

Das Apex-Framework bietet Entwicklern bereits eine beträchtliche Anzahl von Seitenelementen wie „Shuttle“ oder „Date Picker“. Leider gehören Dropdown-Menüs nicht zum Standard-Funktionsumfang. Einzellösungen für solche Menüs in Web-Anwendungen gibt es in vielfältiger Ausprägung. Dieser Artikel zeigt einen generischen Ansatz für die Verwendung von Dropdown-Menüs in Apex-Anwendungen.

Einsatz von Dropdown-Menüs in Apex-Anwendungen

Artur Rollow, OPITZ CONSULTING GmbH

Eine Link-Spalte ist typischerweise ein Bestandteil jedes Interactive Reports. Verbreitet sind beispielsweise Löschk-Aktionen oder die Navigation auf Detailseiten. Für den Anwender ist dies praktisch, da er die gewünschte Aktion direkt auf der Übersichtsseite ausführen kann und nicht auf eine Detailseite navigieren muss. In der Vergangenheit forderten die Anwender deshalb häufig eine Möglichkeit zur Ausführung mehrerer Aktionen pro Datensatz. Erfahrungsgemäß sollten aber maximal zwei Link-Spalten pro Report verwendet werden. Kommt es zur Überschreitung dieses Wertes, wirkt die Funktionsauswahl für den Anwender überladen und unüberschaubar. Zusätzlich verdrängen die Link-Spalten die eigentlichen Report-Daten auf die rechte Seite des Monitors. Eine Alternative hierzu bietet ein sogenanntes „Dropdown-Menü“. In diesem erfolgt eine Zusammenfassung aller Datensatz-Aktionen. Der Vorteil liegt darin, dass der Einsatz von nur einer Link-Spalte notwen-

dig ist. Ein Dropdown-Menü ist keine Standard-Komponente der Apex-Entwicklungsumgebung. Der Artikel stellt deshalb einen eigenen Lösungsansatz vor.

Das vorgestellte Dropdown-Menü (siehe Abbildung 1) wurde im Rahmen eines Projekts zur Verbesserung der Benutzerfreundlichkeit einer Apex-Anwendung erfolgreich eingesetzt. Die Anforderungen an das Menü sind:

- *Wiederverwendbarkeit*
Der entwickelte Quellcode soll für beliebig viele Menüs verwendbar sein. Neue Menüs sollen deklarativ erstellt werden.
- *Konfigurierbarkeit*
Änderungen an der Beschriftung der Einträge sollen ohne Entwickler stattfinden
- *Untermenüs*
Um das Dropdown-Menü möglichst kompakt zu halten, sollten Untermenüs möglich sein
- *Performance*
Das Dropdown-Menü soll sich im Browserfenster flüssig aufbauen

empfängt vom Server den generierten HTML-Code und stellt ihn in dem Browserfenster dar.

2. Die serverseitige Komponente besteht aus weiteren Unterkomponenten – dem Anwendungsprozess und einer „Pipelined Table Function“. Serverseitig wird der HTML-Code für das Menü erstellt. Die Menü-Attribute sind in den Datenbank-Tabellen konfigurierbar hinterlegt. Unter anderem wird eine ausführbare Aktion festgelegt. Diese definiert, was bei einem Klick passiert, beispielsweise EDIT, DELETE etc.

Um den gesamten HTML-Code der Seite nicht unnötig aufzublähen und somit eine gute Performance zu gewährleisten, muss das Dropdown-Menü für den angeklickten Datensatz zur Laufzeit generiert werden.

JavaScript-Funktion

Beim Klick auf ein Menü-Icon wird der clientseitigen Komponente mitgeteilt, welches Menü für welchen Datensatz aufgerufen werden soll. Hierfür wurde die JS-Funktion „ApexMenu.generateMenu“ (siehe Listing 1) erstellt. Folgende Parameterliste liegt vor: Mit „pThis“ wird der Primärschlüssel des angeklickten Datensatzes übergeben. „pMenuId“ identifiziert ein konkretes Menü. Der Client kommuniziert mit dem Server über einen Ajax-Request, um den HTML-Code für das Menü erzeugen zu lassen. Die Verwendung der asynchronen Datenübertragung verhindert beim Klick auf ein Menü-Icon ein erneutes Laden der Seite.

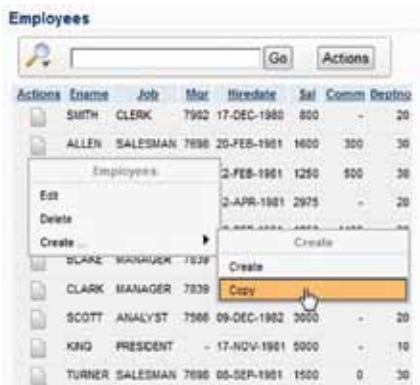


Abbildung 1: Dropdown-Menü im Einsatz

Technische Realisierung

Zur Umsetzung der Anforderungen wurden zwei Komponenten implementiert:

1. Mittels einer clientseitigen Komponente, einer JavaScript-Funktion, wird ein Request an den Server abgesetzt, mit dem eine Prozedur zum Erzeugen des Menü-HTML-Codes aufgerufen wird. Die JS-Funktion

Auf der Serverseite erfolgt der Aufruf des Apex-Anwendungsprozesses „GENERATE_MENU“. Dabei wird die aktuelle Parameterliste der JS-Funktion an den Datenbank-Server weitergereicht. Damit im Server die Parameter verwendet werden können, sind zwei Anwendungselemente angelegt (F101_MENU_ID und F101_OBJ_ID). Der erzeugte HTML-Code wird mit der jQuery-Syntax „\$(, #body').append(gReturn)“ dem aktuellen DOM-Baum hinzugefügt. Durch den weiteren Funktionsaufruf „ApexMenu.appMenuMultiOpen“ wird das Dropdown-Menü im Browser-Fenster dargestellt. Aus Gründen der Performance lohnt sich die Überlegung, beim erneuten Aufruf desselben Menüs den bereits vorliegenden HTML-Code wiederzuverwenden statt diesen über einen Ajax-Request neu zu generieren.

Anwendungsprozess

Der Aufbau des serverseitigen Anwendungsprozesses „GENERATE_MENU“ (siehe Listing 2) ist übersichtlich. Es findet ein Aufruf der Pipelined Table Function „GEN_MENU“ statt. Dabei werden die vom Client übergebenen Parameter mithilfe der Anwendungselemente weitergereicht. Die Menü-Einträge werden zeilenweise zurückgeliefert. Die gesamte Ergebnismenge wird zu einem kompletten Dropdown-Menü zusammengefasst und mit dem „HTP.PRN“-Aufruf zurück an den Client übergeben.

Table Function

Das Listing für den nachfolgenden Text steht unter <http://www.doag.org/go/doagnews/drop> zum Download bereit. Die Table Function „GEN_MENU“ (siehe Listing 3) stellt die Haupt-Komponente des gesamten Dropdown-Mechanismus dar. Jede mit „PIPE ROW“ zurückgelieferte Zeile entspricht einem HTML-Tag. Im PL/SQL-Code ist dafür eine festdefinierte HTML-Vorlage mit „“- und „“-Elementen hinterlegt. Die endgültige Struktur des Menüs hängt von der aktuellen Parameterliste ab. „MENU_ID“ legt die Anzahl der Menü-Einträge fest, die aus der Tabelle „APEX_MENU“ aus-

```
var ApexMenu = function () {
    return {
        generateMenu: function (pThis, pMenuId) {
            var get = new htmldb_Get(null, &APP_ID., ,APPLICATION_PROCESS=GENERATE_MENU', 0);
            get.add(,F101_OBJ_ID', pThis.id);
            get.add(,F101_MENU_ID', pMenuId);
            gReturn = get.get();
            $(, #body').append(gReturn);

            var dropdown_id = pMenuId + pThis.id;
            setTimeout(, ApexMenu.appMenuMultiOpen($x(\'' + pThis.id + ,\'), \\' + dropdown_id + ,\', \\'Bottom\')', 100);
            get = null;
        },
        appMenuMultiOpen: function (pThis, pThat, pPlacement) {
            var l_that = $x(pThis);
            var lMenu = $x(pThat);
            if (lMenu) {
                if (l_that != gCurrentAppMenuImage) {
                    app_AppMenuMultiClose();
                    pThis.className = g_dhtmlMenuOn;
                    dhtml_MenuOpen(l_that, pThat, false, pPlacement);
                    gCurrentAppMenuImage = l_that;
                } else {
                    dhtml_CloseAllSubMenus();
                    app_AppMenuMultiClose();
                }
            }
        }
    };
}();
```

Listing 1: JavaScript-Funktion mit Ajax-Request

```
DECLARE
    menu_html VARCHAR2 (3000);
BEGIN
    FOR item IN (SELECT *
                FROM TABLE (GEN_MENU (
                    :F101_MENU_ID,
                    2,
                    :F101_OBJ_ID)))
    LOOP
        menu_html := menu_html || item.COLUMN_VALUE;
    END LOOP;
    HTP.prn (menu_html);
END;
```

Listing 2: Anwendungsprozess mit dem Aufruf der Pipelined Table Function

gelesen werden. Die OBJ_ID, Primärschlüssel des angeklickten Datensatzes, ersetzt den Platzhalterstring aus dem Action-Eintrag, zum Beispiel wird „apex.submit(DELETE?#1#)“ zur Laufzeit zu „apex.submit(DELETE?4712)“. Beim Aufbau eines Untermenüs erfolgt ein rekursiver Aufruf der Table Function. Dies wird mit der Spalte „CHILD_

MENU_ID“ gesteuert. Für das Aussehen sowie das Ein- und Ausblenden des Menüs sorgen die in dem Apex-Framework vorhandenen CSS- und JS-Klassen.

Beispiel-Menü

Mit der Implementierung der beschriebenen Komponenten ist das Grund-

DOAG 2012 Development

Bonn | 14. Juni 2012

ERSTMALS

Fachkonferenz für
Erfahrungsaustausch
unter Software-
Entwicklern.

Software-Entwicklung auf Basis von Oracle Tools und Technologien – Wohin geht die Reise?

Veranstaltungs-Schwerpunkte

- PL/SQL und Apex
- Java und Open Source
- Forms, Reports, ADF, BI-Publisher
- BPM und Softwarearchitektur



DOAG

Deutsche ORACLE-Anwendergruppe e.V.

Infos unter development.doag.org



professional
it software &
services **pitss**[®]

iJUG
Verbund

SOUG
Swiss Oracle User Group

ORACLE[®]

BASIS
International

OPITZ CONSULTING

t&p software.
our profession.

XDEV³
Rapid Java Development

gerüst für die Verwendung der Drop-down-Menüs in Apex-Anwendungen geschaffen. Um ein Beispiel-Menü zu erstellen, genügt es, Menü-Einträge (siehe Listing 4) in der Tabelle „APEX_MENU“ (siehe Listing 3) abzuspeichern und den Interactive Report um eine Link-Spalte zu erweitern.

Zum Einbau des Dropdown-Menüs in den Interactive Report wird das vorhandene SQL-Statement (siehe Listing 5) mit einem „“-Tag um eine Link-Spalte ergänzt. Der „“-Tag enthält einen „onclick“-Event, sodass beim Anklicken des Menü-Icons die clientseitige JS-Komponente aufgerufen wird. Mit „this“ wird der Parameterliste immer der aktuelle „“-Tag als Objekt übergeben. Da das Attribut „id“ des Tags den Wert des Primärschlüssels enthält, kann in der JS-Funktion der aktuelle Datensatz ermittelt werden. Der zweite Parameter identifiziert das Dropdown-Menü in der Datenbank-Tabelle.

Weitere Features

Der vorliegende Code lässt sich mit wenigen Schritten um weitere Features wie Mehrsprachigkeit und bedingtes Anzeigen der Menü-Einträge erweitern. Das bedingte Anzeigen der Menüeinträge kann mit einer zusätzlichen „CONDITION“-Spalte in der „APEX_MENU“-Tabelle umgesetzt werden. Pro Menü-Eintrag wird eine Bedingung in SQL hinterlegt. Das SQL-Statement wird mit „EXECUTE IMMEDIATE“ zur Laufzeit ausgeführt. Wurde die Bedingung nicht erfüllt, wird der Menü-Eintrag ausgegraut und somit nicht ausführbar. Ähnlich wie bei der Action kann der Primärschlüssel durch einen Ersetzungsstring innerhalb der Bedingung verwendet werden.

Fazit

Das vorgestellte Dropdown-Menü dient als sinnvolle Ergänzung zu den vorhandenen Apex-Bordmitteln. Trotz der Erstellung des Menüs zur Laufzeit und des asynchronen Request bietet dieser Lösungsansatz eine sehr gute Performance. Nach einem einmaligen Aufwand für die Implementierung der

```
CREATE TABLE APEX_MENU
(
  , MENU_ID VARCHAR2(50) NOT NULL
  , CHILD_MENU_ID VARCHAR2(50)
  , ORDER_NO NUMBER(1) NOT NULL
  , ITEM_TEXT VARCHAR(50) NOT NULL
  , ACTION VARCHAR2(255)
);
```

Listing 3: Tabellenstruktur für Menüs

```
INSERT INTO APEX_MENU (MENU_ID,CHILD_MENU_ID,ORDER_NO,ITEM_
TEXT,ACTION)
VALUES ('Employees',null,1,'Edit','f?p=&APP_ID.:2:&APP_
SESSION.:NO::P2_EMPNO:#1#');
INSERT INTO APEX_MENU (MENU_ID,CHILD_MENU_ID,ORDER_NO,ITEM_
TEXT,ACTION)
VALUES ('Employees',NULL,2,'Delete','javascript:apex.
submit(''DELETE?#1#'')');
INSERT INTO APEX_MENU (MENU_ID,CHILD_MENU_ID,ORDER_NO,ITEM_
TEXT,ACTION)
VALUES ('Employees','Create',3,'Create',null);
INSERT INTO APEX_MENU (MENU_ID,CHILD_MENU_ID,ORDER_NO,ITEM_
TEXT,ACTION)
VALUES ('Create',null,1,'Create','javascript:apex.
submit(''CREATE?#1#'')');
INSERT INTO APEX_MENU (MENU_ID,CHILD_MENU_ID,ORDER_NO,ITEM_
TEXT,ACTION)
VALUES ('Create',null,2,'Copy','javascript:apex.
submit(''COPY?#1#'')');
```

Listing 4: Insert-Statements für Beispiel-Menü

```
SELECT ,'
AS Action,
ENAME,
JOB,
MGR,
HIREDATE,
SAL,
COMM,
DEPTNO
FROM EMP;
```

Listing 5: Erweiterung des SQL-Statements

notwendigen Komponenten lässt sich in kurzer Zeit ein Menü in eine Berichtssseite integrieren. Die client- und serverseitigen Komponenten können beliebig um weitere Funktionalitäten erweitert werden.

Artur Rollow
OPITZ CONSULTING GmbH
artur.rollow@opitz-consulting.com

