

Oracle APEX 4.0 mit Plug-ins selbst erweitern!

Patrick Wolf
Oracle Austria GmbH
Wien

Schlüsselworte:

Oracle APEX, Plug-Ins, Erweiterbarkeit

Einleitung

Eines der wesentlichen neuen Features von Oracle Application Express 4.0 sind "Plug-Ins", mit deren Hilfe die existierende Funktionalität von Oracle APEX um zusätzliche deklarative Komponenten erweitert werden kann. Verpassen Sie nicht die Chance alles darüber direkt von der Quelle, dem APEX Entwickler dieses Features, zu erfahren!

Sie erfahren alles notwendige um sofort mit der Entwicklung Ihrer eigenen Plug-Ins starten zu können. Dabei werden Themen wie Dynamic SQL, JavaScript, Performance und eventuelle Sicherheitsaspekte angesprochen und durchleuchtet. Wissenswertes über die Architektur des Plug-In Sub-Systems geben Ihnen zusätzliche, unschätzbare Hintergrundinformationen. Zu guter Letzt lernen Sie noch alle dokumentierten und vielleicht auch ein paar undokumentierte Utility Funktionen kennen, damit Sie sofort produktiv Plug-Ins entwickeln können.

Was sind Plug-Ins?

Sie sind die einfachste Möglichkeit Oracle APEX um neue Item-, Region-, Process- und Dynamic Action Types zu erweitern und diese gleichberechtigt neben den Standard Komponenten zu verwenden. Diese neuen Komponenten wie zum Beispiel ein Star Rating wie wir es von Amazon kennen um Bücher zu bewerten, fügen sich nahtlos in die APEX Entwicklungsumgebung ein und erlauben es dann jedem Entwickler diese einfach, schnell und deklarativ in eigenen Applikationen zu verwenden. Plug-Ins selbst werden in PL/SQL geschrieben.

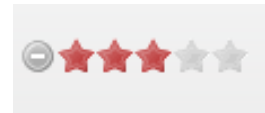


Abb. 1: Star Rating Item Type Plug-In

Plug-Ins verwenden

Bevor Sie jetzt aber sofort loslegen Plug-Ins selbst zu entwickeln, werfen Sie zuvor einen Blick auf die bestehenden Plug-Ins die Ihnen von Oracle und der weltweiten APEX Community fix fertig zur Verfügung gestellt werden. Unter <http://apex.oracle.com/plugins/> und <http://www.apex-plugin.com/> finden Sie für verschiedene Anwendungsgebiete bereits fertige Plug-Ins, welche Sie sofort in Ihren Applikationen einsetzen können.

Plug-Ins sind ganz einfach zu installieren. Alle notwendigen Meta Daten welche das Aussehen und Verhalten im Application Builder definieren, der PL/SQL Code um die neue Komponente dann schliesslich in einer APEX Seite anzuzeigen und die zusätzliche Dateien wie JavaScript-, CSS- und Bild-Dateien befinden sich in einer SQL Datei die wie von Oracle APEX gewohnt über die Import Funktion installiert werden kann. Plug-Ins werden als „Shared Component“ innerhalb einer

Applikation abgespeichert. Das hat den Vorteil, dass bei einem Applikationsexport diese automatisch inkludiert sind und dadurch die Installation der Applikation auf einem anderen System super einfach bleibt und keine weiteren Dateien mühsam installiert werden müssen.

Nach dem Import eines Plug-Ins kann dieses sofort verwendet werden. In den verschiedenen Assistenten zum Erstellen einer neuen Komponente befindet sich die neue Auswahl Option „Plug-Ins“, hinter der sich die Liste der installierten Plug-Ins dieses Komponenten Typs befindet. Aber auch eine bestehende Komponente wie zum Beispiel ein Page Item kann einfach geändert werden. Die Auswahlliste „Display As“ zeigt neben den Standard Item Types auch die installierten Plug-Ins mit der Kennzeichnung *[Plug-Ins]* an. Abhängig vom ausgewählten Plug-In wird dann in der Region „Settings“ noch nach weiteren Plug-In spezifischen Einstellungen gefragt. Falls der Entwickler einen Hilfetext eingegeben hat, steht auch für diese zusätzlichen Attribute eine Online Hilfe zur Verfügung.

Um die Performance eines Plug-Ins zu steigern, kann jederzeit der PL/SQL Code auch in ein Package ausgelagert werden, damit dieser bereits kompiliert zur Verfügung steht. Weiters können die statischen Dateien des Plug-Ins, wie zum Beispiel JavaScript Libraries oder Bilder in ein Verzeichnis am Web Servers kopiert werden. Im Plug-In selbst muss dann nur noch das Attribute „File Prefix“ auf dieses Verzeichnis (z.B. `#IMAGE_PREFIX#/plugins/star_rating_v1_0/`) geändert werden. Alles weitere übernimmt automatisch die Plug-In Infrastruktur.

Als letzter kleiner Tipp: Wird das selbe Plug-In in verschiedenen Applikationen eines Workspaces verwendet, dann sollte das Plug-In in eine Master Applikation installiert werden und in den anderen Applikationen mit der „Subscribe“ Funktionalität referenziert werden. Damit muss bei einer Änderung des Plug-Ins dieses nur in der Master Applikation aktualisiert werden und alle anderen Applikationen werden ohne eine weitere Installation automatisch aktualisiert.

Plug-Ins selbst erstellen

Finden Sie aber nicht das passende Plug-In, dann erstellen Sie einfach Ihr eigenes! Das ist nicht besonders schwierig, da Sie meistens nur einen PL/SQL Wrapper um eine bestehende JavaScript Library schreiben müssen. Gehen Sie zum Beispiel auf <http://plugins.jquery.com> um ein passendes JavaScript basierendes Widget zu finden und dann können Sie auch schon loslegen.

Starten Sie in dem Sie die JavaScript Library mal manuell in Ihre APEX Seite integrieren um sich mit der Funktionalität und den notwendigen Aufrufen und Optionen vertraut zu machen.

Nachdem Ihr neues Widget erfolgreich in Ihrer Seite funktioniert, können Sie dazu übergehen ein APEX Plug-In dafür zu erstellen. Klicken Sie dazu in den *Shared Components > User Interface > Plug-Ins* auf den „Create“ Button um die notwendigen Meta Daten für Ihr Plug-In zu definieren.

1. **Name:** Sichtbare Bezeichnung der neuen Komponente (z.B. Star Rating)
2. **Internal Name:** Name unter dem das Plug-In intern referenziert wird. Dieser Wert muss innerhalb Ihrer Applikation eindeutig sein. Wenn Sie das Plug-In unter einer Open Source Lizenz der APEX Community zur Verfügung stellen, dann sollte dieser Wert weltweit eindeutig sein. Am besten verwenden Sie generell Ihren umgedrehten Domain Namen wie das bereits in der Java Welt bei Klassen üblich ist. Ein Beispiel für so einen Namen wäre `de.ihrfirmenname.star_rating`
3. **Type:** Definiert um welche Art von Plug-In es sich handelt (Item, Region, Process oder Dynamic Action)

4. **PL/SQL Code:** Der hier angegebene Code ist verantwortlich dafür die notwendigen HTML Tags und JavaScript Aufrufe für Ihr Plug-In in die APEX Seite auszugeben. Dieses Feld kann aber auch leer bleiben und gerade bei der Entwicklung ist es angenehmer den Code in einem Package zu speichern, damit man seine vorhandenen Entwicklungstools wie zum Beispiel SQL Developer, TOAD oder PL/Developer verwenden kann.
5. **Callbacks:** Abhängig vom gewählten Plug-In Typ werden hier unterschiedliche Callback Felder angeboten. Bei einem Item Type Plug-In ist das zum Beispiel
 - Render Function Name
 - AJAX Function Name und
 - Validation Function Name

Diese Callbacks bieten Ihnen die Möglichkeit sich in die APEX Engine reinzuhängen, welche dann Ihren Code zu verschiedenen Zeitpunkten automatisch ausführt. Die Callback Funktionen müssen einem vorgegebenen Interface entsprechen, welches in der Online Hilfe des entsprechenden Callbacks dokumentiert ist. Wie Sie dem obigen Beispiel entnehmen können ist es auch sehr einfach eine AJAX Funktionalität in Ihrem Plug-In anzubieten, welches dann on-demand zusätzliche Daten vom Server nachladen kann. Beispiel für das Render Callback Interface eines Item Type Plug-Ins:

```
function render_item_type (
  p_item           in apex_plugin.t_page_item,
  p_plugin         in apex_plugin.t_plugin,
  p_value          in varchar2,
  p_is_readonly   in boolean,
  p_is_printer_friendly in boolean )
  return apex_plugin.t_page_item_render_result
```

6. **Standard Attributes:** Damit haben Sie die Möglichkeit bestehenden Attribute eines Komponenten Typs auch in Ihrem Plug-In zu verwenden. Zum Beispiel ob Ihr Plug-In eine LOV verwendet, Quick Picks anbietet oder Read Only angezeigt werden kann.
7. **Custom Attributes:** Sind die Standard Attributes nicht ausreichend, dann können auch bis zu 10 eigene Attribute im Application Builder angezeigt werden wenn Ihr Plug-In ausgewählt wurde. Bezogen auf unser obiges Beispiel des Star Rating Plug-Ins wäre das zum Beispiel die Anzahl der angezeigten Sterne. Ein Custom Attribute kann folgenden Typ haben
 - Integer
 - Number
 - PL/SQL Code, PL/SQL Expression, PL/SQL Function Body
 - Page Item, Page Items
 - Page Number
 - SQL Query
 - Select List
 - Text (übersetzbar)
 - Textarea (übersetzbar) oder
 - Yes/No

Abhängig vom Typ wird auch für Plug-Ins im Application Builder automatisch überprüft ob es sich um eine gültige SQL Query, ... handelt. Weiters erfolgt auch eine nahtlose Integration mit dem Advisor.

8. **Files:** Um die Installation eines Plug-Ins zu vereinfachen ist es wichtig, dass es nicht notwendig ist irgendwelche Dateien auf einen Web Server zu kopiert. Darum bieten Plug-Ins die Möglichkeit alle relevanten Dateien mit der Plug-In Definition in der Datenbank abzuspeichern. Wenn Sie ein Plug-In entwickeln rate ich aber dazu die Dateien vorerst auf einen Web Server zu legen und erst die finale Version in der Plug-In Definition zu speichern.

Packages welchen Ihnen das Leben erleichtern

Speziell für die Plug-In Entwicklung wurden mehrere Utility Packages erstellt, um die Plug-In Entwicklung möglichst einfach und sicher zu machen.

- **APEX_PLUGIN:** Enthält alle Record Typ Definitionen, welche Sie für die Callback Interface Erstellung benötigen.
- **APEX_PLUGIN_UTIL:** Deckt fast alle Bereiche der Plug-In Entwicklung ab. Damit ist es zum Beispiel möglich Dynamic SQL sicher und möglichst einfach auszuführen.
- **APEX_JAVASCRIPT:** Enthält die notwendigen Funktionen um JavaScript Code sicher in die HTML Seite einzufügen.
- **APEX_CSS:** Müssen CSS Dateien oder Inline CSS eingebunden werden, dann enthält dieses Package alle notwendigen Funktionen um dies möglichst einfach zu machen.

Checkliste für sichere und performante Plug-Ins

1. Testen Sie Ihr Plug-In in dem Sie es mehrmals auf der Seite verwenden
2. Lagern Sie Ihren JavaScript- und CSS Code in externe Dateien aus
3. Testen Sie im Read Only und Printer Friendly Modus
4. Sicherheit! Auf SQL- und XSS Injections prüfen. Validieren Sie die eingegebenen Daten!
5. AJAX Aufrufe: Keine internen Daten (z.B. Tabellename) für den Aufruf in die Seite einfügen!
6. Hilfetexte für das Plug-In und die Attribute eingeben
7. Funktioniert es mit anderen Sprachen?
8. Barrierefreiheit: Funktioniert es auch mit einem Screen Reader?
9. Lizenzbedingungen festlegen

Kontaktadresse:

Patrick Wolf

Oracle Austria GmbH
Wagramerstraße 17-19
A-1223 Wien

Telefon: +43 (0) 1-33 777-0
E-Mail patrick.wolf@oracle.com
Blog: www.inside-oracle-apex.com
Twitter: www.twitter.com/patrickwolf