

Dynamisches Anzeigen von Informationen in APEX mit jQuery UI Dialogs und Tabs

Dr. Gudrun Pabst
Trivadis GmbH
München

Schlüsselworte:

APEX, jQuery UI, Dialogs, Tabs

Einleitung

Die von Apex erstellten HTML-Seiten bieten selbst wenig Dynamik. Durch das Einbinden von jQuery UI-Elementen können die Seiten und ihre Bedienung durch die kontextbezogene Anzeige von Informationen benutzerfreundlicher gestaltet werden.

Im folgenden wird gezeigt, wie mit jQuery UI Dialogs Teile einer Seite modal eingeblendet werden, wenn sie benötigt werden.

Sollen dem Anwender viele Daten übersichtlich zur Verfügung gestellt werden, bietet sich die Darstellung mit jQuery UI Tabs an.

Die Einbindung der jQuery UI – Elemente in APEX lässt sich dabei sehr einfach umsetzen.

Voraussetzungen

Alle in diesem Dokument beschriebenen Funktionalitäten sind mit APEX 4.1, jQuery 1.6.2, jQuery UI 1.8.14 sowie jQuery UI Tabs 1.8.14 durchgeführt worden.

Ein erstes Dialog-Fenster: Bitte warten

Wenn eine Verarbeitung von Daten länger dauert, soll der Anwender einerseits informiert werden, dass die Verarbeitung läuft. Andererseits soll er daran gehindert werden, weitere Aktionen auf der Seite auszulösen oder dieselbe Verarbeitung nochmals zu starten.

Zu diesem Zweck bietet es sich an, durch ein modales Fenster die weitere Benutzung der Seite zu unterbinden, so lange die Verarbeitung andauert. Dieses modale Fenster wird beim Submit eingeblendet und bleibt angezeigt, bis die nächste Page geladen wird:



Damit die Funktionalität auf allen Seiten zur Verfügung steht, werden die benötigten Objekte auf der Page 0 erstellt.

Für dieses Dialog-Fenster sind drei Objekte nötig:

Zunächst wird eine HTML-Region angelegt, die den HTML-Code für die Darstellung des Texts und des Bildes enthält. Damit diese Region später einfach mittels jQuery angesprochen werden kann, wird sie mit der statischen ID BITTE_WARTEN versehen. Für die Region Source wird folgender Code verwendet:

```
<div style="height: 200px; width: 600px;
        display: table-cell; vertical-align: middle;"
    align="center">
    <h2>Verarbeitung läuft ...</h2>
    
</div>
```

Um diese Region als modales Fenster zu verwenden, wird eine Dynamic Action angelegt, die beim Page Load die Region als jQuery Dialog initialisiert. Die Dynamic Action führt dazu diesen jQuery Code aus:

```
$("#BITTE_WARTEN").dialog ({
    autoOpen: false,
    title:     "Bitte warten",
    height:   240,
    width:    650,
    modal:    true,
    open:     function(event, ui) {
                $(".ui-dialog-titlebar-close").hide(); }
    })
```

Zuletzt wird noch eine Dynamic Action angelegt, die dafür sorgt, dass beim Submit („Before Page Submit“) das Dialog-Fenster angezeigt wird:

```
$("#BITTE_WARTEN").dialog ( "open" );
```

Auswahlfenster: Anzeige von Daten im Dialog

In APEX können mit den Popup LOVs dem Anwender Daten zur Auswahl angeboten werden. Allerdings kann nur eine Spalte angezeigt und nur eine Spalte zurückgegeben werden. Das Fenster für die Popup LOV ist dabei ein separates Fenster und nicht modal.

Stellt man dagegen eine Report-Region als jQuery Dialog dar, kann man viele Spalten in einem modalen Fenster präsentieren. Mit Eingabefeldern und einem Refresh des Report ohne Submit wird eine Suche im Auswahlfenster ermöglicht. Die Rückgabe der Daten wird über JavaScript umgesetzt. Damit dies funktioniert, muss der Report auf derselben Seite wie das Ziel für die Daten definiert sein.

Zunächst wird auf der gewünschten Seite die Reports-Region erstellt und die Suchfelder angelegt. Die Where-Bedingung der Abfrage wird um alle Bedingungen aus den Suchfeldern erweitert. Die Suchfelder werden ebenfalls unter „Page Items to Submit“ aufgelistet, damit die Abfrage des Report auf die Werte zugreifen kann, die der Anwender in die Suchfelder eingetragen hat. Zum einfacheren Zugriff über jQuery wird die Reports-Region mit einer statischen ID versehen.

Zum Auslösen der Suche wird ein Button angelegt mit der Action „Defined by Dynamic Action“. Die durch den Button ausgelöste Dynamic Action führt den Refresh auf die Reports-Region durch. Nun muss die Region als jQuery Dialog initialisiert werden. Dies funktioniert wie für das „Bitte warten“-Fenster.

Die Anzeige des Dialogs muss in diesem Fall vom Anwender ausgelöst werden. Daher wird in der Page ein Button angelegt mit der Action „Defined by Dynamic Action“. Die Dynamic Action zum Button enthält den Code zum Anzeigen des Dialogs:

```
$("#REP_MITARBEITER_SUCHE").dialog ( "open" );
```

Zur Rückgabe der Daten in die Zielfelder muss eine JavaScript-Funktion programmiert werden. Diese wird entweder in den Page-Eigenschaften oder in einer JavaScript-Library hinterlegt. Die Function verteilt die ausgewählten Daten in die gewünschten Items und schließt danach das Dialog-Fenster:

```
function WerteSetzen ( pID , pNachname , pVorname ){  
    $s("P20_MIT_ID",pID);  
    $("#P20_VORGESETZTER").text(pNachname+", "+pVorname);  
    $("#REP_MITARBEITER_SUCHE").dialog("close");  
}
```

Im Report wird eine Link-Spalte definiert, die diese Funktion aufruft. Als Parameter werden dabei die Werte der Spalten im aktuellen Datensatz des Report übertragen:

```
javascript:WerteSetzen('#MIT_ID#','#MIT_NACHNAME#','#MIT_VORNAME#');
```

Auswahl	Nachname	Vorname	eMail
✓	King	Steven	SKING
✓	Kochhar	Neena	NKOCHHAR
✓	De Haan	Lex	LDEHAAN
✓	Ernst	Bruce	BERNST
✓	Austin	David	DAUSTIN
✓	Pataballa	Valli	VPATABAL
✓	Lorentz	Diana	DLORENTZ
✓	Greenberg	Nancy	NGREENBE
✓	Faviet	Daniel	DFAVIET
✓	Chen	John	JCHEN
✓	Sciarra	Ismael	ISCIARRA
✓	Urman	Jose Manuel	JMURMAN
✓	Popp	Luis	LPOPP
✓	Raphaely	Den	DRAPHEAL

Darstellung vieler Detail-Daten: Tabs

Sollen zu einer Master-Tabelle mit vielen Detail-Tabellen diese abhängigen Daten alle angezeigt werden, wird der Platzbedarf sehr hoch. Mit Hilfe von Tabs (Karteikarten) können diese Daten dennoch übersichtlich auf der Seite untergebracht werden:



Projekt	Funktion	Prozent	Start	Ende
PR0156-APEX-Entwicklung	Projektleiter	50	15.03.2012	31.12.2012
PR0156-APEX-Entwicklung	Anwendungsdesigner	50	15.03.2012	31.12.2012
PR0157-APEX-Bestellverwaltung	Projektleiter	10	01.01.2012	25.06.2012
PR0157-APEX-Bestellverwaltung	Anwendungsdesigner	40	01.01.2012	25.06.2012
PR0157-APEX-Bestellverwaltung	Programmierer	50	01.01.2012	25.06.2012
PR0160-Konzept APEX-Anwendung	Datenmodellierer	75	17.04.2012	22.09.2012
PR0160-Konzept APEX-Anwendung	Anwendungsdesigner	25	17.04.2012	22.09.2012

Vorarbeiten:

Da APEX die für die Tabs benötigten js – und css – Dateien nicht standardmäßig einbindet, müssen die betroffenen Pages oder Page-Templates um den entsprechenden Aufruf erweitert werden:

```
<link rel="stylesheet" href="/i/libraries/jquery-  
ui/1.8.14/themes/base/minified/jquery.ui.tabs.min.css" type="text/css" />  
  
<script src="/i/libraries/jquery-  
ui/1.8.14/ui/minified/jquery.ui.tabs.min.js" type="text/javascript">  
</script>
```

Um die Tabs in APEX umzusetzen, ist es nötig, zunächst die Struktur des HTML-Codes zu betrachten, die von jQuery für die Tabs erforderlich ist. Aus dem HTML-Code

```
<div id="jQueryTabs">  
  <ul>  
    <li><a href="#tab-1">Erstes Tab</a></li>  
    <li><a href="#tab-2">Zweites Tab</a></li>  
  </ul>  
  <div id="tab-1">  
    Code erstes Tab  
  </div>  
  <div id="tab-2">  
    Code zweites Tab  
  </div>  
</div>
```

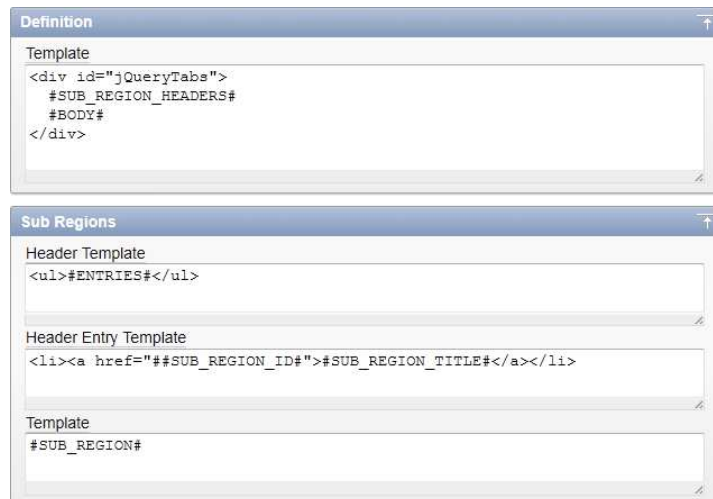
und dem Aufruf

```
$( "#jQueryTabs" ).tabs();
```

erzeugt jQuery die Darstellung



Um diese Struktur in APEX nachzubilden, wird ein neues Region-Template für eine Region mit Subregions benötigt:



Auf der APEX-Page wird dann eine neue HTML-Region mit diesem Template angelegt; alle Regions, die auf Karteikarten dargestellt werden sollen, werden dieser Region als Subregions zugeordnet.

Damit die HTML-Region und ihre Subregions als Karteikarten dargestellt werden, muss sie beim Page Load über eine Dynamic Action als jQuery Tabs initialisiert werden:

```
$( "#jQueryTabs" ).tabs ( ) ;
```

Kontaktadresse:

Dr. Gudrun Pabst
Trivadis GmbH
Lehrer-Wirth-Straße 4
D-81829 München

Telefon: +49 (0) 89-99 27 59 30
Fax: +49 (0) 89-99 27 59 59
E-Mail: gudrun.pabst@trivadis.com
Internet: www.trivadis.com