



Apex – Tipps und Tricks

Lisa Klimesch, Muniqsoft

Es sind oftmals die kleinen Features, die – vielleicht etwas versteckt oder nicht auf den ersten Blick plausibel – übersehen werden oder so manchem Apex-Entwickler gar nicht erst bekannt sind. Features, die das Entwickeln von Applikationen erleichtern und verbessern können. In diesem Artikel sind einige Tipps und Tricks zu eben solchen Features zusammengestellt.

Hidden Items verschlüsseln

Wie der Name bereits sagt, liegen Hidden Items als versteckte Elemente auf Applikationsseiten. Die Apex Session State Protection verhindert zwar mittels Prüfsummen, dass Werte der Hidden Items bei Übergabe von einer Seite zur anderen geändert werden, der Inhalt der Hidden Items kann

dabei allerdings im HTML-Code der Seite problemlos ausgelesen werden. Das ist nicht immer gewünscht. Eine Möglichkeit, die Sicherheit von Hidden Items zu erhöhen, ist, sie zu verschlüsseln. Für dieses Vorgehen stellt die Funktion in *Listing 1* die Grundlage dar.

In der Datenbank sollte dabei der Zeichensatz „UTF8“ verwendet werden. Nun

kann ein Hidden Item mithilfe der Funktion `Crypt` verschlüsselt werden, beispielsweise durch eine Computation mit dem Ausführungspunkt „Before Header“ und dem PL/SQL-Funktion-Body „RETURN crypt('Hallo Welt');“.

Soll der Wert auch wieder entschlüsselt werden, kann dafür ein After-Submit-Prozess mit dem PL/SQL-Code „:P1_

DECRYPT := crypt(:P1_CRYPT, cryptmode => 'D');“ genutzt werden. Eine Alternative zu diesem Vorgehen stellen Application Items dar: Sie werden nicht direkt auf der Seite, sondern auf Applikationsebene angelegt und sind daher sowohl vor neugierigen Blicken als auch vor ungewollter Werte-Änderung geschützt.

Grid Layout in Apex 5.0

Apex 5.0 bietet für das Layout einer Applikation zahlreiche tolle Features. Trotz allem birgt das neue Grid Layout dabei auch ein wenig Verwirrungs-Potenzial, beispielsweise im Kontext einer Fehlermeldung beim Seitenladen (siehe Abbildung 1).

Wer auf diese Meldung stößt, sollte sich im Page Designer mit den Eigenschaften „Column Span“ und „Label Column Span“ (zu finden unter der Rubrik „Grid“) beschäftigen, insbesondere mit dem Element-Typ „Item“. Aber beginnen wir von vorn ...

Grundsätzlich stehen Seiten und darin enthaltenen Komponenten einer Applikation ab Apex 5.0 jeweils zwölf Spalten zur Verfügung. So auch dem Komponenten-Typ „Item“. Hinzu kommt, dass Items aus zwei Bereichen bestehen: „Label“ und „Inhalt“. Der Default für die Breite des Item-Labels („Page Template“-Default) beträgt drei Spalten, während für die Breite des Item-Inhalts standardmäßig der Wert „Automatic“ gesetzt ist. Der Inhalt nimmt also so viele Spalten einer Region ein, wie möglich (maximal 9, siehe Abbildung 2).

Anhand der Attribute „Column Span“ und „Label Column Span“ können diese Bereiche im Page Designer definiert

```
CREATE OR REPLACE FUNCTION crypt (
  text IN VARCHAR2,
  key IN VARCHAR2 DEFAULT lpad('MunigSoft', 32, '#;'),
  cryptmode IN VARCHAR2 DEFAULT 'E')
RETURN VARCHAR2 IS
  p_typ PLS_INTEGER:=4360;
  -- ENCRYPT_AES256+chain_cbc+ pad_pkcs5;
BEGIN
  IF substr(upper(cryptmode),1,1)='E' THEN
  -- Verschlüsselung
    RETURN (sys.dbms_crypto.encrypt(
      src => sys.UTL_I18N.STRING_TO_RAW(text,'AL32UTF8'), typ => p_typ,
      key => sys.utl_raw.cast_to_raw(key));
  ELSE
    -- Entschlüsselung
    RETURN sys.UTL_I18N.RAW_TO_CHAR (sys.dbms_crypto.decrypt(
      src => text,
      typ => p_typ,
      key => sys.utl_raw.cast_to_raw(key)));
  END IF;
END;
```

Listing 1

werden: „Column Span“ meint dabei die Breite des gesamten Items (inklusive Label und Inhalt). Die Eigenschaft „Label Column Span“ bezieht sich dagegen ausschließlich auf die Spaltenbreite des Item-Labels. Wird der Wert für „Column Span“ nun aus Platzgründen oder nach einer Migration auf beispielsweise zwei Spalten reduziert, während für das Label immer noch der Default-Wert von drei Spalten festgelegt ist, kommt es zu obiger Fehlermeldung. Allein das Label würde mehr Spalten einnehmen, als dem Item insgesamt zustehen – das Item kann in diesem Fall nicht mehr angezeigt werden, weil es schlichtweg nicht mehr in die vorgesehene Spaltenanzahl hineinpasst.

Die Lösung: „Column Span“ ist so zu vergrößern, dass das Item mitsamt Label- und Inhaltsbereich wieder in die angegebene

Spaltenanzahl passt. Am einfachsten geht das mit der Einstellung „Automatic“ – in jedem Fall jedoch mit einem höheren Wert als der des „Label Column Span“, sodass

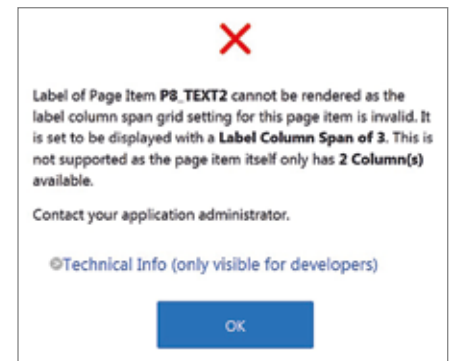


Abbildung 1: Fehlermeldung zum „Invalid Grid Setting“



Abbildung 2: Spalten-Layout im Universal Theme

genug Platz für den Item-Inhalt bleibt (siehe *Abbildung 3*). Alternativ kann auch „Label Column Span“ reduziert oder – sofern gewünscht – auf „0“ gesetzt werden. Das Label wird dann gar nicht mehr angezeigt.

Template-Variablen

In Apex lassen sich Variablen auf fünf verschiedene Arten referenzieren: als sogenannte „Substitution Strings“ in HTML-Umgebungen („&APP_USER.“), als Bind-Variablen für SQL- oder PL/SQL-Blöcke („:APP_ID“), als V/NV-Funktionen im PL/SQL-Kontext („V('APP_USER'“), als Shortcuts für Regionen und Items („HELP_TEXT“) oder als Template-Substitutions („#HOME_LINK#“). Für einige der Universal Theme Templates in Apex 5.0 sind die Template Substitutions besonders interessant.

Ist eine Template-Substitution im aktuellen Template (Page, Region, List, ...) enthalten, kann sie in der Applikation definiert und genutzt werden. Im „Shared Components“-Menü „Edit Application Definition“ lässt sich so beispielsweise der Substitution String „#GLOBAL_NOTIFICATION#“ im Seiten-Template füllen oder ein Favicon (Voraussetzung: „#FAVICON#“ im Seiten-Template) anhand des statischen Substitution Strings „APP_FAVICON“ und eines dazugehörigen Bild-Wertes festlegen (siehe *Abbildung 4*).

Das Universal Theme in Apex 5.0 bringt unter anderem einige neue Report-Templates mit, deren Nutzung unter Verwendung von Substitution Strings am Beispiel des Report-Templates „Timeline“ erklärt wird. Ein Blick in das Template (Shared Components → Templates → Report Timeline) verrät, welche Substitution Strings grundsätzlich verfügbar sind und durch einen Report gefüllt werden können. Die Strings in schwarzer Farbe stehen für Daten, die später im Report sichtbar dargestellt sind. In unserem Beispiel sind das „#USER_AVATAR#“, „#USER_NAME#“, „#EVENT_DATE#“, „#EVENT_TYPE#“, „#EVENT_TITLE#“ und „#EVENT_DESC#“. Substitution Strings in roter Farbe dagegen sind Teil von HTML-Klassen und können zur HTML- und CSS-Bearbeitung genutzt werden, hier „#EVENT_MODIFIERS#“, „#USER_COLOR#“, „#EVENT_STATUS#“ und „#EVENT_ICON#“ (siehe *Abbildung 5*).

Erstellt man nun damit und mit der simplen SQL-Query „select 1 from dual“ einen Bericht, so werden alle im Template definierbaren Substitution Strings angezeigt (siehe *Abbildung 6*). In der Report Query können die Substitution Strings nun als Spalten-Alias verwendet und damit gefüllt werden (siehe *Listing 2*).

Schließlich fehlt noch die Definition von CSS-Klassen-Selektoren, die sich auf die HTML/CSS-Substitution-Strings „#USER_

COLOR#“ und „#EVENT_STATUS#“ beziehen. Sie erfolgt zum Beispiel im „Custom CSS“-Bereich des Theme Rollers (siehe *Listing 3*). *Abbildung 7* zeigt den Report.

Hilfetexte verwalten

Auch für Item-Hilfetexte ist es praktisch, statische Substitution Strings festzulegen, etwa wenn ein Hilfe-Text für mehrere Items genutzt werden soll. Dafür werden zunächst ein String (zum Beispiel „EMAIL_HILFE“) und der dazugehörige Wert (zum Beispiel „Geben Sie Ihre E-Mail-Adresse ein“) im „Shared Components“-Menü unter „Edit Application Definition“ definiert. Anschließend wird im Page Designer der Substitution String für das Item unter der Eigenschaft „Help“ mit der Syntax „&EMAIL_HILFE.“ aufgerufen.

Eine Übersicht über die Hilfetexte einer Applikation findet sich unter „Utilities“ im Application Builder. Über die rechte Taskleiste („Page Specific Utilities“) und den Menüpunkt „Item Utilities“ gelangt man zum Bereich „Grid Edit of all Item Help Text“. Dort sind alle Hilfetexte mit Angaben zu Seite und Item aufgelistet und können beliebig verändert werden. Ein Default-Text für Items, denen noch kein Hilfetext zugewiesen wurde, lässt sich über den Button „Seed Item Help Text“ definieren (siehe *Abbildung 8*).

In einer Beispiel-Applikation sind zwei der Tipps nachgebaut (siehe „<https://apex.oracle.com/pls/apex/f?p=96478>“, Benutzername und Passwort lauten „demo“).

Hinweis: Die Abbildungen 3 bis 8 finden sie online unter: www.doag.org/go/redstack/klimesch_abb

```
select
  'IQ'           as USER_AVATAR,
  'Munisoft'    as USER_NAME,
  '14.03.2016'  as EVENT_DATE,
  'Kurs'        as EVENT_TYPE,
  'Apex I'     as EVENT_TITLE,
  'Grundlagen der Anwendungsentwicklung mit Apex'
  as EVENT_DESC,
  'user_blue'  as USER_COLOR,
  'kurs_status' as EVENT_STATUS,
  'fa-book'   as EVENT_ICON
from dual
```

Listing 2

```
.user_blue {
  background-color: #58D3F7;
}
.kurs_status {
  background-color: #82FA58;
}
```

Listing 3



Lisa Klimesch
lisa.klimesch@muniqsoft.de