

Erweiterte Interaktion mit Trees und Charts in APEX

Dr. Gudrun Pabst

Trivadis GmbH
Lehrer-Wirth-Straße 4
81829 München

gudrun.pabst@trivadis.com

BASEL BERN LAUSANNE ZÜRICH DÜSSELDORF FRANKFURT A.M. FREIBURG I.BR. HAMBURG MÜNCHEN STUTTGART WIEN

1

2012 © Trivadis

Erweiterte Interaktion mit Trees und Charts in APEX

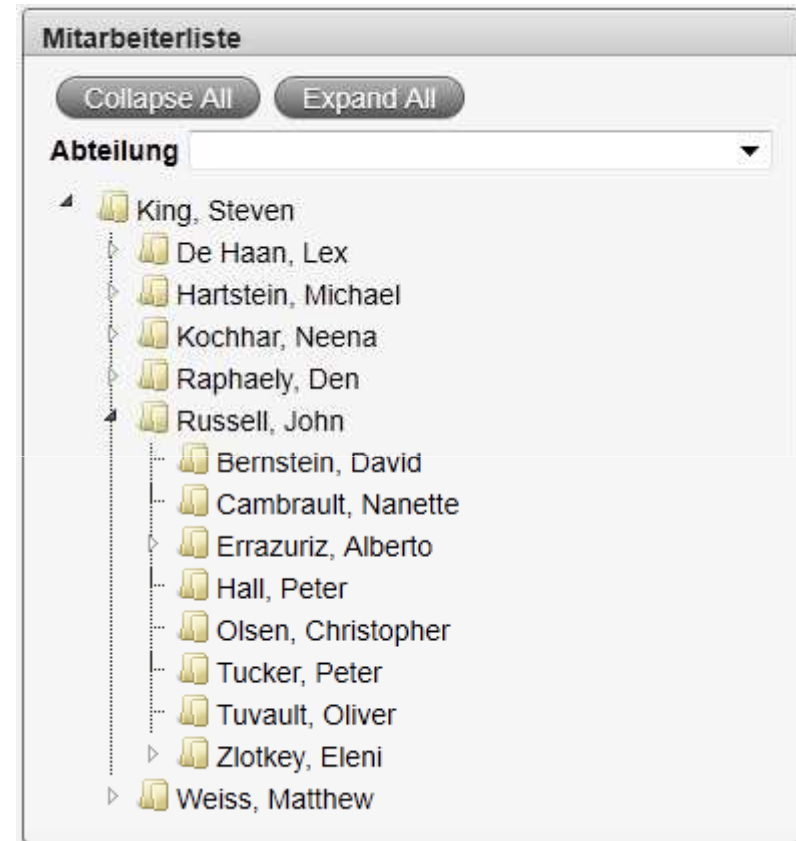
trivadis
makes IT easier. ■ ■ ■

AGENDA

- **Trees**
- **Charts**
 - Ein- und Ausblenden von Datenreihen
 - Dynamische Anzahl von Datenreihen

Trees

- Hierarchische Darstellung von Daten in APEX
 - derzeit noch kein Refresh ohne Submit möglich
 - hier: bei Auswahl einer Abteilung Anzeige der Mitarbeiter der Abteilung
- Lösung:
 - Austausch der Tree-Knoten mit einer Dynamic Action



Trees

- Struktur des Tree (vereinfacht ohne Links, ...)

- Daten liegen als JSON-Objekt vor:

```
[{attributes:{"id":"0"},data:{"title":"Txt0", attributes:{}},  
  children:[{  
    attributes:{"id":"1"},data:{"title":"Txt1", attributes:{}},  
    children: [ ... ]} ] ]
```

- Konvertierung in Tree-Darstellung durch Aufruf von `apex.widget.tree.init`

- Struktur der Daten anschließend:

```
<ul class="ltr">  
  <li id="0" class="last open">  
    <a class="" style="" href=""><ins>&nbsp;</ins>Txt0</a>  
    <ul>  
      <li id="1" class="open">  
        <a class="" style="" href=""><ins>&nbsp;</ins>Txt1</a>  
        <ul>  
          ...
```

Trees

- Konvertierte Daten

- Listeneinträge `` in HTML-Listen ``
- Start-Liste hat die Klasse `ltr`
- Klasse des Eintrags gibt den Status des Eintrags an:

<code>open</code>	geöffneter Knoten mit Unterknoten
<code>leaf</code>	Knoten ohne Unterknoten
<code>closed</code>	geschlossener Knoten mit Unterknoten

- Zusätzliche Klasse `last` für den letzten Eintrag eines Zweigs



Trees

- Beim Refresh jetzt: Aufbau des benötigten HTML-Code
- Bei kleinen Ergebnismengen:
 - Dynamic Action vom Typ „Execute PL/SQL Code“
 - Rückgabe des HTML-Code in ein Item der aktuellen Seite
- Bei Ergebnissen mit mehr als 4000 Zeichen:
 - On-Demand-Prozess
 - Rückgabe des Ergebnisses mittels http.p
 - Aufruf des On-Demand-Prozess über eine Dynamic Action vom Typ „Execute JavaScript Code“

Auslöser hier: Änderung der Select-Liste

AGENDA

- Trees
- **Charts**
 - **Ein- und Ausblenden von Datenreihen**
 - Dynamische Anzahl von Datenreihen

Charts

- Region Source beim Erstellen: Object

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"  
        codebase="#HOST_PROTOCOL#://download.macromedia.com/  
                pub/shockwave/cabs/flash        ...  
</object>
```

- Kein Zugriff auf das Objekt:

- Ein- und Ausblenden von Datenreihen nicht umsetzbar
- Dynamische Anzahl von Datenreihen mit Standard-Methoden nicht möglich

- Alternative: AnyChart JavaScript Integration Library

- Erstellen des Charts mit JavaScript
- Chart steht als JavaScript-Objekt zur Verfügung und kann manipuliert werden



2012 © Trivadis

Erweiterte Interaktion mit Trees und Charts in APEX

Einbinden des Chart mit JavaScript

- Voraussetzung: AnyChart - JavaScript-Bibliothek einbinden

- Einbinden in der Chart – Page oder im Page Template

- Code:

```
<script src="#IMAGE_PREFIX#flashchart/anychart_5/js/AnyChart.js"
      type="text/javascript">
</script>
```

- Ersetzen der Region Source durch JavaScript – Code:

```
<div id="c_#REGION_ID#"></div>
<script type="text/javascript" language="javascript">
    gvChart = new AnyChart('#IMAGE_PREFIX#flashchart/anychart_5/
                          swf/#CHART_TYPE#.swf');
    ...
    gvChart.write("c_#REGION_ID#");
</script>
<!-- #FLOW_FLASH_CHART5_R#
      wichtig, sonst wird neue Source nicht übernommen -->
```

Ein- und Ausblenden von Datenreihen

- Problem: Generiertes XML enthält keine ID für Series

- Lösung

- XML auslesen:

```
$.ajax ( {  
  url:      "#HOST#apex_util.flash?p=&APP_ID.:&APP_PAGE_ID.: " +  
            "&APP_SESSION.:FLOW_FLASH_CHART5_R#REGION_ID#" ,  
  type:     "GET" ,  
  datatype: "xml" ,  
  success:  function(pXML) { vXML = pXML; }    });
```

- IDs nachtragen:

```
$vXMLObjekt.find("series").each(  
  function() { var vSeriesName = $(this).attr("name");  
                $(this).attr("id", vSeriesName);    });
```

- XML an Chart übergeben:

```
gvChart.setData ( vXMLString );
```



Ein- und Ausblenden von Datenreihen

- Zugriff auf Chart über globale JavaScript – Variable
- Zugriff auf Datenreihe über ID
- Ein- und Ausblenden der Datenreihe :

```
gvChart.showSeries ( "Electronics" , false );  
gvChart.refresh() ;
```

- Nutzung der Standardfunktionalitäten des Chart
- Für dynamische Anzahl von Datenreihen zu wenig

AGENDA

- Trees
- **Charts**
 - Ein- und Ausblenden von Datenreihen
 - **Dynamische Anzahl von Datenreihen**

Dynamische Anzahl von Datenreihen

- Problem:
Hinterlegte Selects legen Anzahl der Datenreihen fest

- Lösung

- Custom XML => YES
- Platzhalter #DATA# ersetzen durch Referenz auf Page Items (hier für Page 30):

```
<data>
&P30_XML.
</data>
<data_sets>
  <csv_data_set name="data"><![CDATA[
&P30_DATEN.
]]></csv_data_set>
</data_sets>
```

- Page Item P30_XML füllen mit XML für AnyChart (inklusive IDs für Series)
- Page Item P30_DATEN füllen mit Daten im CSV-Format



Dynamische Anzahl von Datenreihen

- XML für AnyChart (Daten werden als CSV bereitgestellt)

```
<series id="Electronics" type="Line" data_source="CSV"
        color="0x1D8BD1" name="Electronics">
  <csv_data_mapping data_set="data">
    <field column="0" name="name" />
    <field column="1" name="y" />
  </csv_data_mapping>
</series>

<series id="Hardware" type="Line" data_source="CSV"
        color="0xF1683C" name="Hardware">
  <csv_data_mapping data_set="data">
    <field column="0" name="name" />
    <field column="2" name="y" />
  </csv_data_mapping>
</series>
```

Dynamische Anzahl von Datenreihen

- Dynamische Anzahl von Datenreihen mittels der Page Items
 - "Series" – Blöcke dynamisch erstellen und in das XML-Item schreiben
 - CSV-Daten dynamisch erstellen und in das Daten-Item schreiben

```
...
v_select := 'select listagg(text,chr(10)) within group (order by qrtr)' ||
            ...;
for r_kat in c_kat loop
    v_series_xml := v_series_xml ||
        '<series id="' || r_kat.kat_name || '" ...>' ||
        ' <csv_data_mapping data_set="data">' ||
        ' <field column="0" name="name" />' ||
        ' <field column="' || v_wert_col || '" name="y" />' ||
        '</csv_data_mapping>' ||
        '</series>';
--
    v_select := v_select || ',sum ( ... )';
end loop;
...
:P30_XML := v_series_xml;
execute immediate ( v_select ) into :P30_DATEN;
end;
```

Zusammenfassung

- Trees :
 - Einfaches Nachladen durch Ersetzen des HTML-Code
 - Kenntnis der Struktur nötig

- Charts :
 - Umstellen auf Erstellen durch JavaScript
 - Zugriff auf das Chart über eine globale JavaScript-Variable

 - Erweitern des Standard-XML durch IDs für Datenreihen
 - Danach Ein- und Ausblenden von Datenreihen möglich

 - Custom XML mit eigenen Informationen über Daten und Datenreihen
 - Dadurch dynamische Anzahl von Datenreihen einbaubar

VIELEN DANK

Dr. Gudrun Pabst

Trivadis GmbH

Lehrer-Wirth-Straße 4

81829 München

gudrun.pabst@trivadis.com

BASEL

BERN

LAUSANNE

ZÜRICH

DÜSSELDORF

FRANKFURT A.M.

FREIBURG I.BR.

HAMBURG

MÜNCHEN

STUTTGART

WIEN

17

2012 © Trivadis

Erweiterte Interaktion mit Trees und Charts in APEX

trivadis
makes IT easier. ■ ■ ■