

Oracle hat mit Apex Version 4.2 nun auch die Möglichkeit zum deklarativen Erstellen von mobilen Applikationen geschaffen. Genauso wie beim Schreiben herkömmlicher Apex-Web-Anwendungen erfolgt dies stark Wizard-gestützt und macht erste Ergebnisse schnell sichtbar.

Apex für Smartphones

Peter Raganitsch, click-click IT Solutions

Um es gleich vorwegzunehmen, hier geht es natürlich um die Erstellung von mobilen Webseiten, nicht um native Applikationen. Darunter versteht man ganz normale HTML-Seiten, die allerdings für die Darstellung auf (kleinen) mobilen Browsern optimiert sind und somit auf (fast) allen Geräten laufen. Native Applikationen hingegen müssen für jedes Betriebssystem (iOS, Android etc.) in der jeweiligen Sprache programmiert werden und sind dann über den entsprechenden App-Store verfügbar. Oracle Apex konzentriert sich hier ganz auf mobile Webseiten und versucht damit, einen einfachen Zugang zu den eigenen Business-Daten zu ermöglichen.

Mobile Webseiten vs. „Responsive Web Design“

Mobile Webseiten sind, wie eingangs erwähnt, für mobile Browser optimier-

te Webseiten. Das bedeutet im Gegenzug, dass diese Webseiten in einem normalen Desktop-Browser meist etwas plump und klobig wirken, da der überschüssige Bildschirmplatz nicht genutzt wird. Also baut man einmal eine normale Webseite für alle Desktop-Browser und einmal eine spezielle für mobile Endgeräte. Das „Responsive Web Design“ eröffnet neue Möglichkeiten. Ziel ist es, eine einzige Webseite zu schreiben, die sich – je nach Endgerät – unterschiedlich darstellt. Genau genommen muss man sagen: Das jeweilige Endgerät entscheidet, wie diese „Responsive Seite“ dargestellt wird, blendet bei Platzmangel Teile aus, verkleinert Überschriften sowie Bilder und ordnet den Inhalt neu an (siehe Abbildungen 1 bis 3).

Beide Methoden haben Vor- und Nachteile und es muss im Bedarfsfall

eine gewählt werden. Die gute Nachricht jedenfalls ist, dass es diese beiden Möglichkeiten gibt und dass mit Apex 4.2 auch beide Varianten von Haus aus unterstützt werden.

Mobile Applikationen

Um spezielle mobile Seiten beziehungsweise Applikationen zu erstellen, hat Apex mit Version 4.2 die mobile Bibliothek „jQuery Mobile“ integriert. Diese kümmert sich um die optimale Darstellung auf allen mobilen Endgeräten. Je nachdem, welche Features (HTML5, CSS3 etc.) das jeweilige Endgerät unterstützt, wird die Darstellung der Seite angepasst. Unter <http://jquerymobile.com/gbs/> ist aufgelistet, welche Endgeräte mit welcher Güte-Klasse unterstützt werden.

Klasse „A“ beschreibt aktuelle Smartphones mit voller Unterstützung, hier

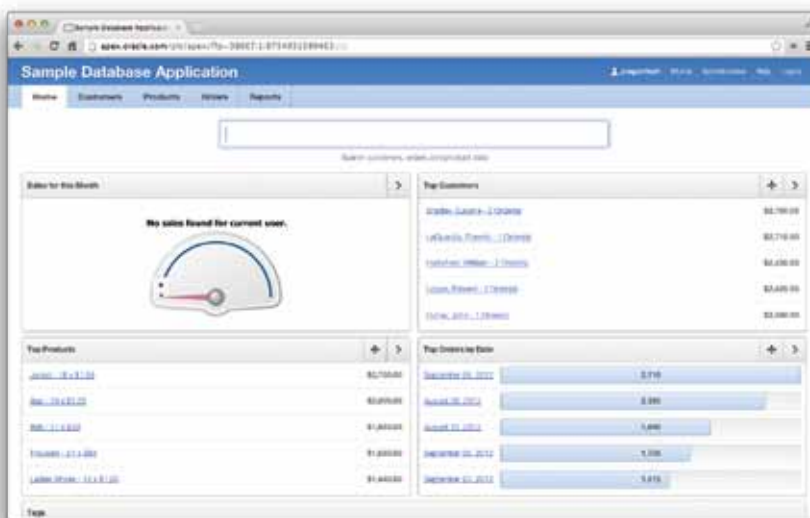


Abbildung 1: Darstellung einer „Responsive Web Design“-Seite auf einem Desktop-Browser mit jeweils zwei Regionen nebeneinander



Abbildung 2: Darstellung einer „Responsive Web Design“-Seite auf einem Smartphone-Browser. Die einzelnen Regionen sind nun untereinander dargestellt



Abbildung 3: Mobile Webseite optimiert auf Smartphones mit Konzentration auf das Wesentliche und einfache Bedienung



Abbildung 4: Startseite der vom Wizard erstellten mobilen Applikation



Abbildung 5: Anzeige eines Bilds aus einer „BLOB“-Spalte

ist also die Oberfläche so gut umgesetzt, dass es fast wie eine installierte Applikation wirkt. Bei Klasse „B“ fehlen den Endgeräten manche Funktionen, dadurch wird die Oberfläche etwas einfacher dargestellt. So sind zum Beispiel die Button-Ecken nicht abgerundet und anstelle von Hintergrund-Farbschattierungen ist eine flächige Farbe eingesetzt.

Unter Klasse „C“ fallen Geräte ohne besondere Eigenschaften, dort wird die Oberfläche minimalistisch dargestellt – lediglich einfacher, schwarzer Text auf weißem Hintergrund und normale Links anstelle von Buttons. Dadurch ist immerhin noch sichergestellt, dass die Grundfunktionen vorhanden sind, auch wenn es optisch weit weg von einer echten mobilen Applikation ist.

Das bedeutet: Apex erstellt das HTML-Grundgerüst für die Applikation und jQuery Mobile optimiert dies im Browser des darstellenden Smartphones. Dabei wird nicht nur auf die Darstellung geachtet, sondern auch auf die Bedienung durch Touch-Events (mit dem Finger Links aktivieren, Wischen etc.).

Damit die Unterscheidung zwischen mobilen und „normalen“ Apex-Seiten eindeutig ist, hat man sogenannte „User Interface Types“ eingeführt. Davon gibt es derzeit genau zwei: „Desktop“ und „jQuery Mobile Smartphone“. Eine Apex-Seite ist einem UI-Typ zugeordnet; dadurch wird bestimmt,

welches Template zum Einsatz kommt („mobile“ oder „normal“) und welche Arten von Komponenten auf dieser Seite verwendet werden können. So gibt es etwa für mobile Seiten eine neue Komponente namens „Listview“, die auf Desktop-Seiten nicht angelegt werden kann. Das bietet die Möglichkeit, innerhalb einer Applikation sowohl normale Desktop-Seiten als auch spezielle, mobile Seiten zu haben. Beim Start einer Applikation, die beide UI-Typen enthält, wird aufgrund des aufrufenden Browsers festgestellt, welcher UI-Typ der passendere ist, und der Benutzer auf die normale oder mobile Log-in-Seite weitergeleitet.

Elemente einer mobilen Applikation

Wie von Apex gewohnt, werden auch mobile Applikationen über den Wizard erstellt. Dabei muss lediglich im ersten Schritt die Auswahl „User Interface“ auf „jQuery Mobile Smartphone“ gestellt sein. Der Rest funktioniert wie gewohnt: Seiten zur Applikation hinzufügen, Authentifizierung wählen, dann wird am Ende des Wizard die mobile Applikation erstellt (siehe Abbildung 4).

Am Screenshot der erstellten Applikation sehen wir auch schon die wichtigsten Bestandteile: eine Kopfleiste mit Navigation, der Hauptbereich mit der neuen Listview-Komponente und (nicht sichtbar, da leer) eine Fußleis-

te, die wiederum Buttons oder Text beinhalten kann. Kopf- und Fußleiste werden vom Wizard automatisch als HTML-Regionen auf der globalen Seite (vormals: Page Zero) angelegt. Hier können wir über das gewählte Region-Template das Verhalten dieser Leisten beeinflussen.

In der Kopfleiste sind zwei Region-Buttons, an deren Beispiel man sehr einfach die Funktionsweise von jQuery Mobile erkennen kann. Im Home-Button (links mit dem Haus-Symbol) wird über „Button Attributes“ das Aussehen gesteuert (siehe Listing 1). Listing 2 zeigt, welchen HTML-Code Apex daraus erzeugt.

jQuery Mobile orientiert sich an diesen sogenannten „data“-Attributen“ und beeinflusst die Darstellung entsprechend. Wobei an dieser Stelle anzumerken ist, dass „data“-Attribute HTML5-Standard sind, aber keinerlei bezeichnende Eigenschaften haben. Ein Browser ignoriert also diese Attribute einfach und wartet auf eine JavaScript-Bibliothek (in unserem Fall jQuery Mobile), um sie auszuwerten.

In der jQuery-Mobile-Dokumentation (siehe <http://jquerymobile.com/demos/1.1.0/docs/api/data-attributes.html>) steht eine gute Übersicht über alle „data“-Attribute. Auch sonst kann der Autor nur empfehlen, die jQuery-Dokumentation zur Hand zu nehmen, um zu herauszufinden, was alles mög-



Abbildung 6: List-Divider und Advanced-Formatierung



Abbildung 7: Virtuelle Tastatur für E-Mail-Adressen



Abbildung 8: Virtuelle Tastatur für Telefonnummern

```
data-icon="home" data-iconpos="notext" data-direction="reverse"
```

Listing 1

```
<a href="javascript:apex.navigation.redirect(&#x27;f?p=34099:1:12315047222879::N0::&#x27;);" id="B6890179116905031168" class="" data-icon="home" data-iconpos="notext" data-direction="reverse">Home</a>
```

Listing 2

lich ist und wie diese Ergebnisse erzielt werden können. Apex stellt an dieser Stelle nur den Rahmen zur Verfügung, die Feinjustierung muss der Anwendungsentwickler selbst übernehmen.

In der Browser-Darstellung ist zwischen Kopf- und Fußleiste eine Apex-Liste als Navigationsmenü unter ZuhilfenahmedesListen-Templates „Listview“ zu erkennen. Neben dieser einfachen Listen-Darstellung gibt es noch eine eigene Komponente „Listview“, die im mobilen Umfeld auch als Ersatz für klassische Reports verwendet wird. Basierend auf einem „SELECT“-Statement kann diese deklarativ eingestellt sein und dann komplexere Inhalte darstellen (siehe Abbildungen 5 und 6).

jQuery Mobile Layout

jQuery Mobile bietet eine sehr einfache und effiziente Möglichkeit, um

das Layout mobiler Seiten anzupassen. Über das Attribut „datatheme“ kann man auf jedem Element (Header, Footer, Listview, Button, Link, Item etc.) einen sogenannten „swatch-letter“ definieren. Dabei handelt es sich um die Buchstaben „a“ bis „z“, die jQuery Mobile jeweils einem Layout zuordnet. „a“, „b“ und „c“ sind mitgeliefert, man kann dies selbst bis „z“ erweitern beziehungsweise auch die mitgelieferten überschreiben. Am einfachsten geht dies mit dem „jQuery Mobile Theme Roller“, einer Online-Applikation unter „http://jquerymobile.com/themeroller“. Hier kann man sich ein neues Layout zusammenklicken und anschließend die fertigen CSS-Dateien herunterladen (siehe Abbildung 12).

Die neuen CSS-Dateien können nun auf dem eigenen Webserver abgelegt

und in die gewünschten Seiten eingebunden werden. Dies geschieht entweder durch die Seiten-Definition oder direkt im Seiten-Template, jeweils im Bereich „CSS – File URLs“.

Responsive Web Design

Entscheidet man sich für eine Seite, die gleichzeitig für Desktop, Tablet und Smartphone funktionieren soll, dann ist „Responsive Web Design“ (RWD) die richtige Wahl. Hier wird die Seite jeweils vollständig auf das Endgerät geladen und erst dort durch den Browser anhand von im Design festgelegten CSS-Regeln entschieden, was wie angezeigt wird.

Ein gutes Beispiel für RWD ist die Startseite der Oracle Cloud unter http://cloud.oracle.com. Betrachtet man diese Seite auf einem Desktop-Browser und verkleinert das Fenster, so sieht man, wie zuerst manche Überschriften und Bilder kleiner werden, danach die einzelnen Regionen schmaler, bis der Platz so eng wird, dass manche Bilder ausgeblendet und die Regionen nicht mehr nebeneinander, sondern untereinander angezeigt sind. Wichtig zu wissen ist, dass die gesamte Seite immer übertragen und nur gewisse Teile aus- beziehungsweise ausgeblendet werden. Daraus ergibt sich auch der Nachteil gegenüber speziellen mobilen Seiten: Bei RWD werden mehr Daten übertragen.



Abbildung 9: Virtuelle Tastatur für Web-Adressen (URL)

Für Apex sind RWD-Seiten ganz normale Desktop-Seiten, hier wird lediglich ein spezielles Theme verwendet. Mit Oracle Apex 4.2 wird Theme „25“ als „Responsive Theme“ mit-



Abbildung 10: HTML5-Balkendiagramm

geliefert. Es verwendet den neuen „Grid-Layout“-Mechanismus, der ein (Theme-gesteuertes) unsichtbares und flexibles Raster zur Ausrichtung von Regionen und Feldern anbietet. Damit



Abbildung 11: Kalender

lässt sich bei Regionen und Feldern deklarativ angeben, ob sie in einer neuen Zeile oder in der aktuellen neben dem anderen Inhalt angezeigt werden sollen. Das gilt allerdings nur so lan-

PROLICENSE[®]
OPTIMIZING SOFTWARE ASSETS
Kompetent – Unabhängig – Erfolgsbasiert

ORACLE-SOFTWARE IST JEDEN CENT WERT!

Unsere Mandanten zahlen trotzdem weniger.
Sprechen Sie mit uns!

Wir sind nur unseren Mandanten verpflichtet.

- > **Compliance sichern**
- > **Audit vermeiden**
- > **Kosten senken**

ProLicense GmbH

Friedrichstraße 191 | 10117 Berlin

Tel: +49 (0)30 60 98 19 230 | www.prolicense.com

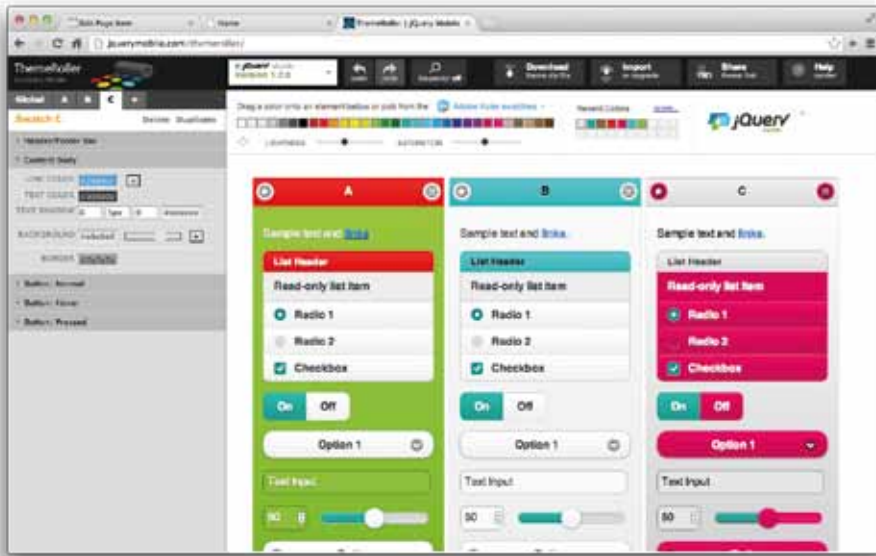


Abbildung 12: Farbenprächtige Layouts mit wenigen Maus-Klicks

ge, wie ausreichend horizontaler Platz vorhanden ist. Beim Unterschreiten einer bestimmten Grenze werden die Regionen beziehungsweise Felder untereinander angezeigt.

Einfache Eingabe

Die Darstellung von Daten auf kleinen, mobilen Bildschirmen ist mit den zuvor genannten „Listviews“ auch relativ einfach umzusetzen. Die Eingabe von Daten ist eine andere Sache. Hier entscheidet der Komfort schnell darüber, ob die Applikation tatsächlich verwendet oder als zu kompliziert abgetan wird.

Für Eingabefelder gibt es die Möglichkeit, einen Platzhalter zu definieren, damit der Anwender erkennt, welche Art der Eingabe von ihm erwartet wird. Viel wichtiger ist allerdings die genaue Definition des Datentyps. Neben „Number“ und „Text“ lassen sich über den Sub-Typ auch E-Mail, URL und Telefon definieren. Durch die Angabe dieses Sub-Typs hat der Browser nun die Möglichkeit, eine entsprechende virtuelle Tastatur einzublenden (siehe Abbildungen 7 bis 9).

Charts

Neben einfachen Listen-Darstellungen ist ein Chart oft wesentlich aus-

sagekräftiger, denn mit einem übersichtlichen Bild lassen sich komplexe Zahlen und Zusammenhänge oft besser abbilden. Zu den bisher üblichen Flash-Charts der in Apex enthaltenen AnyChart-Bibliothek gibt es nun auch HTML5-Charts. Das ist wichtig, da nicht jedes Smartphone auch Flash unterstützt.

Mit Apex 4.2 erfolgt das Upgrade auf eine neue Version von AnyChart, die (fast alle) Flash-Charts auch als HTML5 anbietet. Der Clou dabei ist, dass man sich nicht festlegen muss. Stellt man das Chart auf „Flash“, wird automatisch auf HTML5 gewechselt, falls das Endgerät kein Flash unterstützt (siehe Abbildung 10).

Wer noch kein Chart mit Apex erstellt hat, sollte dies unbedingt mal ausprobieren. So schnell und einfach kommt man sonst nie zu einer passenden Grafik. Der Wizard fragt dabei den Chart-Typ (Balken, Linien, Torten etc.) sowie ein „SELECT“-Statement als Datenquelle ab – schon ist das Chart fertig.

Kalender

Daten mit Tages- beziehungsweise Zeit-Bezug lassen sich mit einer Kalenderansicht mobil darstellen. Ist an einem Tag zumindest ein Datensatz vor-

handen, wird dies durch einen blauen Punkt symbolisiert. Klickt man darauf, kann man auf eine Detailliste weiterverzweigen. Alles, was man dazu benötigt, ist eine Abfrage (SELECT), die die entsprechenden Daten liefert; um die Darstellung kümmert sich – wie gewohnt – Apex selbst (siehe Abbildung 11).

Fazit

Die aktuelle Version von Apex hat sich ganz dem Thema „Mobile“ verschrieben und bringt einige praktische Funktionen mit. Man kann nun für jeden Einsatzzweck die passende Anwendung schreiben, immer begleitet von effizienten Wizards und komfortablen Einstellungsmöglichkeiten.

Wer Daten aus seiner Oracle-Datenbank mobil verfügbar machen möchte, findet in Apex sicher das passende Tool. Schneller und einfacher lassen sich mobile Applikationen nicht erstellen, noch dazu mit allen Sicherheits-Features.

Dies werden sicher nicht die letzten mobilen Features sein; für die kommenden Apex-Versionen werden Tablet-Support und noch mehr mobile Komponenten erwartet. Je weiter sich das Internet entwickelt, desto mehr ist auch in Apex möglich.

Peter Raganitsch
peter.raganitsch@click-click.at

