

PL/SQL-Web-Services (Teil I)

Autor: Markus Fiegler, ORDIX AG, Paderborn

Web-Services spielen in der heutigen Software-Architektur eine immer wichtigere Rolle. Ihr großer Vorteil ist eine vollständige Unabhängigkeit von Plattform und Programmiersprache. Greifen Web-Services auf bestehende PL/SQL-Programme zu, so spricht man von PL/SQL-Web-Services. In diesem Beitrag wird anhand von Beispielen aufgezeigt, wie existierende PL/SQL-Programme als Web-Services zur Verfügung gestellt werden.

Web-Services sind Internet-Dienste, die über das Web angefordert und im Web ausgeführt werden können. Im

Gegensatz zu Web-Anwendungen mit einem HTML-Formular ist der Zugriff auf Web-Services ohne Browser möglich. Dadurch ist eine automatische Übertragung von Informationen zwischen Anwendungen gewährleistet.

Service-orientierte Architektur

Web-Services orientieren sich an der Service-orientierten Architektur (SOA) und vereinen daher verteilte und objekt-orientierte Programmierstandards. Mit ihrer Hilfe können betriebswirtschaftliche Anwendungen im Internet erstellt werden. Die Zerlegung eines Software-Systems in Services erleichtert die mehrfache Verwendung einer einmal implementierten Geschäftsfunktionalität (Business-Logik). Damit können die Wiederverwendbarkeit eines Software-Systems verbessert sowie die Kosten und die Entwick-

lungszeit minimiert werden. Durch die Kapselung der anwendungsspezifischen Geschäftsobjekt-Sichten in Services findet außerdem eine Entkopplung der Software-Systeme statt. Dadurch werden Abhängigkeiten reduziert.

Im Umfeld von Web-Services lassen sich folgende Teilnehmer/Instanzen identifizieren:

- Konsument
- Anbieter
- Verzeichnis

Der Anbieter veröffentlicht die Beschreibungen seiner Dienste in einem Verzeichnis. Ein Konsument durchsucht das Verzeichnis des Anbieters und wählt einen gewünschten Dienst aus. Danach findet eine dynamische Anbindung des Konsumenten an den Anbieter statt. Schließlich greift der Konsument auf die Methoden des Anbieters zu (siehe Abbildung 1).

Grundlage für Web-Services sind folgende Standards:

- Universal Description, Discovery and Integration (UDDI) ist ein Verzeichnisdienst zur Registrierung von Web-Services. Ein Verzeichnisdienst ermöglicht das dynamische Finden des Web-Services durch den Konsumenten.

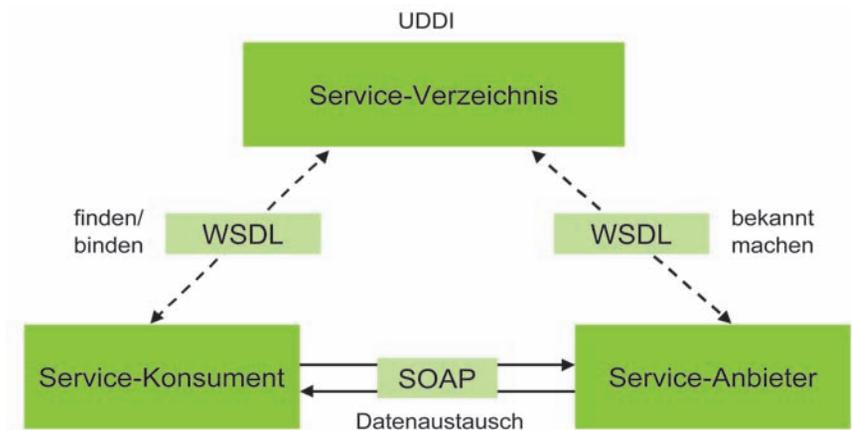


Abbildung 1: Funktionsweise von Web-Services.

- Web Services Description Language (WSDL) definiert einen Plattform-, Programmiersprachen- und Protokoll-unabhängigen XML-Standard zur Beschreibung von Web-Services, die einen Austausch von Nachrichten ermöglichen.
- Simple Object Access Protocol (SOAP) ist ein Protokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote Procedure Calls durchgeführt werden können.

Integration und Datenaustausch

Unternehmenslösungen werden oft mit verschiedenen Programmiersprachen wie Java, Visual Basic (VB) oder C++ entwickelt. Meistens ist eine Integration der einzelnen Systeme und ein Datenaustausch zwischen den Systemen notwendig, wobei ein sehr hoher Aufwand erforderlich ist. Abhilfe verschaffen hier die Web-Services. Mit deren Hilfe können Clients und Server unabhängig von der Plattform und der Programmiersprache über Internet-Standardprotokolle wie HTTP, HTTPS, SMTP oder FTP kommunizieren.

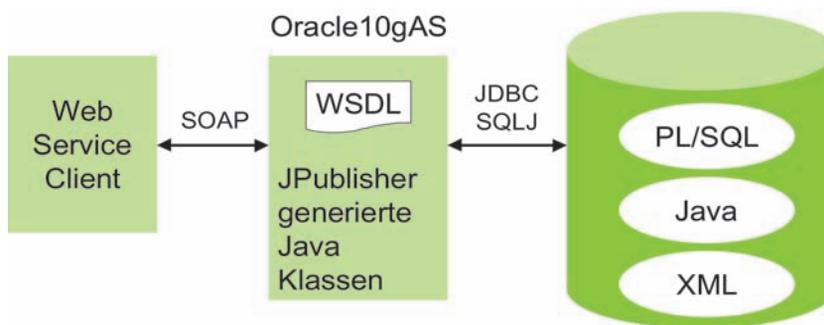


Abbildung 2: Datenbank als Web-Services Provider (Web-Services Call-in).

Die entwickelten Anwendungen müssen lediglich einigen Standards entsprechen, damit diese als Web-Services zur Verfügung gestellt und genutzt werden können. Durch die Nutzung der Internet-Standardprotokolle treten außerdem nur selten Probleme mit Firewalls auf, da unter anderem das HTTP-Standardprotokoll eingesetzt werden kann.

Wiederverwendung von vorhandenen PL/SQL-Programmen

Mit Oracle10g ist es möglich, die Funktionalität für Web-Services mit der Programmiersprache PL/SQL zu erstellen. Damit können vorhandene PL/SQL-Programme mit der neuen Web-Services-Architektur wiederverwendet werden.

```

CREATE OR REPLACE PACKAGE pck_seminar IS
    FUNCTION hole_preis(p_name IN VARCHAR2) RETURN NUMBER;
END pck_seminar;
/
CREATE OR REPLACE PACKAGE BODY pck_seminar IS
    FUNCTION hole_preis(p_name IN VARCHAR2) RETURN NUMBER IS v_preis seminar.preis%TYPE;
    CURSOR c_seminar IS SELECT preis FROM seminar WHERE name = p_name;
BEGIN
    OPEN c_seminar;
    FETCH c_seminar INTO v_preis;
    CLOSE c_seminar;
    RETURN v_preis;
END hole_preis;
END pck_seminar;
/
    
```

Abbildung 3: PL/SQL-Package pck_seminar.

```

<data-source
    class="com.evermind.sql.DriverManagerDataSource"
    name="OracleDS"
    location="jdbc/OracleCoreDS"
    xa-location="jdbc/xa/OracleXADS"
    ejb-location="jdbc/OracleDS"
    connection-driver="oracle.jdbc.driver.OracleDriver"
    username="mf"
    password="mf"
    url="jdbc:oracle:thin:@localhost:1521:ORCL10"
    inactivity-timeout="30"
/>
    
```

Abbildung 4: Ein Auszug aus der Datei data-sources.xml zur Einrichtung von Datenquellen in der OC4J-Umgebung.

Web-Services können unter anderem auf PL/SQL-Packages, Stored Procedures, Stored Functions, Triggers, SQL-Abfragen und Java-Stored-Procedures zugreifen (siehe Abbildung 2). Die notwendigen Schritte, um beispielsweise ein PL/SQL-Package (siehe Abbildung 3) als Web-Service zu veröffentlichen, werden im Folgenden beschrieben.

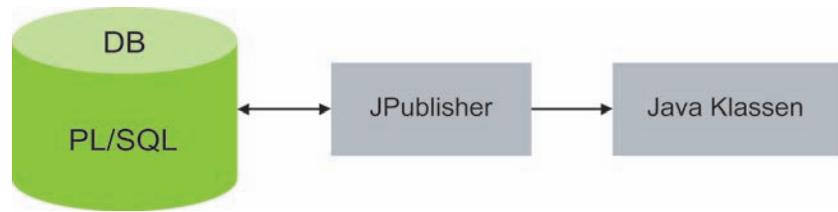


Abbildung 5: Mit JPublisher lassen sich Java-Wrapper-Klassen für PL/SQL generieren.

Datenquellen in OC4J einrichten

Um ein PL/SQL-Package als Web-Service zur Verfügung zu stellen, muss zunächst im OC4J eine Datenquelle eingerichtet werden. Eine Datenquelle im OC4J kann in der Datei data-sources.xml im Verzeichnis ORACLE_HOME\j2ee\home\config eingerichtet werden. Die Abbildung 4 stellt einen Auszug aus der Datei data-sources.xml dar.

Java-Wrapper-Klassen generieren

Im nächsten Schritt werden für die Prozeduren des PL/SQL-Packages die so genannten Wrapper-Klassen benötigt. Diese Wrapper-Klassen werden mithilfe des Programms JPublisher in der Programmiersprache Java generiert (siehe Abbildung 5). Der Vorteil der Wrapper-Klassen ist, dass die vorhandenen PL/SQL-Prozeduren nicht in einer anderen Programmiersprache codiert werden müssen und so in PL/SQL wiederverwendet werden können. Um die Wie-

derverwendbarkeit der PL/SQL-Prozeduren in der Service-orientierten Architektur zu ermöglichen, verfügen die Wrapper-Klassen über so genannte Wrapper-Methoden, die zum Aufrufen der eigentlichen PL/SQL-Prozeduren dienen.

WebServicesAssembler

Zur Erstellung der Wrapper-Klassen mithilfe des Programms JPublisher kann das Programm WebServicesAssembler verwendet werden. Das Programm WebServicesAssembler ruft implizit das Programm JPublisher auf und fasst die mit JPublisher erzeugten Wrapper-Klassen in einem EAR-Archiv zusammen. Bei dem EAR-Archiv handelt es sich um ein spezialisiertes Archiv, mit dem J2EE-konforme Unternehmensanwendungen für das Deployment auf einen Application Server verpackt werden.

```
<web-service>
  <display-name>PLSQL Web Service</display-name>
  <description>PLSQL Beispiel ORDIX Seminarpreise</description>
  <destination-path>./seminarPreiseService.ear</destination-path>
  <temporary-directory>/tmp4</temporary-directory>
  <context>/seminarpreisews</context>
  <stateless-stored-procedure-java-service>
    <jar-generation>
      <schema>mf/mf</schema>
      <db-url>jdbc:oracle:thin:@localhost:1521:ORCL10</db-url>
      <db-pkg-name>pck_seminar</db-pkg-name>
    </jar-generation>
    <uri>/seminarPreiseService</uri>
    <database-JNDI-name>jdbc/OracleDS</database-JNDI-name>
  </stateless-stored-procedure-java-service>
  <wsdl-gen>
    <wsdl-dir>wsdl</wsdl-dir>
    <option name="force">>true</option>
    <option name="httpServerURL">http://localhost:8891</option>
  </wsdl-gen>
  <proxy-gen>
    <proxy-dir>proxy</proxy-dir>
    <option name="include-source">>true</option>
  </proxy-gen>
</web-service>
```

Abbildung 6: Auszug aus einer Konfigurationsdatei config.xml für das Programm WebServicesAssembler.

```
java -jar %ORACLE_HOME%\webservices\lib\WebServicesAssembler.jar -config
config.xml
```

Abbildung 7: Aufruf von WebServicesAssembler.

```
java -jar %ORACLE_HOME%\j2ee\home\admin.jar ormi://localhost admin welcome
-deploy -file ./seminarPreiseService.ear -deploymentName seminarPreiseService

java -jar %ORACLE_HOME%\j2ee\home\admin.jar ormi://localhost admin welcome
-bindWebApp seminarPreiseService seminarPreiseService_web default-web-site
/seminarpreisews
```

Abbildung 8: Aufruf von admin.jar, um in einer OC4J-Umgebung einen Web-Service bereitzustellen.

```
Syntax
=> http://host:port/context/URI

Beispiel
=> http://localhost:8891/seminarpreisews/seminarPreiseService
```

Abbildung 9: URL-Syntax zum Testen von Web-Services.

Das EAR-Archiv beinhaltet neben den Wrapper-Klassen noch weitere Dateien wie `application.xml` oder ein WSDL-Dokument. Die Datei `application.xml` gibt an, über welche Komponenten das EAR-Archiv verfügt. Das WSDL-Dokument enthält dagegen unter anderem Informationen über die Parameter, Datentypen und Rückgabewerte der Methode, die als Web-Service aufgerufen werden können.

Aufruf von `WebServicesAssembler`

Für den Aufruf von `WebServicesAssembler` wird eine XML-Konfigurationsdatei benötigt. Diese enthält Informationen über den zu veröffentlichenden PL/SQL-Web-Service. In dieser Datei werden unter anderem die Datenbank-Verbindungsinformationen und der Prozedurname des PL/SQL-Packages hinterlegt. Die Abbildung 6 stellt eine XML-Beispielkonfigurationsdatei dar, und Abbildung 7 zeigt den Aufruf von `WebServicesAssembler`.

Web-Services auf den Oracle Application Server übertragen

Das mithilfe des Programms `WebServicesAssembler` erzeugte EAR-Archiv wird anschließend auf dem Application Server bereitgestellt. Ein Web-Service kann auf einen Application Server unter anderem mit den Programmen `Enterprise Manager` oder `DCMCTL` übertragen werden. In einer Standalone OC4J-Umgebung kann zusätzlich noch das Programm `admin.jar` verwendet werden (siehe Abbildung 8).

Bereitgestellte Web-Services testen

Zur Überprüfung des bereitgestellten Web-Services kann die in der Abbildung 9 dargestellte URL-Syntax verwendet

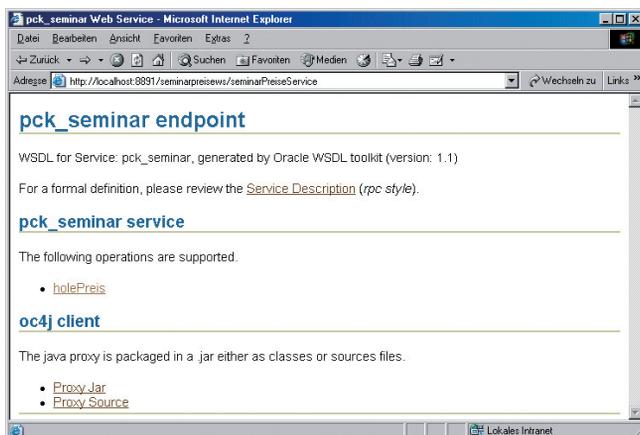


Abbildung 10: Web-Services Testseite des "pck_seminar" Services.

werden. Durch die Eingabe der URL in den Web-Browser erscheint eine Testseite mit allen Operationen des Web-Services (siehe Abbildung 10). Diese Operationen können einzeln aufgerufen und getestet werden (siehe Abbildung 11). Das Ergebnis wird in Form einer Simple Object Access Protocol (SOAP) Response geliefert (siehe Abbildung 12).

Im nächsten Teil werden wir erläutern, wie Web-Services mit der Datenbank-Programmiersprache PL/SQL aufgerufen werden können.

Glossar	
AS	Application Server. AS sind Middleware-Komponenten, in denen die Geschäftslogik implementiert wird.
DCMCTL	DCMCTL ist ein Werkzeug zur Administration von Oracle Application Servern. Mit diesem Werkzeug ist unter anderem eine Automatisierung von administrativen Aufgaben wie dem Starten und Beenden eines Application Servers oder auch das Deployment von Anwendungen möglich.
Deployment	Installation einer Application Server-Anwendung auf einem AS.
EAR	Enterprise Application Archive. EAR ist ein spezialisiertes Archiv, mit dem J2EE-konforme Unternehmensanwendungen für das Deployment auf einen AS verpackt werden.
Firewall	Eine Firewall ist eine Software oder auch Hardwarelösung, die ein Computernetzwerk oder einen einzelnen Computer vor unerwünschten Zugriffen aus dem Internet schützen soll.
OC4J	Oracle Application Server Containers for J2EE. OC4J ist die Java-Komponente eines Application Servers, in der Java-Anwendungen ausgeführt werden.
SOAP	Simple Object Access Protocol. SOAP ist ein Protokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote Procedure Calls durchgeführt werden können.
UDDI	Universal Description, Discovery and Integration. UDDI ist ein Verzeichnisdienst zur Registrierung von Web-Services, der das dynamische Finden des Web-Services durch den Konsumenten ermöglicht.
WSDL	Web Services Description Language. WSDL definiert einen Plattform-, Programmiersprachen- und Protokoll-unabhängigen XML-Standard zur Beschreibung von Web-Services, die einen Austausch von Nachrichten ermöglichen.
XML	Extensible Markup Language. XML ist eine so genannte META-Sprache zur Beschreibung von Dokumenten. Ein Vorteil von XML ist der vereinfachte Austausch von Daten, da XML-Formate in einer strengen Syntax definiert werden können und so die Implementierung von zuverlässigen Schnittstellen erlauben.

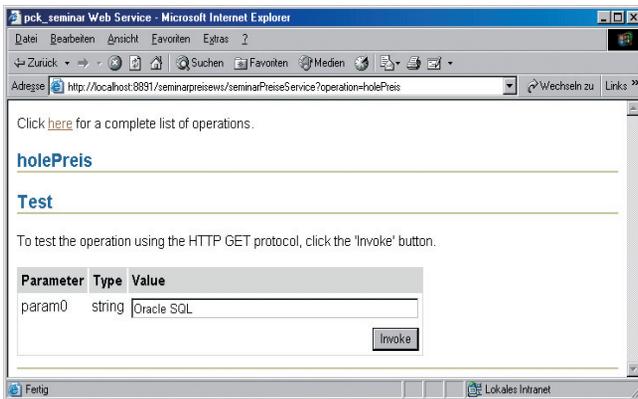


Abbildung 11: Web-Services Testseite zum Testen der "holePreis," Operation.

Um die in diesem Beitrag aufgeführten Beispiele zu testen, muss der Oracle Application Server Containers for J2EE 10g (OC4J) installiert sein. Alternativ kann auch der von Oracle zur Verfügung gestellte OC4J Standalone Distributions verwendet werden (s. <http://technet.oracle.com>). Dieser zeichnet sich durch eine einfache und mit wenig Aufwand verbundene Installation aus. Außerdem wird eine Oracle 10g Datenbank benötigt.

Kontakt:
 Markus Fiegler
 info@ordix.de

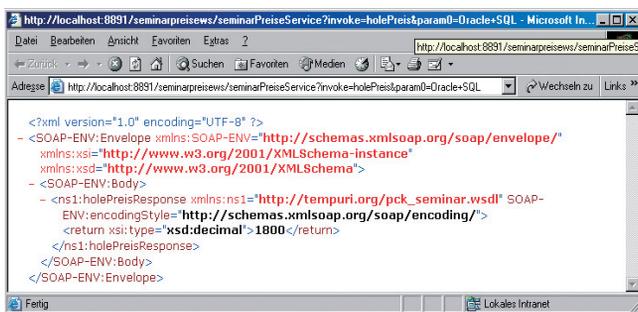


Abbildung 12: Beispiel einer SOAP-Response.