Das APEX QS-Plugin

Oliver Lemm
MT AG
Ratingen

Schlüsselworte
APEX, Qualitätssicherung, Plugin

Einleitung
Mit dem Vortrag wird ein Ansatz vorgestellt, mit welchem man innerhalb der Entwicklung direkt den Entwickler unterstützt und auf Fehler hinweist. Die Logik wird dabei innerhalb eines Plugin gekapselt um bei jedem Projekt schnell und einfach die verschiedensten Prüfungen durchführen zu können.

Die Anforderungen
Um die Wiederverwendbarkeit zu gewährleisten galt als primäre Prämisse, die Logik und die nötigen Daten so zu entwerfen, dass möglichst große Flexibilität und kaum Einschränkungen vorhanden sind. Trotz der Flexibilität darf die Funktionalität nicht eingeschränkt werden. Es sollen dabei jegliche Daten und Metadaten des jeweiligen Schemas ausgewertet werden können. Kann eine Prüfung nicht durchgeführt werden, so soll dies an den Benutzer zurückgemeldet werden, ohne dass die anderen Prüfungen davon beeinflusst werden.
Weiter soll es dem Benutzer beim Darstellen von Fehlern ermöglicht werden, direkt an die Stelle im Application Builder zu springen, wo das Problem behoben werden kann.
Innerhalb einer Menge von Prüfungen, soll weiter definiert werden können, falls Abhängigkeiten bestehen, oder ob eine Prüfung aktiv ist oder nicht.

Das Konzept
Die Definition aller Prüfungen, welche definiert werden können, sollen als SQL-Anfrage hinterlegt werden.
Die einzelnen Prüfungen besitzen folgende Eigenschaften:

- Name: Name der Prüfung
- Category: APEX, DDL, DATA
- Objecttype:
  - bei APEX => ITEM,REGION (oder andere APEX Elemente, siehe Repository Views)
  - bei DDL => TABLE, PACKAGE, TRIGGER (oder weitere Datenbankobjekte)
  - bei DATA => TABLE, REF_TABLE
- Message: Fehlermeldung, welche ausgegeben werden soll.
- Comment: Kommentar zur Prüfung.
- Exclude: Objekte als Doppelpunkt getrennte Liste, welche von der Prüfung ausgeschlossen werden.
- Errortlevel: 1 (ERROR), 2(WARNING), 4 (INFO)
- Active: 1 (YES) / 0 (NO)
- SQL: Abfrage, die alle Datensätze ermittelt welche fehlerhaft sind. Die Abfrage muss zwingend folgende Eigenschaften enthalten
  - Name: Name des Objekts was geprüft wird
  - Value: Wert, welcher bei Fehlschlag mit ausgegeben wird.

    - ID (optional): Falls das Objekt über einen eindeutigen Schlüssel identifizierbar ist diesen mit ausgeben.
    - Errortlevel (optional): Kann das Errorlevel auf Prüfungsebene überschreiben.
    - Objecttype (optional): Kann den Objecttype auf Prüfungsebene überschreiben.
    - Message (optional): Kann die Message auf Prüfungsebene überschreiben.
    - change_user (optional): Ausgabe des Entwicklers, welcher die letzte Änderung durchgeführt hat
    - change_date (optional): Ausgabe des Datums der letzten Änderung
- Predecessor: Falls eine Prüfung abhängig von einer weiteren Prüfung ist, so wird dies in einer Doppelpunkt separierten Liste hinterlegt.
- Layer: 1 (PAGE), 2 (APPLICATION), 3(DATABASE)

Die obigen Eigenschaften, werden dabei in einer Tabelle hinterlegt.
Die komplette Verarbeitungslogik wird dabei in einem Package realisiert, welches die SQL Anfrage einer Prüfung dynamisch ausführt (über execute immediate).
Die einzelnen Prüfungen werden nur durchgeführt wenn:

1. Layer <= dem eingestellten Layer (im Plugin)
2. das Errorlevel <= dem eingestellten Errorlevel (im Plugin)
3. Active = 1
4. die abhängigen Prüfungen/Vorgänger (PREDECESSOR) liefern keine Fehler zurück

Das Ergebnis dieser Prüfungen soll dabei dem Entwickler dargestellt werden können, als auch für weitere Abfragen in einer Collection zur Verfügung stehen.

Die Umsetzung
Da es innerhalb APEX mehrere verschiedene Typen von Plugins gibt, die für diese Funktionalität in Frage kommen kann es je nach Einsatzzweck entweder als Region Plugin oder als Process Plugin sinnvoll sein.
Will man das Ergebnis der Prüfungen entweder selbst verarbeiten oder es per eigenen Report ausgeben, so würde ein Process Plugin sinnvoll sein.
Will man hingegen ohne weiteren Aufwand in einer Region sehen welche Fehler existieren, so ist ein Region Plugin die besser Auswahl.
Da vor allem der schnelle Einsatz und die effektive Nutzbarkeit im Focus diese Plugins steht, ist das Region-Plugin zu favorisieren.

In beiden Fällen sind die Eigenschaften beim Hinzufügen des Plugins zu einer Anwendung folgendermaßen:

- Errorlevel
  Hier wird bestimmt, welche Fehler basierend auf dem Errorlevel ausgegeben werden sollen. Wählt der Benutzer ERROR aus, so werden keine Prüfungen mit dem Level INFO oder WARNING angezeigt
- Active
  Hier werden die Namen aller vorhandenen Prüfungen angezeigt. Es können dabei über ein Shuttle Validierungen aktiviert und deaktiviert werden.
- Layer
  Hierdurch wird angegeben welche Ebenen (Datenbank, APEX oder APEX Seite) überprüft werden sollen. So kann das Plugin gezielt nur eines oder mehrerer dieser Layer gleichzeitig prüfen.

- User
  Sollen die Benutzer, welcher die letzte Änderung durchgeführt werden mit angezeigt werden bei APEX Komponenten?
- Date
  Soll das Datum der letzten Änderung angezeigt werden?

Insgesamt kann so recht fein justiert werden, welche Prüfungen wann ausgeführt werden sollen.

Im Einsatz
Will man das Plugin effektiv einsetzen, so empfiehlt sich die Global Page einer APEX Anwendung. Die jeweilige Region oder das Process Plugin kann dann entweder über eine Build Option, oder per Condition verwendet werden. Die folgende PLSQL Expression (Condition) bietet sich an, da sie nur dann die Region anzeigt, wenn man als Entwickler aktiv ist.
APEX_APPLICATION.G_EDIT_COOKIE_SESSION_ID IS NOT NULL

Da innerhalb der Packagelogik optional die Daten in eine Collection geschrieben werden können besteht zusätzlich die Möglichkeit, diese per Mail zu versenden. Dabei kann man die Daten zum Beispiel per Job an eine Menge von Emailadressen auf Basis der Collection, oder mit dem Inhalt, welcher in die Region des Plugins geschrieben werden nutzen.

Neben dem Einsatz in der Seite, oder zum Mailen kann man recht einfach auch die Daten über ein Logging in eine eigene LOG-Tabelle schreiben.

Der Ausblick
In der Zukunft könnte man Export und Import einbauen um nur die Regeln einfach zu transportieren. Optional kann für die Zukunft geprüft werden, ob eine Einbindung des APEX Advisors möglich ist, um neben eigens definierten Prüfungen auf die gesamte Prüfmenge zugegriffen werden kann.
Neben der reinen Ausgabe der Fehler können diese auch aggregiert und über Grafiken dargestellt werden. Ob dies innerhalb des Plugins oder separat sinnvoll ist lässt sich zum jetzigen Zeitpunkt schwer einschätzen, aber birgt Potential für die Zukunft.

Fazit
Schneller und einfacher können keine Prüfungen innerhalb der Entwicklung erfolgen. Auch die Portierbarkeit als Plugin und die Flexibilität durch die Nutzung von SQL-Abfragen stellt die sinnvolle Nutzung des Plugins als besonders hilfreich dar.
Auch die Erweiterbarkeit zum späteren Zeitpunkt und die vielfältigen Filtermöglichkeiten helfen genau die Fehler und Informationen auszugeben um frühzeitig Korrekturen durchzuführen. Da APEX und die zugehörigen Anwendungskomponenten in einer Datenbank liegen kann der Großteil der Vorgaben innerhalb APEX und in der Datenbank geprüft werden.

Kontaktadresse:
Oliver Lemm
MT AG
Balcke-Dürr-Allee 9
D-40882 Ratingen

Telefon:	+49 (0) 2102 309 61 - 164
Fax:	+49 (0) 2102 309 61 - 10
E-Mail	oliver.lemm@mt-ag.com
Internet:	www. mt-ag.com
Twitter:	https://twitter.com/OliverLemm
Blog:	http://oliverlemm.blogspot.de