

Mit PL/SQL auf's iPad

DOAG Konferenz, 20.11.2012

Martin Friemel
Tönisvorst

Schlüsselworte

jQuery Mobile, PL/SQL, iPad, iPhone, Android

Einleitung

Wie entwickelt man PL/SQL-Webanwendungen für mobile Geräte?

Der Vortrag zeigt, wie man als Datenbank-Entwickler mit Hilfe von jQuery Mobile schöne Webanwendungen für iPad, iPhone und Android programmieren kann. Er vermittelt die wichtigsten Grundlagen über die Verwendung von jQuery Mobile in eigenen PL/SQL-Programmen.

Als Demo wird der PL/SQL-Sourcecode einer wunderschönen Webanwendung besprochen, die für die Bedienung auf dem iPad optimiert wurde. Die Beispielanwendung ist eine Sportler-Datenbank zur Erfassung und statistischen Auswertung von Trainingsergebnissen. An diesem Beispiel zeigt der Vortrag anschaulich, wie aus der Oracle-Datenbank heraus mit wenig Aufwand eine Anwendung entsteht, die optimal mit den Fingern bedient werden kann.

Ich zeige, wie man sich an das Thema "jQuery Mobile" heran arbeitet, wie man die Online-Dokumentation zur Hilfe nimmt, welche Voraussetzungen auf dem eigenen Webserver zu treffen sind, welche Javascript- und CSS-Dateien geladen und auf welche Weise sie eingebunden werden müssen.

Das Wissen aus diesem Vortrag ist auch wertvoll als Vorbereitung für den Einsatz von APEX 4.2, das ebenfalls jQuery Mobile einsetzt, um Anwendungen für mobile Geräte zu erzeugen. Die vermittelten jQuery Mobile-Grundlagen helfen beim Verständnis und bei der Anpassung der APEX-Webdesign-Templates für mobile Geräte.

Die Besucher werden nach dem Vortrag jede Scheu vor der Verwendung von jQuery Mobile verloren haben und sie können genau einschätzen, was in den APEX 4.2-Vorstellungen gemeint ist, wenn von der Unterstützung für mobile Geräte gesprochen wird.

Was ist jQuery Mobile, was hat es mit jQuery zu tun?

jQuery ist eine JavaScript Library, mit der interaktive Webseiten programmiert werden können. Leider ist jQuery in erster Linie für Desktop-Browser entwickelt worden. Es gibt kaum Funktionen für die Entwicklung von mobilen Websites.

Das **jQuery Mobile** Framework löst dieses Problem: Auf Basis von jQuery bietet jQuery Mobile Funktionen, Themes und HTML-Erweiterungen an, mit denen Benutzeroberflächen für mobile Anwendungen erstellt werden können.

Der Einstieg in jQuery Mobile

Der Einstieg in die jQuery Mobile-Programmierung ist denkbar einfach: Lediglich ein paar Dateien in den HTML-Header einbinden, danach einige "data"-Attribute in den HTML-Code streuen und schon nach wenigen Minuten steht die erste mobile Webseite.

jQuery Mobile in die Webseite einbinden

Auf der jQuery Mobile Website kann man im „Quick start guide“ auf der Seite <http://jquerymobile.com/demos/1.1.1/docs/about/getting-started.html> nachlesen, welche drei Dateien in den Header der mobilen Webseiten eingebunden werden müssen.

jQuery Mobile besteht aus einem Stylesheet und einer Javascript-Datei. Dazu muss noch die Basis jQuery eingebunden werden.

Sie müssen drei Dateien in Ihre Webseite einbinden:

- Das jQuery Mobile Stylesheet (jquery.mobile-1.1.1.min.css)
- Die jQuery Library (jquery-1.7.1.min.js)
- Die jQuery Mobile Library (jquery.mobile-1.1.1.min.js)

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet"
        href="http://code.jquery.com/mobile/1.1.1/
              jquery.mobile-1.1.1.min.css" />

  <script src="http://code.jquery.com/jquery-1.7.1.min.js"></script>

  <script src="http://code.jquery.com/mobile/1.1.1/
              jquery.mobile-1.1.1.min.js"></script>
</head>
<body>
  (...)
```

Die drei benötigten Dateien werden in diesem Code-Beispiel nicht auf dem eigenen Webserver gehostet, sondern jeweils vom Webserver **code.jquery.com** geladen. Das hat einige Vorteile:

- Sie müssen sich nicht um die Ablage der Dateien auf Ihrem Webserver kümmern.
- Die Ladezeit Ihrer Seite sinkt. Viele andere mobile Webseiten laden die jQuery Mobile-Dateien ebenfalls vom code.jquery-com-Server. Daher befinden sie sich mit hoher Wahrscheinlichkeit bereits im Cache des Browsers.

Es gibt jedoch auch Gründe, die jQuery-Dateien unter <http://jquerymobile.com/download/> herunter zu laden und auf dem eigenen Webserver zu speichern:

- Es kann sein, dass Ihre Benutzer innerhalb des Firmennetzes im eigenen Intranet surfen und u.U. keinen Zugang zum Internet haben und damit keine Dateien von code.jquery.com laden

können.

- Ihre Verfügbarkeits-Zusagen für die mobile Webanwendung lassen es u.U. nicht zu, dass Sie sich auf die Bereitstellung der Dateien auf code.jquery.com verlassen können.

Das Gerüst einer jQuery Mobile-Webseite

```
<body>

<div data-role="page">

  <div data-role="header">
    <h1>DOAG 2012</h1>
  </div>

  <div data-role="content">
    <p>Mein Seiteninhalt</p>
  </div>
</div>
</body>
```

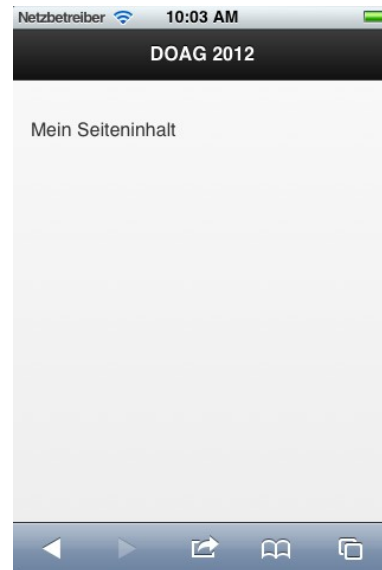


Abb. 1: iPhone-Webseite

Was hat es mit den data-role-Attributen auf sich?

jQuery Mobile nutzt eigene data-Attribute (wie z.B. **data-role="header"**), um `<div>`-Blöcke in der Webseite zu beschreiben.

Die Einbindung von eigenen data-Attributen ist keine Erfindung der jQuery-Mobile-Entwickler, sondern ist in der HTML5-Spezifikation vorgesehen. In der HTML5-Dokumentation werden sie auf [dev.w3.org](http://dev.w3.org/html5/spec/global-attributes.html#embedding-custom-non-visible-data-with-the-data-*-attributes) hier beschrieben:

http://dev.w3.org/html5/spec/global-attributes.html#embedding-custom-non-visible-data-with-the-data-*-attributes

(...)

3.2.3.9 Embedding custom non-visible data with the data-* attributes

A custom data attribute is an attribute in no namespace whose name starts with the string "data-" ...

(...)

JavaScript libraries may use the custom data attributes, ...

(...)

Die data-Attribute geben der jQuery Mobile-Library also die Möglichkeit, div-Blöcke mit eigenen Attributen zu kennzeichnen.

Beispiel:

Anhand des data-role-Attributes in `<div data-role="header">` erkennt jQuery Mobile, dass es sich bei dem div-Bereich um den Kopf der iPhone-Webseite handelt und den Inhalt entsprechend formatiert darstellen.

Für Webseiten-Entwickler ist das sehr komfortabel, denn

- jQuery Mobile sorgt dafür, dass der Kopf auf jedem mobilen Gerät korrekt dargestellt wird. Sie müssen Ihr Design nicht für die verschiedenen Zielgeräte (iPhone, iPad oder Android) anpassen.
- die Benutzer-Oberfläche wird komplett über das HTML-Markup beschrieben. Sie müssen keine einzige Zeile Javascript selbst schreiben.
- mit "Themes" können Sie das Aussehen der mobilen Webseite ändern, ohne den HTML-Code ändern zu müssen. Mit dem Online-Tool "Theme-Roller" <http://jquerymobile.com/themeroller/> können Sie den einzelnen Elementen Ihrer mobilen Webseiten per Mausklick Fonts, Farben, Schattierungen u.a. zuweisen. Der Theme-Roller generiert daraus eine neue CSS-Datei, die Sie einfach (wie bereits beschrieben) in den Header Ihrer Webseiten einbinden. jQuery Mobile liefert aber bereits im Standard-Stylesheet verschiedene Themes mit. Sie können sie über data-Attribute direkt ansprechen.

Beispiel:

```
<div data-role="header" data-theme="a"> erzeugt einen schwarzen Seitenkopf  
<div data-role="header" data-theme="b"> erzeugt einen blauen Seitenkopf
```

- jQuery berücksichtigt automatisch die auf Handys und Tablets auftretenden Ereignisse, die sonst nur sehr mühsam zu programmieren wären, wie z.B. das Drehen des Gerätes um 90 Grad. Sie haben die Möglichkeit, mit eigenem Javascript-Code auf diese Ereignisse zu reagieren. Dazu stehen Events wie *tap*, *taphold*, *swipe*, *swipeleft*, *swiperight*, *orientationchange*, *scrollstart*, *scrollstop* zur Verfügung. Mit diesen Events beschäftigen wir uns in diesem Vortrag aber nicht, denn wir wollen zum Start ohne eigenes Javascript auskommen.

Die wichtigsten data-Attribute

data-position gibt an, ob das Element eine feste Position im Browser-Fenster bekommt, oder ob es aus dem Focus scrollen darf. Es wird i.d.R. für den Header und den Footer der Seite verwendet.

data-role definiert das Aussehen des Elements. Kann wie im o.a. Code-Beispiel die Werte “page”, “header”, “content” und “footer” haben. Für Listen-Darstellungen gibt es zwei weitere data-role-Werte:

- **data-role="listview"** für eine Listenzeile
- **data-role="list-divider"** für eine Listen-Trennzeile, z.B. bei einem Gruppenwechsel

Beispiel: Listendarstellung ohne data-role-Attribute

Das ist die herkömmliche Darstellung einer Liste von Links ohne jede jQuery Mobile-Unterstützung:

```
<ul>

  <li><a href="#">Peter</a></li>
  <li><a href="#">Paul</a></li>
  <li><a href="#">Mary</a></li>

</ul>
```

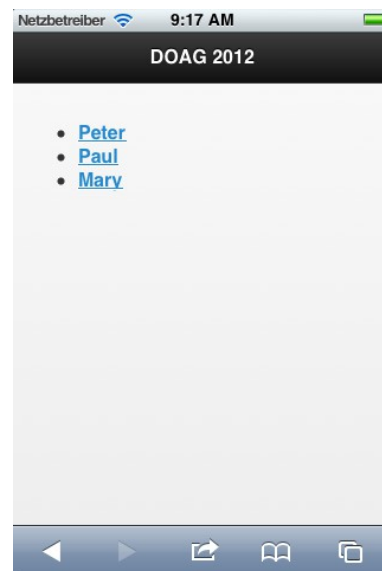


Abb. 2: Liste ohne jQuery Mobile

Beispiel: Listendarstellung mit data-role-Attribut

Fügen Sie lediglich das data-Attribut data-role="listview" hinzu. jQuery Mobile formatiert die Liste so, wie Sie es von Ihrem Smartphone gewöhnt sind. Links sind keine blau unterstrichenen Texte mehr, sondern werden durch den Pfeil am rechten Rand dargestellt:

```
<ul data-role="listview">

  <li><a href="#">Peter</a></li>
  <li><a href="#">Paul</a></li>
  <li><a href="#">Mary</a></li>

</ul>
```

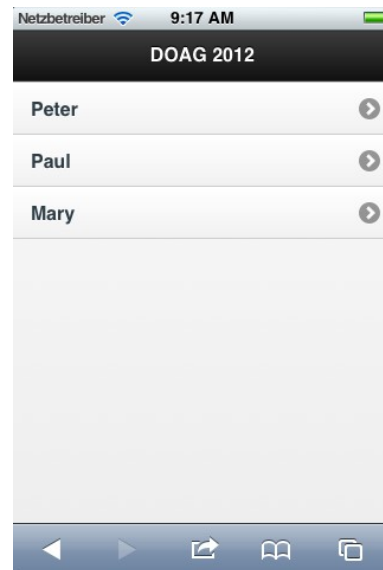


Abb. 3: iPhone-Liste

data-inset definiert, ob der Block entweder mit Rand in die Seite eingebettet wird (data-inset="true"), oder ob der Block die gesamte Seite ausfüllen darf (data-inset="false").

Beispiel: Listendarstellung mit data-inset

```
<ul data-role="listview"
    data-inset="true">

  <li><a href="#">Peter</a></li>
  <li><a href="#">Paul</a></li>
  <li><a href="#">Mary</a></li>

</ul>
```

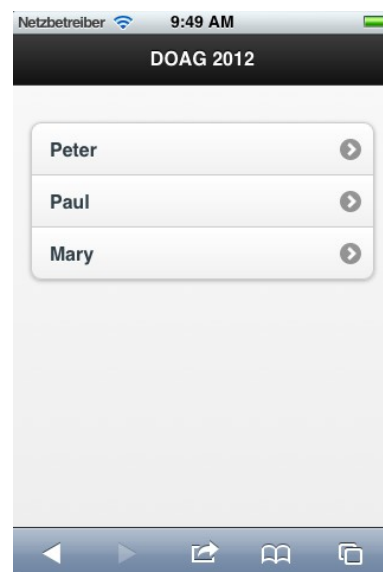


Abb. 4: data-inset

Beispiel mit Gruppenwechsel

In diesem Beispiel werden zwei *list-divider*-Zeilen als Rubrik-Überschrift eingefügt:

```
<ul data-role="listview"
    data-inset="true">

    <li data-role="list-divider">
        Namen
    </li>

    <li><a href="#">Peter</a></li>
    <li><a href="#">Paul</a></li>
    <li><a href="#">Mary</a></li>

    <li data-role="list-divider">
        Orte
    </li>

    <li><a href="#">Bonn</a></li>
    <li><a href="#">Nürnberg</a></li>

</ul>
```

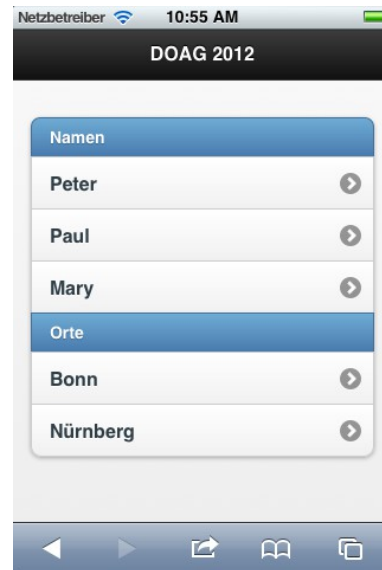


Abb. 5: list-divider

data-theme gibt Ihnen die Möglichkeit, das Aussehen eines Elementes zu variieren. Wählen Sie eines der vorgegeben Themes durch die Angabe der a, b, c, d, e.

data-dividertHEME definiert das Aussehen der Zeilen, die mit `data-role="list-divider"` markiert sind.

Beispiel für die Verwendung von Themes

In diesem Beispiel wird die Liste mit dem Theme „e“ auf gelbem Hintergrund dargestellt, während die Gruppenwechsel-Zeilen mit Theme „a“ in schwarz ausgegeben werden:

```
<ul data-role="listview"
    data-inset="true"
    data-theme="e"
    data-divider-theme="a">

    <li data-role="list-divider">
        Namen
    </li>

    <li><a href="#">Peter</a></li>
    <li><a href="#">Paul</a></li>
    <li><a href="#">Mary</a></li>

    <li data-role="list-divider">
        Orte
    </li>

    <li><a href="#">Bonn</a></li>
    <li><a href="#">Nürnberg</a></li>

</ul>
```

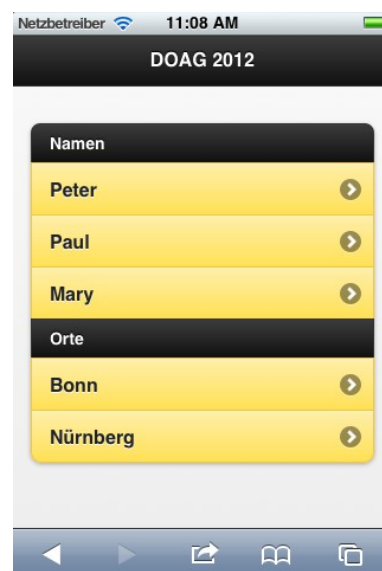


Abb. 6: Themes

Die gesamte Webseite

Das ist der HTML-Code der gesamten Webseite aus dem zuvor entwickelten Listen-Beispiel. Der blau gefärbte Text zeigt die Anpassungen, mit denen aus herkömmlichem HTML für Desktop-Browser eine mobile Webseite erzeugt wurde:

```
<!DOCTYPE html>

<html>

  <head>
    <link rel="stylesheet"
      href="http://code.jquery.com/mobile/1.1.1/jquery.mobile-1.1.1.min.css" />
    <script src="http://code.jquery.com/jquery-1.7.1.min.js"></script>
    <script src="http://code.jquery.com/mobile/1.1.1/jquery.mobile-1.1.1.min.js"></script>
  </head>

  <body>
    <div data-role="page">

      <div data-role="header">
        <h1>DOAG 2012</h1>
      </div>

      <div data-role="content">

        <ul data-role="listview"
          data-inset="true"
          data-theme="e"
          data-divider-theme="a">

          <li data-role="list-divider">
            Namen
          </li>

          <li><a href="#">Peter</a></li>
          <li><a href="#">Paul</a></li>
          <li><a href="#">Mary</a></li>

          <li data-role="list-divider">
            Orte
          </li>

          <li><a href="#">Bonn</a></li>
          <li><a href="#">Nürnberg</a></li>

        </ul>

      </div>
    </div>
  </body>
</html>
```


Tiefer einsteigen

Das sieht alles sehr einfach aus, aber es bleiben viele Fragen offen, wie z.B.:

- Was gibt es noch für data-Attribute?
- Wie bekomme ich nach dieser Einführung einen Überblick über alle jQuery Mobile-Features?
- Welche Geräte werden unterstützt?
- Wie werden Formulare und Buttons formatiert?

Antworten auf alle Fragen finden Sie in den Online-Beispielen auf der jQuery Mobile Website. Klicken Sie auf jquerymobile.com einfach am oberen Rand auf den Link „Docs“.



Abb. 7: jQuery Mobile Website



Abb. 8: jQuery Mobile Demos

In den Demos finden für jedes Feature den HTML-Code und ein Online-Beispiel. Sie können diese Beispiele mit einem Desktop-Browser wie Firefox aufrufen. Mit dem Firefox-Addon „Firebug“ können Sie dann den HTML-Code des Beispiels analysieren herausfinden, wie sie die Code-Schnipsel am besten in Ihre Anwendung übernehmen.

PL/SQL und JQuery Mobile

Der Vortrag heißt „Mit PL/SQL auf's iPad“ - wie also nutzen wir die jQuery Mobile-Möglichkeiten aus der Oracle Datenbank heraus?

Die erste Wahl sollte (natürlich) APEX ab Version 4.2 sein, denn APEX verwendet jQuery Mobile, um Webseiten für mobile Geräte zu generieren. Dieser Vortrag soll Grundlagenwissen vermitteln – daher programmieren wir hier mit „nativem“ PL/SQL und erzeugen unsere dynamischen Webseiten ohne die Hilfe von APEX.

Jede Oracle 11-Installation ab „Standard-Edition One“ darf ohne weitere Lizenzkosten den Oracle- (Apache) Webserver mit modPLSQL verwenden.

Im Download-Bereich finden Sie die benötigten Komponenten in den „Fusion Middleware Web Tier Utilities“:

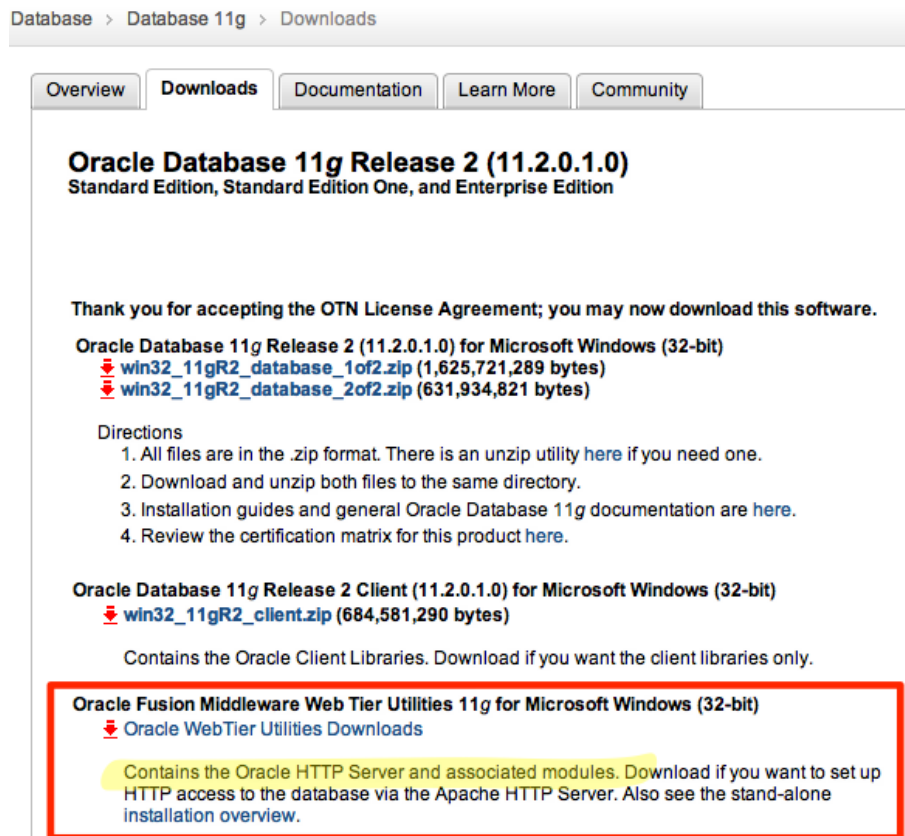


Abb. 9: Oracle Download

Auf die modPLSQL-Konfiguration im Apache Webserver wird in diesem Vortrag nicht eingegangen. Sie müssen einen DAD („Database Access Descriptor“) konfigurieren. Dort geben Sie an, an welches Schema sich modPLSQL beim Aufruf einer dynamischen Webseite anmelden soll, um in diesem Schema Oracle-PL/SQL-Packages aufzurufen. Außerdem wird der virtuelle Pfad für diese DAD-Konfiguration definiert.

Beispiel:

`www.webag.com/pls/doag/mein_package.meine_procedure?p_name=Peter`

Der virtuelle Pfad `/pls/doag` zeigt auf die DAD-Konfiguration. Das Apache-Modul modPLSQL meldet sich an das entsprechende Schema an und ruft die Procedure `meine_procedure` im Package `mein_package` auf. Der Parameter `p_name` wird an die Procedure übergeben.

Trainingcenter Beispielanwendung

Mit der modPLSQL-Technik habe ich die Trainingcenter-Anwendung für Desktop-Browser programmiert. Es handelt sich um eine privat genutzte Webanwendung, mit der Ausdauertrainings-Ergebnisse erfasst und statistisch ausgewertet werden.

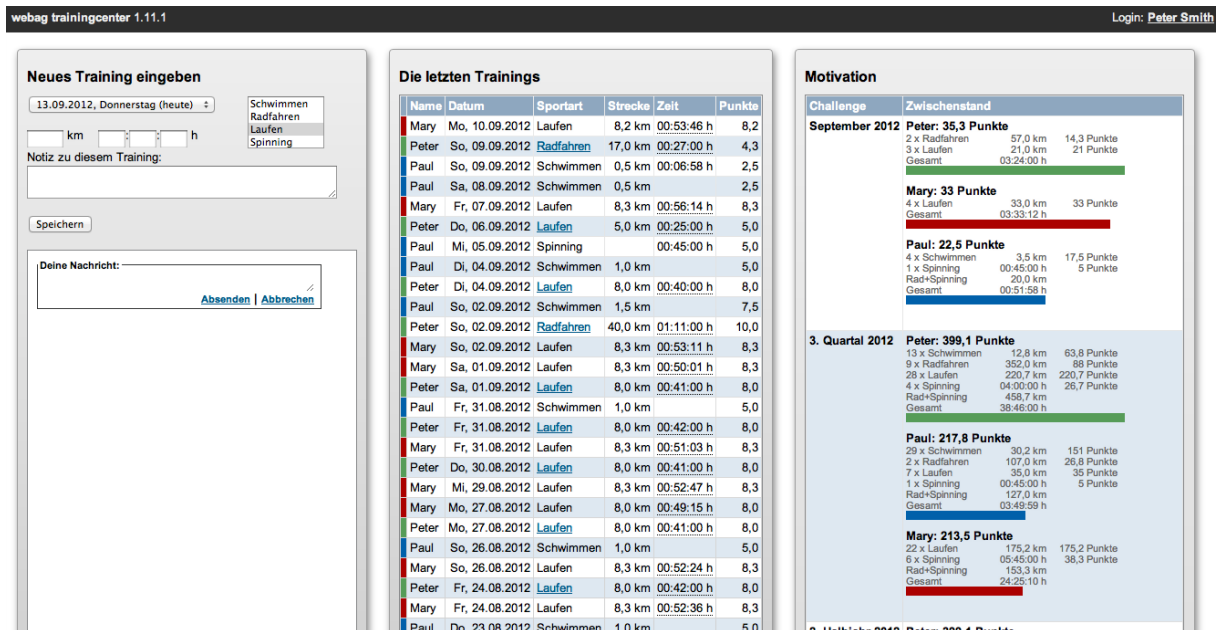


Abb. 10: Trainingcenter – Ursprüngliche Anwendung für herkömmliche Desktop-Browser

Schnell kam der Wunsch auf, die Anwendung auf dem Sofa mit dem iPad oder dem iPhone nutzen zu können. Natürlich sehen normale Webseiten auf dem 1024x768 Pixel großen Bildschirm des iPad 2 einigermaßen gut aus, aber der Komfort bei der Eingabe neuer Trainings kann nicht begeistern:

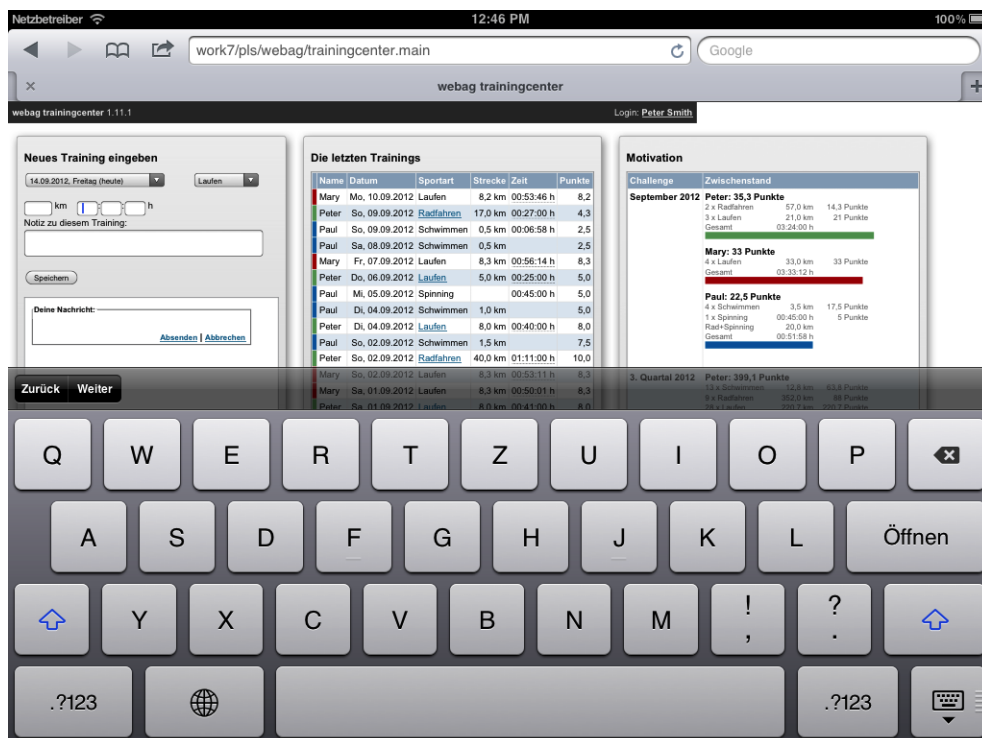


Abb. 11: Desktop-Webseite auf dem iPad

Aufgrund des überschaubaren Funktionsumfangs eignet sich diese Anwendung sehr gut für eine Portierung in eine mobile Webanwendung. Als Framework kam für mich als APEX-Programmierer nur jQuery Mobile in Frage, weil ich mich auf diese Weise unkompliziert in die Technik einarbeiten konnte, die Oracle für APEX gewählt hat.

Man erkennt schnell, dass es nicht nur darum geht, Listendarstellungen mit data-role-Attributen „iPad-fein“ zu machen, sondern man muss auch die Navigation und Menüführung an die Gepflogenheiten der iPad-Benutzer anpassen.

Die neue Benutzeroberfläche orientiert sich sehr stark an das Design der jQuery Mobile Dokumentations-Website mit einer Menüleiste auf der linken Seite.

Während des Vortrags auf der DOAG-Konferenz werden einige Source-Code-Beispiele aus der Beispielanwendung besprochen.

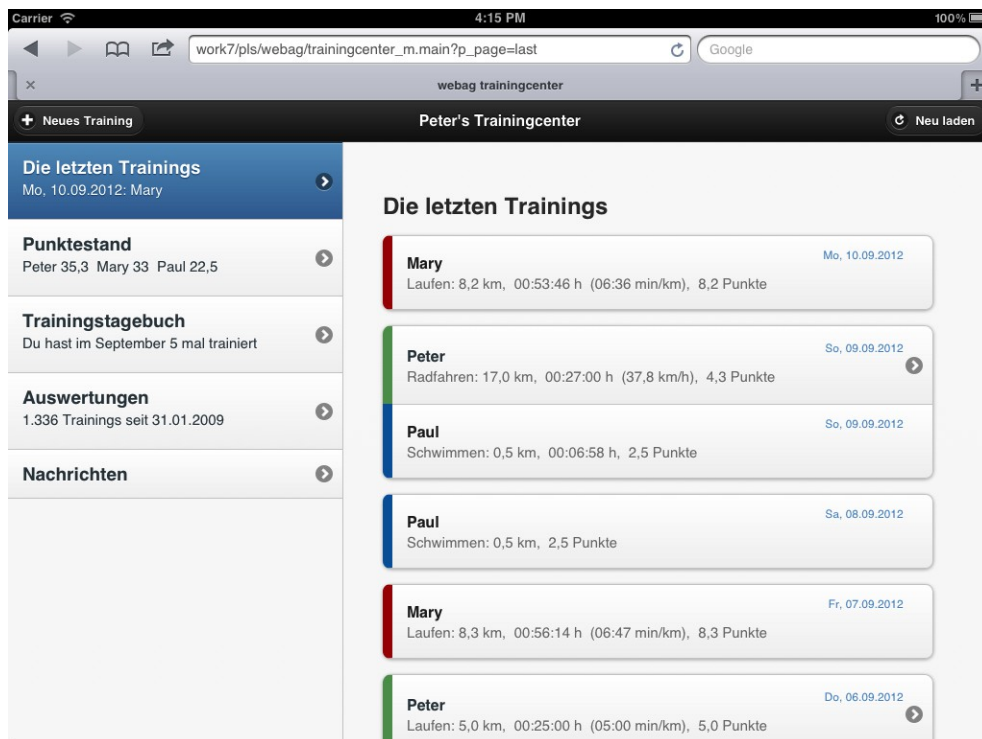


Abb. 12: Trainingcenter Mobile – Liste der letzten Trainingseinheiten

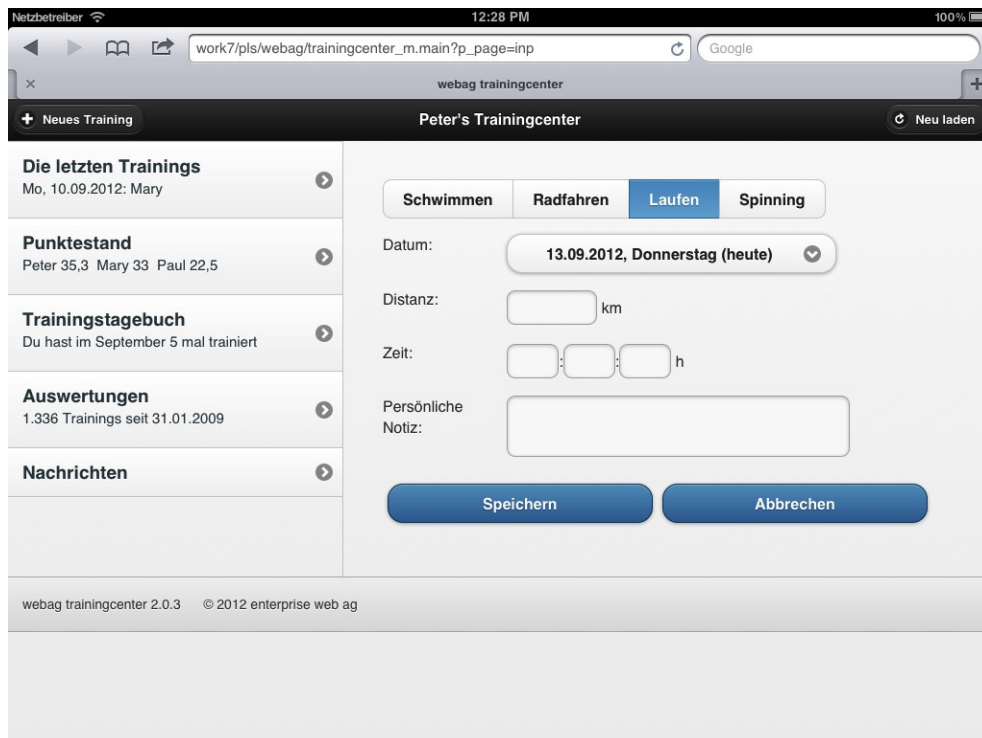


Abb. 13: Trainingcenter Mobile – Neues Training eingeben

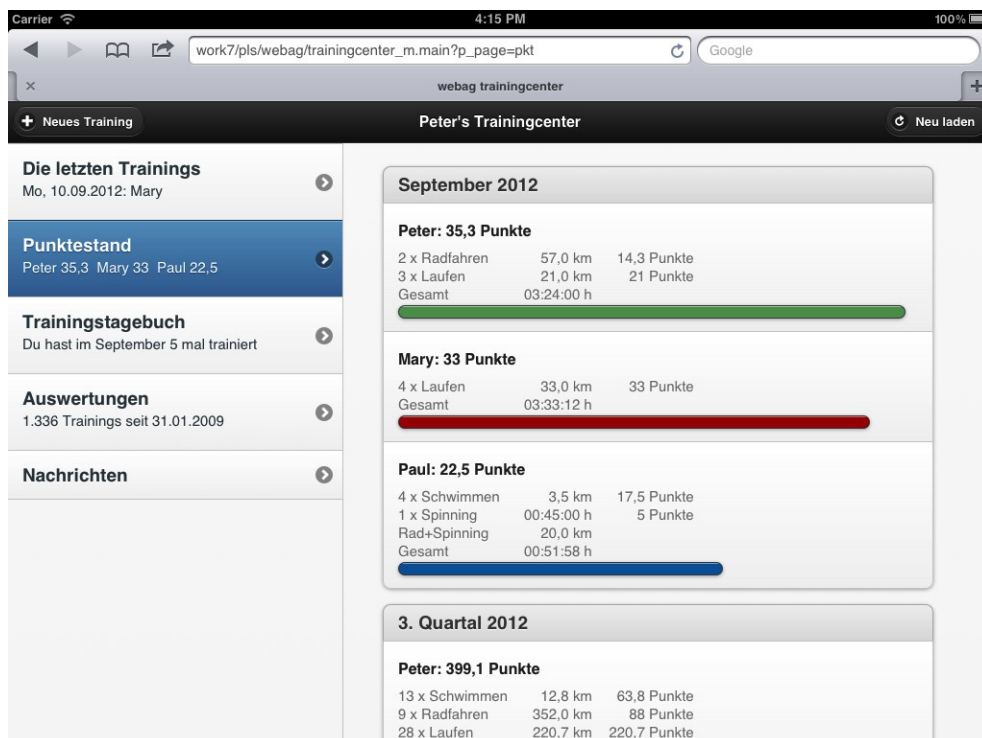


Abb. 14: Trainingcenter Mobile - Punktestand

Fazit

Insgesamt kann ich das positive Fazit ziehen, dass die Programmierung mit jQuery Mobile und den unkomplizierten data-Attributen zu einem einfacheren Programm-Code führen als die bisherige Programmierung für Desktop-Browser!

Kontaktadresse:

Martin Friemel
Rosenstraße 44 E
D-47918 Tönisvorst

Telefon:	+49 (0) 160 949 848 78
E-Mail	mfriemel@webag.com
Internet:	www.webag.com