

# Die APEX APIs Schatzkiste

**November 2015**

**DOAG 2015**

**Ulrike Brenner**

- Seit 1999 Oracle Entwicklerin
- Oracle DB 8-12c
- SQL  
PL/SQL  
Oracle Forms/Reports  
APEX
- Entwicklerin  
Teamleiterin  
Kundenbetreuerin



# API

Application Programming Interface

Schnittstelle zur  
Anwendungsprogrammierung

# APEX-APIs

Oracle Application Express API Reference is intended for application developers who are building database-centric web applications using Oracle Application Express.

# APEX-APIs

Oracle Application Express API Reference is intended for application developers who are building database-centric web applications using Oracle Application Express.

# APEX-APIs

- PL/SQL - Packages
  - direkt in APEX in einem PL/SQL-Block
  - in einem Package in der DB gekapselt

# APEX-APIs

- in APEX in Verwendung
- getestet
- supported

# APEX-APIs

- APEX 5 (jede neue APEX-Version)
- New Features
- Deprecated and Desupported Features



# APEX-APIs

- APEX 5
  - 28 APEX\_Packages  
ca. 500 Prozeduren/Funktionen
  - JavaScript APIs
- Application Express API Reference:  
[https://docs.oracle.com/cd/E59726\\_01/doc.50/e39149/toc.htm](https://docs.oracle.com/cd/E59726_01/doc.50/e39149/toc.htm)

# Collections in der DB

- **Associative Arrays (index-by tables)**
- **Nested Tables**
- **Varrays**

# APEX\_COLLECTION

- temporäre Speicherung (innerhalb einer Session) von
  - Zeilen
  - Spalten
- insert / update / delete

# APEX\_COLLECTION

- Abbildung eines Workflows  
z.B.: Upload einer csv-Datei
  - ) Laden der Datei
  - ) Verbindung Spalten csv-&Tabellen
  - ) Analyse der Daten (insert/update/delete)
- Performance  
z.B.: mehrfach verwendete zeitaufwändige  
Selects

# APEX\_COLLECTION

- Name d. Collection: max 255 Zeichen (wird immer in Großbuchstaben umgewandelt)
- SEQ\_ID = eindeutige ID einer row pro Collection
- C001 -> C050: Character - Spalten
- N1->N5 / D1->D5: Number / Date
- CLOB001, BLOB001, XMLTYPE001

# APEX\_COLLECTION

- CREATE\_COLLECTION
- CREATE\_OR\_TRUNCATE\_COLLECTION
- CREATE\_COLLECTION\_FROM\_QUERY
- CREATE\_COLLECTION\_FROM\_QUERY2
- CREATE\_COLLECTION\_FROM\_QUERY\_B
- CREATE\_COLLECTION\_FROM\_QUERYB2

# APEX\_COLLECTION

APEX\_COLLECTION.

CREATE\_COLLECTION\_FROM\_QUERY

```
( p_collection_name =>
    'MY_COLLECTION'
, p_query            =>
    'SELECT DEPTNO
      , DNAME
      FROM DEPT'
);
```

# APEX\_COLLECTION

View: APEX\_COLLECTIONS (Plural!)

```
SELECT SEQ_ID
       , C001 AS DEPTNO
       , C002 AS DNAME
FROM   APEX_COLLECTIONS
WHERE  COLLECTION_NAME =
        'MY_COLLECTION'
;
```



# APEX\_COLLECTION

- Add / Update Member
- Resequence Collection / Sort Members
- Truncate Collection
- Delete | / all Collections
- ...

# APEX\_COLLECTION

- Packaged Applications
  - Sample Collections

# APEX\_IR

## Interactive Report

- Query, die der Enduser ausführt inklusive aller Filter
  - Interactive Report als Filter
  - Performanceanalyse

# APEX\_IR

- Query, die der Enduser ausführt
- Zurücksetzen von Report Settings
- Löschen von Saved Reports
- Ändern von Subscriptions Email/Language
- Delete Report (saved IR)
- Change Report Owner (saved IR)

# APEX\_IR

Empfohlen nur im Submit-Prozess  
(Report Download Issue)

- Filter hinzufügen
- Clear Report
- Reset Report

# APEX\_IR

```
vIRReport APEX_IR.t_report;
```

```
vIRReport := APEX_IR.Get_Report  
            ( p_page_id    IN NUMBER  
              , p_region_id IN NUMBER  
            );
```

tatsächliche Query:

```
vIRReport.sql_query
```

angewandte Filter:

```
vIRReport.binds(i).name
```

```
vIRReport.binds(i).value
```

# APEX\_IR

# Interactive Report

- Packaged Applications
  - Sample Reporting
  - page 11

# APEX\_ZIP

- neu mit APEX 5
- zip/unzip von Dateien



# APEX\_ZIP

- **ADD\_FILE**
  - hängt einzelne BLOBs in eine ZIP-Datei
- **FINISH**
  - nach der letzten einzufügenden Datei aufzurufen um die ZIP-Datei fertig zu erstellen

# APEX\_ZIP

- GET\_FILES
  - Liefert in einem Array die Dateinamen einer ZIP-Datei (BLOBs)
  - inklusive Pfad
- GET\_FILE\_CONTENT
  - Retourniert die einzelne Datei aus der ZIP-Datei

# APEX\_DEBUG

Erweiterung des Debug-Message-Logs  
in PL/SQL Code

- Vereinfachung der Fehlersuche
- seit APEX 4.2: **APEX\_DEBUG\_MESSAGE**  
-> **APEX\_DEBUG**

# APEX\_DEBUG

- Error (Level 1)  
wird immer geschrieben
- Message (Def. Level 4)  
nur wenn Debug enabled ist
  - Substitution Strings (%s)

# APEX\_DEBUG

```
APEX_DEBUG.ERROR  
  ( p_message IN VARCHAR2  
    , p0      IN VARCHAR2  
                                     DEFAULT NULL  
    , p1      IN VARCHAR2  
                                     DEFAULT NULL  
    . . . . .  
    , p9      IN VARCHAR2  
                                     DEFAULT NULL  
    , p_max_length IN PLS_INTEGER  
                                     DEFAULT 1000  
  ) ;
```

# APEX\_DEBUG

- Selection der generierten Texte/Messages/  
Fehler
- via GUI im Entwicklermodus
- View:APEX\_DEBUG\_MESSAGES

# APEX\_DEBUG

- Enable (Level 1-9) / Disable
- Remove debug by age / by app

# APEX\_ITEM

- APEX-Felder dynamisch erstellen
  - Checkbox, Radiogroup
  - Date-Popup
  - Select-List
  - Textfelder, Textareas
  - Hidden Items



# APEX\_JSON

- JavaScript Object Notation
- Textbasiertes Datenformat zum Datenaustausch und Speichern von Daten

# APEX\_JSON

- neu mit APEX 5
- kann aber auch direkt in der DB verwendet werden
- auch ohne APEX nutzbar

# natives JSON

- natives JSON ab DB-Version 12.1.0.2
- JSON\_VALUE, JSON\_QUERY, JSON\_TABLE
- kein natives Erzeugen von JSON

<http://sql-plsql-de.blogspot.co.at/2015/05/apexjson-ohne-apex-json-erzeugen-mit.html>

# APEX\_JSON

- Parsen
- Auslesen
- Generieren
  
- XML <-> JSON

# Installation einer APEX- Applikation

- Import der Applikation über GUI
- Ausführen eines SQL-Skripts über ein command-line Entwicklertools (z.B: SQL-Plus)

# Installation einer APEX- Applikation

- selber Workspace
- selbe Workspace-ID
- @f1234.sql

# APEX\_APPLICATION INSTALL

- Änderung von Attributen einer Applikation bei der Installation, z.B.:
  - Workspace ID
  - Application ID
  - Offset
  - parsing Schema ...

# APEX\_APPLICATION INSTALL

- Import einer Applikation in einen anderen Workspace mit einem anderen parsing Schema



# APEX\_APPLICATION\_

# INSTALL

- Applikation:  
1234
- neuer Workspace:  
MY\_WS\_PROD
- neues parsing Schema:  
MY\_PROD\_SCHEMA

# APEX\_APPLICATION\_

# INSTALL

```
SELECT WORKSPACE_ID  
      INTO vWorkspaceID  
      FROM APEX_WORKSPACES  
      WHERE WORKSPACE = 'MY_WS_PROD'  
;
```

```
APEX_APPLICATION_INSTALL.  
      SET_WORKSPACE_ID(vWorkspaceID) ;
```

# APEX\_APPLICATION\_

# INSTALL

```
APEX_APPLICATION_INSTALL.  
  GENERATE_OFFSET;
```

```
APEX_APPLICATION_INSTALL.  
  SET_SCHEMA ( 'MY_PROD_SCHEMA' );
```

```
@f1234.sql
```

# APEX\_APPLICATION\_

# INSTALL

Automatische Installation von Supporting  
Objects:

```
APEX_APPLICATION_INSTALL.  
SET_AUTO_INSTALL_SUP_OBJ  
  
( p_auto_install_sup_obj => true );
```

# APEX-APIs

- Stöbern Sie in der Dokumentation!
- [https://docs.oracle.com/cd/E59726\\_01/doc.50/e39149/toc.htm](https://docs.oracle.com/cd/E59726_01/doc.50/e39149/toc.htm)



# Ulrike Brenner

click-click IT Solutions

Aspettenstraße 48  
A-2380 Perchtoldsdorf

[ulrike.brenner@click-click.at](mailto:ulrike.brenner@click-click.at)  
<http://www.click-click.at>