

„Wolkig bis heiter“ – APEX als Drehkreuz für Web-Service-Anwendungen

Andreas Wismann
WHEN OTHERS
D-41564 Kaarst

Schlüsselworte

APEX, Web Service, PL/SQL, API, APEX_WEB_SERVICE, REST, SOAP

Einleitung

„APEX beherrscht Web Services“ – interessant.

Was kennzeichnet einen Web Service? Welche Möglichkeiten eröffnen sich, wenn wir Oracle Application Express zusammen mit Web Services verwenden? Welche Verfahren beherrscht APEX und wie aufwändig ist es, sie zu implementieren?

Diese Fragen klärt die Präsentation, zusammen mit vielen Live-Beispielen.

Web Services: ein kurzer Abriss

Grundsätzlich ist ein Web Service einfach eine Datenquelle, die über das Internetprotokoll angesprochen wird. Der Client stellt eine Anfrage und wählt dabei die Art der Information aus, und der Server liefert die Daten zurück. Das Datenformat für die Kommunikation reicht von simplen ASCII-Zeichenfolgen über kommaseparierten Text, HTML, XML, JSON, RSS bis zu binären Codierungsformen. Mit anderen Worten: Solange Client und Server miteinander auf einer Wellenlänge sind, können sie „alles Mögliche“ vereinbaren und übertragen. Es gibt natürlich anerkannte Standards für typische Anwendungsfälle, und die beherrscht („natürlich“) auch APEX.

Hauptsächlich muss zwischen SOAP („*Simple Object Access Protocol*“) und RESTful Web Services („*REpresentational State Transfer*“) unterschieden werden. Erstere haben sozusagen „den Stein ins Rollen“ gebracht, und zwar schon vor der Jahrtausendwende. Ursprünglich ins Leben gerufen durch Microsoft, stellte SOAP die Alternative zu Serviceprotokollen wie DCOM und CORBA dar, die auf binären Formaten beruhen und sich daher nicht ideal über das Internet vermitteln lassen. Stattdessen benutzt SOAP XML als Transportmittel. Alle Anfragen und Antworten sind immer in eine XML-Struktur eingebettet. Das SOAP-Protokoll ist sehr mächtig, streng formalisiert und verzeiht keine Fehler - daher ist es vergleichsweise aufwändig zu implementieren und eher umständlich zu handhaben. Schon einfache SOAP-Konversationen erfordern eine Menge Code. Typische Beispiele für SOAP-basierte Web Services stammen aus der Finanzbranche, etwa um Wertpapierkurse abzurufen. Glücklicherweise bietet APEX die Generierung von Formularen und Report auf Basis SOAP assistentengesteuert an (siehe weiter unten)

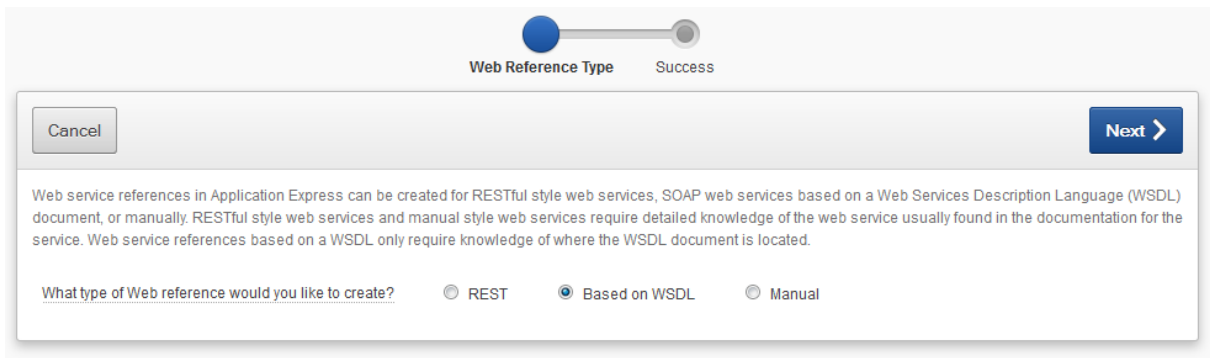
Um das Jahr 2002 kamen die RESTful Web Services hinzu. Ein REST-Request kann schon aus dem Aufruf einer URL mit oder ohne weitere Parameter bestehen. In der Antwort können sich vielfältige Formate befinden, die entweder auf Standards wie XML oder JSON basieren, aber auch einfach strukturierte Textmitteilungen bis hin zu kompletten „Eigenkreationen“.

APEX-Bordmittel für Web Services

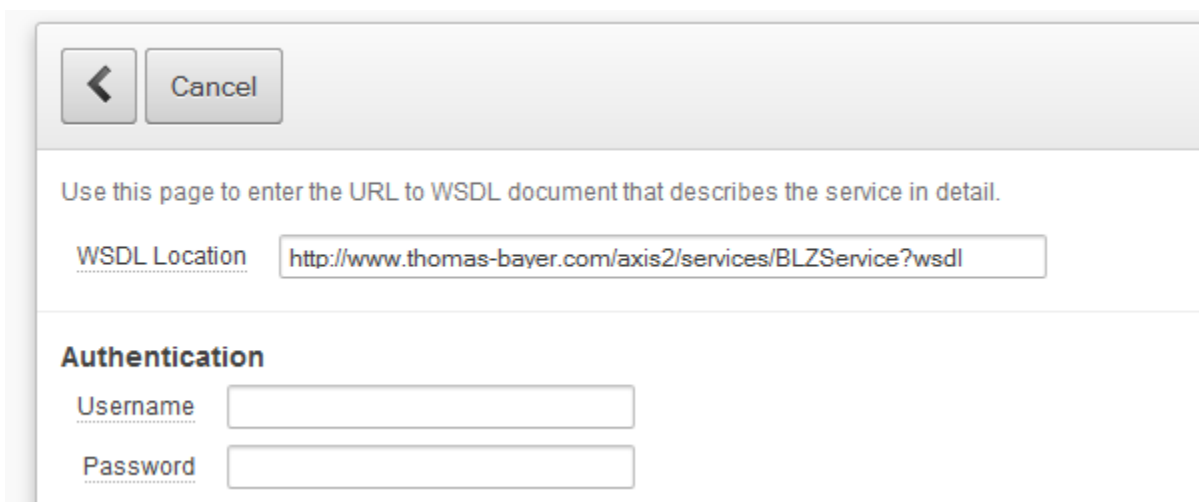
Mit dem Package APEX_WEB_SERVICE stellt Application Express Version 4.2 PL/SQL-Routinen für den Abruf von Web Services bereit. Dabei werden sowohl SOAP als auch REST unterstützt. Dies ist also die Toolbox für alle, die sich eingehender mit den Parametern und Daten beschäftigen wollen, die es bei der Web Service Kommunikation zu beachten gilt (weiter unten stelle ich Ihnen aber auch den „einfachen“ Ansatz über einen Assistenten vor). In APEX_WEB_SERVICE existieren Methoden, um Web Services aufzurufen, Cookies und HTTP-Header zu speichern und die Antworten zu parsen. Obwohl das SOAP-Format eigentlich immer XML ist, kann dieses Package die Antwortdaten des entfernten Servers praktischerweise (neben XML) auch als VARCHAR2 und CLOB an die PL/SQL-Umgebung zurückliefern.

Einen SOAP Web Service abfragen

In den Shared Components > Web Service References führt ein Wizard (Sie sehen, ich verwende hauptsächlich die englischsprachige APEX-Installation) ohne eine einzige Codezeile zu einem Formular, das auf einem Web Service basiert. Dazu nehme ich hier als Beispiel einen freien Web SOAP Web Service, der bei Eingabe der (alten) Bankleitzahl den Banknamen, die BIC, den Ort und die Postleitzahl der Bankfiliale zurückgibt:



Der Knackpunkt ist hierbei immer die WSDL („Web Service Description Language“), also die Datei, die eine Auflistung aller in diesem Web Service verfügbaren Dienste beinhaltet. APEX ist in der Lage, die WSDL automatisch zu analysieren. Die URL gibt der Anbieter des Services bekannt.



Der Wizard liest die WSDL, wertet sie live aus und registriert diesen Web Service für die weitere Verwendung in APEX. Sie können sogar unmittelbar anschließend weiterklicken und erhalten innerhalb weniger Sekunden ein funktionstüchtiges Formular, mit dem der Web Service abgefragt werden kann:

getBank		Submit
Blz	<input type="text" value="30040000"/>	
Bezeichnung	<input type="text" value="Commerzbank"/>	
Bic	<input type="text" value="COBAEEDDXXX"/>	
Ort	<input type="text" value="Düsseldorf"/>	
Plz	<input type="text" value="40002"/>	

(In diesem Fall gibt man die BLZ ein und klickt auf Submit). Das Formular besitzt den vom Wizard generierten Prozess „Web Service Request“, der beim Abschicken der Seite den Web Service kontaktiert, die Antwort des Servers entgegennimmt, diese parst und in die Page Items einträgt.

Für den Anwendungsprogrammierer stellt sich der Vorgang fast genauso dar, als hätte er die Werte aus der Datenbank geholt.

Web Services über AJAX per Application Process aufrufen

Der eigentliche Clou von Web Services besteht natürlich darin, dass sie aus einer Webseite heraus per AJAX aufgerufen werden können. Es ist also nicht unbedingt nötig, ein APEX-Formular zu submitten, um an die Ergebnisse heranzukommen. Dies wird lediglich durch den Assistenten so vorgegeben.

Ein Web Service kann entweder komplett per JavaScript angesprochen und ausgewertet werden, oder man bemüht PL/SQL (zum Beispiel unter Zuhilfenahme des Package APEX_WEB_SERVICE) und lässt das Ergebnis wiederum durch JavaScript parsen. Der jQuery-Code, der auf ein bestimmtes Ereignis in der APEX-Seite wie etwa einen Button-Klick hören könnte, sieht im Prinzip so aus:

```
$.ajax({
  type: "POST",
  url: "wwv_flow.show",
  data: {
    p_flow_id : $('#pFlowId').val(),
    p_flow_step_id : $('#pFlowStepId').val(),
    p_instance : $('#pInstance').val(),
    x01 : 'ein_Wert', // apex_application.g_x01
    x02 : 'ein_anderer_Wert', // apex_application.g_x02
    // ... bis .g_x10
    p_request : "APPLICATION_PROCESS=XYZ"
  },
  dataType: "json", // oder xml, text
  success: function (data) {
    /* Resultat verarbeiten ... */
  }
});
```

Der Weg über den Server (im Beispiel: APPLICATION_PROCESS=XYZ) scheint zwar zunächst ein Umweg zu sein, hat aber einen entscheidenden Vorteil gegenüber rein clientseitiger Verarbeitung: Sie können Geschäftslogik auf der Serverseite hinterlegen und müssen solche Programmteile nicht in JavaScript nachprogrammieren. So könnte beispielsweise ein Web Service eine Reihe von

Kundennummern zurückliefern, Ihr APEX-Anwendungsprozess filtert aber noch vor dem clientseitigen Parsen des Ergebnisses bestimmte Zeilen mit PL/SQL-Mitteln heraus, je nach Kontext.

Einen APEX-Report zur Datenquelle für einen Web Service machen

APEX kann selbst auch Informationen als RESTful Web Service bereitstellen. Damit das funktioniert, muss die APEX-Instanz zunächst für RESTful Web Services konfiguriert werden. Hierzu wird bei Installationen, die noch den APEX Listener alter Schule verwenden, das Skript `apex_rest_config.sql` ausgeführt werden (siehe Dokumentation des Listeners). In der Datenbank 11g (Release 1 oder 2) zusammen mit APEX 4.2.x müssen über das Package `DBMS_NETWORK_ACL_ADMIN` das `connect`-Privileg an den APEX-User `APEX_040200` vergeben werden, bzw. es kommt der direkte Nachfolger des APEX Listeners, Oracle REST Data Services (kurz ORDS) zum Einsatz.

Anschließend lässt sich quasi „mit einem Klick“ ein klassischer APEX-Report zu einer RESTful Web-Service-Datenquelle umfunktionieren.

The screenshot shows the 'Attributes' configuration page in APEX. The 'Static ID' field contains 'demoProductInfo'. Below it are several empty text input fields for 'Region Attributes', 'Region Image', and 'Image tag attributes'. The 'Region Display Selector' is a dropdown menu set to 'No'. The 'Region HTML table cell attributes' field is highlighted with a blue arrow. The 'Enable RESTful Access' dropdown is set to 'Yes'. At the bottom, the 'REST Access URL' is displayed as 'http://apex.oracle.com/pls/apex/apex_rest.getReport?app=25499&page=1000&reportid=demoProductInfo'.

Die XML-Ausgabe eines solchen REST-fähigen Reports, der von externen Anwendungen unter der angegebenen APEX-URL aufgerufen werden kann, liest sich wie folgt:

```
<?xml version="1.0"?>

<ROWSET>
  <ROW>
    <PRODUCT_ID>41</PRODUCT_ID>
    <PRODUCT_NAME>Business Shirt</PRODUCT_NAME>
    <PRODUCT_DESCRIPTION>Wrinkle-free cotton business
shirt</PRODUCT_DESCRIPTION>
    <CATEGORY>Mens</CATEGORY>
    <PRODUCT_AVAIL>Y</PRODUCT_AVAIL>
    <LIST_PRICE>50</LIST_PRICE>
    <MIMETYPE>image/jpeg</MIMETYPE>
    <FILENAME>shirt.jpg</FILENAME>
    <IMAGE_LAST_UPDATE>18.09.2014</IMAGE_LAST_UPDATE>
  </ROW>
  <ROW>
    <PRODUCT_ID>42</PRODUCT_ID>
```

```
<PRODUCT_NAME>Trousers</PRODUCT_NAME>
<PRODUCT_DESCRIPTION>Black trousers suitable for every business
man</PRODUCT_DESCRIPTION>
<CATEGORY>Mens</CATEGORY>
<PRODUCT_AVAIL>Y</PRODUCT_AVAIL>
<LIST_PRICE>80</LIST_PRICE>
<MIMETYPE>image/jpeg</MIMETYPE>
<FILENAME>pants.jpg</FILENAME>
<IMAGE_LAST_UPDATE>18.09.2014</IMAGE_LAST_UPDATE>
</ROW>
</ROWSET>
```

APEX rendert den Report mit Hilfe des hier erkennbaren Musters:

```
<ROWSET><ROW><SPALTENNAME>Wert</SPALTENNAME></ROW></ROWSET>
```

Auf diese Weise können Sie zum Beispiel firmeninterne Web Services innerhalb weniger Minuten bereitstellen.

Die Präsentation

Weitere Beispiele und Live-Demonstrationen zum Thema „Web Services in APEX“ warten auf Sie in meinem Vortrag auf der DOAG 2014 in Nürnberg am Donnerstag, 20. November, um 13:00 Uhr im Raum „Istanbul“. Die Folien und ein aktualisiertes Manuskript stehen auf meiner Website zum Download zur Verfügung.

Kontaktadresse:

Andreas Wismann
WHEN OTHERS
Hirschstraße 10
D-41564 Kaarst

Telefon: +49 (0) 2131-314 9966
E-Mail: wismann@when-others.com
Internet: when-others.com
Twitter: @whenothers