Standardwerte übernommen werden.
12. Klicken Sie auf **OK**. Der Tokenkonsument wird unter **Konfigurationsdetails zur Anforderungskonsumentenbindung > Tokenkonsument** angezeigt.

HTTP-Basisauthentifizierung ohne Zuhilfenahme eines Assembliertools hinzufügen:

Sichern Sie einen Service in einem gesicherten Netz wie HTTPS oder einem Intranet mithilfe der HTTP-Basisauthentifizierung. Zum Aktivieren der HTTP-Basisauthentifizierung bearbeiten Sie die im SOAP-Bindungsmodell definierten J2EE-Implementierungsdeskriptoren.

Vorbereitende Schritte

- Service erstellen
- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren

Informationen zu diesem Vorgang

Sie bearbeiten den EJB-Implementierungsdeskriptor, um einem Service eine Authentifizierung hinzuzufügen. Sie können den Implementierungsdeskriptor in der XML-Datei bearbeiten oder ein Assembliertool wie IBM Rational Application Developer verwenden.

Vorgehensweise

1. Entpacken Sie den Inhalt Ihrer EAR-Datei in einem temporären Arbeitsverzeichnis.
2. Bearbeiten Sie den Deskriptor für die Router-Servlet-Webimplementierung `soaprouter.war#WEB-INF/web.xml` und fügen Sie die folgenden Sicherheitselemente unter dem Stammelement **<web-app>** hinzu.

   security-constraint
   Dieses Element ist erforderlich. Definiert die Berechtigungen für den Zugriff auf eine Ressourcengruppe, die vom Unterelement **<web-resource-collection>** definiert wird.

   web-resource-collection
   Dieses Element ist ein untergeordnetes Element des Elements **<security-constraint>** und ist erforderlich. Definiert die Komponenten der Webanwendung, auf die diese Integritätsbedingung für die Sicherheit angewendet wird.

   web-resource-name
   Dieses Element ist ein untergeordnetes Element des Unterelements **<web-resource-collection>** und ist erforderlich. Definiert den Namen dieser Webressourcengruppe.

   description
   Dieses Element ist ein untergeordnetes Element des Unterelements **<web-resource-collection>** und ist optional. Eine Textbeschreibung dieser Webressourcengruppe.

   url-pattern
   Dieses Element ist ein untergeordnetes Element des Unterelements **<web-resource-collection>** und ist erforderlich. Definiert, welche URL-Muster diese Integritätsbedingung für die Sicherheit anwendet. Im Fall des

SOAP-Router-Servlets sollte das URL-Muster /* verwendet werden. Der Stern (*) allein ist keine gültige Eingabe.

http-method
Dieses Element ist ein untergeordnetes Element des Unterelements **<web-resource-collection>**. Dieses Element ist optional und kann mehrmals angegeben werden. Deklarieren Sie mithilfe mindestens eines Elements **<http-method>**, welche HTTP-Methoden (beispielsweise GET oder POST) die Integritätsbedingung für die Sicherheit betrifft. Standardmäßig wird die Integritätsbedingung auf alle HTTP-Methoden angewendet. Im Fall des SOAP/HTTP-Routerservlets werden Anforderungen mithilfe der Methode POST erstellt. Wenn Sie das Element **<http-method>** angeben, müssen Sie sicherstellen, dass die Methode POST aufgelistet ist. Wenn Sie die WSDL-Datei schützen wollen, muss außerdem die Methode GET angegeben sein.

auth-constraint
Dieses Element ist ein untergeordnetes Element des Elements **<security-constraint>** und ist erforderlich. Definiert, welche Gruppen oder Principals Zugriff auf die Webressourcengruppe haben, die in dieser Integritätsbedingung für die Sicherheit definiert wurde.

description
Dieses Element ist ein untergeordnetes Element des Unterelements **<auth-constraint>** und ist optional. Eine Textbeschreibung dieser Integritätsbedingung für die Sicherheit.

role-name
Dieses Element ist ein untergeordnetes Element des Unterelements **<auth-constraint>**. Dieses Element ist erforderlich und kann mehrmals angegeben werden. Definiert, welche Sicherheitsrollen auf Ressourcen zugreifen können, die in dieser Integritätsbedingung für die Sicherheit definiert wurden (beispielsweise SuiteUser oder SuiteAdmin). Repliziert die Rollen, die für die Serviceimplementierung definiert wurden.

user-data-constraint
Dieses Element ist ein untergeordnetes Element des Elements **<security-constraint>** und ist optional. Definiert, wie der Client mit dem Server kommuniziert. Wird im Allgemeinen nur für die SSL-Aktivierung verwendet.

description
Dieses Element ist ein untergeordnetes Element des Unterelements **<user-data-constraint>** und ist optional. Eine Textbeschreibung dieser Integritätsbedingung für Benutzerdaten.

transport-guarantee
Dieses Element ist ein untergeordnetes Element des Unterelements **<user-data-constraint>** und ist optional. Gibt den Typ der Integritätsbedingung für die Sicher-

heit an, der für Daten erzwungen werden soll, die zwischen Client und Server übertragen werden:

- NONE (keine Integritätsbedingung)
- INTEGRAL (Datenintegrität)
- CONFIDENTIAL (im Wesentlichen SSL)

login-config
Dieses Element ist erforderlich. Definiert, wie ein Benutzer authentifiziert wird.

auth-method
Dieses Element ist ein untergeordnetes Element des Elements **<login-config>** und ist erforderlich. Gibt die Methode an, mit der ein Benutzer authentifiziert wird. In diesem Fall nur auf BASIC gesetzt.

realm-name
Dieses Element ist ein untergeordnetes Element des Elements **<login-config>** und ist optional. Name des Realm, auf den verwiesen wird, um Benutzerberechtigungen zu authentifizieren.

security-role
Dieses Element ist erforderlich und kann mehrmals angegeben werden. Deklariert Sicherheitsrollen. Für jede unter dem Element **<auth-constraint>** aufgelistete Rolle muss ein Element **<security-role>** vorhanden sein.

description
Dieses Element ist ein untergeordnetes Element des Elements **<security-role>** und ist optional. Eine Textbeschreibung dieser Sicherheitsrolle.

role-name
Dieses Element ist ein untergeordnetes Element des Elements **<security-role>** und ist erforderlich. Name einer Rolle, der in diesem Webmodul Berechtigungen erteilt wurden.

Anmerkung: Erforderliche Elemente müssen im Kontext der Aktivierung der HTTP-Basisauthentifizierung betrachtet werden. Ein erforderliches Element wird möglicherweise zwar nicht für das `web.xml`-XML-Schema, aber für die Aktivierung der HTTP-Basisauthentifizierung benötigt.

3. Bearbeiten Sie den Implementierungsdeskriptor für die EJB-Bindung `soapbinding.jar#META-INF/ejb-jar.xml` und fügen Sie unter dem Element **<assembly-descriptor>** die folgenden Sicherheitselemente hinzu.

security-role
Dieses Element ist erforderlich und kann mehrmals angegeben werden. Deklariert Sicherheitsrollen, die in dieser EJB definierte Methoden aufrufen dürfen. Wenn Sie eine Rolle angeben, die von den **<method-permission>**-Elementen verwendet wird, müssen Sie sicherstellen, dass die Rolle im Element **<security-role>** deklariert wird. Das Element **<method-permission>** definiert genau, welche Rollen welche EJB-Methode aufrufen können.

description
Dieses Element ist ein untergeordnetes Element des Elements **<security-role>** und ist optional. Eine Textbeschreibung dieser Sicherheitsrolle.

role-name
Dieses Element ist ein untergeordnetes Element des Elements **<security-role>** und ist erforderlich. Ein Rollenname, dem einige Berechtigungen für den EJB-Zugriff erteilt wurden.

method-permission
Dieses Element ist erforderlich und kann mehrmals angegeben werden. Definiert, welche Rollen welche EJB-Methoden aufrufen können.

role-name
Dieses Element ist ein untergeordnetes Element des Elements **<method-permission>**. Dieses Element ist optional und kann mehrmals angegeben werden. Name einer Rolle, die die durch die Elemente **<method>** definierten Methoden aufrufen kann. Kann nicht mit einem Element **<unchecked>** kombiniert werden.

unchecked
Dieses Element ist ein untergeordnetes Element des Elements **<method-permission>** und ist optional. Gibt an, dass für den Aufruf der Methoden, die über die **<method>**-Elemente definiert wurden, kein Zugriffsberechtigung erforderlich ist. Kann nicht mit einem Element **<role-name>** kombiniert werden.

method Dieses Element ist ein untergeordnetes Element des Elements **<method-permission>**. Dieses Element ist erforderlich und kann mehrmals angegeben werden. Definiert eine EJB-Methode, für die die Zugriffsberechtigung **<role-name>** oder **<unchecked>** gilt.

ejb-name
Dieses Element ist ein untergeordnetes Element des Unterelements **<method>** und ist erforderlich. Der Name der EJB, in der diese Methode definiert ist.

method-name
Dieses Element ist ein untergeordnetes Element des Unterelements **<method>** und ist erforderlich. Der Name der EJB-Methode.

Anmerkung: Erforderliche Elemente müssen im Kontext der Aktivierung der HTTP-Basisauthentifizierung betrachtet werden. Ein erforderliches Element wird möglicherweise zwar nicht für das `web.xml`-XML-Schema, aber für die Aktivierung der HTTP-Basisauthentifizierung benötigt.

4. Paketieren Sie die bearbeiteten Implementierungsdeskriptoren zusammen mit den übrigen unveränderten Dateien wieder in der Datei *Anwendungsname*`.ear`.

Beispiel

Das folgende Beispiel zeigt eine SOAP-über-HTTP-Bindung für die Dateien `web.xml` und `ejb-jar.xml`, wobei die HTTP-Basisauthentifizierung (ohne SSL-Verschlüsselung) für alle Benutzer aktiviert, die den Rollen SuiteUser und SuiteAdmin zugeordnet sind:

Beispiel für die Datei `web.xml`:

```
soaprouter.jar#WEB-INF/web.xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd" id="WebApp_ID" version="2.4">
  <display-name>MyAppSOAPBinding_HTTPRouter</display-name>
```

```
  <servlet>
    <display-name>Web services Router servlet for port-componentMyServiceSOAP
    </display-name>
    <servlet-name>MyServiceSOAP</servlet-name>
    <servlet-class>com.ibm.ws.webservices.engine.transport.http.WebServicesServlet
    </servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>MyServiceSOAP</servlet-name>
    <url-pattern>MyService</url-pattern>
  </servlet-mapping>
  <security-constraint>
    <display-name>SoapRouterConstraints</display-name>
    <web-resource-collection>
      <web-resource-name>SecuredSoapRouterServlet</web-resource-name>
      <url-pattern>/*</url-pattern>
      <http-method>GET</http-method>
      <http-method>POST</http-method>
    </web-resource-collection>
    <auth-constraint>
      <description>Authorized roles</description>
      <role-name>SuiteUser</role-name>
      <role-name>SuiteAdmin</role-name>
    </auth-constraint>
    <user-data-constraint>
      <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
  </security-constraint>
  <login-config>
    <auth-method>BASIC</auth-method>
  </login-config>
  <security-role>
    <role-name>SuiteUser</role-name>
  </security-role>
  <security-role>
    <role-name>SuiteAdmin</role-name>
  </security-role>
</web-app>
```

Beispiel für die Datei ejb-jar.xml:

```
soapbinding.jar#META-INF/ejb-jar.xml
<?xml version="1.0" encoding="UTF-8"?>
<ejb-jar xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" id="ejb-jar_ID"
version="2.1" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/ejb-jar_2_1.xsd">
  <enterprise-beans>
    <session id="MyServiceSOAP">
      <description></description>
      <ejb-name>MyServiceSOAP</ejb-name>
      <home>com.ibm.isd.MyApp.MyService.server.MyServiceHome
      </home>
      <remote>com.ibm.isd.MyApp.MyService.server.MyServiceRemote
      </remote>
      <service-endpoint>com.ibm.isd.MyApp.MyService.MyService
      </service-endpoint>
      <ejb-class>com.ibm.isd.MyApp.MyService.MyServiceSOAPBindingBean
      </ejb-class>
      <session-type>Stateless</session-type>
      <transaction-type>Container</transaction-type>
      <ejb-ref id="EjbRef_MyService_Ref">
        <ejb-ref-name>ejb/MyService</ejb-ref-name>
        <ejb-ref-type>Session</ejb-ref-type>
        <home>com.ibm.isd.MyApp.MyService.server.MyServiceHome</home>
        <remote>com.ibm.isd.MyApp.MyService.server.MyServiceRemote</remote>
      </ejb-ref>
```

```
  </session>
</enterprise-beans>
  <assembly-descriptor>
    <security-role>
      <role-name>SuiteUser</role-name>
    </security-role>
    <security-role>
      <role-name>SuiteAdmin</role-name>
    </security-role>
    <method-permission>
      <role-name>SuiteUser</role-name>
      <role-name>SuiteAdmin</role-name>
      <method>
        <ejb-name>MyServiceSOAP</ejb-name>
        <method-intf>Remote</method-intf>
        <method-name>doNothing</method-name>
        <method-params>
        <method-param>int</method-param>
        </method-params>
      </method>
    </method-permission>
    <method-permission>
      <unchecked/>
      <method>
        <ejb-name>MyServiceSOAP</ejb-name>
        <method-intf>Home</method-intf>
        <method-name>create</method-name>
        <method-params>
        </method-params>
      </method>
      <method>
        <ejb-name>MyServiceSOAP</ejb-name>
        <method-intf>Home</method-intf>
        <method-name>remove</method-name>
      </method>
    </method-permission></assembly-descriptor>
  </assembly-descriptor>
</ejb-jar>
```

HTTP-Basisauthentifizierung mithilfe eines Assembliertools hinzufügen:

Sichern Sie einen Service in einem gesicherten Netz wie HTTPS oder einem Intranet mithilfe der HTTP-Basisauthentifizierung. Zum Aktivieren der HTTP-Basisauthentifizierung bearbeiten Sie die im SOAP-Bindungsmodell definierten J2EE-Implementierungsdeskriptoren.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren" auf Seite 9 beschrieben konfigurieren
- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren
- Service-EAR-Datei wie in „EAR-Datei eines nicht gesicherten Service in ein Assembliertool importieren" auf Seite 11 beschrieben in ein Assembliertool importieren, wenn die Sicherheitsfeatures mithilfe des Tools aktiviert werden sollen

Informationen zu diesem Vorgang

Sie bearbeiten den EJB-Implementierungsdeskriptor, um einem Service eine Authentifizierung hinzuzufügen. Sie können den Implementierungsdeskriptor in der XML-Datei bearbeiten. Alternativ können Sie auch ein Assembliertool wie IBM Rational Application Developer verwenden, wie in dieser Prozedur beschrieben.

Vorgehensweise

1. Erweitern Sie das Modul **soaprouter** auf der Registerkarte **Projektexplorer**.
2. Klicken Sie doppelt auf die Deskriptordatei für die Webimplementierung, um sie zu bearbeiten.
3. Klicken Sie auf die Registerkarte **Seiten**.
4. Wählen Sie **Authentifizierungsmethode** > **BASIC** im Abschnitt **Anmeldung** aus und speichern Sie die Änderung.
5. Klicken Sie auf die Registerkarte **Sicherheit**.
6. Geben Sie im Abschnitt **Sicherheitsrollen** den Namen der hinzuzufügenden Rolle in das Feld **Name** ein und klicken Sie auf **Hinzufügen**.
7. Klicken Sie im Abschnitt **Integritätsbedingungen für die Sicherheit** auf **Hinzufügen**.

   Legen Sie hier dieselben Integritätsbedingungen für die Sicherheit fest, die Sie auch definieren, wenn Sie dem Service die Autorisierung hinzufügen.
8. Geben Sie einen Namen für die Integritätsbedingung ein und klicken Sie auf **Weiter**.
9. Geben Sie einen Webressourcennamen ein, wählen Sie **POST** und **GET** für **HTTP-Methoden** aus und fügen Sie `/*` dem Feld **Muster** hinzu. Wenn Sie die Service-WSDL (Web Services Description Language) schützen wollen, müssen Sie die HTTP-Methode GET schützen.
10. Klicken Sie auf **Fertig stellen**.
11. Klicken Sie im Abschnitt **Autorisierte Berechtigungsklassen** auf **Hinzufügen**.
12. Wählen Sie für **Rollenname** den Namen aus, den Sie im Abschnitt **Sicherheitsrollen** hinzugefügt haben, und klicken Sie auf **Fertig stellen**.
13. Wählen Sie **NONE** im Abschnitt **Benutzerdateneinschränkung** aus und speichern Sie die Änderungen. Diese Integritätsbedingung wird nur für SSL-Verbindungen verwendet.
14. Erweitern Sie das Modul **soapbinding** im Abschnitt **EJB-Projekte** der Registerkarte **Projektexplorer**.
15. Klicken Sie doppelt auf die Deskriptordatei für die EJB-Implementierung, um sie zu bearbeiten.
16. Klicken Sie auf die Registerkarte **Assembly**.
17. Klicken Sie im Abschnitt **Sicherheitsrollen** auf **Hinzufügen** und geben Sie denselben Rollennamen ein, den Sie auch dem Deskriptor für die Webimplementierung hinzugefügt haben.
18. Klicken Sie im Bereich **Methodenberechtigungen** auf **Hinzufügen**.
19. Wählen Sie die Sicherheitsrolle im Fenster **Method Permission** aus und klicken Sie auf **Weiter**.
20. Wählen Sie die Enterprise-Bean für Ihren Service im Fenster **Enterprise-Bean-Auswahl** aus und klicken Sie auf **Weiter**.
21. Wählen Sie die zu schützenden Methoden im Fenster **Methodenelemente** aus. In einer typischen Situation wählen Sie alle Methoden aus.
22. Klicken Sie auf **Fertig stellen** und speichern Sie die Änderungen.

Datenschutz mithilfe der SSL-Verschlüsselung hinzufügen

Fügen Sie Datenschutz als eine Möglichkeit zum Schutz eines Service hinzu.

Informationen zu diesem Vorgang

Fügen Sie Datenschutz mithilfe des SSL-Protokolls (Secure Sockets Layer) hinzu. Führen Sie diese Task aus, um SSL auf der Serverseite des Service zu aktivieren. Sie müssen dieselben Änderungen anschließend an der Clientkonfiguration vornehmen. Wenn Sie beispielsweise die Serverkonfiguration so ändern, dass anstelle von SSL das Protokoll TSL (Transport Secure Layer) verwenden wird, müssen Sie den Client ebenfalls für die Verwendung von TSL konfigurieren.

Das SSL-Protokoll basiert auf der PKI-Infrastruktur (Public Key Infrastructure). In dieser Funktion müssen private und öffentliche Schlüssel für den Client und für den Server jeweils in Keystores bzw. Truststores generiert und gespeichert werden, damit die SSL-Verbindung vollständig hergestellt werden kann. In dieser Task werden Beispielkeystores und selbst signierte Zertifikate verwendet. Verwenden Sie Beispielkeystores ausschließlich für Testzwecke. Implementieren Sie in einer echten Produktionsumgebung Ihre eigene PKI-Infrastruktur.

Es folgt eine Übersicht über den Prozess zur SSL-Aktivierung:

- Der Client authentifiziert den Server, woraufhin eine Sitzung initialisiert wird. Diese Authentifizierung wird als SSL-Handshakeprotokoll bezeichnet. Der Client kann sich optional auch beim Server authentifizieren.
- Wenn der SSL-Handshakevorgang erfolgreich ausgeführt wurde, wird anschließend die Datenübertragung initialisiert und verschlüsselt. Dies wird als SSL-Berichtsprotokoll bezeichnet.
- Wenn an einem beliebigen Punkt während der Sitzung ein Alarm auftritt, wird ein Alertpaket an den entsprechenden Empfänger gesendet. Dieses Alertpaket wird als SSL-Alertprotokoll bezeichnet.

In der folgenden Task wird beschrieben, wie Datenschutz mithilfe der SSL-Verschlüsselung hinzugefügt wird:

Service für die Verwendung der SSL-Verschlüsselung konfigurieren:

Nachdem Sie ein SSL-Repertoire (Secure Sockets Layer) in IBM WebSphere Application Server erstellt haben, führen Sie diese Task aus, um SSL auf der Serverseite des Service zu aktivieren. Sie müssen dieselben Änderungen anschließend an der Clientkonfiguration vornehmen. Wenn Sie beispielsweise die Serverkonfiguration so ändern, dass anstelle von SSL das Protokoll TSL (Transport Secure Layer) verwenden wird, müssen Sie den Client ebenfalls für die Verwendung von TSL konfigurieren.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren" auf Seite 9 beschrieben konfigurieren
- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren

- Service-EAR-Datei wie in „EAR-Datei eines nicht gesicherten Service in ein Assembliertool importieren" auf Seite 11 beschrieben in ein Assembliertool importieren, wenn die Sicherheitsfeatures mithilfe des Tools aktiviert werden sollen

Vorgehensweise

1. Erweitern Sie **soaprouter** auf der Registerkarte **Projektexplorer** von IBM Rational Application Developer und klicken Sie zum Bearbeiten doppelt auf den Implementierungsdeskriptor.
2. Wählen Sie die Registerkarte **Sicherheit** aus.
3. Klicken Sie im Abschnitt **Integritätsbedingungen für die Sicherheit** auf **Hinzufügen**.
4. Geben Sie auf der Seite **Integritätsbedingungen hinzufügen** einen Namen für die Integritätsbedingung ein und klicken Sie auf **Weiter**.
5. Geben Sie auf der Seite **Webressource hinzufügen** einen Namen für die Ressource ein.
6. Wählen Sie eine HTTP-Methode in der Liste **HTTP-Methoden** aus. Wählen Sie mindestens die Methode POST aus, um Ihren Service zu schützen. Wenn die Integritätsbedingung für die Sicherheit nicht auf die WSDL-Datei (Web Services Description Language) angewendet werden soll, wählen Sie die Methode GET nicht aus.
7. Fügen Sie `/*` dem Feld **Muster** hinzu, um das URL-Muster anzugeben, das auf die Integritätsbedingung für die Sicherheit angewendet wird.
8. Klicken Sie auf **Fertig stellen**.
9. Wählen Sie **CONFIDENTIAL** im Feld **Typ** des Abschnitts **Integritätsbedingung für Benutzerdaten** aus.
10. Speichern Sie die Änderungen.

Nächste Schritte

Datenintegrität hinzufügen

Fügen Sie Datenintegrität mithilfe einer digitalen XML-Signatur als eine Möglichkeit zum Schutz eines Service hinzu.

Informationen zu diesem Vorgang

Übersicht über den Prozess der digitalen Unterzeichnung:

- Der Client (Sender) unterzeichnet Teile der Anforderungsnachricht mithilfe eines privaten Schlüssels. Der private Schlüssel des Clients wird im Client-Keystore gespeichert.
- Der Server (Empfänger) prüft die digitale Signatur des Clients in der Anforderungsnachricht mithilfe des öffentlichen Schlüssels des Clients. Der öffentliche Schlüssel des Clients wird im Server-Keystore gespeichert.

Digitale XML-Signatur hinzufügen:

Wenn Sie Ihrem Service Datenintegrität mithilfe einer digitalen XML-Signatur hinzufügen wollen, bearbeiten Sie die Implementierungsdeskriptordatei über ein Assembliertool wie IBM Rational Application Developer.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren" auf Seite 9 beschrieben konfigurieren

- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren
- Service-EAR-Datei wie in „EAR-Datei eines nicht gesicherten Service in ein Assembliertool importieren" auf Seite 11 beschrieben in ein Assembliertool importieren, wenn die Sicherheitsfeatures mithilfe des Tools aktiviert werden sollen

Informationen zu diesem Vorgang

Einschränkungen

In dieser Task bearbeiten Sie die Implementierungsdeskriptordatei mithilfe des Web-Services-Editors in IBM Rational Application Developer. Verwenden Sie zur Bearbeitung der Implementierungsdeskriptordatei nicht den Sicherheitsassistenten von Rational Application Developer. Der Assistent ermöglicht nicht die Steuerung der Funktionen für digitale XML-Signaturen, wie z. B. das Unterzeichnen und das Umsetzen von Algorithmen.

Diese Task zeigt, wie digitale Signaturen für Anforderungsnachrichten, nicht für Antwortnachrichten, unterzeichnet und geprüft werden. Die Prozedur für das Unterzeichnen und Prüfen digitaler Signaturen für Antwortnachrichten ist jedoch praktisch identisch und wird daher hier nicht dokumentiert. Zusammengefasst heißt dies, dass Sie für diese Task den Antwortkonsumenten (Clientseite) und den Antwortgenerator (Serverseite) in gleicher Weise konfigurieren, wie der Anforderungsgenerator (Clientseite) und der Anforderungskonsument (Serverseite) in diesem Abschnitt konfiguriert werden.

Vorgehensweise

1. Erweitern Sie **soapbinding** > **Services** auf der Registerkarte **Projektexplorer** von Rational Application Developer und klicken Sie mit der rechten Maustaste auf Ihren Service. Wählen Sie **Anzeigen** aus und klicken Sie dann auf **Implementierungsdeskriptor**. Der Web-Services-Editor wird geöffnet.
2. Klicken Sie im Web-Services-Editor auf die Registerkarte **Erweiterungen**.
3. Erweitern Sie **Erweiterung für Web-Service-Beschreibung** und wählen Sie die Beschreibung für Ihren Service aus.
4. Erweitern Sie **Portkomponentenbindung** und wählen Sie die Bindung für Ihren Service aus.
5. Erweitern Sie **Konfigurationsdetails zum Anforderungskonsumentenservice > Erforderliche Integrität**.
6. Klicken Sie im Fenster **Erforderliche Integrität** auf **Hinzufügen**.
7. Geben Sie einen Namen in das Feld **Name der erforderlichen Integrität** im Fenster **Erforderliche Vertraulichkeit** ein.
8. Wählen Sie **Erforderlich** im Feld **Verwendungstyp** aus, wenn die Integrität erforderlich sein soll. Soll die Integrität optional sein, wählen Sie **Optional** aus.
9. Wählen Sie im Feld **Nachrichtenabschnitte** die Nachrichtenabschnitte aus, die unterzeichnet werden sollen. Zum Unterzeichnen der Nachrichtenabschnitte können Sie Schlüsselwörter oder XPath-Ausdrücke verwenden. Gehen Sie wie folgt vor, wenn Sie Nachrichtenabschnitte hinzufügen wollen:
   a. Klicken Sie auf **Hinzufügen**.

b. Mithilfe von Schlüsselwörtern können Sie generische Abschnitte Ihrer Nachricht angeben, **http://www.ibm.com/websphere/webservices/wssecurity/dialect-was** in der Spalte **Abschnittsdialekt** auswählen und den zu unterzeichnenden Abschnitt der Nachricht in der Spalte **Abschnittskennwort** auswählen.

c. Mithilfe von XPath-Ausdrücken können Sie genauere Abschnitte Ihrer Nachricht angeben und **http://www.w3.org/TR/1999/REC-xpath-19991116** in der Spalte **Abschnittsdialekt** auswählen.

d. Optional: Geben Sie im Feld **Nachrichtenabschnitte** eine generierte Zufallszahl (Nonce) an, um Antwortattacken zu verhindern.

e. Optional: Geben Sie im Feld **Nachrichtenabschnitte** eine Zeitmarke an, um der Nachricht eine Laufzeit zuzuweisen.

10. Klicken Sie im Web-Services-Editor auf die Registerkarte **Binding-Konfigurationen**.
11. Erweitern Sie **Konfigurationsdetails zur Anforderungskonsumentenbindung** > **Trust-Anchor** und klicken Sie auf **Hinzufügen**.
12. Im Fenster **Trust Anchor**:
    a. Geben Sie einen Trust-Anchor-Namen ein.
    b. Geben Sie ein Kennwort für den Keystore ein.
    c. Geben Sie den vollständigen Pfad zum serverseitigen Keystore ein, wie z. B. `C:\sampleks\receiverkeys.jks`.
    d. Wählen Sie den Keystoretyp aus.
    e. Klicken Sie auf **OK**.
13. Erweitern Sie **Konfigurationsdetails zur Anforderungskonsumentenbindung** > **Tokenkonsument** im Web-Services-Editor und klicken Sie auf **Hinzufügen**.
14. Im Fenster **Tokenkonsument**:
    a. Geben Sie einen Tokenkonsumentennamen ein.
    b. Wählen Sie **com.ibm.wssip.wssecurity.token.X509TokenConsumer** als Tokenkonsumentenklasse aus.
    c. Wählen Sie kein Sicherheitstoken aus. Für das Unterzeichnen ist es nicht erforderlich, ein Sicherheitstoken in den Serviceerweiterungen anzugeben.
    d. Wählen Sie zuerst **Wertetyp verwenden** und dann den Wertetyp **X509 certificate token v3** aus.
    e. Wählen Sie **jaas.config verwenden** aus und geben Sie `system.wssecurity.X509BST` in das Feld **Name von jaas.config** ein.
    f. Wählen Sie **Zertifikatspfadeinstellungen verwenden** und **Referenz des Zertifikatspfads** oder **Jedes Zertifikat anerkennen** aus. Wenn Sie **Zertifikatspfadeinstellungen verwenden** auswählen, wählen Sie den Trust-Anchor-Namen aus, den Sie in einem der früheren Schritte erstellt haben.
    g. Klicken Sie auf **OK**.
15. Erweitern Sie **Konfigurationsdetails zur Anforderungskonsumentenbindung** > **Key-Locator** im Web-Services-Editor und klicken Sie auf **Hinzufügen**.
16. Im Fenster **Key-Locator**:
    a. Geben Sie den Namen eines Key-Locators ein.
    b. Wählen Sie eine Key-Locator-Klassenimplementierung aus. Verwenden Sie **com.ibm.wsspi.wssecurity.keyinfo.X509TokenKeyLocator**, wenn Sie zum Prüfen digitaler Signaturen das X.509-Sicherheitstoken aus der Sendernachricht nutzen.
    c. Klicken Sie auf **OK**.

17. Erweitern Sie **Konfigurationsdetails zur Anforderungskonsumentenbindung > Schlüsseldaten** im Web-Services-Editor und klicken Sie auf **Hinzufügen**.
18. Im Fenster **Schlüsseldaten**:
    a. Geben Sie einen Schlüsseldatennamen ein.
    b. Wählen Sie einen Schlüsseldatentyp aus. Verwenden Sie **STRREF** als Schlüsseldatentyp, wenn mithilfe einer URI in der Sendernachricht direkt auf das Sicherheitstoken verwiesen wird, mit dem die digitale Signatur geprüft wird.
    c. Wählen Sie eine Schlüsseldatenklasse aus. Wenn Sie **STRREF** als Schlüsseldatentyp ausgewählt haben, wird **com.ibm.ws.webservices.wssecurity.keyinfo.STRReferenceContentConsumer** automatisch ausgewählt.
    d. Wählen Sie zuerst **Key-Locator verwenden** und dann den Namen des Key-Locators aus, den Sie im Abschnitt zum Key-Locator erstellt haben.
    e. Wählen Sie zuerst **Token verwenden** und dann den Namen des Tokenkonsumenten aus, den Sie im Abschnitt zum Tokenkonsumenten erstellt haben.
    f. Klicken Sie auf **OK**.
19. Erweitern Sie **Konfigurationsdetails zur Anforderungskonsumentenbindung > Signaturdaten** im Web-Services-Editor und klicken Sie auf **Hinzufügen**.
20. Im Fenster **Signaturdaten**:
    a. Geben Sie einen Signaturdatennamen ein.
    b. Wählen Sie einen Algorithmus für die Kanonisierungsmethode aus. Der Standardwert ist **http://www.w3.org/2001/10/xml-exc-c14n#**.
    c. Wählen Sie einen Algorithmus für die Signaturmethode aus. Der Standardwert ist **http://www.w3.org/2000/09/xmldsig#rsa-sha1**.
    d. Klicken Sie auf **Hinzufügen**, wählen Sie das Schlüsseldatenelement aus, das Sie im Abschnitt zu den Schlüsseldaten erstellt haben, und geben Sie einen Schlüsseldatennamen ein.
    e. Klicken Sie auf **OK**.
21. Erweitern Sie **Konfigurationsdetails zur Anforderungskonsumentenbindung > Abschnittsreferenz** im Web-Services-Editor und klicken Sie auf **Hinzufügen**.
22. Im Fenster **Abschnittsreferenz**:
    a. Geben Sie einen Abschnittsreferenznamen ein.
    b. Wählen Sie einen erforderlichen Integritätsabschnitt aus, den Sie im Abschnitt zur erforderlichen Integrität erstellt haben.
    c. Wählen Sie einen Algorithmus für die Digestmethode aus. Der Standardwert ist **http://www.w3.org/2000/09/xmldsig#sha1**.
    d. Klicken Sie auf **OK**.
23. Erweitern Sie **Konfigurationsdetails zur Anforderungskonsumentenbindung > Umsetzungen** im Web-Services-Editor und klicken Sie auf **Hinzufügen**. Der Abschnitt **Umsetzungen** wird verfügbar, nachdem Sie eine Abschnittsreferenz erstellt haben. Umsetzungselemente enthalten Informationen zu den Operationen, die vor dem Signieren für Daten ausgeführt wurden.
24. Im Fenster **Umsetzung**:
    a. Geben sie einen Umsetzungsnamen ein.
    b. Wählen Sie einen Algorithmus aus, um die Operation auszuführen. Der Standardwert ist **http://www.w3.org/2002/10/xml-exc-c14n#**.
    c. Klicken Sie auf **OK**.
25. Speichern Sie Ihre Änderungen.

EAR-Datei eines gesicherten Service aus einem Assembliertool exportieren

Nachdem Sie für einen Service mithilfe eines Assembliertools Sicherheitsfunktionen wie z. B. Authentifizierung, Autorisierung oder Vertraulichkeit aktiviert haben, können Sie den gesicherten Service als EAR-Datei (Enterprise Archive) exportieren.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren" auf Seite 9 beschrieben konfigurieren
- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren
- Service-EAR-Datei wie in „EAR-Datei eines nicht gesicherten Service in ein Assembliertool importieren" auf Seite 11 beschrieben in ein Assembliertool importieren, wenn die Sicherheitsfeatures mithilfe des Tools aktiviert werden sollen
- Service wie in „Service schützen" auf Seite 12 beschrieben schützen

Informationen zu diesem Vorgang

Diese Prozedur zeigt, wie eine Service-EAR-Datei aus IBM Rational Application Developer exportiert wird. Die Schritte sind ähnlich, wenn Sie IBM WebSphere Application Server Toolkit zum Exportieren der Service-EAR-Datei verwenden.

Vorgehensweise

1. Klicken Sie auf der Registerkarte **Projektexplorer** mit der rechten Maustaste auf die EAR-Datei und wählen Sie **Exportieren** > **EAR-Datei** aus.
2. Klicken Sie im Fenster **Exportieren** auf **Durchsuchen**, um die Speicherposition der EAR-Datei auszuwählen.
3. Wählen Sie das Kontrollkästchen zum Überschreiben vorhandener Dateien aus, um die nicht gesicherte Service-EAR-Datei zu überschreiben.
4. Klicken Sie auf **Fertig stellen**.

Gesicherten Service mit der IBM WebSphere Application Server-Konsole erneut implementieren

Nachdem Sie für einen Service mithilfe eines Assembliertools Sicherheitsfunktionen wie z. B. Authentifizierung, Autorisierung oder Vertraulichkeit aktiviert und den gesicherten Service als EAR-Datei (Enterprise Archive) exportiert haben, implementieren Sie diesen gesicherten Service mithilfe der IBM WebSphere Application Server-Konsole erneut.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren" auf Seite 9 beschrieben konfigurieren
- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren

- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren
- Service-EAR-Datei wie in „EAR-Datei eines nicht gesicherten Service in ein Assembliertool importieren" auf Seite 11 beschrieben in ein Assembliertool importieren, wenn die Sicherheitsfeatures mithilfe des Tools aktiviert werden sollen
- Service wie in „Service schützen" auf Seite 12 beschrieben schützen
- EAR-Datei, die den Service enthält, wie in „EAR-Datei eines gesicherten Service aus einem Assembliertool exportieren" auf Seite 31 beschrieben exportieren

Vorgehensweise

1. Wählen Sie **Anwendungen > Unternehmensanwendungen** aus.
2. Wenn Ihre Anwendung nicht gestoppt wurde (angezeigt durch ein rotes X in der Statusspalte), wählen Sie die Anwendung aus und klicken auf **Stoppen**.
3. Wählen Sie die Anwendung aus und klicken Sie auf **Deinstallieren**.
4. Klicken Sie im Fenster **Anwendung deinstallieren** auf **OK**.
5. Klicken Sie auf **Speichern**, um die Änderungen auf die Hauptkonfiguration anzuwenden, und klicken Sie in dem daraufhin angezeigten Fenster auf **Speichern**.
6. Wählen Sie die Anwendung im Fenster **Unternehmensanwendungen** aus und klicken Sie auf **Installieren**.
7. Klicken Sie im Fenster **Vorbereitung der Anwendungsinstallation** auf **Durchsuchen**, um die Speicherposition der EAR-Datei des gesicherten Service auszuwählen, und klicken Sie auf **Weiter**.
8. Klicken Sie so oft auf **Weiter**, bis die Anwendungszusammenfassung angezeigt wird.
9. Klicken Sie auf **Fertig stellen**. Es werden Nachrichten angezeigt, die den Fortschritt der Installation angeben.
10. Klicken Sie zuerst auf **In Masterkonfiguration speichern** und in dem daraufhin angezeigten Fenster auf **Speichern**.
11. Wählen Sie im Fenster **Unternehmensanwendung** die neu installierte Anwendung aus und klicken Sie auf **Starten**.

Gesicherten Service über die IBM InfoSphere Information Server-Konsole aktivieren

Nachdem der gesicherte Service über die IBM WebSphere Application Server-Konsole erneut implementiert wurde, aktivieren Sie den Service über die IBM InfoSphere Information Server-Konsole.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren" auf Seite 9 beschrieben konfigurieren
- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren

- Service-EAR-Datei wie in „EAR-Datei eines nicht gesicherten Service in ein Assembliertool importieren" auf Seite 11 beschrieben in ein Assembliertool importieren, wenn die Sicherheitsfeatures mithilfe des Tools aktiviert werden sollen
- Service wie in „Service schützen" auf Seite 12 beschrieben schützen
- EAR-Datei, die den Service enthält, wie in „EAR-Datei eines gesicherten Service aus einem Assembliertool exportieren" auf Seite 31 beschrieben exportieren
- Service-EAR-Datei wie in „Gesicherten Service mit der IBM WebSphere Application Server-Konsole erneut implementieren" auf Seite 31 beschrieben erneut implementieren

Vorgehensweise

1. Melden Sie sich an der IBM InfoSphere Information Server-Konsole an und öffnen Sie ein Projekt, das den zu aktivierenden Service enthält.
2. Wählen Sie **Betrieb > Implementierte Informationsserviceanwendungen** aus.
3. Wählen Sie im Teilfenster **Sicht** die Anwendung aus, die den zu aktivierenden Service enthält.
4. Klicken Sie auf **Bearbeiten**.
5. Klicken Sie unten im Teilfenster **Sicht** auf **Inaktivieren** und wählen Sie **Aktivieren** aus.
6. Klicken Sie auf **Speichern** und **Schließen**.

Testclient generieren

Sie können einen Testclient generieren, um Ihre Services zu testen. Den Testclient können Sie mit aktivierter oder inaktivierter Sicherheit generieren.

Testclient mit inaktivierter Sicherheit generieren

Mit dieser Task können Sie einen Testclient für einen Service mit inaktivierter Sicherheit generieren.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren" auf Seite 9 beschrieben konfigurieren

Informationen zu diesem Vorgang

Wenn Sie einen Service, der über die Authentifizierung anhand von Benutzernamenstokens gesichert ist, mit einem nicht gesicherten Testclient aufrufen, wird Ihnen die folgende Sicherheitsausnahmebedingung angezeigt:

```
exception: com.ibm.wsspi.wssecurity.SoapSecurityException:
WSEC5509E: Es ist ein Sicherheitstoken mit dem Typ
http://myapp.wisd.ibm.com#UsernameTokenhttp:
//docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-
token-profile-1.0#UsernameToken] erforderlich.
```

Bei einem X.509-Zertifikattoken wird Ihnen die folgende Sicherheitsausnahmebedingung angezeigt:

```
exception: com.ibm.wsspi.wssecurity.SoapSecurityException:
WSEC5509E: Es ist ein Sicherheitstoken mit dem Typ
[X509Tokenhttp://docs.oasis-open.org/wss/2004
/01/oasis-200401-wss-x509-token-profile-1.0#X509] erforderlich.
```

Vorgehensweise

1. Klicken Sie mit der rechten Maustaste in das Fenster der Registerkarte **Server** von IBM WebSphere Application Server und klicken Sie auf **Neu** > **Server**.
2. Geben Sie auf der Seite **Neuen Server definieren** des Fensters **Neuer Server** einen Namen für den Server-Host ein, klicken Sie auf **IBM > WebSphere v6.1-Server** oder **WebSphere v7.0-Server** als Servertyp und klicken Sie auf **Information Server WAS v6.1** oder **Information Server WAS v7.0** als Serverlaufzeitumgebung.
3. Klicken Sie auf **Weiter** und stellen Sie sicher, dass die Informationen auf der daraufhin angezeigten Seite einschließlich des Profilnamens und der Informationen zur Authentifizierung korrekt sind. Sie können optional auf den Link **Test Connection** klicken, um sicherzustellen, dass IBM Rational Application Developer mithilfe der von Ihnen gerade angegebenen Authentifizierungsinformationen eine Verbindung zu WebSphere Application Server herstellen kann.
4. Klicken Sie auf **Weiter**.
5. Klicken Sie im Feld **Verfügbare Projekte** der Seite **Projekte hinzufügen und entfernen** auf die Datei *MyApp*`.ear` und klicken Sie **Hinzufügen>** an, um die Datei der Liste der konfigurierten Projekte hinzuzufügen.
6. Klicken Sie auf **Weiter** und dann auf **Fertig stellen**. Nach der Definition des Server können Sie diesem Server jetzt ein Projekt hinzufügen, indem Sie auf der Registerkarte **Server** mit der rechten Maustaste auf den Server klicken und **Projekte hinzufügen und entfernen** auswählen.
7. Wählen Sie den von Ihnen erstellten Server auf der Registerkarte **Server** aus und klicken Sie auf das Symbol **Server starten**.
8. Wählen Sie **Datei > Neu > Projekt > Dynamisches Webprojekt** aus.
9. Geben Sie im Fenster **Neues dynamisches Webprojekt** einen Projektnamen ein, beispielsweise MeinAnwclient, wählen Sie die entsprechende WebSphere Application Server-Version als Ziellaufzeitumgebung aus, wählen Sie **Projekt einer EAR hinzufügen** aus und geben Sie einen EAR-Dateiprojektnamen, beispielsweise MeinAnwclient.ear, für die EAR-Datei ein, die den Client-Code enthalten wird.
10. Klicken Sie auf **Fertig stellen**.
11. Klicken Sie auf **Datei** > **Neu** > **Sonstige**.
12. Erweitern Sie **Web-Services** im Fenster **Neu**, wählen Sie **Web-Service-Client** aus und klicken Sie auf **Weiter**.
13. Geben Sie im Fenster **Web-Services-Client** den WSDL-URL (Web Services Description Language Uniform Resource Locator; beispielsweise `http://localhost:9080/wisd/MyApp/MyService?wsdl`)) in das Feld **Servicedefinition** der Seite **Web-Services** ein.
14. Wählen Sie **Testclient** als Automatisierungsstufe aus.
15. Verifizieren Sie in der Konfigurationsanzeige die folgenden Einstellungen:

    Server Websphere v6.1-Server oder Websphere v7.0-Server

    Web-Service-Laufzeit
    IBM WebSphere JAX-RPC

    Clientprojekt
    MeinAnwclient

    Client-EAR-Projekt
    MeinAnwclient.ear
16. Klicken Sie auf **Fertig stellen**.

17. Wählen Sie auf der Seite **Web-Service-Clienttest** die Testfunktion **Web-Service-Muster-JSPs** und eine beliebige zu testende Methode aus. Verwenden Sie für die restlichen Angaben auf dieser Seite die Standardeinstellungen und klicken Sie auf **Fertig stellen**.
18. Wenn Sie den Testclient in **Projektexplorer** unter **JSR-109 Web Services** > **Clients** anzeigen wollen, müssen Sie IBM Rational Application Developer erneut starten. Die Anzeige des Testclients ist erforderlich, wenn Sie mit seiner Hilfe die Sicherheitsassistenten ausführen wollen.
19. Optional: Wählen Sie auf der Registerkarte **Web-Services-Testclient**, die geöffnet wird, eine Methode aus, geben Sie eine beliebige Eingabenummer ein und klicken Sie auf **Aufrufen**, um Ihren Service mithilfe des Testclients zu testen. Das Ergebnis wird im Ergebnisteilfenster angezeigt.
20. Optional: Wenn Sie den Testclient ohne Zuhilfenahme des Assembliertools aufrufen wollen, verwenden Sie einen Browser und geben Sie die URL `https://`*`Hostname`*`:`*`WAS-SSL-Kanalport`*`/`*`Anwendungsname`*`/sample`*`Servicename`*`PortTypeProxy/TestClient.jsp` ein.

    Hostname
    Der Name des Servers, der als Host für Ihren Testclient fungiert (beispielsweise localhost).

    WAS-SSL-Kanalport
    Die vom SSL-Transportkanal verwendet Portnummer (Standardwert: 9443)

    Anwendungsname
    Der Name des Testclients

    Servicename
    Der Name des Service, den Sie aufrufen wollen

Testclient mit aktivierter Authentifizierung generieren

Mit dieser Task können Sie einen Testclient für einen Service mit aktivierter Authentifizierung generieren.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren" auf Seite 9 beschrieben konfigurieren
- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren
- Service-EAR-Datei wie in „EAR-Datei eines nicht gesicherten Service in ein Assembliertool importieren" auf Seite 11 beschrieben in ein Assembliertool importieren, wenn die Sicherheitsfeatures mithilfe des Tools aktiviert werden sollen
- Dem Service wie in „Authentifizierungsverfahren hinzufügen" auf Seite 15 beschrieben Sicherheit hinzufügen
- EAR-Datei, die den Service enthält, wie in „EAR-Datei eines gesicherten Service aus einem Assembliertool exportieren" auf Seite 31 beschrieben exportieren
- Service-EAR-Datei wie in „Gesicherten Service mit der IBM WebSphere Application Server-Konsole erneut implementieren" auf Seite 31 beschrieben erneut implementieren

- Gesicherten Service wie in „Gesicherten Service über die IBM InfoSphere Information Server-Konsole aktivieren” auf Seite 32 beschrieben aktivieren

Informationen zu diesem Vorgang

In dieser Task wird mithilfe des Sicherheitsassistenten von IBM Rational Application Developer gezeigt, wie die WS-Security-Authentifizierung mit Benutzernamenstokens auf dem Testclient aktiviert wird. Alternativ können Sie mithilfe des Implementierungsdeskriptor-Editors von Rational Application Developer die Sicherheit auf dem Client aktivieren.

Wenn Sie einen Service aufrufen, für den die Authentifizierung erforderlich ist, aber keinen Benutzernamen und kein Kennwort angeben, wird die folgende oder eine ähnliche Sicherheitsausnahmebedingung zurückgegeben:

```
exception: com.ibm.wsspi.wssecurity.SoapSecurityException:
WSEC5509E:
Es ist ein Sicherheitstoken mit dem Typ [http://docs.oasis-open.org/wss/2004/
01/oasis-200401-wss-username-token-profile-1.0#UsernameToken] erforderlich.
```

In gleicher Weise wird die folgende oder eine ähnliche Ausnahmebedingung zurückgegeben, wenn Sie einen Service aufrufen, für den die Authentifizierung anhand von Benutzernamenstokens erforderlich ist, Sie aber einen ungültigen Benutzernamen oder ein ungültiges Kennwort angeben:

```
exception: com.ibm.wsspi.wssecurity.SoapSecurityException:
WSEC6521E:
Die Anmeldung ist fehlgeschlagen. Ausnahme: javax.security.auth.login.LoginException:
WSEC6690E: Fehler beim Überprüfen des Benutzernamens [admin] und des Kennworts in der
UserRegistry: UserRegistryProcessor.checkRegistry()=false
```

Vorgehensweise

1. Klicken Sie mit der rechten Maustaste in das Fenster der Registerkarte **Server** von IBM WebSphere Application Server und wählen Sie **Neu** > **Server** aus.
2. Geben Sie auf der Seite **Neuen Server definieren** des Fensters **Neuer Server** einen Namen für den Server-Host ein und wählen Sie **IBM > WebSphere v6.1-Server oder WebSphere v7.0-Server** als Servertyp sowie die Serverlaufzeitumgebung aus. Wenn Sie die Task zum Konfigurieren von IBM Rational Application Developer mithilfe von IBM InfoSphere Information Server ausgeführt haben, lautet die korrekte Laufzeitumgebung **Information Server WAS v6.1** oder **Information Server WAS v7.0**.
3. Klicken Sie auf **Weiter** und stellen Sie sicher, dass die Informationen auf der daraufhin angezeigten Seite einschließlich des Profilnamens und der Informationen zur Authentifizierung korrekt sind.
4. Klicken Sie auf **Weiter**.
5. Klicken Sie im Feld **Verfügbare Projekte** der Seite **Projekte hinzufügen und entfernen** auf die gesicherte EAR-Datei und klicken Sie **Hinzufügen >** an, um die Datei der Liste der konfigurierten Projekte hinzuzufügen.
6. Klicken Sie auf **Weiter** und dann auf **Fertig stellen**.
7. Wählen Sie den von Ihnen erstellten Server auf der Registerkarte **Server** aus und klicken Sie auf das Symbol **Server starten**.
8. Wählen Sie **Datei > Neu > Projekt > Dynamisches Webprojekt** aus.
9. Geben Sie im Fenster **Neues dynamisches Webprojekt** einen Projektnamen ein, beispielsweise MeinAnwclient, wählen Sie **Information Server WAS v6.1** oder **Information Server WAS v7.0** als Ziellaufzeitumgebung aus, wählen Sie

Projekt einer EAR hinzufügen aus und geben Sie einen EAR-Projektnamen, beispielsweise MeinAnwclient.ear, für die EAR-Datei ein, die den Client-Code enthalten wird.

10. Klicken Sie auf **Fertig stellen**.
11. Wählen Sie **Datei > Neu > Sonstige** aus.
12. Erweitern Sie **Web-Services** im Fenster **Neu**, wählen Sie **Web-Service-Client** aus und klicken Sie auf **Weiter**.
13. Geben Sie im Fenster **Web-Services-Client** den WSDL-URL (Web Services Description Language Uniform Resource Locator; beispielsweise `http://localhost:9080/wisd/MyApp/MyService?wsdl`) in das Feld **Servicedefinition** der Seite **Web-Services** ein.
14. Wählen Sie **Testclient** als Automatisierungsstufe aus.
15. Verifizieren Sie in der Konfigurationsanzeige die folgenden Einstellungen:

    Server Websphere v6.1-Server oder Websphere v7.0-Server

    Web-Service-Laufzeit
    IBM WebSphere JAX-RPC

    Clientprojekt
    MeinAnwclient

    Client-EAR-Projekt
    MeinAnwclient.ear
16. Klicken Sie auf **Fertig stellen**.
17. Wählen Sie auf der Seite **Web-Service-Clienttest** die Testfunktion **Web-Service-Muster-JSPs** und die zu testende Methode aus. Verwenden Sie für die restlichen Angaben auf dieser Seite die Standardeinstellungen und klicken Sie auf **Fertig stellen**.
18. Erweitern Sie im Projektexplorer **JSR-109-Web-Services > Clients**.
19. Klicken Sie mit der rechten Maustaste auf den gerade erstellen Testclient und wählen Sie **Gesicherter Web-Service-Client** > **Eigenständiges Sicherheitstoken hinzufügen** aus (es wird vorausgesetzt, dass es sich bei der im Service aktivierten Sicherheit um ein Benutzernamensicherheitstoken für Authentifizierungszwecke handelt). Wenn sich der Testclient nicht im Ordner **Clients** befindet, starten Sie Rational Application Developer erneut.
20. Wählen Sie im Assistenten für die WS-Security den Tokentyp **Benutzernamenstoken** und den Callback-Handler **com.ibm.wsspi.wssecurity.auth.callback.NonPromptCallbackHandler** aus. Die Anmeldeberechtigungen geben Sie auf der nächsten Seite des Assistenten an.
21. Klicken Sie auf **Weiter**.
22. Geben Sie eine gültige Benutzer-ID und ein gültiges Kennwort für den Service ein und klicken Sie auf **Fertig stellen**.
23. Speichern Sie die Änderungen und starten Sie den Client erneut.
24. Optional: Wählen Sie auf der Registerkarte **Web-Services-Testclient**, die geöffnet wird, eine Methode aus, geben Sie eine beliebige Eingabenummer ein und klicken Sie auf **Aufrufen**, um Ihren Service mithilfe des Testclients zu testen. Das Ergebnis wird im Ergebnisteilfenster angezeigt.
25. Optional: Wenn Sie den Testclient nicht mithilfe des Assembliertools aufrufen wollen, geben Sie in das Adressfeld des Web-Browsers den URL `https://`*`Hostname`*`:`*`WAS-SSL-Kanalport`*`/`*`Anwendungsname`*`/sample`*`Servicename`*`PortTypeProxy/TestClient.jsp` ein.

Hostname
Der Name des Servers, der als Host für Ihren Testclient fungiert (beispielsweise localhost).

WAS-SSL-Kanalport
Die vom SSL-Transportkanal verwendet Portnummer (Standardwert: 9443).

Anwendungsname
Der Name des Testclients.

Servicename
Der Name des Service, den Sie aufrufen wollen.

Beispielkeystores, Schlüssel und Zertifikate generieren

Für jeden Verschlüsselungstyp, einschließlich SSL, ist es erforderlich, dass Sie Keystores auf Clientseite und auf Serverseite generieren. Keystores enthalten Schlüssel, die Nachrichten unterzeichnen und verschlüsseln. Mit den Beispielkeystores in diesem Thema können Sie Ihre Verschlüsselung testen. Alternativ können Sie mithilfe des Java-Dienstprogramms 'keytool' das Script ausführen, um eigene Beispielkeystores zu generieren.

Anmerkung: Zur Vereinfachung dieses Beispiels wurden die Beispielkeystores über selbst signierte Zertifikate generiert und nicht über Zertifikate, die von einer Zertifizierungsstelle signiert wurden.

Verwenden Sie diese Beispielkeystores ausschließlich für Testzwecke. Implementieren Sie in einer echten Produktionsumgebung Ihre eigene PKI-Infrastruktur (Public Key Infrastructure).

Sie können Keystores mithilfe verschiedener Methoden generieren. Die Beispielkeystores in diesem Thema wurden mithilfe des Java-Befehlszeilentools `keytool` und des folgenden Scripts generiert. Sie können auch das IBM Schlüsselmanagementtool 'iKeyman' verwenden, eine grafische Benutzerschnittstelle für die Verwaltung von Keystores und Schlüsseln. Es ist mit IBM WebSphere Application Server im Verzeichnis *`Installationsverzeichnis`*`/WebSphere/AppServer/bin/ikeyman.bat` installiert.

SSL-Verschlüsselung

- Client-Keystore (clientsslkeys.jceks):
  - Privater Clientschlüssel
- Client-Truststore (clientssltrusts.jceks):
  - Selbst signiertes Serverzertifikat mit nur öffentlichem Schlüssel
- Server-Keystore (serversslkeys.jceks):
  - Privater Serverschlüssel
- Server-Truststore (serverssltrusts.jceks):
  - Selbst signiertes Clientzertifikat mit nur öffentlichem Schlüssel

Die Beispielkeystores weisen außerdem die folgenden Merkmale auf.

Tabelle 1. Merkmale für Client-Keystore `clientsslkeys.jceks`

Feld	Wert
Dateiname	`C:\sampleks\clientsslkeys.jceks`
storepass-Kennwort	Kennwort

Tabelle 1. Merkmale für Client-Keystore `clientsslkeys.jceks` (Forts.)

Feld	Wert
Storetyp	JCEK
Definierter Name	CN=wasclient, O=IBM, C=US
keypass-Kennwort	Kennwort
Aliasname	wasclient

Tabelle 2. Merkmale für Client-Truststore `clientssltrusts.jceks`

Feld	Wert
Dateiname	`C:\sampleks\clientssltrusts.jceks`
storepass-Kennwort	Kennwort
Storetyp	JCEK
Zertifikatsdatei	wasserver.cert

Tabelle 3. Merkmale für Server-Keystore `serversslkeys.jceks`

Feld	Wert
Dateiname	`C:\sampleks\serversslkeys.jceks`
storepass-Kennwort	Kennwort
Storetyp	JCEK
Definierter Name	CN=wasserver, O=IBM, C=US
keypass-Kennwort	Kennwort
Aliasname	wasserver

Tabelle 4. Merkmale für Server-Truststore `serverssltrusts.jceks`

Feld	Wert
Dateiname	`C:\sampleks\serverssltrusts.jceks`
storepass-Kennwort	Kennwort
Storetyp	JCEK
Zertifikatsdatei	wasclient.cert

Client-Keystore erstellen

Über das Java-Tool `keytool` können Sie den Client-Keystore mit dem folgenden Script generieren.

```
REM  Generate the client key store and private key
keytool -genkey -v -alias wasclient -keypass password -keystore
clientsslkeys.jceks -storepass password -storetype jceks -dname
"CN=wasclient, O=IBM, C=US" -keyalg "RSA"
REM  Generate the client self-signed certificate
keytool -export -v -alias wasclient -file wasclient.cert -rfc
-keystore
clientsslkeys.jceks -storepass password -storetype jceks
```

Server-Keystore erstellen

Über das Java-Tool `keytool` können Sie den Server-Keystore mit dem folgenden Script generieren.

```
REM  Generate the server key store and private key
keytool -genkey -v -alias wasserver -keypass password -keystore
serversslkeys.jceks -storepass password -storetype jceks -dname
"CN=wasserver, O=IBM, C=US" -keyalg "RSA"
REM  Generate the server self-signed certificate
keytool -export -v -alias wasserver -file wasserver.cert -rfc
-keystore
serversslkeys.jceks -storepass password -storetype jceks
```

Client-Truststore erstellen

Über das Java-Tool `keytool` können Sie den Client-Truststore mit dem folgenden Script generieren.

```
REM  Import the server certificate in the client truststore
keytool -import -v -noprompt -alias wasserver -file wasserver.cert -
keystore clientssltrusts.jceks -storepass password -storetype jceks
```

Server-Truststore erstellen

Über das Java-Tool `keytool` können Sie den Server-Truststore mit dem folgenden Script generieren.

```
REM  Import the client certificate in the server truststore
keytool -import -v -noprompt -alias wasclient -file wasclient.cert -
keystore serverssltrusts.jceks -storepass password -storetype jceks
```

Testclient mit aktiviertem SSL generieren

Mit dieser Task können Sie einen Testclient für einen Service generieren, der Datenschutz über die SSL-Verschlüsselung (Secure Sockets Layer) ermöglicht.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren” auf Seite 9 beschrieben konfigurieren
- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren” auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren” auf Seite 10 beschrieben exportieren
- Service-EAR-Datei wie in „EAR-Datei eines nicht gesicherten Service in ein Assembliertool importieren” auf Seite 11 beschrieben in ein Assembliertool importieren, wenn die Sicherheitsfeatures mithilfe des Tools aktiviert werden sollen
- Dem Service wie in „Datenschutz mithilfe der SSL-Verschlüsselung hinzufügen” auf Seite 26 beschrieben Vertraulichkeit hinzufügen
- EAR-Datei, die den Service enthält, wie in „EAR-Datei eines gesicherten Service aus einem Assembliertool exportieren” auf Seite 31 beschrieben exportieren
- Service-EAR-Datei wie in „Gesicherten Service mit der IBM WebSphere Application Server-Konsole erneut implementieren” auf Seite 31 beschrieben erneut implementieren
- Gesicherten Service wie in „Gesicherten Service über die IBM InfoSphere Information Server-Konsole aktivieren” auf Seite 32 beschrieben aktivieren

Informationen zu diesem Vorgang

In einem Teil dieser Task erstellen Sie über IBM WebSphere Application Server ein neues SSL-JSSE-Repertoire (Java Secure Socket Extension). Zum Erstellen dieses neuen Repertoires müssen Sie die Beispiel-Keystores aus „Beispielkeystores, Schlüssel und Zertifikate generieren" auf Seite 38 verwenden.

Vorgehensweise

1. Melden Sie sich an der IBM WebSphere Application Server-Administrationskonsole an.
2. Erweitern Sie **Sicherheit** im Navigationsteilfenster und klicken Sie auf **SSL**.
3. Klicken Sie im Fenster **SSL-Konfigurationsrepertoires** auf **Neues JSSE-Repertoire**.
4. Auf der Registerkarte **Konfiguration**:
   a. Geben Sie einen Aliasnamen für das Repertoire ein. Beispiel: `WASClientSSL`.
   b. Wählen Sie das Kontrollkästchen **Clientauthentifizierung** nicht aus. Das SSL-Repertoire fungiert als SSL-Client und nicht als Server. Daher gibt es keinen zu authentifizierenden Client.
   c. Wählen Sie die Sicherheitsstufe **HIGH** aus.
   d. Optional: Wählen Sie Verschlüsselungen aus der Liste **Cipher Suites** aus, wenn Sie die vordefinierten SSL-Verschlüsselungen entsprechendend der von Ihnen ausgewählten Sicherheitsstufe überschreiben wollen.
   e. Optional: Wählen Sie das Kontrollkästchen **Verschlüsselungstoken** aus.
   f. Optional: Wählen Sie einen vordefinierten oder angepassten JSSE-Provider aus.
   g. Wählen Sie das Protokoll **SSL** aus.
   h. Geben Sie die Speicherposition, den Namen und das Kennwort für die Schlüsseldatei ein und wählen Sie ein Format aus. Verwenden Sie die Schlüsseldatei aus dem Thema zu Beispiel-Keystorereferenzen.
   i. Geben Sie die Speicherposition, den Namen und das Kennwort für die Trust-Datei ein und wählen Sie ein Format aus. Verwenden Sie die Trust-Datei aus dem Thema zu Beispiel-Keystorereferenzen.
   j. Klicken Sie auf **OK**.
5. Speichern Sie die Änderungen und starten Sie IBM WebSphere Application Server erneut.
6. Klicken Sie auf der Registerkarte **Server** von IBM Rational Application Developer mit der rechten Maustaste auf eine beliebige Position im Fenster und wählen Sie **Neu** > **Server** aus.
7. Geben Sie auf der Seite **Neuen Server definieren** des Fensters **Neuer Server** einen Namen für den Server-Host ein und wählen Sie **IBM > WebSphere v6.1-Server oder WebSphere v7.0-Server** als Servertyp sowie die Serverlaufzeitumgebung aus. Wenn Sie die Task zum Konfigurieren von IBM Rational Application Developer mithilfe von IBM InfoSphere Information Server ausgeführt haben, lautet die korrekte Laufzeitumgebung **Information Server WAS v6.1** oder **Information Server WAS v7.0**.
8. Klicken Sie auf **Weiter** und stellen Sie sicher, dass die Informationen auf der daraufhin angezeigten Seite einschließlich des Profilnamens und der Informationen zur Authentifizierung korrekt sind.
9. Klicken Sie auf **Weiter**.

10. Klicken Sie im Feld **Verfügbare Projekte** der Seite **Projekte hinzufügen und entfernen** auf die gesicherte EAR-Datei und klicken Sie **Hinzufügen >** an, um die Datei der Liste der konfigurierten Projekte hinzuzufügen.
11. Klicken Sie auf **Weiter** und dann auf **Fertig stellen**.
12. Wählen Sie den von Ihnen erstellten Server auf der Registerkarte **Server** aus und klicken Sie auf das Symbol **Server starten**.
13. Wählen Sie **Datei > Neu > Projekt > Dynamisches Webprojekt** aus.
14. Geben Sie im Fenster **Neues dynamisches Webprojekt** einen Projektnamen ein, beispielsweise MeinAnwclient, wählen Sie **Information Server WAS v6.1** oder **Information Server WAS v7.0** als Ziellaufzeitumgebung aus, wählen Sie **Projekt einer EAR hinzufügen** aus und geben Sie einen EAR-Projektnamen, beispielsweise MeinAnwclient.ear, für die EAR-Datei ein, die den Client-Code enthalten wird.
15. Klicken Sie auf **Fertig stellen**.
16. Wählen Sie **Datei > Neu > Sonstige** aus.
17. Erweitern Sie **Web-Services** im Fenster **Neu**, wählen Sie **Web-Service-Client** aus und klicken Sie auf **Weiter**.
18. Geben Sie im Fenster **Web-Services-Client** den WSDL-URL (Web Services Description Language Uniform Resource Locator; beispielsweise http://localhost:9080/wisd/MyApp/MyService?wsdl) in das Feld **Servicedefinition** der Seite **Web-Services** ein.
19. Wählen Sie **Testclient** als Automatisierungsstufe aus.
20. Verifizieren Sie in der Konfigurationsanzeige die folgenden Einstellungen:

    Server Websphere v6.1-Server oder Websphere v7.0-Server

    Web-Service-Laufzeit
    IBM WebSphere JAX-RPC

    Clientprojekt
    MeinAnwclient

    Client-EAR-Projekt
    MeinAnwclient.ear
21. Klicken Sie auf **Fertig stellen**.
22. Wählen Sie auf der Seite **Web-Service-Clienttest** die Testfunktion **Web-Service-Muster-JSPs** und die zu testende Methode aus. Verwenden Sie für die restlichen Angaben auf dieser Seite die Standardeinstellungen und klicken Sie auf **Fertig stellen**.
23. Erweitern Sie **JSR-109 Web Services** > **Clients** im **Projektexplorer** von IBM Rational Application Developer und klicken Sie doppelt auf den generierten Testclient, um den Implementierungsdeskriptor zu bearbeiten. Starten Sie Rational Application Developer erneut, wenn sich der Testclient nicht im Ordner befindet.
24. Klicken Sie auf die Registerkarte **WS-Bindung** im Fenster **Webimplementierungsdeskriptor** und erweitern Sie den Abschnitt **Details für die portqualifizierte Namensbindung**.
25. Geben Sie den Namen des Client-SSL-Repertoires in das Feld für den Namen der HTTP-SSL-Konfiguration ein. Beispiel: *Name2Knoten01/WASClientSSL*.
26. Speichern Sie die Änderungen und starten Sie den Client erneut.
27. Optional: Wählen Sie auf der Registerkarte **Web-Services-Testclient**, die geöffnet wird, eine Methode aus, geben Sie eine beliebige Eingabenummer ein und klicken Sie auf **Aufrufen**, um Ihren Service mithilfe des Testclients zu testen. Das Ergebnis wird im Ergebnisteilfenster angezeigt.

28. Optional: Wenn Sie den Testclient nicht mithilfe des Assembliertools aufrufen wollen, geben Sie in das Adressfeld des Web-Browsers den URL `https://`*`Hostname`*`:`*`WAS-SSL-Kanalport`*`/`*`Anwendungsname`*`/sample`*`Servicename`*`PortTypeProxy/TestClient.jsp` ein.

    Hostname
    Der Name des Servers, der als Host für Ihren Testclient fungiert (beispielsweise localhost).

    WAS-SSL-Kanalport
    Die vom SSL-Transportkanal verwendet Portnummer (Standardwert: 9443).

    Anwendungsname
    Der Name des Testclients.

    Servicename
    Der Name des Service, den Sie aufrufen wollen.

Beispiel

Sie können die folgenden Fehler vermeiden, indem Sie SSL über IBM WebSphere Application Server aktivieren:

- Wenn Sie einen gesicherten SSL-Service über HTTP anstatt über HTTPS aufrufen, wird die folgende Ausnahmebedingung zurückgegeben:

  ```
  exception: WSWS3499W: Neue Umleitungsposition:
  http://localhost:9080/wisd/MyApp/MyService
  ```

- Wenn Sie einen gesicherten SSL-Service über einen nicht für SSL konfigurierten Testclient aufrufen, wird die folgende Ausnahmebedingung zurückgegeben:

  ```
  exception: WSWS3713E: Die Verbindung zum fernen Host
  localhost ist fehlgeschlagen. Der folgende Fehler wurde empfangen:
  Handshake terminated SSL engine: CLOSED
  ```

- Wenn Sie einen gesicherten SSL-Service über einen Testclient aufrufen, der nicht korrekt für SSL konfiguriert ist (der Client-Truststore wurde nicht korrekt konfiguriert), wird die folgende Ausnahmebedingung zurückgegeben:

  ```
  exception: com.ibm.wsspi.channel.framework.exception
  ChannelException:com.ibm.wsspi.channel.framework.exception.
  ChannelException: Invalid trust file name of null
  ```

SOAP-Nachrichten verfolgen

Sie können SOAP-Nachrichten verfolgen, die zwischen Client und Server ausgetauscht werden, um Fehler bei einem Service zu beheben. Zur Verfolgung von SOAP-Nachrichten können Sie das TCP/IP-Überwachungstool in IBM Rational Application Developer verwenden.

Vorbereitende Schritte

- Service erstellen
- Assembliertool wie in „Assembliertool mit IBM InfoSphere Information Server konfigurieren" auf Seite 9 beschrieben konfigurieren
- Nicht gesicherten Service wie in „Nicht gesicherten Service über die IBM InfoSphere Information Server-Konsole inaktivieren" auf Seite 10 beschrieben inaktivieren
- Service zum Erstellen einer Service-EAR-Datei wie in „Service mithilfe der WebSphere Application Server-Konsole exportieren" auf Seite 10 beschrieben exportieren

- Service-EAR-Datei wie in „EAR-Datei eines nicht gesicherten Service in ein Assembliertool importieren" auf Seite 11 beschrieben in ein Assembliertool importieren, wenn die Sicherheitsfeatures mithilfe des Tools aktiviert werden sollen
- Service wie in „Service schützen" auf Seite 12 beschrieben schützen
- EAR-Datei, die den Service enthält, wie in „EAR-Datei eines gesicherten Service aus einem Assembliertool exportieren" auf Seite 31 beschrieben exportieren
- Service-EAR-Datei wie in „Gesicherten Service mit der IBM WebSphere Application Server-Konsole erneut implementieren" auf Seite 31 beschrieben erneut implementieren
- Gesicherten Service wie in „Gesicherten Service über die IBM InfoSphere Information Server-Konsole aktivieren" auf Seite 32 beschrieben aktivieren

Vorgehensweise

1. Wählen Sie **Fenster** > **Sicht anzeigen** > **Weitere** aus.
2. Erweitern Sie **Debug** im Fenster **Sicht anzeigen**, wählen Sie **TCP/IP-Monitor** aus und klicken Sie auf **OK**.
3. Klicken Sie mit der rechten Maustaste in das TCP/IP-Monitorfenster und wählen Sie **Eigenschaften** aus.
4. Klicken Sie auf **Hinzufügen** und geben Sie die folgende Information in das Fenster **Neuer Monitor** ein bzw. wählen Sie sie aus:
   a. Geben Sie eine noch nicht verwendete lokale Überwachungsportnummer ein. Beispiel: *13557*. Der Client sendet SOAP-Anforderungen dann an diesen Port.
   b. Geben Sie den Namen des Host-Computers ein, der den Server ausführt. Wenn sich beispielsweise Client und Server auf demselben Computer befinden, lautet der Hostname `localhost`.
   c. Geben Sie die Nummer des zu überwachenden Ports ein. Dabei handelt es sich um den Port, an den der Client Anforderungen senden würde, wenn die TCP/IP-Überwachung nicht aktiviert wäre. Die Standardportnummer für IBM WebSphere Application Server ist beispielsweise `9080`.
   d. Wählen Sie als Überwachungstyp **HTTP** oder **TCP/IP** aus. Mit **TCP/IP** werden die vollständigen HTTP-Nachrichten angezeigt, einschließlich der HTTP-Header und der SOAP-Nachrichten. Mit **HTTP** werden nur die SOAP-Nachrichten angezeigt.
   e. Klicken Sie auf **OK**.
5. Wählen Sie in dem Fenster mit den Benutzervorgaben für den TCP/IP-Monitor, das geöffnet wird, das Überwachungsprogramm aus und klicken Sie auf **Starten**, um mit der Überwachung zu beginnen.
6. Erweitern Sie **JSR-109 Web Services** > **Clients** im **Projektexplorer**, klicken Sie mit der rechten Maustaste auf den Testclient und wählen Sie **Öffnen** aus.
7. Wählen Sie die Methode **getEndpoint()** aus und klicken Sie auf **Aufrufen**. Kopieren Sie die zurückgegebene Endpunkt-URL.
8. Wählen Sie die Methode **setEndpoint()** aus, fügen Sie die Endpunkt-URL ein, die zuvor von der Methode **getEndpoint()** zurückgegeben worden ist, und ändern Sie die Portnummer, sodass sie auf die Portnummer des TCP/IP-Monitors verweist. Klicken Sie auf **Aufrufen**. Wenn beispielsweise localhost der Hostname und 13557 die lokale Portnummer ist, lautet der Endpunkt `http://localhost:13557/wisd/`*`MeineAnwendung`*`/`*`MeinService`*.
9. Rufen Sie Ihren Service über eine der folgenden Methoden mit dem Testclient auf:

- Wählen Sie in **Web-Service-Testclient** eine im Service definierte Methode aus, geben Sie eine beliebige Eingabenummer ein und klicken Sie auf **Aufrufen**.
- Wenn Sie den Testclient nicht mithilfe des Assembliertools aufrufen wollen, geben Sie in das Adressfeld des Web-Browsers den URL `https://`*`Hostname`*`:`*`WAS-SSL-Kanalport`*`/`*`Anwendungsname`*`/sample`*`Servicename`*`PortTypeProxy/TestClient.jsp` ein.

  Hostname
  Der Name des Servers, der als Host für Ihren Testclient fungiert (beispielsweise localhost).

  WAS-SSL-Kanalport
  Die vom SSL-Transportkanal verwendet Portnummer (Standardwert: 9443).

  Anwendungsname
  Der Name des Testclients.

  Servicename
  Der Name des Service, den Sie aufrufen wollen.

10. Analysieren Sie nach dem Aufrufen des Service die HTTP- und SOAP-Nachrichten im TCP/IP-Monitorfenster, um Fehler zu beheben, die im Service aufgetreten sind.

Sicherheitstechnologien im Vergleich

Zum Auswählen einer Technologie, die Ihren Anforderungen entspricht, können Sie Sicherheitstechnologien anhand zweier Parameter vergleichen: Grad der bereitgestellten Sicherheit und Schwierigkeitsgrad beim Konfigurieren.

Im Allgemeinen ist eine Technologie schwieriger zu implementieren, wenn sie auf der Public-Key-Verschlüsselung basiert, die das Konfigurieren einer PKI-Infrastruktur (Public Key Infrastructure) erfordert. Dies trifft auf jeden Typ digitaler Unterzeichnung und Verschlüsselung zu. Diese Technologien bieten jedoch gewöhnlich ein höheres Sicherheitsniveau als andere Technologien, die möglicherweise weniger Infrastrukturarbeit erfordern.

Tabelle 5. Authentifizierungsverfahren und zugehörige Sicherheitsstufen

Authentifizierungsverfahren	**Stufe**	**Sicherheitsgrad**	**Schwierigkeitsgrad der Konfiguration**	**Hinweise**
HTTP-Basisauthentifizierung	Transport	niedrig	niedrig	Benutzername und Kennwort werden als Klartext (base64) über das Netz übertragen. Wird nur in sicheren Netzwerken verwendet (HTTPS oder Intranet).

Tabelle 5. Authentifizierungsverfahren und zugehörige Sicherheitsstufen (Forts.)

Authentifizierungsverfahren	**Stufe**	**Sicherheitsgrad**	**Schwierigkeitsgrad der Konfiguration**	**Hinweise**
HTTP-Digest-Authentifizierung (nicht unterstützt von IBM WebSphere Application Server)	Transport	mittel	mittel	Benutzername und Kennwort werden verschlüsselt.
Benutzernamenstoken	Nachricht	niedrig	mittel	Benutzername und Kennwort werden als Klartext über das Netz übertragen. Wird nur in sicheren Netzwerken verwendet (HTTPS, Intranet oder bei XML-Verschlüsselung).
X.509-Zertifikattoken	Nachricht	hoch	mittel	Erfordert eine PKI-Infrastruktur (Public Key Infrastructure).
Kerberos-Ticket-Token	Nachricht	hoch	hoch	Erfordert eine Kerberos-Infrastruktur.

Tabelle 6. Autorisierungsverfahren und zugehörige Sicherheitsstufe

Autorisierungsverfahren	**Stufe**	**Sicherheitsgrad**	**Schwierigkeitsgrad der Konfiguration**	**Hinweise**
Deklarative J2EE-Integritätsbedingungen für die Sicherheit	Anwendung	mittel	mittel	Erfordern keine Sicherheitsinfrastruktur.

Deklarative J2EE-Integritätsbedingungen für die Sicherheit lassen sich leicht konfigurieren (lediglich Änderungen an den J2EE-Implementierungsdeskriptoren sind erforderlich) und bieten gleichzeitig ein gutes Zugriffssteuerungsniveau. Sie bilden oft den Ausgangspunkt beim Schutz von Services.

Tabelle 7. Vertraulichkeitsverfahren und zugehörige Sicherheitsstufe

Vertraulichkeits-verfahren	**Stufe**	**Sicherheitsgrad**	**Schwierigkeits-grad der Konfi-guration**	**Hinweise**
SSL	Transport	mittel	mittel	Erfordert eine PKI-Infrastruk-tur (Public Key Infrastructure). Vertraulichkeit wird von Netzpunkt zu Netzpunkt be-reitgestellt.

SSL und TLS (Transport Layer Security) stellen über die Verschlüsselung hinaus zusätzliche Sicherheitsfunktionen zur Verfügung, einschließlich Authentifizierung, Datenschutz und Unterstützung von Verschlüsselungstokens für sichere HTTP-Verbindungen.

Die Verschlüsselung ist eine rechenintensive Operation, die die Leistung beeinträchtigen kann. Außerdem wird der Verwaltungsaufwand erhöht. Sie sollte daher erst nach sorgfältiger Erwägung eingesetzt werden.

Tabelle 8. Digitales Integritätsverfahren und zugehörige Sicherheitsstufe

Daten-integritäts-verfahren	**Stufe**	**Sicherheitsgrad**	**Schwierigkeits-grad der Konfi-guration**	**Hinweise**
Digitale XML-Signatur	Nachricht	hoch	mittel	Erfordert eine PKI-Infrastruk-tur.

Die Verschlüsselung ist eine rechenintensive Operation, die die Leistung beeinträchtigen kann. Außerdem wird der Verwaltungsaufwand erhöht. Sie sollte daher erst nach sorgfältiger Erwägung eingesetzt werden.

Sicherheitstechnologien und -bindungen

Nicht alle Sicherheitstechnologien sind für alle Bindungen verfügbar.

Für einige Bindungen ist möglicherweise nur eine Untergruppe der vollständigen Liste vorhandener Sicherheitstechnologien verfügbar (bzw. anwendbar). Beispielsweise ist keine der Web-Service-Sicherheitstechnologien für die EJB-Bindung anwendbar. Die folgende Tabelle enthält eine Zusammenfassung der für die unterschiedlichen Bindungen verfügbaren Sicherheitstechnologien.

Tabelle 9. Sicherheitstechnologien und -bindungen

	Sicherheits-technologie	**EJB**	**SOAP über HTTP**	**SOAP über JMS**	**Text über JMS**
Authentifizierung	JAAS/EJB	Verfügbar			
Authentifizierung	HTTP-Basic		Verfügbar		
Authentifizierung	HTTP-Digest		Nicht unterstützt von IBM WebSphere Application Server 6.0 und höher		
Authentifizierung	Benutzernamenstoken für Web-Service-Sicherheit		Verfügbar	Verfügbar	
Authentifizierung	Sonstige Web-Service-Sicherheit (Kerberos, X509, LTPA, SAML)		Verfügbar	Verfügbar	
Autorisierung	J2EE	Verfügbar	Verfügbar	Verfügbar	Verfügbar
Datenschutz	HTTPS und SSL	Verfügbar	Verfügbar	Verfügbar	Verfügbar

Unterstützung für behindertengerechte Bedienung in den Produkten

Sie können Informationen zum Status von IBM Produkten hinsichtlich der Unterstützung für behindertengerechte Bedienung abrufen.

Die Produktmodule und Benutzerschnittstellen von IBM InfoSphere Information Server sind nicht uneingeschränkt für behindertengerechte Bedienung geeignet. Das Installationsprogramm installiert die folgenden Produktmodule und -komponenten:

- IBM InfoSphere Business Glossary
- IBM InfoSphere Business Glossary Anywhere
- IBM InfoSphere DataStage
- IBM InfoSphere FastTrack
- IBM InfoSphere Information Analyzer
- IBM InfoSphere Information Services Director
- IBM InfoSphere Metadata Workbench
- IBM InfoSphere QualityStage

Informationen zum Status von IBM Produkten hinsichtlich der Unterstützung für behindertengerechte Bedienung finden Sie auf der entsprechenden Website unter der folgenden Adresse: http://www.ibm.com/able/product_accessibility/index.html.

Dokumentation im behindertengerechten Format

Dokumentation im behindertengerechten Format für die InfoSphere Information Server-Produkte steht in einem Information Center zur Verfügung. In diesem Information Center wird zur Darstellung der Dokumentation das Format XHTML 1.0 verwendet, das mit den meisten Web-Browsern geöffnet werden kann. XHTML ermöglicht es Ihnen, die gewünschten Anzeigeeinstellungen in Ihrem Browser festzulegen. Darüber hinaus ist der Einsatz von Sprachausgabeprogrammen und anderen Unterstützungseinrichtungen für den Zugriff auf die Dokumentation möglich.

IBM und Unterstützung für behindertengerechte Bedienung

Im IBM Human Ability and Accessibility Center finden Sie weitere Informationen zum Engagement von IBM hinsichtlich der Unterstützung für behindertengerechte Bedienung.

Auf Produktdokumentation zugreifen

Die Produktdokumentation steht in einer Reihe unterschiedlicher Formate zur Verfügung und kann über verschiedene Zugriffsmöglichkeiten abgerufen werden, zum Beispiel in Hilfetexten, die direkt über die Clientschnittstelle des Produkts geöffnet werden, in einem die gesamte Suite umfassenden Information Center und in PDF-Büchern.

Das Information Center wird als allgemeiner Service mit IBM InfoSphere Information Server installiert. Es enthält Hilfetexte für den Großteil der Produktschnittstellen sowie eine umfassende Dokumentation für alle Produktmodule in der Suite. Sie können das Information Center über das installierte Produkt oder über einen Web-Browser öffnen.

Zugriff auf das Information Center

Zum Öffnen des installierten Information Center stehen Ihnen die nachfolgend beschriebenen Methoden zur Verfügung.

- Klicken Sie auf den Link **Hilfe** rechts oben in der Clientschnittstelle.

  Anmerkung: Über IBM InfoSphere FastTrack und IBM InfoSphere Information Server Manager wird im Haupthilfeelement eine lokale Hilfefunktion geöffnet. Wählen Sie **Hilfe > Information Center öffnen** aus, um das Information Center mit vollem Funktionsumfang zu öffnen.
- Drücken Sie die Taste F1. Mit der Taste F1 wird normalerweise das Thema geöffnet, das den momentan in der Clientschnittstelle angezeigten Kontext beschreibt.

  Anmerkung: Die Taste F1 kann in Web-Clients nicht verwendet werden.
- Verwenden Sie einen Web-Browser, um auf das installierte Information Center zuzugreifen, ohne beim Produkt angemeldet zu sein. Geben Sie dazu im Web-Browser die folgende Adresse ein: http://host_name:port_number/infocenter/topic/com.ibm.swg.im.iis.productization.iisinfsv.home.doc/ic-homepage.html. Hierbei steht 'host_name' für den Namen des Computers der Service-Ebene, in der das Information Center installiert ist, und 'port_number' für die Portnummer von InfoSphere Information Server. Die Standardportnummer lautet 9080. Auf einem Microsoft® Windows® Server-Computer mit dem Namen 'iisdocs2' weist die Webadresse zum Beispiel das folgende Format auf: http://iisdocs2:9080/infocenter/topic/com.ibm.swg.im.iis.productization.iisinfsv.nav.doc/dochome/iisinfsrv_home.html.

Ein Teil des Information Center ist auch auf der IBM Website verfügbar und wird in regelmäßigen Abständen aktualisiert: http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r7/index.jsp.

PDF- und Hardcopydokumentation abrufen

- Ein Teil der PDF-Bücher wird über das Softwareinstallationsprogramm von InfoSphere Information Server sowie über die Verteilerdatenträger bereitgestellt. Die übrigen PDF-Bücher sind online verfügbar und können über das folgende Support-Dokument aufgerufen werden: https://www.ibm.com/support/docview.wss?uid=swg27008803&wv=1.

- Sie können IBM Veröffentlichungen auch in Hardcopyformat online oder über den zuständigen IBM Ansprechpartner bestellen. Wenn Sie Veröffentlichungen online bestellen möchten, rufen Sie das IBM Publications Center unter http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss auf.

Feedback zur Dokumentation senden

Kommentare zur Dokumentation können Sie uns wie folgt zukommen lassen:

- Über das Onlineformular: www.ibm.com/software/data/rcf/.
- Per E-Mail: comments@us.ibm.com.

Links auf Websites anderer Anbieter

Dieses Information Center enthält möglicherweise Links oder Verweise auf Websites und Ressourcen anderer Anbieter.

Es bestehen keine Zusicherungen, Gewährleistungen oder Verpflichtungen von IBM hinsichtlich der Websites oder Ressourcen anderer Anbieter (einschließlich Websites von Lenovo), auf die über eine IBM Site verwiesen wird, Zugriff besteht oder Links vorhanden sind. Ein Link auf eine Website eines anderen Anbieters bedeutet nicht, dass IBM den Inhalt und die Verwendung dieser Website billigt oder deren Eigentümer anerkennt. Darüber hinaus ist IBM nicht an Transaktionen beteiligt und übernimmt keine Verantwortung für Transaktionen zwischen Ihnen und anderen Anbietern, auch wenn die Informationen (oder Links) zu diesen Anbietern auf einer IBM Website zur Verfügung stehen. IBM ist nicht für die Verfügbarkeit solcher externen Sites oder Ressourcen verantwortlich und übernimmt keine Verantwortung oder Haftung für Inhalte, Services, Produkte oder sonstiges Material, die bzw. das auf diesen oder über diese Sites oder Ressourcen verfügbar sind.

Wenn Sie auf eine Website eines Fremdanbieters zugreifen, handelt es sich um eine unabhängige Site, deren Inhalt nicht von IBM kontrolliert wird. Dies gilt auch dann, wenn diese Site möglicherweise das IBM Logo enthält. IBM sieht es als Ihre Aufgabe an, sicherzustellen, dass Sie sich vor Viren, Würmern, trojanischen Pferden oder sonstigen zerstörerischen Programmen schützen; dies gilt auch für den Schutz Ihrer Informationen.

Bemerkungen und Marken

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Bemerkungen

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für in dieser Dokumentation beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Dokument aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung dient nur zu Planungszwecken. Die in dieser Veröffentlichung enthaltenen Informationen können geändert werden, bevor die beschriebenen Produkte verfügbar sind.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Musteranwendungsprogramme, die in Quellensprache geschrieben sind und Programmiertechniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Musterprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Die Musterprogramme werden ohne Wartung (auf "as-is"-Basis) und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM übernimmt keine Haftung für Schäden, die durch die Verwendung der Musterprogramme entstehen.

Kopien oder Teile der Musterprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Musterprogrammen der IBM Corporation abgeleitet. © Copyright IBM Corp. _Jahr/Jahre angeben_. Alle Rechte vorbehalten.

Marken

IBM, das IBM Logo und ibm.com sind Marken der IBM Corp. in den USA und/oder anderen Ländern. Weitere Unternehmens-, Produkt- oder Servicenamen können Marken von IBM oder anderer Hersteller sein. Eine aktuelle Liste der IBM Marken finden Sie im Web unter www.ibm.com/legal/copytrade.shtml.

Die folgenden Namen sind Marken oder eingetragene Marken anderer Unternehmen:

Adobe ist eine eingetragene Marke von Adobe Systems Incorporated in den USA und/oder anderen Ländern.

IT Infrastructure Library ist eine eingetragene Marke der Central Computer and Telecommunications Agency. Die Central Computer and Telecommunications Agency ist nunmehr in das Office of Government Commerce eingegliedert worden.

Intel, das Intel-Logo, Intel Inside, das Intel Inside-Logo, Intel Centrino, das Intel Centrino-Logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium und Pentium sind Marken oder eingetragene Marken der Intel Corporation oder deren Tochtergesellschaften in den USA oder anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Microsoft, Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

ITIL ist eine eingetragene Marke und eine eingetragene Gemeinschaftsmarke des Office of Government Commerce, welche beim US Patent and Trademark Office registriert sind.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Cell Broadband Engine wird unter Lizenz verwendet und ist eine Marke von Sony Computer Entertainment, Inc. in den USA und/oder anderen Ländern.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.

United States Postal Service ist Inhaber der folgenden Marken: CASS, CASS Certified, DPV, LACS^{Link}, ZIP, ZIP + 4, ZIP Code, Post Office, Postal Service, USPS und United States Postal Service. Die IBM Corporation ist ein nicht ausschließlicher Lizenznehmer für DPV und LACS^{Link}.

Weitere Unternehmens-, Produkt- oder Servicenamen können Marken anderer Hersteller sein.

Kontaktaufnahme mit IBM

Sie können sich an IBM wenden, um Unterstützung, Informationen zu Software-Services, Produktinformationen sowie allgemeine Informationen zu erhalten. Darüber hinaus können Sie Feedback zu Produkten und zur Dokumentation an IBM abgeben.

In der folgenden Tabelle sind die Quellen aufgeführt, unter denen Sie Unterstützung, Informationen zu Software-Services, Produktinformationen sowie Informationen zu Lösungen erhalten können.

Tabelle 10. IBM Quellen

Quelle	**Beschreibung und Position**
IBM Support Portal	Sie können Unterstützungsinformationen anpassen, indem Sie die Produkte und Themen, die für Sie von Interesse sind, unter www.ibm.com/support/entry/portal/Software/Information_Management/InfoSphere_Information_Server auswählen.
Software-Services	Informationen zu Software-, IT- und Unternehmensberatungsservices erhalten Sie auf der Site 'Lösungen' unter www.ibm.com/businesssolutions/de.
Meine IBM	Auf der Site 'Meine IBM' unter www.ibm.com/account/de/de/ können Sie ein Konto einrichten und so Links auf IBM Websites und Informationen Ihren speziellen Anforderungen an die technische Unterstützung entsprechend verwalten.
Schulung und Zertifizierung	Unter http://www.ibm.com/software/sw-training/ können Sie Informationen zu technischen Schulungen und Weiterbildungsservices für Einzelpersonen, Unternehmen und öffentliche Organisationen erhalten, mit denen IT-Kenntnisse erzielt, beibehalten und optimiert werden können.
IBM Ansprechpartner	Sie können unter www.ibm.com/connect/ibm/us/en/ Kontakt zum IBM Ansprechpartner aufnehmen, um Informationen zu Lösungen zu erhalten.

Feedback abgeben

Die folgende Tabelle beschreibt, wie Feedback zu Produkten und zur Produktdokumentation an IBM abgegeben werden kann.

Tabelle 11. Feedback an IBM abgeben

Art des Feedbacks	Aktion
Feedback zum Produkt	Sie können die Umfrage zur Verbraucherfreundlichkeit von Software nutzen, um allgemeines Feedback zu Produkten zu geben (Consumability Survey unter www.ibm.com/software/data/info/consumability-survey, landessprachliche Version unter https://www-950.ibm.com/survey/oid/wsb.dll/studies/consumabilitywebform.htm?renderlang=de).
Feedback zur Dokumentation	Sie können einen Kommentar zum Information Center senden, indem Sie auf den Link 'Feedback' klicken, der sich rechts oben in jedem Information Center-Thema befindet. Darüber hinaus können Sie Kommentare zu den PDF-Büchern, dem Information Center und sonstiger Dokumentation wie folgt senden: • Über das Onlineformular: www.ibm.com/software/data/rcf/. • Per E-Mail: comments@us.ibm.com.

Index

P

R

S

T

U

V

W

SC12-4672-00