Apex & iPhone - Entwicklung mobiler Webanwendungen und Apps

Dr. Harry W. Trummer / Alexander Elsas
Goethe-Universität
Frankfurt am Main

Schlüsselworte

Apex, mobile Webanwendungen, Apps

1 Einleitung

Ein wichtiges Qualitätskriterium für eine moderne Hochschule ist das Ranking in den verschiedenen, auch internationalen, Vergleichen. Dabei wird als ein Faktor, neben der Bewertung von Lehre und Forschung, auch die Qualität der Bindung von Ehemaligen (Alumni) an die Hochschule bewertet. Deshalb organisieren die Hochschulen in Form von Förder- und Alumniorganisationen ihre Ehemaligen.

Je nach Größe der Alumniorganisation (beispielsweise bei der Goethe Finance Association (GFA) an der Universität Frankfurt über 2.000 Mitglieder) ist dann das Management von Mitgliedern und Planung und Durchführung von Veranstaltungen mit den Förderunternehmen eine umfangreiche Aufgabe, die sich mit Ad-Hoc-Excel-Tabellen nicht mehr realisieren lässt.

Bereits auf der DOAG 2010 wurde deshalb die Alumni-Management-Lösung GFA3 vorgestellt,[1] die auf Grundlage der Oracle Express Edition mit Application Express (Apex) implementiert wurde. Im Mittelpunkt der Anwendung steht dabei die enge Verzahnung von internen Prozessen (wie z. B. der Planung eines Events) und die Darstellung und Bewerbung auf der öffentlich verfügbaren Website http://www.gfa-frankfurt.org.

Die Anforderungen an eine derartige Website ändern sich natürlich im Laufe der Zeit: die meisten Studenten sind heutzutage mit Smartphone oder Notebook auf dem Campus präsent, insbesondere das iPhone (und auch die Android-Konkurrenz) hat eine derartige Verbreitung gefunden, dass der natürliche nächste Schritt in der Evolution die Präsenz auf dem mobilen Endgerät, sozusagen am „Point-of-Sale“, ist.

Dieser Beitrag soll deswegen aufzeigen, wie man von einer bestehenden Web-Präsenz zu einer für mobile Endgeräte optimierten Version und letztlich zu einer App auf dem Smartphone kommt.

---

[1] Vgl. Alexander Elsas / Dr. Harry W. Trummer: Erfolgreiches Alumni-Management mit Oracle Application Express, DOAG 2010.

2 Von Web zu App
2.1 Vorgehensweise

Für den Weg von einer herkömmlichen Webseite zu einer App bietet sich ein Vorgehen in vier Schritten an:

1. Funktionalitätsreduktion
2. Umsetzung als Web-App
3. Publizieren der Daten mit RSS-Feeds
4. Entwickeln der nativen App

Diese 4 Schritte werden in den folgenden Kapiteln näher dargestellt.

2.2 Funktionalitätsreduktion

Ausgangspunkt ist die in Abbildung 1 gezeigte Webseite der GFA.

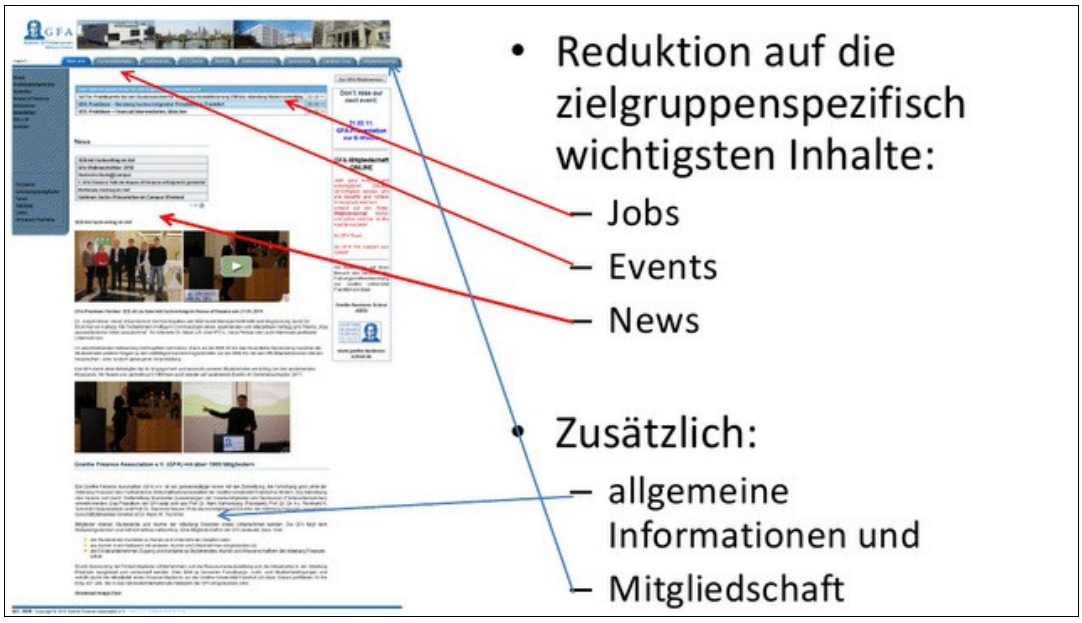

Abbildung 1: Funktionalitätsreduktion.

Erster Schritt ist dann die zielgruppenspezifische Einschränkung der gebotenen Informationsvielfalt auf die Besonderheiten eines mobilen Endgerätes. Unter Berücksichtigung der Zugriffszahlen auf verschiedene Bereiche der GFA-Seite und der Zielgruppe Studenten sind das die Bereiche

- News,
- Events,
- Jobs,

zusätzlich allgemeine Informationen zu GFA und Mitgliedschaft.

Somit sind 5 Elemente ausgewählt, die zunächst in einer geräteneutralen, aber für kleine Bildschirme optimierten Web-App-Version implementiert wurden.

2.3 Web-App
2.3.1 Überblick

Ein großer Vorteil aller relevanten mobilen Endgeräte ist die große Browser-Homogenität: sowohl für die iPhone-Plattform (iOS) als auch Android und beinahe alle anderen mobilen Plattformen (z. B. Bada) basiert der Webbrowser auf dem von KHTML abgeleiteten Webkit – eine Berücksichtigung von Browser-Besonderheiten entfällt somit, es wird letztlich nur für eine Browser-Plattform entwickelt.

Abbildung 2 zeigt die Seitenstruktur, die der bestehenden Anwendung als mobile Seiten hinzugefügt wurde.

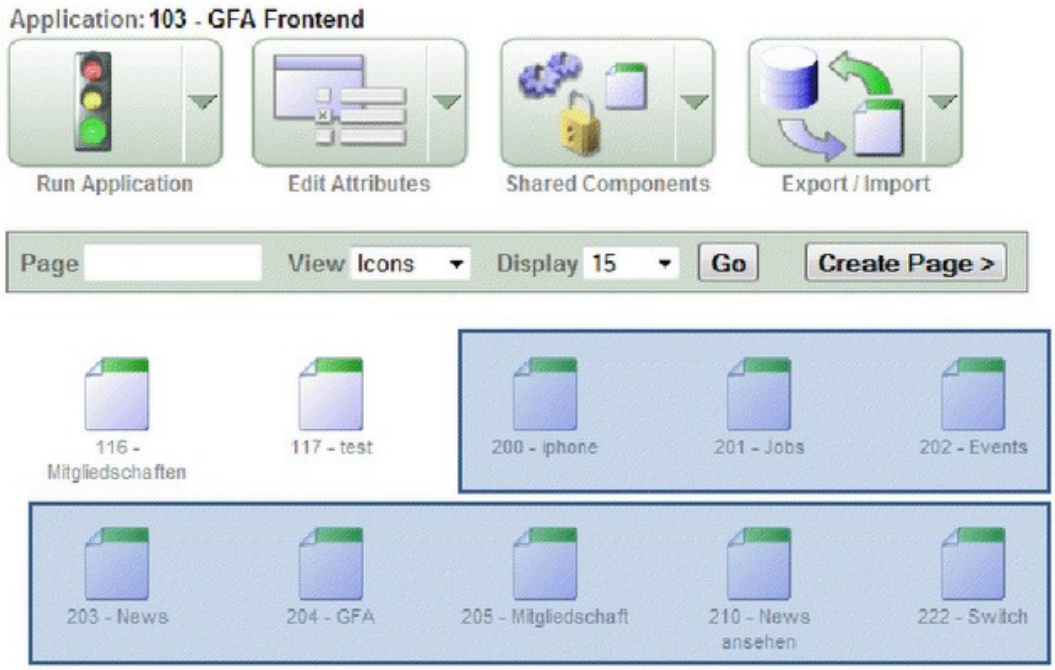

Abbildung 2: Mobile Seiten.

Detailaspekte zu den Seiten werden in den folgenden Kapiteln dargestellt.

2.3.2 Seiten 200 - 210

Die Seiten 200 – 210 bilden die eigentlichen Inhalte der Web-App ab, die wesentlichen Implementierungsdetails werden im Folgenden dargestellt.

Allen Seiten gemeinsam ist das Seitentemplate „Printer Friendly“, der HTML-Header ist jeweils wie folgt:

```
<meta name="viewport"
content="width=device-width,
minimum-scale=1.0, maximum-scale=1.0" />

<meta name="apple-mobile-web-app-capable"
content="yes" />
```

Abbildung 3 zeigt die Seite 201 (Jobs) mit den entsprechenden Regionen.

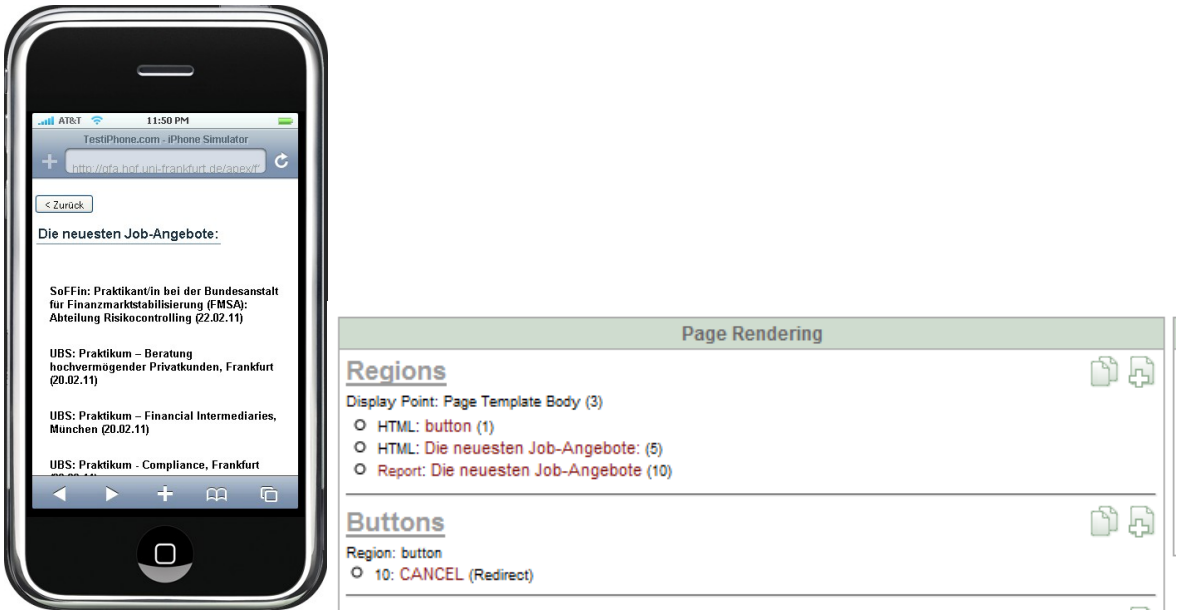

Abbildung 3: Seite 201 (Jobs).

Abbildung 4 zeigt den Report „Die neuesten Job-Angebote“, Abbildung 5 die dazugehörenden Reportattribute.

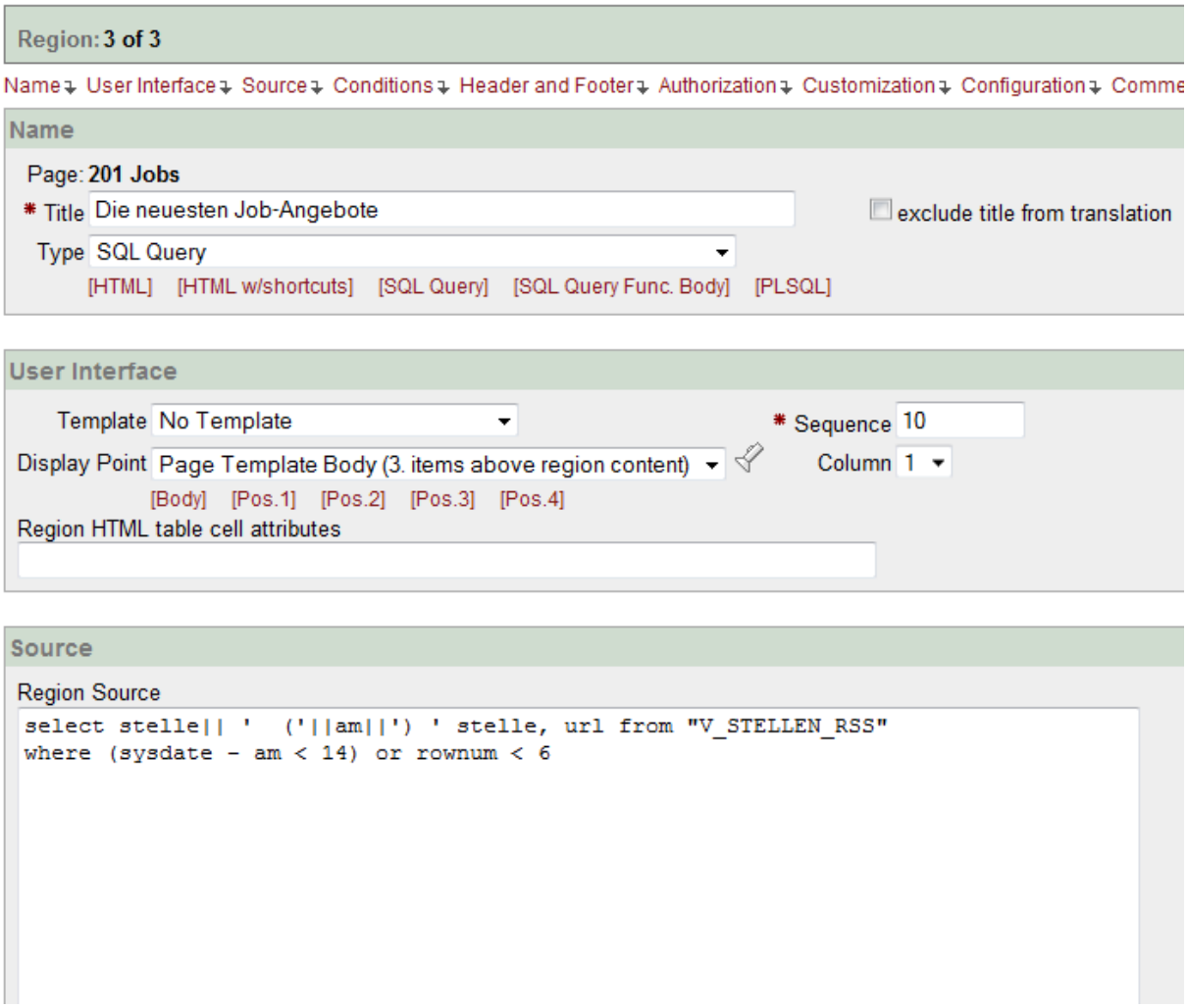

Abbildung 4: Report „Die neuesten Job-Angebote“.

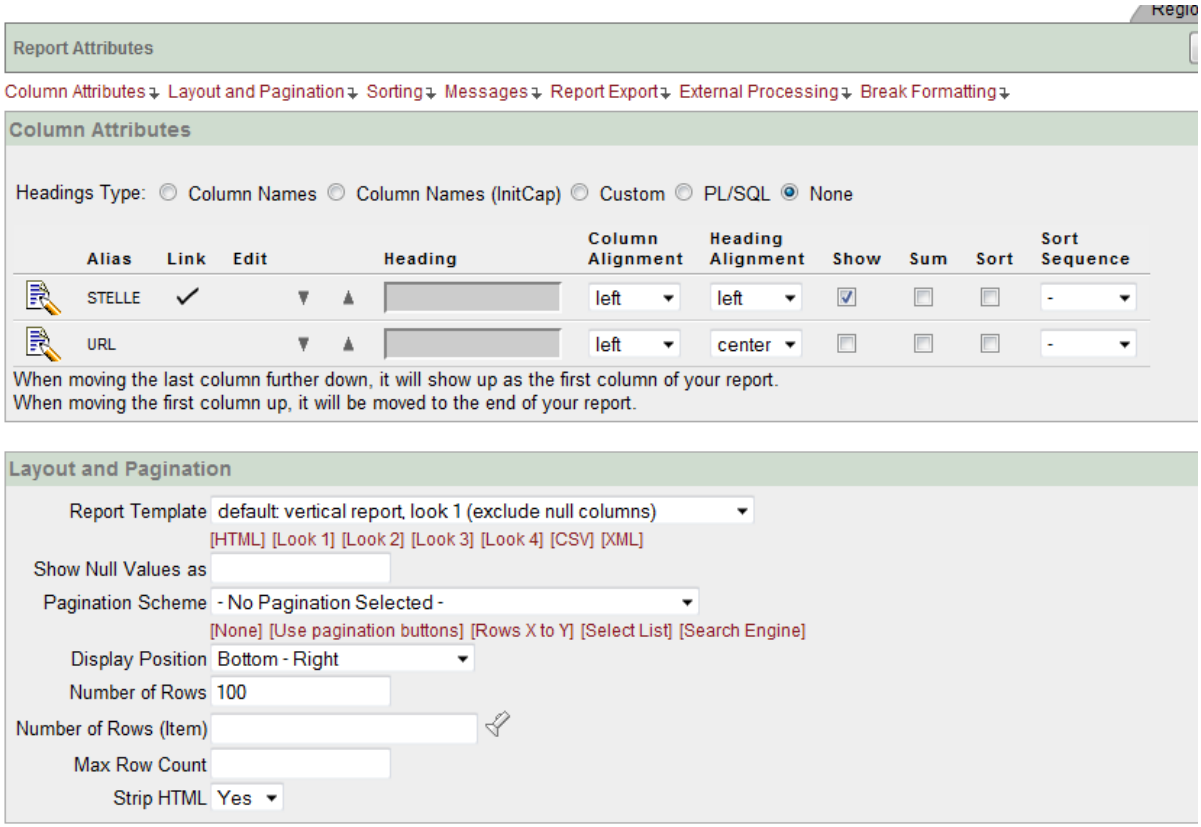

Abbildung 5: Reportattribute.

Die anderen Seiten sind entsprechend aufgebaut, Abbildung 6 zeigt alle Seiten im Überblick.

Abbildung 6: Seiten 200 - 210.

2.3.3 Seite 222 - Switch

Je nach aufrufendem Endgerät (normaler PC und Tablet vs. Smartphone) muss entschieden werden, ob die normale Webseite oder die Web-App-Version gezeigt werden soll. Dafür sind verschiedene Ansätze denkbar, z. B.:

- Unterschiedliche URLs, wie z. B. *m.gfa-frankfurt.org* oder *gfa-frankfurt.mobi*.
- Eine Weiche, die den User-Agent-String des Browsers auswertet und auf die richtigen Seiten der Anwendung weiterleitet.

Da erster Ansatz trivial, aber typischerweise mit Zusatzkosten versehen ist, soll hier der zweite näher erläutert werden.

Die Domain gfa-frankfurt.org wird auf die Seite 222 (Switch) weitergeleitet. Auf dieser Seite existieren verschiedene Verzweigungen (Branches), die jeweils zur „normalen“ Startseite (1) bzw. zur „mobilen“ Startseite (200) weiterleiten. Abbildung 7 stellt dies dar.

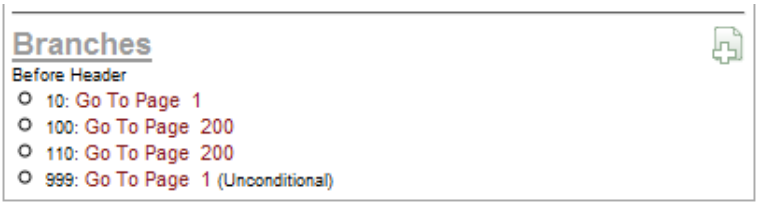

Abbildung 7: Verzweigungen.

Die Verzweigungen 10, 100 und 110 haben dann jeweils eine Bedingung vom Typ „Exists“ die überprüft, ob für den User-Agent-String des Aufrufenden ein Eintrag in der Tabelle MOB_DEVS vorhanden ist. Innerhalb der Einträge wird auf das Vorhandensein eines identifizierenden Teilstrings geprüft. Die Struktur der mehrfachen Abfragen erlaubt es, die Besonderheiten der unterschiedlich strukturierten User-Agent-Strings sinnvoll zur Verzweigung abzubilden.

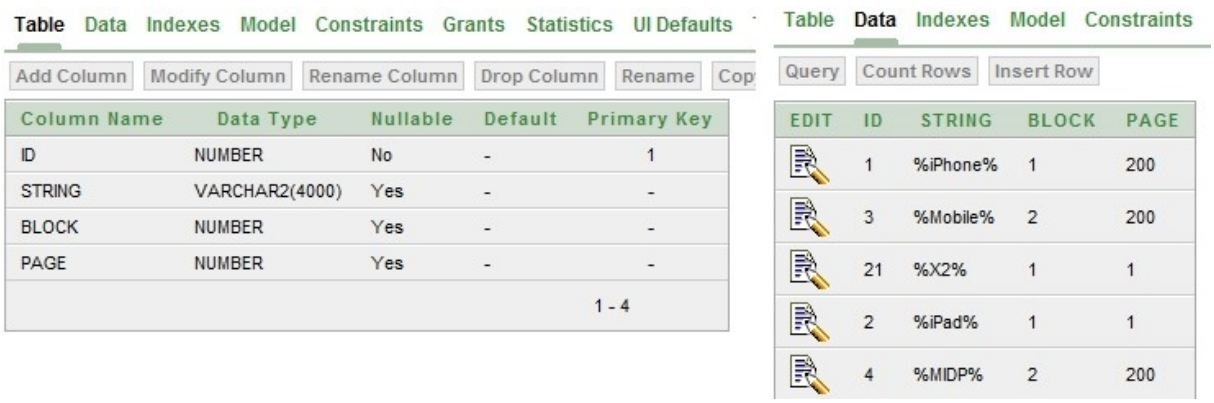

Abbildung 8: MOB_DEVS.

Abbildung 8 zeigt Struktur und Beispieldaten der Tabelle MOB_DEVS, Abbildung 9 die Abfrage zur Weiterleitung 10 zur Seite 1, Abbildung 10 die beiden Weiterleitungen zur Seite 200.

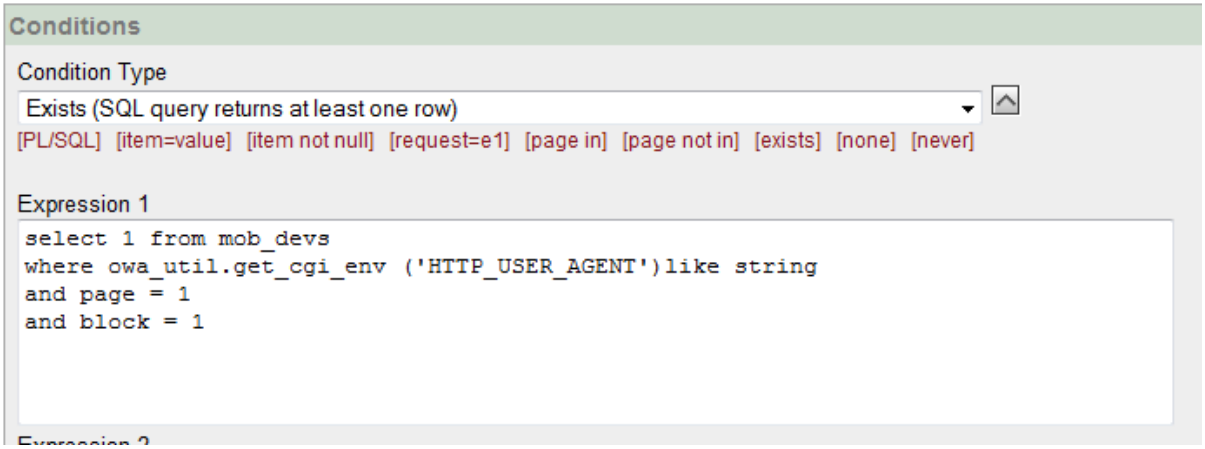

Abbildung 9: Weiterleitung zur Seite 1.

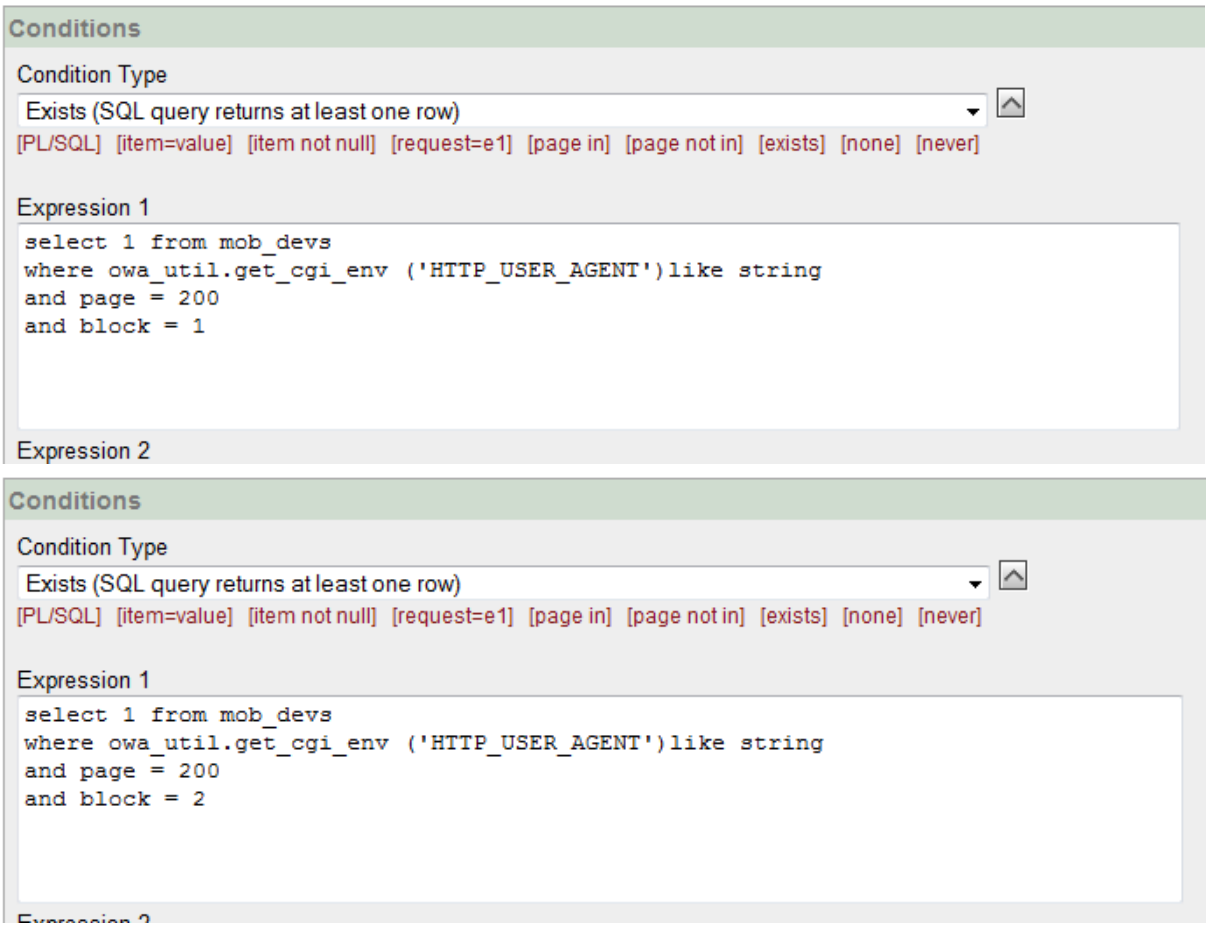

Abbildung 10: Weiterleitungen zur Seite 200.

Voraussetzung für die Weiterleitung auf Grundlage der User-Agent-Strings ist natürlich eine gut gepflegte Tabelle MOB_DEVS mit entsprechenden Einträgen an User-Agent-Strings. Möglichkeiten, an entsprechende Informationen zu gelangen, sind z. B.:

- Eine Liste auf zytrax.com (http://www.zytrax.com/tech/web/mobile_ids.html),
- die mit dem *Qualified User Agent Capture Kit* gesammelten Einträge auf databaselab.org/quack (Anklicken der Grafik),
- selbst sammeln, z. B. indem die Umschaltbuttons („Zur Mobil-Version" bzw. „Zur Standard-Webseite") einen entsprechenden Eintrag in MOB_DEVS anlegen, der dann noch analysiert und zugeordnet werden muss.

Das Vorhandensein der Möglichkeit des manuellen Umschaltens zwischen Standard- und Mobilversion durch diese Umschaltbuttons erlaubt es dem Aufrufer, die für sein Endgerät passende Version auszuwählen, falls die automatische Entscheidung auf Seite 222 nicht zum richtigen Ergebnis führt. Dies wird typischerweise durch neue Endgeräte notwendig, die in MOB_DEVS noch nicht bekannt sind.

Inhalte, die angezeigt werden, sind auf Seite 222 nicht vorhanden.

2.4 RSS-Feeds

Aufbauend auf der Web-App wird eine iPhone-App entwickelt, die aus der bestehenden APEX-Anwendung per RSS-Feeds mit den Daten versorgt wird. Die Wahl auf die iPhone-Plattform fiel dabei aus Popularitätsgrünen der iOS-Geräte, die Vorgehensweise für eine Android-App wäre identisch.

RSS-Feeds werden in APEX als Procedures definiert, Abbildung 11 zeigt den Quelltext eines Beispielfeeds zur News-Anzeige.

Dieser ist eine leicht modifizierte Version des von Carsten Czarski in http://www.oracle.com/webfolder/technetwork/de/community/apex/tipps/provide-rss/index.html publizierten.

Die Spaltenbezeichnungen sind dabei zur Verdeutlichung wie bei Carsten Czarski belassen worden, in der Anwendung findet eine „Übersetzung" (KUNDENNAME beinhaltet die jeweils relevanten Informationen) durch einen entsprechenden View V_NEWS statt.

Der Procedure ist noch das Recht „EXECUTE to Public" mit

```
Grant execute on RSSFEED_NEWS to public
```

zu gewähren.

Unter Oracle XE ist schließlich noch zu berücksichtigen, dass in der Function WWV_FLOW_EPG_INCLUDE_MOD_LOCAL ein Eintrag erstellt werden muss (siehe Dietmar Aust: http://daust.blogspot.com/2006/04/xe-calling-stored-procedures.html).

Der Aufruf des RSS-Feeds erfolgt dann z. B. über die URL http://gfa.hof.uni-frankfurt.de/apex/GFA.RSSFEED_JOBS?P_JOBS=0 .

```
CREATE OR REPLACE PROCEDURE RSSFEED_NEWS(
  p_news IN NUMBER
) AS
  v_xml blob;
BEGIN
  with jobs as (
    select id, kundenname, umsatz, datum
    from (
      select id, kundenname, umsatz, datum
      from v_news

      order by id desc
    )

  )
  select
    XMLElement("rss",
      XMLAttributes('2.0' as "version"),
      XMLElement("channel",
        XMLElement("title", 'GFA-News: RSS Feed'),
        XMLElement("link",'http://gfa-frankfurt.org'),
        XMLElement("description", 'News'),
        XMLElement("language", 'de_de'),
        (
        XMLAgg(
         XMLElement("item",
           XMLElement(
             "title",
            j.kundenname
           ),
           XMLElement(
             "link",
             j.umsatz
           )
         )
        )
        )
      )
    ).getblobval(nls_charset_id('AL32UTF8')) into v_xml
  from jobs j;
  owa_util.mime_header('text/xml');
  wpg_docload.download_file(v_xml);
  dbms_lob.freetemporary(v_xml);
END;
```

Abbildung 11: RSS-Feed.

2.5 iOS-App

Auf Grundlage der Datenbereitstellung per RSS-Feed kann jetzt eine native iOS-App (oder auch Android-App) entwickelt werden. Diese Entwicklung hat aber nichts mehr mit APEX zu tun, deswegen sollen hier nur einige Möglichkeiten angedeutet werden:

- Programmierung auf der Macintosh-Plattform mit Objective C (bzw. JAVA bei Android),
- Nutzung eines App-Baukastens im Web, wie er z. B. unter http://www.appmakr.com/ und http://www.totallyapp.com/ angeboten wird,
- Beauftragung eines externen Dienstleisters mit der App-Erstellung.

Im Falle der hier präsentierten GFA-App wurde ein externer Dienstleister (neosofttech.com) mit der App-Erstellung beauftragt.

3 Fazit

Das Erstellen einer Web-App auf Grundlage einer Apex-Anwendung ist ein relativ einfacher Vorgang, der sich mit den seit Apex 2.1 vorhandenen Bordmitteln „out of the box“ realisieren lässt.

Ein erster Ansatz kann darin bestehen, auf Grundlage des Seiten-Templates „Printer Friendly“ und des Berichts-Templates „Vertical Report“ eine optimierte Version zu erstellen.

Mit Zusatzsoftware wie jQuery und neueren Apex-Versionen ist eine noch größere Annäherung an das typische Aussehen einer nativen App möglich, Hinweise dazu findet man bei Peter Raganitsch (http://www.oracle.com/webfolder/technetwork/de/community/apex/tipps/peter-raganitsch-mobile/index.html und http://www.oracle.com/webfolder/technetwork/de/community/apex/tipps/peter-raganitsch-mobile/index-teil2.html).

Durch Publikation der relevanten Daten in Form von RSS-Feeds kann dann die Apex-Anwendung als Datengrundlage einer nativen App dienen.

Kontaktadresse:

Alexander Elsas
Goethe-Universität
Grüneburgplatz 1
D-60323 Frankfurt

Telefon: +49 (0) 69-798 33636
Fax: +49 (0) 69-798 33639
E-Mail aelsas@finance.uni-frankfurt.de
Internet: gfa-frankfurt.org