

Einen HTML-Anker in eine Apex-Applikation einbauen

Philip Ton, OPITZ CONSULTING GmbH

Jeder Endbenutzer von Application Express kennt das: Man möchte sich mal eben einen Datensatz im Detail ansehen, doch beim Zurückspringen auf die Master-Ansicht landet man statt auf der Höhe des vorher ausgewählten Datensatzes wieder am Seitenanfang.

Dem Endbenutzer die vorherige Scroll-Höhe anzubieten, ist eine Komfort-Funktion, die die Arbeit mit Apex-Anwendungen angenehmer macht. Dieser Artikel zeigt, wie man einen Anker schnell in bestehende oder neue Reports einbauen und ihn bei Bedarf aktivieren kann, ohne dazu Frontend und Code deutlich anpassen zu müssen.

Anwender können mit Oracle Application Express (Apex) Tabellen als Report betrachten. An sich ein schönes Feature, gäbe es da nicht einen kleinen Schönheitsfehler: Da die Zeilen meist auf Datensatz-Ebene angeordnet sind, stellen sie nur einen Bruchteil der möglichen Ansicht dar. Dazu kommt, dass es häufig so viele Datensätze auf einer Seite gibt, dass der Nutzer nach unten scrollen muss, um weitere In-

formationen zu sehen. Um sich ein Element in der Master-Ansicht genauer anzuschauen, steht eine Detail-Ansicht zur Verfügung. Springt man von dort aus allerdings wieder zurück auf die Master-Ansicht, wird die Seite neu geladen und, sofern der Datensatz außerhalb des Sichtbereichs liegt, ist nur der obere Bereich zu sehen.

Bei diesem Problem hilft ein klassisches HTML-Element. Das als „Anker“ bekannte Element wird auf Datensatz-Ebene eingebaut und gewährleistet beim Zurückspringen die komplette Sichtbarkeit des vorher ausgewählten Datensatzes (siehe [Abbildung 1](#)).

In vier leicht zu implementierenden Schritten wird gezeigt, wie schnell man seinen Kunden oder Endbenutzern auf diese Weise eine Freude machen kann.

Erstellen eines Hidden Items

Im ersten Schritt wird ein Hidden Item angelegt, um die Datensatz-ID zwischenspeichern. Die ID für den Anker muss zwischengespeichert werden, wenn die Navigation von der Masterseite auf die Detailseite stattfindet. Damit dieses Page Item nicht auf jeder betroffenen Seite angelegt werden muss, wird es einmalig auf der Page Zero (Global Page) erstellt. Von jeder Seite aus wird dann auf dieses Page Item zugegriffen. Es sollte als Hidden Item angelegt sein, damit die ID nicht zwangsweise vom Endbenutzer gesehen wird. Ein möglicher Name für das Page Item ist „PO_ANCHOR_ID“.

Anpassen der Lade-Prozesse zum Speichern der Datensatz-ID

Im Regelfall befindet sich auf einer Detailseite ein Ladeprozess, der die Daten mithilfe eines erhaltenen Primärschlüssels (die Datensatz-ID) in entsprechende Page Items lädt. An dieser Stelle wird angeknüpft und dem in Schritt 1 angelegten Hidden Item ein Wert zugewiesen (siehe [Listing 1](#)).

Einbau des Ankers im Interactive Report

Der vorher ausgewählte Datensatz soll beim Rücksprung von der Detailseite auf die Masterseite wieder im Sichtfeld des Browsers erscheinen. Damit dies funktioniert, sind die Anker auf Datensatz-Ebene im Interactive Report der Masterseite einzubauen. Im Vorfeld ist zu überlegen, in welcher Spalte der Anker eingebaut werden soll. Folgende Kriterien sind dabei zu beachten:

- Es empfiehlt sich, eine Spalte auszuwählen, die nie vom Endbenut-

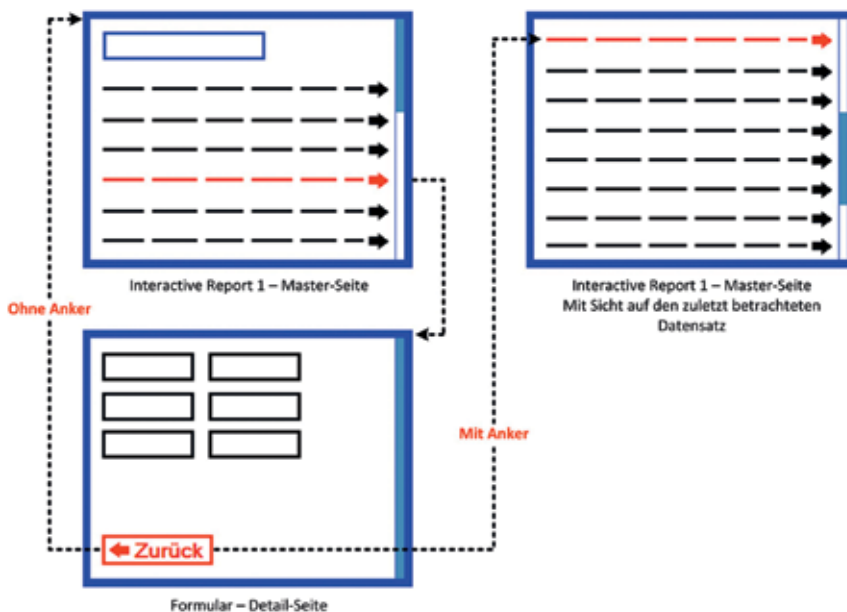


Abbildung 1: Apex-Anker

```

/* Standard Lade-Prozess */
BEGIN
  SELECT Kunden_Name,
         Kunden_Vorname,
         Kunden_Ort
         --[...]
  INTO :P5_KUNDEN_NAME,
       :P5_KUNDEN_VORNAME,
       :P5_KUNDEN_ORT
       --[...]
  FROM Kunden
  WHERE Kunden_ID = :P5_KUNDEN_ID;
END;

/* Lade-Prozess mit Wertzuweisung vom Hidden Item */
BEGIN
  SELECT Kunden_Name,
         Kunden_Vorname,
         Kunden_Ort
         --[...]
  INTO :P5_KUNDEN_NAME,
       :P5_KUNDEN_VORNAME,
       :P5_KUNDEN_ORT
       --[...]
  FROM Kunden
  WHERE Kunden_ID = :P5_KUNDEN_ID;

  -- Es folgt die Wertzuweisung vom Hidden Item:
  PO_ANCHOR_ID := :P5_KUNDEN_ID;
END;

```

Listing 1

zer ausgeblendet beziehungsweise deaktiviert wird, damit der HTML-Code dem Anker auch in jedem Fall zur Verfügung steht.

- Die Spalte, in die der Anker eingebaut wird, sollte von vornherein dem Datentyp „String“ entsprechen. Der Einbau des Ankers in eine Spalte, die später den Datentyp „Number“ oder „Date“ enthält, wird den Typ zunächst unweigerlich zu einem String wandeln. Auch wenn der HTML-Code später nicht mehr sichtbar ist, könnte diese Entscheidung die von Apex angebotene Sortierung dieser Datentypen fatal verändern.
- Die ID des Ankers sollte so eindeutig sein, dass auf den richtigen Datensatz zurückgesprungen werden kann. Deshalb empfiehlt es sich, als ID eine Spalte auszuwählen, die auf Datensatz-Ebene eindeutig ist. Dies trifft zum Beispiel auf den Primär-Schlüssel zu, wie das Quellcode-Beispiel zeigt.

Der Einbau des Ankers ist schnell programmiert und anhand des Quell-

codes leicht ersichtlich (siehe Listing 2).

Die Spalte würde den HTML-Code in der Anwendung demnach zunächst nicht als Code, sondern als String ausgeben. Dafür muss die Spalten-Eigenschaft noch verändert werden. Ziel ist es, die Column-Attribute für die betroffene Spalte zu ändern (siehe Abbildung 2). Dieser Vorgang

```

/* Standard SQL-Query*/
SELECT Kunden_ID,-->Primärschlüssel
       Kunden_Name,
       Kunden_Vorname,
       Kunden_Ort
       --[...]
  FROM Kunden;

/* SQL-Query mit HTML-Anker (id) */
SELECT Kunden_ID,-->Primärschlüssel
       ,<a id="" || Kunden_ID || ','>' || Kunden_Name Kunden_Name,
       Kunden_Vorname,
       Kunden_Ort
       --[...]
  FROM Kunden;

```

Listing 2

wird durch die folgenden Schritte erreicht:

1. Eigenschaften des Interactive Reports öffnen (zum Beispiel per Doppelklick).
2. Wechsel auf die Registerkarte „Report Attributes“
3. Im Abschnitt „Column Attributes“ die entsprechende Spalte aussuchen und unter „Display Text As“ die Eigenschaft „Standard Report Column“ auswählen. Dies hat zur Folge, dass der HTML-Code auch als solcher wahrgenommen wird, sich in die Seite integriert und verhindert, dass der Code als String oder Text identifiziert wird und damit lediglich auf der Seite ausgegeben würde.
4. Abschließend mittels „Apply Changes“ die Änderung speichern.

Umstellen der Rückwärts-Navigation auf den Anker

Abschließend ist die Rückwärts-Navigation so umzustellen, dass sie bei Aktivierung in der generierten URL den Anker mit sich führt, der dafür sorgt, dass der Nutzer auf den Datensatz der Masterseite zurückspringen kann. Dafür wird zunächst der Branch oder Button auf der Detailseite ausgewählt, der für die Rückwärts-Navigation auf die Masterseite zuständig ist. Beide haben gemeinsam, dass sie Page-Item-Werte zuweisen können. Diese Zuweisung wird mit der generierten URL wirksam.

Um den technischen Hintergrund des Ankers besser verständlich zu machen, wird an dieser Stelle kurz der

Aufbau der URL in Apex-Anwendungen erläutert, von der bei diesem Vorgang Gebrauch gemacht wird: Die Stelle vor dem letzten Doppelpunkt in der URL dient der Auswahl der Variablen, die zu deklarieren sind. Die Stelle danach beinhaltet die damit zu füllenden Werte in der jeweiligen Reihenfolge. Dazu als Beispiel die Adresse „http://123.1.123.123:7778/pls/apex/f?p=100:2:5498012442492::NO::P2_TEST:4711“. Beim Öffnen dieser URL würde der Benutzer auf Seite 2 gelangen. Das Page Item „P2_TEST“ auf dieser Seite würde mit dem Wert „4711“ deklariert sein.

Auch zu den Anker in URLs an dieser Stelle ein kleiner Exkurs. Ein Anker würde in einer gewöhnlichen URL so aussehen: „http://www.adresse.org#AnkerName“, „http://www.doag.org#top“ würde also bezwecken, dass der Anfang der Seite „www.doag.org“ zu sehen ist.

Dem Beispiel folgend, würde ein Anker in der URL der Apex-Anwendung wie folgt lauten: „http://123.1.123.123:7778/pls/apex/f?p=100:2:5498012442492::NO::#123“. Beim Öffnen dieser URL würde der Benutzer abermals auf die Seite 2 gelangen. Sollte der Entwickler einen HTML-Anker mit der ID „123“ im Interactive Report eingebaut haben, würde der Betrachter auf die Höhe dieses Ankers springen und ihn damit in den sichtbaren Bereich befördern.

In einem Branch („Action“ – „Page in this application“) oder Button („Action when button clicked“ – „Redirect to page in this application“) besteht die Möglichkeit, Page Items einen Wert zuzuweisen. Das Page Item, dessen Wert in der URL als String aufgelöst werden soll, kann man unter „With these values“ auswählen. In diesem Fall wird das Page Item genommen, das die Datensatz-ID zwischengespeichert hat. Zusätzlich wird vor das Item noch eine Raute geschrieben, um den Anker in der URL zu erhalten, zum Beispiel „#&P0_ANCHOR_ID“.

Beim Speichern der Einstellung mittels „Apply Changes“ und beim erneuten Öffnen des Branch sollte sich der Target-Type in URL geändert haben und unter „URL Target“ die URL in der gleichen Schreibweise zu sehen sein, in der sie auch beim Aktivieren des Branch erscheinen würde.

Column Attributes				
To change the column display order run the report as a developer. Click Select Columns from the Actions Menu and move the displayed columns using				
	Heading	Type	Link	Display Text As
	KUNDEN_NAME	Kunden Name	STRING	Standard Report Column
	KUNDEN_VORNAME	Kunden Vorname	STRING	Display as Text (escape special characters)
	KUNDEN_ORT	Kunden Ort	STRING	Display as Text (escape special characters)

Abbildung 2: Column-Attribute

Achtung: Wenn die oben genannten Änderungen beim Button wirksam werden und funktionieren, jedoch alle Felder in der betreffenden Region leer sind, liegt die Vermutung nahe, dass es sich um den gleichen Prozess wie beim Branch handelt. Nur wurde dann die entsprechende Auswahl nicht in der Entwicklung berücksichtigt. Im Zweifelsfall sollte deshalb immer ein Branch benutzt werden, wie er zum Beispiel beim Aktivieren eines Buttons ausgelöst wird (siehe Abbildung 3).

Noch zu beachten

Beim Navigieren von einer Detailseite auf eine andere kommt es zu Pro-

blemen, weil die „P0_ANCHOR_ID“ überschrieben wird. Um dem vorzubeugen, bedarf es keiner komplexeren Logik. Schon ein weiteres „Page Zero Page Item“ schafft hier Abhilfe. Für einen Endbenutzer ist es verlockend, in Apex-Anwendungen die Rückwärts-Navigation des Browsers zu benutzen. Auch dann muss nicht zwingend der zu bearbeitende Datensatz zu sehen sein.

Fazit

Die Implementierung eines Ankers mit dem Ziel, den zuvor betrachteten Datensatz wieder ohne Scrollen im Blickfeld zu haben, ist kein Hexenwerk. Die

Abbildung 3: Eigenschaften des Branch-Buttons

Anpassungen sind auch bei schon bestehenden Anwendungen sehr gering und als Entwickler können Sie diese Änderungen sehr schnell und routiniert vornehmen. Dieses Vorgehen empfiehlt sich besonders auf Seiten, auf denen sehr viele Datensätze auf einmal angezeigt werden oder wenn

es dazu noch ein Action-Menü gibt, in dem weitere Einstellungen vorgenommen werden können. Die Endbenutzer werden Ihnen für dieses kleine Feature dankbar sein, wenn sie zukünftig nicht mehr für jeden Datensatz umständlich scrollen müssen.

Philip Ton

philip.ton@opitz-consulting.com



Oracle SQL – Das umfassende Handbuch

gelesen von Bernd Tuba, regionaler DOAG-Repräsentant Trier/Saarland/Luxemburg

Gleich vorweg: Es handelt sich nicht um eine „SQL Referenz“, die wie in vielen Fällen mehr oder weniger eine deutsche Übersetzung der Oracle-Dokumentation sowie etwaiger Notes von Oracle Technical Network (OTN) beinhaltet. Vielmehr dient das Buch als praxisnaher Leitfaden mit vielen Beispielen und Übungen.

Das Buch ist gut strukturiert, was sich direkt im Inhaltsverzeichnis prüfen lässt. Auch der Index ist logisch aufgebaut. Ich habe ihn des Öfteren genutzt und bin immer fündig geworden.

Es gibt an den verschiedensten Stellen Hinweise zu speziell in 11g R2 eingeführten Funktionalitäten und auch bereits Ausblicke auf die nächste Datenbank-Version 12c. Gerade Letzteres steigert schon die Neugierde auf diese neue Version.

Idealerweise sind direkt nach Inhalt, Einführung und dem Dankeschön an die Ehefrau in Kapitel 2 die zum Erlernen von SQL technischen Voraussetzungen Schritt für Schritt aufgezeigt. Es ist beschrieben, von welchen Online-Quellen man die (Übungs-)Datenbank und etwaige Tools (SQL Developer) herunterladen sowie installieren und konfigurieren kann. Dabei wird auch auf feine Details wie die Zeichensatz-

Codierung eingegangen. Zur weiteren Vorbereitung zählt auch das Installieren (Ausführen) der Beispielskripte, die zum Download bereitstehen und nicht manuell eingetippt werden müssen. Zum Abschluss dieses Kapitels wird auch noch auf OTN und die Oracle-Online-Dokumentation verwiesen.

Wie Kapitel 2 liefern auch die beiden Folgekapitel für Einsteiger notwendige Grundlagen. Dies sind zum einen die Basics der relationalen Datenbanken und zum anderen die grundlegenden Architekturen und Prozesse einer Oracle-Datenbank. Gerade bei Letzterem wären ein paar Grafiken hilfreich gewesen. Allerdings rundet die Erklärung der Datentypen zum Abschluss dann wieder perfekt ab. Somit ist Teil I des Buches „Einführung und Grundlagen“ abgeschlossen.

Anschließend wird dann auch erstmals der direkte Kontakt zu SQL gesucht, der dann in den Folgekapiteln auch entsprechend gepflegt und ausgebaut wird. So kann man die Kapitel 5, 6, 7 und 8 als unabdingbares Grundlagenwissen bezeichnen. Gerade die in Kapitel 7 ausreichend kurz und mit verständlichen Beispielen aufgeführten Zeilen-Funktionen werden in der Praxis sicherlich immer wieder als



Nachschlagewerk dienen. Nicht nur für diese Kapitel gilt, dass die spannenden Übungen eine ideale Möglichkeit bieten, das gerade Erlernte zu üben und vertiefen. Ab der Beschreibung der Gruppen-Funktionen in Kapitel 8 wird der Stoff dann allerdings anspruchsvoller, aber auch interessanter.

So zum Beispiel die analytischen Funktionen: Obwohl ich in der Vergangenheit bereits ein paar (DOAG-)Vorträge zu diesem Thema besucht habe, war meine praktische Anwendung eher gering. Und dennoch war Kapitel 9 für mich eines der absoluten Highlights des Buches. Wie der Autor treffend erklärt, sollte man dieses Thema nicht nur im Bereich eines DWH ansiedeln. Die Beispiele sind teilweise komplex, aber detailliert beschrieben. Auch gibt der Autor einen „Step by Step“-Guide, der nochmals die einzelnen Schritte aufführt. Besser kann man dem Leser ein anspruchsvolles Thema nicht nahebringen.

Das abschließende Kapitel „Unterabfragen“ bietet einen gelungenen