

Der Artikel zeigt, welche Layouts mit Apex möglich sind, welche Einschränkungen existieren und worauf man achten sollte. Er gibt Antwort darauf, wie eine funktionierende Vermischung von Design und Funktionalität aussehen kann und wann der Einsatz von jQuery sinnvoll ist.

Layout und dynamische Elemente für Apex-Anwendungen

Markus Dötsch, MuniQSoft GmbH

Apex kommt bei immer mehr Firmen in den Fachbereichen, aber auch bei der zentralen IT-Abteilung an. Die Frage nach den möglichen Web-Layouts, die in einer Apex-Umgebung machbar sind, lässt sich einfach und schnell beantworten: Alles ist möglich – was auch mit anderen webbasierten Systemen machbar ist. Die Frage nach den Einschränkungen wirft sofort eine Gegenfrage auf: Welche Vorkenntnisse bringt man mit und welche Schwerpunkte sind wichtig?

Apex aus Sicht der Oracle-Welt

Oft kommen die Anforderungen aus einer Oracle-Umgebung. Dort ist man Client-Server und eventuell Forms gewohnt. Hier lässt sich die Frage oft umformulieren: Welche Einschränkungen bestehen in einem browserbasierten System?

Die Beschreibungssprache HTML ist per se zwar keine Einschränkung, bringt aber aus Sicht einer lokal installierbaren Anwendung mit einem klassischen User-Interface ein paar Hürden mit. Die Anordnung von Elementen fällt am Anfang schwer, der Workflow einer Applikation ist unübersichtlich und es gibt zu viele Einschränkungen bei der Entwicklung – auch wenn man später feststellt, dass die Funktionalität oft nicht fehlt, sondern nur an anderer Stelle auffindbar ist.

Apex fühlt sich für den Anwendungsentwickler genauso gut oder schlecht an wie jedes andere auf Web-Technologie getrimmte System. Die Funktionen, die man im Web schon an anderer Stelle entdeckt hat, lassen sich auch mit Apex umsetzen – wenn

nicht mit Bordmitteln, dann sicher durch die Integration anderer Systeme oder Bibliotheken. Apex ist, genauso wie die darunterliegende Oracle-Datenbank, flexibel anpassbar und versteht sich mit vielen Schnittstellen und Web-Standards.

Apex aus der Web-Sicht

Aus der Sicht eines erfahrenen Web-Entwicklers ist Apex in der Regel zuerst relativ unübersichtlich. Man versteht jedoch sehr schnell, wie die Abläufe in der Darstellung von Webseiten funktionieren, da sie ja nicht anders erfolgen als in der gewohnten Umgebung. Nur die Darstellung der Entwicklungsoberfläche sieht unterschiedlich aus. Sobald man in das Geheimnis um Themes und Templates eingeweiht ist, sind alle Möglichkeiten klar strukturiert.

Was man zu schätzen lernt, ist der einfache Umgang mit den Daten. Report und Formular sind in Windeseile erstellt. Ein an die Corporate Identity angepasstes Template lässt sich immer wieder verwenden. Alles funktioniert modular und logisch. Erste Erfolge sind schnell erreicht.

Gedanken zum Thema „Layout“

Oft heißt es zu Beginn eines Projekts, dass eine neue Apex-Anwendung so aussehen soll wie das Intranet oder die Webseite. Aber auch wenn man mit einer Apex-Anwendung auf einer „grünen Wiese“ beginnen kann, stellt sich vielleicht die Frage: Können wir uns ein Webseiten-Layout von einem Profi erstellen lassen?

Rein theoretisch ist das kein Problem, aber in der Praxis gibt es ein paar kleine Stolpersteine.

Grundsätzlich muss dabei zwischen dem Rahmen einer Webseite, also dem Kopf-, Fuß- und Seitenbereich auf der einen Seite sowie dem Inhalt einer Webseite auf der anderen Seite unterschieden werden. Für den Anwender gehören diese beiden Welten zusammen. In der Kombination kann der Betrachter entscheiden, ob die Seite gefällt oder nicht. Das Gerüst einer Webseite kann in Apex vollkommen kopiert werden. Aber eine auf Layout getrimmte Webseite, die von Marketing-Profis erstellt wurde, sieht in der Regel auch im Content-Bereich gut aus. Viele Bilder, wenig Text, alles aufeinander abgestimmt.

Bei der Apex-Anwendung wirkt das nicht ganz so gut: Der eingebundene Report ist durch die vielen Spalten viel zu breit, das dahinterliegende Formular viel zu schmal und zu hoch.

Sind Layout und Funktionalität nun Gegensätze oder doch Verbündete? Die folgenden Beispiele zeigen auf, wie Apex trotz Layout noch benutzbar ist, beziehungsweise wie gut Apex auch mit Funktionalität aussehen kann.

Es verschmelzen hier sehr schnell die Ideen hinter Layout und dynamischen Elementen. Es soll gut aussehen und trotzdem benutzbar sein. Also verwenden wir doch die Möglichkeiten, die uns das Web bietet. Ein Baustein dieser Lösung ist das JavaScript-Framework jQuery. Ab der Version 4 von Apex wird diese Bibliothek bereits mitgeliefert. Einige der von Apex mitgelieferten Themes nutzen die Möglichkeiten aus dieser Sammlung.

Beispiel: Breiter Report

Eine designte Webseite hat in der Regel eine feste Breite. Der heutige Standard beträgt 1024 oder 1280 Pixel, je nach Anforderung und Benutzergruppe. Oft werden die Seiten auch zentriert dargestellt. Was passiert mit der Webseite, wenn man einen interaktiven Report einer Tabelle mit 30 Spalten oder mehr hinzufügt?

Auch bei einem Standard-Report, bei dem der Anwender nicht in der Lage ist, die Spalten zu verändern, kann es zu einem nicht gewünschten Aussehen kommen. Denn auch hier gilt: Der Inhalt bestimmt die Tabellendarstellung. Was passiert, wenn lange Inhalte – eventuell noch ohne Leerzeichen, was einen Zeilenumbruch verhindert – die gewünschte Darstellung zerstören? Entweder das Layout der Apex-Anwendung wird „zerrissen“, was zur Folge haben kann, dass nachfolgende Elemente, wie zum Beispiel der Footer, rechts daneben dargestellt werden oder das Logo seinen vorgesehenen Platz verlässt. Andernfalls bleibt das Layout weiterhin stabil, dann werden Teile des Reports nicht zu sehen sein oder sie sind überdeckt von anderen Bereichen wie zum Beispiel einer Sidebar (siehe Abbildung 1).

Eine Lösung kann darin bestehen, dem Anwender die Möglichkeit zu geben, die aktuelle Seite wahlweise in der vollen Breite darzustellen. Dies er-

```
$(document).ready(function() {
    $('#schalter').hide();
    $('#ausblenden').click(function() {
        $('#sidebar').toggle();
        $('#schalter').toggle();
        $('#main').css('width', '100%');
    });
    $('#schalter').click(function() {
        $('#sidebar').toggle();
        $('#schalter').toggle();
        $('#main').css('width', '.816px');
    });
});
```

Listing 1

reicht man durch einen immer erreichbaren Schalter, der zwischen voller Breite und designtem Layout mit fester Breite umschaltet. Zusätzlich könnte man eine eventuell störende Side- oder Navigations-Bar ein- und ausblenden. Hier übernimmt jQuery diese ansprechende Aufgabe für uns. Die Elemente müssen eindeutig über IDs bezeichnet sein. Über die Seite 0 oder eine externe JavaScript-Datei, die über ein Template geladen wird, sorgen wir global dafür, dass die notwendigen Code-Zeilen auf jeder Seite zur Verfügung stehen. Soll die Funktion nur auf einer Seite zur Verfügung gestellt werden, können wir den Code auch in der Seite unter „JavaScript -> Execute when Page Loads“ eintragen (siehe Listing 1).

In diesem Beispiel ist der Schalter in die Sidebar integriert. Sobald man darauf klickt, wird die Sidebar ausgeblen-

det und das Layout auf 100 Prozent ausgedehnt. Damit der Anwender die Sidebar wieder einblenden kann, wird stattdessen eine kleine Grafik als Schalter dargestellt. Die jQuery-Funktion „toggle“ blendet Sidebar und Schalter jeweils ein oder aus.

Eine weitere Idee besteht darin, die grundsätzliche Darstellung einer Seite bereits bei der Auslieferung der Applikation festzulegen. Ist der Inhalt zu breit, wird standardmäßig die Sidebar ausgeblendet, der Anwender hat aber über den Schalter immer die Möglichkeit, sie wieder anzuschalten. Es muss dann nur dafür gesorgt werden, dass dann die Sidebar komplett überlappt und nicht durchsichtig ist. Zu guter Letzt kann noch die Einstellung pro Seite im Benutzerkonto abgespeichert werden, damit der Anwender die Seite so wiederfindet, wie er sie beim letzten Mal verlassen hat.

Eine andere Möglichkeit für einen zu breiten Report

Eine weitere Lösung für einen zu breiten Report besteht in der Möglichkeit, zusätzliche Spalten nur auf Anforderung des Anwenders per Klick darzustellen. Mit einem sogenannten „Named Column“-Template für einen Report können wir eine beliebige, selbst definierte Anordnung der Spalten eines Select-Statements in einem Report wählen. Dieser Vorteil wird mit dem Nachteil erkauft, dass mit dieser Technik für jedes Select-Statement ein eigenes Template erstellt werden muss, da im Template jede Spalte mit Namen angesprochen wird.

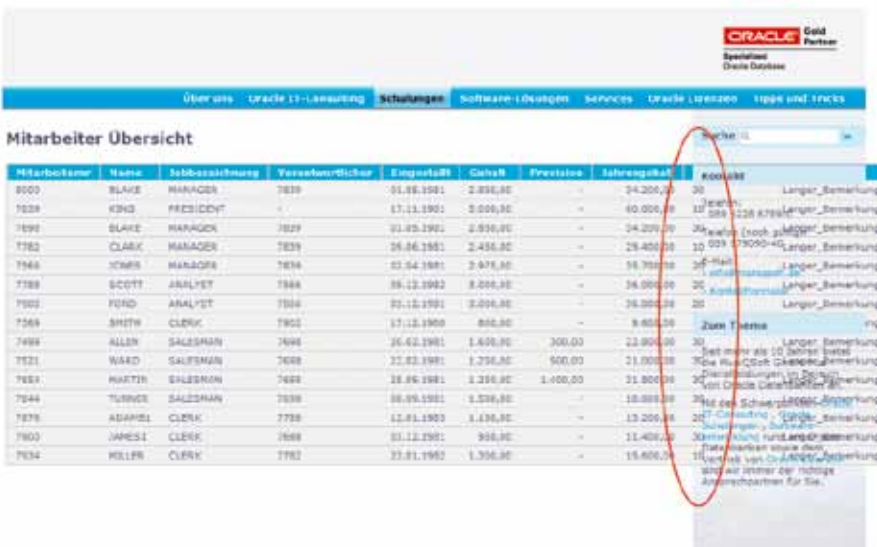


Abbildung 1: Da sich Report und Sidebar überlappen, ist der Text an dieser Stelle kaum mehr zu lesen



bi.doag.org

DOAG 2012 Business Intelligence München | 18. April 2012

Erfolgreicher durch BI!

Business Intelligence Technology
Big Data
Data Warehouse & ETL
Business Solutions & Management

Sponsoren



dimensional
informatics



MicroStrategy



PROMATIS

Reply
riverland

ORACLE

TRE
visto

trivadis
makes IT easier.

SHS VIVEON

Kooperations-
partner

SOUG
Swiss Oracle User Group

DOAG
Deutsche ORACLE-Anwendergruppe e.V.



Mitarbeiter Übersicht

Mitarbeiter	Name	Abkürzung	Personalnummer	Umsatz	Lohn	Prämie	Jahresgehalt	Abfertigung	Leistung
1000	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1001	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1002	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1003	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1004	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1005	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1006	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1007	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1008	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1009	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1010	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1011	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1012	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1013	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1014	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1015	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1016	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1017	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1018	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1019	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...
1020	BLAKE	MANAGER	7539	25.00.1981	9.000,00	-	36.000,00	30	Langer, Betriebszugehörigkeit, sehr hoch, andauernd, hoch, sehr...

Abbildung 2: Die Darstellung in voller Breite zerstört zwar das Layout, man kann jedoch mehr Daten darstellen

Aufklappbarer Report

EMPNO	ENAME	JOB	MGR	COMM	DEPTNO
8000	BLAKE	MANAGER	7539	-	30
7829	KING	PRESIDENT	-	-	10
7598	BLAKE	MANAGER	7539	-	30

Einstellungsdatum: 01.05.1981 - Gehalt: 2.850,00 - Jahresgehalt: 34.200,00

7782	CLARK	MANAGER	7539	-	30
7566	JONES	MANAGER	7539	-	30

Einstellungsdatum: 01.04.1981 - Gehalt: 2.975,00 - Jahresgehalt: 35.700,00

7788	SCOTT	ANALYST	7566	-	30
7902	FORD	ANALYST	7566	-	30
7369	SMITH	CLERK	7566	-	30
7499	ALLEN	SALESMAN	7566	300,00	30
7811	WARD	SALESMAN	7566	500,00	30
7654	MARTIN	SALESMAN	7566	1.400,00	30
7844	TURNER	SALESMAN	7566	-	30
7876	ADAMS	CLERK	7566	-	30
7900	JAMES	CLERK	7566	-	30
7934	MILLER	CLERK	7566	-	30

Abbildung 3: Über den Klick auf eine Zeile werden weitere Informationen dargestellt

Formular

Form on EMP1

Abbrechen Anlegen

ENAME: SCOTT
JOB: ANALYST
SAL: 3000
DEPTNO: 30

Abbildung 4: Das Formular füllt die Seite nicht aus und sieht daher deplatziert aus

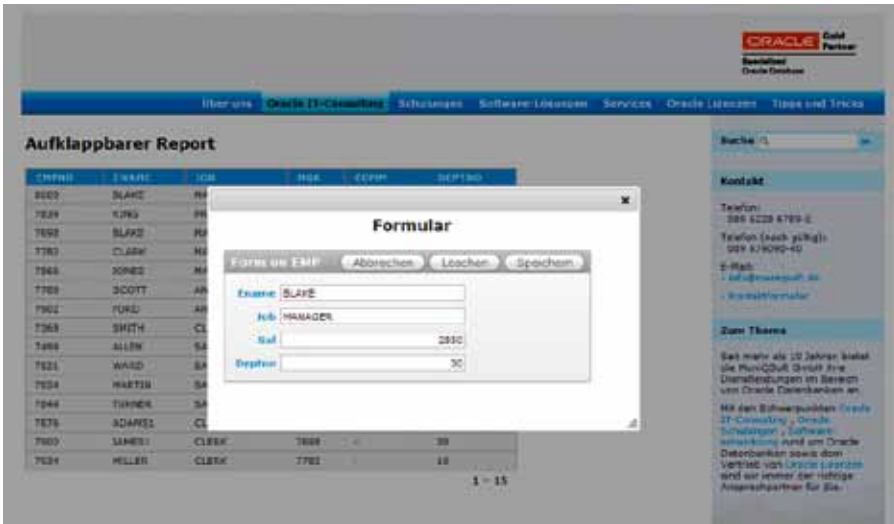


Abbildung 5: Das Formular wird über dem Report „schwebend“ dargestellt

Wir können dadurch jedoch mehrzeilige Ausgaben erreichen. Die Idee bietet sich vor allem in den Fällen an, in denen nicht alle Spalten notwendigerweise dargestellt werden müssen, um einen Datensatz zu identifizieren. Zusätzlich gibt es eventuell Spalten, die viel Text beinhalten. Also werden in der immer dargestellten Zeile die „kurzen“ Spalten ausgegeben, während der lange Text nur für den Datensatz angezeigt wird, der angeklickt wurde. Die unterhalb befindlichen Datensätze rutschen nach unten und machen Platz für den zusätzlich dargestellten Text.

Auch hier kommt jQuery zum Einsatz, das die Zeilen „anklickbar“ macht und für die Darstellung des zusätzlichen Inhalts sorgt (siehe Abbildung 3). Eine detaillierte Anleitung für die Erstellung einer solchen Lösung ist unter <http://www.muniqsoft.de/tipps/monatstipps/jahr-2011/februar-2011.htm> beschrieben.

Beispiel für ein kleines Formular

In unserer ausdesignten Apex-Anwendung kommen wir nun an den Punkt, an dem wir ein paar Daten über ein Formular eingeben möchten. Mit dem Apex-Wizard ist das Formular schnell erstellt, aber entweder benutzen die Formularfelder und die dazugehörigen Labels nur einen Teil des voll aufgezo-genen Fensters oder es klebt alles links oben im Content-Bereich. Der Region eine feste Breite zu verpassen, hilft hier zwar erst einmal, sieht allerdings trotzdem noch nicht gut aus (siehe Abbildung 4).

Eine Lösung könnte sein, das Formular nicht auf einer eigenen Seite, sondern gewissermaßen über dem Report darzustellen. Erstellt man dieses Overlay mit der Ausprägung „modal“, lässt sich der darunter grau dargestellte Report erst wieder bearbeiten, wenn das Formular geschlossen wurde. Eine feste Breite und eine zentral ausgerich-tete Position lassen das kleine Formular nun gut aussehen.

Aus technischer Sicht betrachtet, handelt es sich dabei eigentlich um eine Seite, die mit dem Formular-Wizard erstellt wurde. Zusätzlich wird auf der Seite der Report eingehängt. Mit

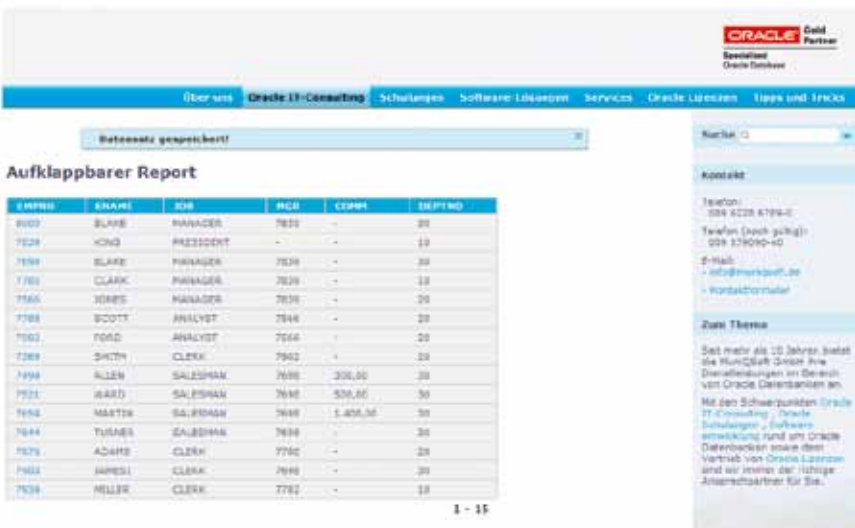


Abbildung 6: Der Report rutscht aufgrund der Message ein paar Pixel nach unten

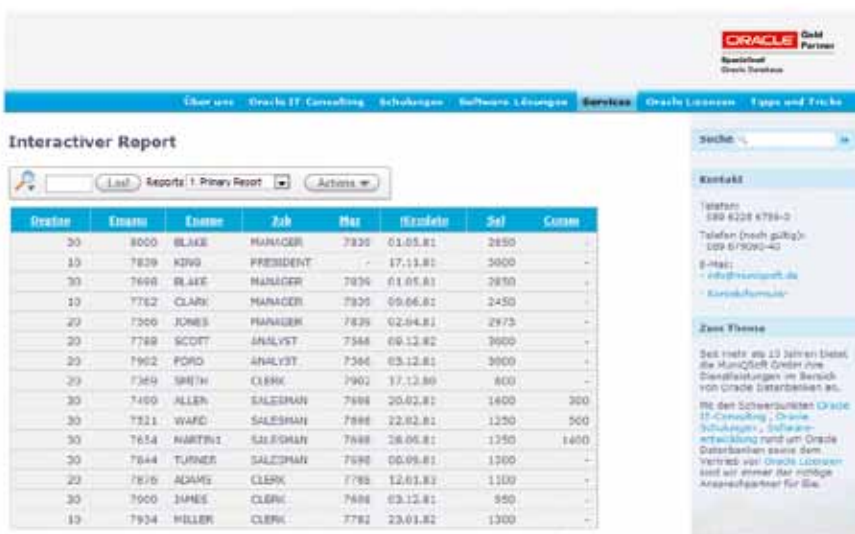


Abbildung 7: Eine kleinere Textbox und veränderte Button-Beschriftung in der Suchmaske des interaktiven Reports

ein paar „Conditions“ und dem Einsatz von jQuery wird nun dafür gesorgt, dass das Formular nur auf Klick im Report dargestellt und beim Speichern oder Abbrechen wieder ausgeblendet wird. Mit dem Einsatz der AJAX-Möglichkeiten lässt sich die Funktionalität sogar ohne eigentlichen „Submit“ der Seite durchführen. Das ist jedoch nicht in allen Fällen zwingend notwendig.

Mit jQuery wird die eindeutig benannte Formular-Region (Stichwort Static ID) mittels der Dialog-Funktion als manipulierbares Overlay dargestellt. Position, Größe und das Aussehen lassen sich dabei über Optionen steuern (siehe Abbildung 5).

Beispiel für die Message Areas

Nach so viel Speichern von Formularen stören die standardmäßig eingeblendeten Message Areas. Der gesamte restliche Content rutscht nach unten und verschiebt alles. Die Meldungen über erfolgreiches oder nicht erfolgreiches Speichern von Datensätzen

sind ja eigentlich ganz nützlich, aber dass die Oberfläche dabei immer verrutscht, ist nicht nur ein designtechnisches Problem, sondern auch eine Einschränkung der Ergonomie, da das Auge auf der Suche nach der Position derselben Formular-Elemente immer wandern muss.

Auch hier können jQuery und die Dialog-Funktion helfen. Zum einen wird die Nachricht über dem Report zentriert im oberen Bereich dargestellt, zum anderen verschwindet die Message-Box entweder automatisch nach ein paar Sekunden oder bei einem Klick außerhalb der Nachricht (siehe Abbildung 6 und Listing 2).

Damit der Dialog nur dargestellt wird, wenn gespeichert wurde, ist die Request-Variable beim Speichern belegt, was dann in JavaScript abgefragt werden kann. Die „Fade Out“-Funktion blendet innerhalb von drei Sekunden den Dialog sanft aus und schließt ihn danach.

Beispiel: Interaktiver Report ist Blackbox

Bei der Benutzung des interaktiven Reports merkt man als Entwickler sehr schnell, dass hier nicht alle Dinge verändert werden können, wie das bei einem normalen Report der Fall ist. Das liegt sicherlich daran, dass durch die

```
if (,&REQUEST.' == ,saved') {
    $('#messages').dialog({ position: 'top' });
    $('#messages').fadeOut(3000, function () {
        $('#messages').dialog('close')
    });
};
});
```

Listing 2

MuniQSoft GmbH – Datenbanken mit iQ

IT-Consulting	Schulungen	Software-Lösungen	Oracle Lizenzen
<ul style="list-style-type: none"> › Performance Tuning <ul style="list-style-type: none"> • Oracle Datenbank Tuning • Oracle SQL + PL/SQL Tuning › Real Application Clusters › Data Guard + Fail Safe › Datenbank Management <ul style="list-style-type: none"> • Konfiguration • Backup & Recovery • Migration und Upgrade › OEM Grid Control › Oracle Security › Services <ul style="list-style-type: none"> • Remote DBA Services • Telefon-/Remotesupport <p>Nutzen Sie unsere Kompetenz für Ihre Oracle Datenbanken.</p>	<ul style="list-style-type: none"> › Oracle SQL › Oracle PL/SQL › Oracle DBA › Oracle APEX › Backup & Recovery › RMAN › Neuerungen 10g/11g › Datenbank Tuning › Datenbank Monitoring › Datenbank Security <p>Wir bieten Ihnen öffentliche Kurse sowie Inhouse-Schulungen.</p>	<ul style="list-style-type: none"> › Individualsoftware <ul style="list-style-type: none"> • .NET und Visual Basic • Java › Oracle APEX › PL/SQL <p>Unser Ziel: Individuelle Softwareentwicklung mit Fokus auf Ihre Zufriedenheit.</p>	<ul style="list-style-type: none"> › Oracle Datenbanken <ul style="list-style-type: none"> • Standard Edition One • Standard Edition • Enterprise Edition • Personal Edition › Oracle Produkte <ul style="list-style-type: none"> • Enterprise Manager • Oracle Tools <p>Optimale Lizenzierung durch individuelle Beratung.</p>



```
$(,button.apexir-go-button > span').text(,Los!');
$(,apexir_SEARCH').css(,width', ,50px');
```

Listing 3

Interaktivität viele Möglichkeiten festgesetzt sein müssen, um das Ganze nicht zu komplex zu machen.

Konkret bedeutet es zum Beispiel, dass wir keine Möglichkeit haben, das Template des Reports zu ändern. Die Änderung des Suchfelds oder die Beschriftung des Buttons ist nicht deklarativ möglich. Man könnte jetzt auf die Idee kommen, die darunter liegenden Quelldateien zu verändern. Schließlich besteht auch der interaktive Report nur aus HTML, CSS und ein bisschen JavaScript. Aber Änderungen an diesen

Stellen wirken sich in der Regel global aus, nicht nur in der Applikation, auch im Workspace oder sogar serverweit.

Auch hier bietet sich wieder einmal der Einsatz von jQuery an. Wir nutzen die Möglichkeit, Änderungen an der Seite durchzuführen, nachdem die Seite eigentlich fertig gerendert wurde, aber kurz bevor sie dem Anwender dargestellt wird. Über einen geeigneten Selektor zeigen wir auf das zu ändernde Element und verändern dann über jQuery-Funktionen die CSS-Eigenschaften oder die Texte. Aber Vor-

sicht, auch der interaktive Report selbst nutzt diese Technik und ändert vielleicht ebenfalls dynamisch nachträglich den Inhalt der Webseite. Und wer zuletzt kommt, gewinnt (siehe Abbildung 7 und Listing 3).

Über den Selektor „button.apexir-go-button > span“ finden wir einen Button, der die Klasse „apexir-go-button“ hat und der als Inhalt einen „“ besitzt. Den Text des Span-Elements ersetzen wir mit „Los!“.

Zusätzlich verpassen wir der Textbox, die mit der ID \$apexir_SEARCH\$ versehen ist, eine weitere CSS-Eigenschaft mit der Breite von 50 Pixeln.

Fazit

Ob eine Webseite ein gutes Layout hat oder nicht, ist oft Geschmackssache. Benutzerfreundlichkeit und Übersichtlichkeit sind eher messbar. Aber nur die harmonische Zusammenarbeit von Design und Funktionalität lässt eine inhaltlich gute Applikation auch zu einer akzeptierten Anwendung werden. Manchmal gilt das im Apex-Umfeld geflügelte Wort: „Akzeptanz durch Firlefanz!“

Markus Dötsch
MuniQSoft
markus.doetsch@muniqsoft.de



Kleines Beispiel jQuery

Die JavaScript-Bibliothek jQuery ist ein Open-Source-Framework, mit dem sich viele JavaScript-Funktionen auf einfache Weise nutzen lassen. In Apex ist jQuery seit der Version 4.0 in den Themes automatisch eingebunden. Grundsätzlich wird zuerst der Bereich auf der Webseite ausgewählt, der manipuliert werden soll, der sogenannte „Selektor“. Mit Punkt getrennt wird die Funktion angegeben, die auf das selektierte Element angewendet werden soll, zum Beispiel „\$(,#schalter').hide();“ (siehe Tabelle 1).

\$	Der Funktionsname von jQuery, alternativ kann auch „jQuery“ angegeben werden
#schalter	Bezeichnung des zu manipulierenden HTML-Elements „(<div id=“schalter“></div>)“. Dabei steht die Raute (#) für IDs und der Punkt für Klassen (class). Es gibt noch viele weitere Möglichkeiten der Identifizierung von Elementen.
hide()	Funktion, um Element zu verstecken. In der Klammer können weitere Parameter angegeben werden.

Tabelle 1

Dazu ein weiteres Beispiel: „\$(,content').css(,border', ,solid 3px red');“. Der Klasse „content“ wird dabei eine zusätzliche CSS-Eigenschaft mitgegeben, ein roter Rahmen mit drei Pixeln Breite. Bei den Regeln für die Syntax handelt es sich um die CSS-Schreibweise, mit dem einzigen Unterschied, dass Parametername und Wert kommasetrennt angegeben werden. Das Äquivalent zur oberen Zeile in einer CSS-Datei könnte so aussehen: „.content { border: solid 3px red; }“. Weiterführende Informationen und Tutorials für den Einsatz von jQuery sind unter <http://www.jquery.com> zu finden.

Newsticker

MySQL Cluster 7.2 ab sofort verfügbar
Die neue Version bietet laut Oracle bis zu sieben Mal mehr Leistung für komplexe Abfragen. Mit SQL- und NoSQL-Zugriff über eine neue Memcached API soll MySQL Cluster 7.2 das Beste aus „beiden Welten“ beinhalten: Es erlaubt Schlüssel/Wert-Operationen und komplexe SQL-Abfragen innerhalb derselben Datenbank.