

Rich-Internet-Applications mit jQuery und dem APEX Listener

Klaus Friemelt
MT AG
Ratingen

Schlüsselworte: APEX Listener, Oracle GlassFish, Oracle WebToolkit, jQuery, RIA

Einleitung

Seit Zeiten der Oracle WebDB, dem Urahn des aktuellen APEX, existiert das Oracle Webtoolkit – die drei Datenbankpackages SYS.HTP, SYS.HTF und SYS.OWA_UTIL, die im Zusammenspiel mit der Apache-Komponente mod_plsql Webseiten aus der Datenbank erzeugen.

Mit dem Application Express Listener steht ein java-basierter Nachfolger des bewährten mod_plsql bereit. Der Listener ist lauffähig in verschiedenen J2EE-Umgebungen wie dem Oracle Weblogic Server (WLS), dem Oracle GlassFish Server und dem Oracle OC4J-Container. Im early-adopter-Release des APEX Listeners wurde noch der Tomcat Server unterstützt, welcher aber im aktuellen Release dem GlassFish gewichen ist.

Die Möglichkeit, das „embedded plsql gateway“ (EPG) für WebToolkit-Anwendungen zu verwenden, ist für produktive Umgebungen aus Sicherheitsgründen nicht zu empfehlen.

jQuery ist ein JavaScript-Framework, um browserunabhängige, Ajax-getriebene Oberflächen einfach zu erzeugen. Durch jQuery-Erweiterungen kommt man schnell zu echten Rich-Internet-Applications (RIA).

Es gibt nun einige Szenarien, für die ein einfaches Webfrontend hilfreich ist, ohne dass man direkt eine komplexe Applikation anstrebt. Denkbare Beispiele seien hier die Visualisierung von Applikations-Logfiles oder eine einfache Dateiverwaltung im Browser.

Hier wird gezeigt, wie einfach mit dem WebToolkit, dem GlassFish und dem APEX Listener datenbankgetriebene Webanwendungen realisiert werden können.

jQuery

jQuery ist ein JavaScript-Framework, welches immer weitere Verbreitung findet. Es arbeitet browserunabhängig, so dass sich der Entwickler keine Gedanken machen muss, wie sich die Anwendung in Anbetracht der wachsenden Browservielfalt darstellt und verhält. Darüber

hinaus bietet es einfachste Ajax-Integration, welche der Grundstein für moderne Anwendungen ist.

Auch ein anderes Problem wird mit jQuery einfach gelöst: oftmals startet eine Webseite beim Laden mit dem onLoad()-Event JavaScripte, die auf Objekte der Seite zugreifen, die zum Eventzeitpunkt noch gar nicht existieren. jQuery stellt einen Einstiegspunkt bereit, an dem garantiert alle Objekte des DOM-Baumes (der ja eine Webseite beschreibt) verfügbar sind.

Der Zugriff auf diese Elemente des DOM-Baumes erfolgt mit Selektoren, über die mit jQuery alle Eigenschaften ermittelt und modifiziert werden können.

Für jQuery existieren unzählige Erweiterungen, mit denen man schnell zu leistungsfähigen, interaktiven Webanwendungen kommt, welche häufig als „rich internet applications“ (RIA) bezeichnet werden.

Auf jQuery aufbauende Frameworks sind beispielsweise jQueryUI (u.a. „Drag & Drop“-Unterstützung, Animationen, etc) und jsTree zur Visualisierung hierarchischer Daten.

Downloads, Dokumentation und Demos sind jeweils zu finden unter <http://jquery.com>, <http://jqueryui.com> und <http://www.jstree.com>.

Installation und Einrichtung des Oracle GlassFish Servers

Es wird hier der Oracle GlassFish Server als schlankere Lösung gegenüber dem Oracle WebLogic-Server betrachtet. Zum Einsatz kommt die Version 3.0.1. Der Download findet sich unter <http://www.oracle.com/technetwork/middleware/GlassFish/overview/index.html>.

Zur Installation die zip-Datei ins Zielverzeichnis entpacken und von der Konsole den Server starten mit „<Zielverzeichnis>/glassfishv3/glassfish/bin sh startsrv“. Anschließend lässt sich die Administrationskonsole im (lokalen) Browser starten unter <http://localhost:4848/>. Standardmäßig erscheint die Konsole anmeldungsfrei. Zum Absichern unter dem Menüpunkt Enterprise Server – Administrator Password ein Kennwort vergeben.

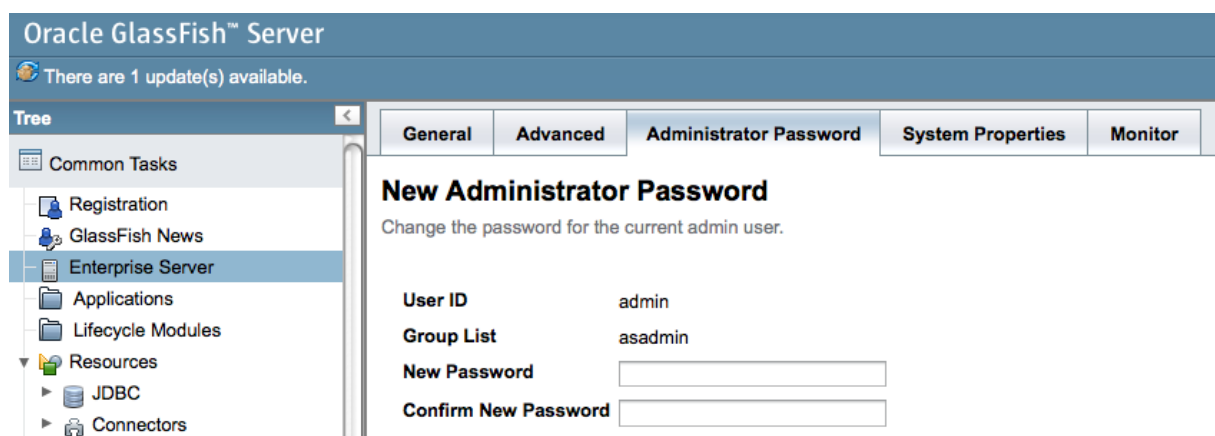


Abb. 1: Administrator Kennwort vergeben

Die nächsten Schritte zum Deployment des APEX Listeners für alle drei möglichen J2EE-Umgebungen sind detailliert beschrieben in dem PDF-Dokument „Installation Guide“, welches verfügbar ist unter:

<http://www.oracle.com/technetwork/developer-tools/apex-listener/downloads/index.html>.

Auch der Listener selbst steht dort zum Herunterladen bereit.

Installation und Einrichtung des APEX Listeners

Der APEX Listener besteht aus einer 4.2 MB großen war-Datei, die auf den Server verteilt wird. Zuvor müssen auf dem GlassFish Server einige Benutzer angelegt werden. Dies geschieht auf der Admin-Konsole unter dem Menüpunkt „Security“. Zunächst ist aber die Checkbox „Default Principal To Role Mapping“ zu aktivieren.

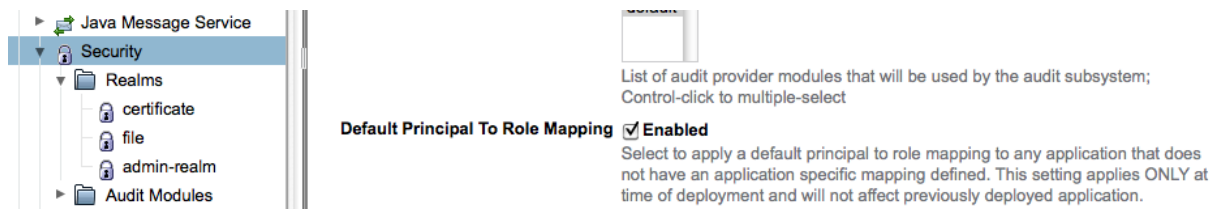


Abb. 2: Aktivierung des „Default Principal To Role Mappings“

Diese bewirkt, dass für die Authentifizierung letztlich nur die Datei web.xml angepasst werden muss. Ansonsten ist manuell eine Datei sun-web.xml anzulegen mit security-role-mapping-Einträgen. Als nächstes für den Realm „file“ unter dem Button „Manage Users“ drei Benutzer anlegen. Die User ID ist dabei frei wählbar, wichtig ist aber die Gruppenbezeichnung!

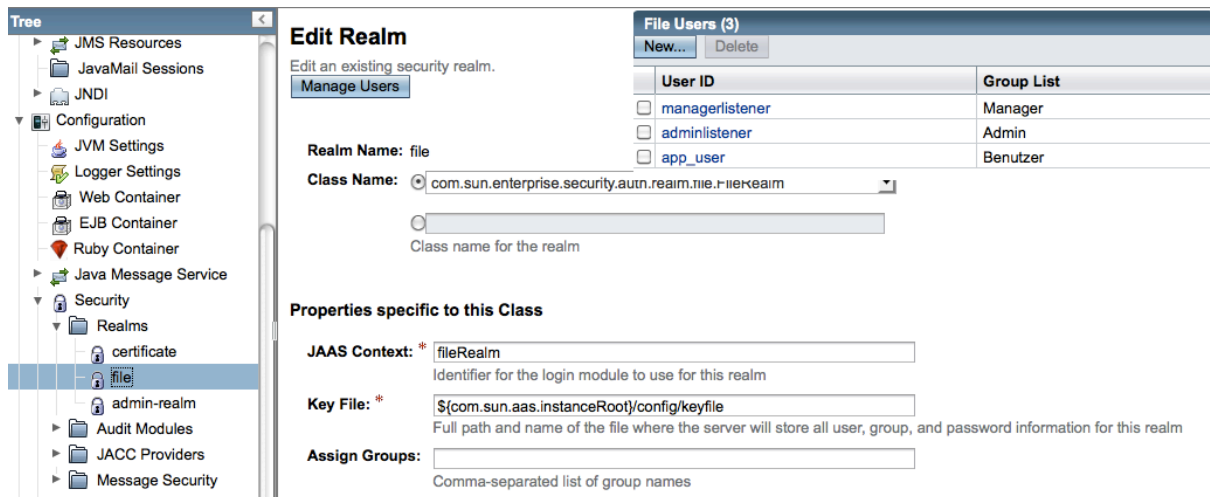


Abb. 3: Anlegen der nötigen Benutzer

Nun kann die war-Datei verteilt werden. Dazu unter dem Menüpunkt „Applications“ auf den Button „Deploy“ klicken und unter „Location“ den Speicherort der Datei apex.war auswählen und mit „OK“ bestätigen.

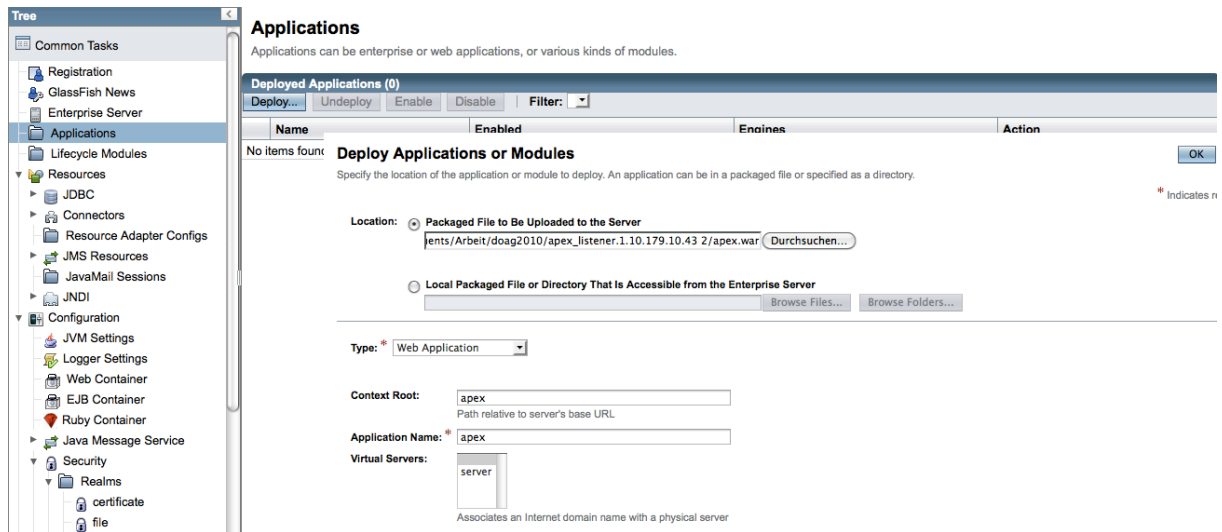


Abb. 4: Deployment der apex.war-Anwendung

Anschließend kann der Listener über seine eigene Oberfläche administriert werden. Der Aufruf erfolgt über <http://localhost:8080/apex/listenerConfigure>.

ORACLE® Application Express Listener

Administration

Connection
Security
Caching
Pre-Post Processing
Status
Miscellaneous

Database Connection

Username:

Password:

Connection Type: Basic

Hostname:

Port:

SID

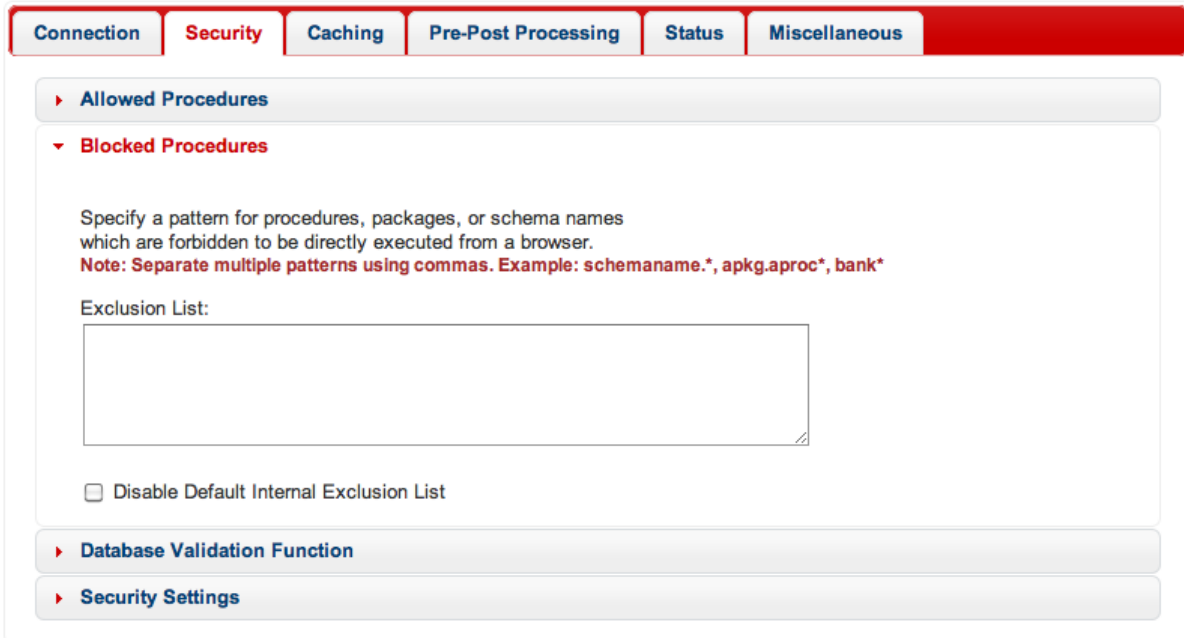
Service name

▶ JDBC Settings

Apply Cancel

Abb. 5: Start der Administrationsoberfläche

Zunächst werden die Verbindungsdaten zur Datenbank erfasst, darüber hinaus gibt es nahezu die gleichen Optionen wie bei einem database access descriptor (dad) unter mod_plsql.



The screenshot shows the Oracle APEX Administration interface with the 'Security' tab selected. The interface has a red header bar with tabs for 'Connection', 'Security', 'Caching', 'Pre-Post Processing', 'Status', and 'Miscellaneous'. Under the 'Security' tab, there are three expandable sections: 'Allowed Procedures', 'Blocked Procedures', and 'Database Validation Function'. The 'Blocked Procedures' section is expanded, showing instructions to specify a pattern for procedures, packages, or schema names that are forbidden to be directly executed from a browser. A note states: 'Note: Separate multiple patterns using commas. Example: schemaname.*, apkg.aproc*, bank*'. Below this is an 'Exclusion List' text area and a checkbox labeled 'Disable Default Internal Exclusion List'. The 'Security Settings' section is also visible at the bottom.

Abb. 6: Sicherheit durch Blocken gelisteter Datenbank-Prozeduren

Es zeigt sich eine gravierende Abweichung vom alten mod_plsql: dort konnte man die Dokumenten-Tabelle beliebig deklarativ angeben, hier ist es die APEX-Tabelle `www_flow_file_objects$` im Schema `FLows_FILES`. Wenn man also gar kein APEX installiert hat, muss man für die Verwendung von Datei-Uploads mindestens diese Tabelle anlegen.

Nun muss noch ein Verzeichnis angelegt werden für alle Grafiken, Stylesheets und JavaScripte, die von der Anwendung zur Laufzeit benötigt werden. Dazu im Verzeichnis `glassfishv3/glassfish/domains/domain1/docroot/` den Ordner „i“ anlegen.

Anpassungen in der web.xml

Die im Paket `apex.war` enthaltene Datei `/WEB-INF/web.xml` ist für die Verwendung mit APEX ausgelegt. Sind beispielsweise alternative Namen und Orte für das Bilderverzeichnis nötig, müsste diese `web.xml`-Datei entsprechend angepasst werden. Auch werden alle Servlet-Aufrufe außer `/listenerAdmin` und `/listenerStatus` ohne Authentifizierung an das `ModApexServlet` durchgereicht, weil von einer APEX-internen Benutzerverwaltung ausgegangen wird. Möchte man für seine reine WebToolkit-Anwendung eine Authentifizierung erreichen, ist in der `web.xml` ein entsprechender `security-constraint` anzulegen.

Die Security-Constraints werden schon beim Deployment der apex.war ausgewertet. Deshalb muss für diese Art der Anpassung zunächst die apex.war entpackt werden in ein neues Verzeichnis mit dem Namen apex. Darin kann die Datei WEB-INF/web.xml editiert werden. Zum Neuverpacken in der Konsole in das Verzeichnis navigieren und das folgende Kommando absetzen: `jar cf ../apex.war *`

Die neue apex.war-Datei kann dann auf den GlassFish-Server neu verteilt werden. Wenn man nur nicht benötigte servlet-mappings entfernen möchte, geht das durch direktes Editieren einer deployten web.xml, es muss dann lediglich die Anwendung neu gestartet werden.

```

- <web-app id="WebApp_ID" version="2.4" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/
  <display-name>ApexMod</display-name>
+ <servlet></servlet>
+ <servlet-mapping></servlet-mapping>
+ <servlet-mapping></servlet-mapping>
  <!-- Admin -->
+ <servlet></servlet>
+ <servlet-mapping></servlet-mapping>
+ <servlet-mapping></servlet-mapping>
+ <servlet></servlet>
- <servlet-mapping>
  <servlet-name>Images</servlet-name>
  <url-pattern>/i/*</url-pattern>
</servlet-mapping>
+ <context-param></context-param>
+ <welcome-file-list></welcome-file-list>
- <security-constraint>
  - <web-resource-collection>
    <web-resource-name>ALLES</web-resource-name>
    <url-pattern>/*</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
  - <auth-constraint>
    <role-name>Benutzer</role-name>
  </auth-constraint>
</security-constraint>
- <login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>Anmeldung beim Glassfish auf MBP</realm-name>
</login-config>
- <security-role>
  <role-name>Benutzer</role-name>
</security-role>
</web-app>

```

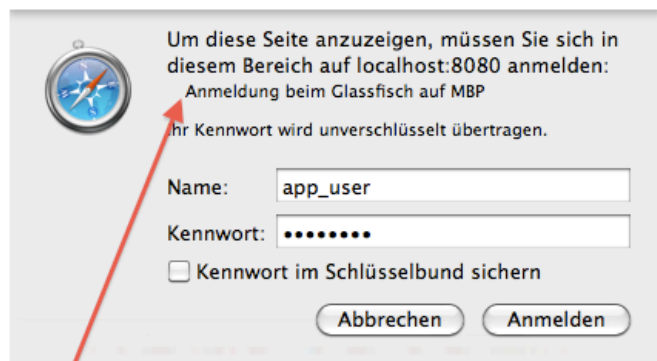


Abb. 7: Anpassung von Grafik-Verzeichnis und security-constraint in der web.xml-Datei

Anwendungsbeispiele für jQuery

Das erste Beispiel für die Verwendung von jQuery haben wir bereits vorab gesehen: Die Administrations-Oberfläche des APEX Listeners verwendet die jQueryUI-Komponenten „Accordion“ und „Tabs“, zu finden unter <http://jqueryui.com/demos/>.

Das zweite Beispiel zeigt die elementaren Ajax-Fähigkeiten von jQuery anhand einer Bankleitzahlensuche. Beim Betätigen des Buttons „Bank suchen“ wird zur Bankleitzahl im

Hintergrund per Ajax nach dem dazugehörigen Banknamen gesucht und dieser im Ergebnisbereich angezeigt.

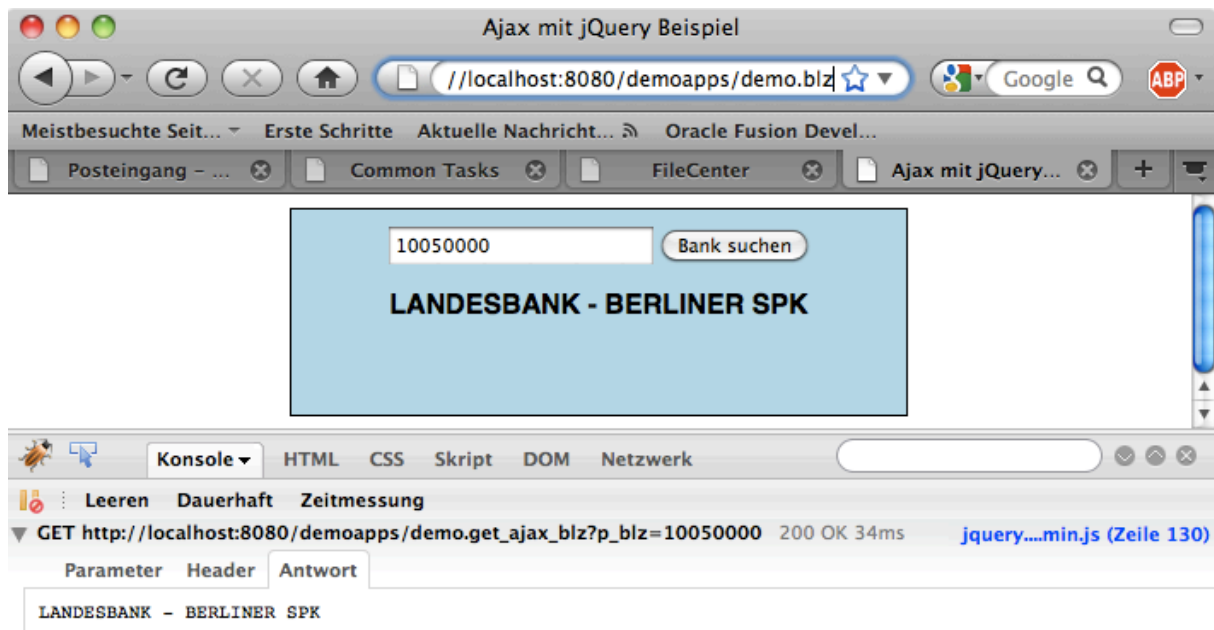


Abb. 8: Bankleitzahlensuche im Browser per Ajax

In der Firebug-Konsole sieht man die ausgehende Anfrage und die entsprechende Antwort. Die kommentierte Quelltextansicht zeigt die verwendeten jQuery-Methoden.

```

PROCEDURE blz
IS
BEGIN
  http.p('<html><head>
<title>Ajax mit jQuery Beispiel</title>
<script src="/i/jquery-1.4.2.min.js"></script>
<script>
$(document).ready( function(){
$("#suchen").click(function(){
$("#bankname p").load("demo.get_ajax_blz?p_blz="+$("#blz").val());
});
});
</script>

<style type="text/css">
#wrapper {
width: 340px; height: 100px;
padding: 10px; margin-top: 10px; margin:auto;
border: 1px solid black; text-align:center;
font-family:helvetica;font-weight:bold;
background:lightblue;color:black;
}
</style>
</head>
<body>
<div id="wrapper">
<input type="text" id="blz" value="" maxlength="8">
<input type="button" id="suchen" value="Bank suchen">
<div id="bankname"><p></p></div>
</div>
</body>
</html>');
END blz;

```

Annotations in the image:

- Startet, wenn Seite gerendert ist (points to `$(document).ready()`)
- onClick() für Button definieren (points to `$("#suchen").click()`)
- Ajax-Aufruf mit Parameter BLZ starten, und Ergebnis anzeigen (points to `$("#bankname p").load()`)
- Selektor \$ = jQuery (points to `$("#suchen")` and `$("#bankname p")`)
- Eingabefeld BLZ (points to `<input type="text" id="blz" value="" maxlength="8">`)
- Suchbutton (points to `<input type="button" id="suchen" value="Bank suchen">`)
- Ergebnisbereich „Bankname“ (points to `<div id="bankname"><p></p></div>`)

Abb. 9: PL/SQL-Prozedur zur Erzeugung der Seite mit htp.p()

Die serverseitigen Prozeduren, die von einem Ajax-Request angesprochen werden, sollten den korrekten „Mime-Type“ im HTTP-Header liefern. In unserem einfachen Beispiel wird das Ergebnis (die gefundene Bankbezeichnung) als einfacher Text erwartet, deshalb wird mit der WebToolkit-Prozedur OWA_UTIL.mimeheader() als Wert text/plain geliefert.

Würden komplexere Datenstrukturen im Format XML oder json („JavaScript object notation“) zurückgeliefert, ist entsprechend der Mime-Type text/xml oder application/json zu wählen.

```
PROCEDURE get_ajax_blz(  
    p_blz IN VARCHAR2)  
IS  
    CURSOR bank_csr  
    IS  
        SELECT UPPER (kurzbezeichnung) bank  
        FROM blz WHERE bankleitzahl = p_blz AND merkmal = 1;  
    r_bank_csr bank_csr%ROWTYPE;  
BEGIN  
    OPEN bank_csr;  
    FETCH bank_csr INTO r_bank_csr;  
    CLOSE bank_csr;  
    -- text/plain oder text/xml oder text/json  
    OWA_UTIL.mime_header ('text/plain', FALSE);  
    HTP.p ('Cache-Control: no-cache');  
    HTP.p ('Pragma: no-cache');  
    OWA_UTIL.http_header_close;  
    --  
    HTP.p (NVL(r_bank_csr.bank, 'unbekannt'));  
END get_ajax_blz;
```

Abb. 10: PL/SQL-Prozedur als Antwortfunktion für den Ajax-Request zur Bankleitzahlsuche

Zusammenfassung

Der Oracle GlassFish-Server läuft unter diversen Betriebssystemen wie z.B. Linux, OS X, Solaris oder Windows und ist unkompliziert in Installation und Einrichtung.

Beim APEX Listener zeigt sich, dass er aktuell noch sehr speziell auf APEX zugeschnitten ist und für beliebige WebToolkit-Anwendungen keinen vollständigen Ersatz für mod_plsql darstellt. Gerade das parallele Betreiben mehrerer DADs, wie man es unter mod_plsql kennt, lässt sich mit dem APEX Listener nur schwer abbilden.

jQuery vereinfacht als etabliertes Framework die Entwicklung von Ajax-getriebenen Anwendungen mit reichhaltigen Oberflächen extrem und ist sowohl für J2EE- als auch WebToolkit-Anwendungen geeignet. Besonderes Augenmerk verdient auch das Plugin-Repository für jQuery, welches unter <http://plugins.jquery.com/> zu finden ist.

Kontaktadresse:

Klaus Friemelt

MT AG

Balcke-Dürr-Allee 9

D-40882 Ratingen

Telefon: +49(0)2102-309 61-0
Fax: +49(0)2102-309 61-50
E-Mail klaus.friemelt@mt-ag.com
Internet: www.mt-ag.com