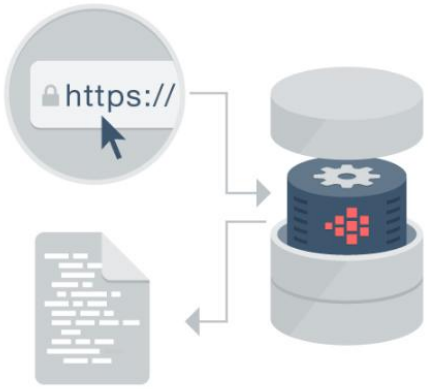
# Mehr als "*Report on Table*":
# Application Express visualisiert Daten aller Art!

*Autor: Carsten Czarski, ORACLE Deutschland B.V. & Co KG*




*In nahezu allen Geschäftsanwendungen werden Daten aus relationalen Tabellen - als Tabelle (Bericht) dargestellt. Schaut man sich ein wenig im Internet oder in mobilen Apps um – so sieht die Welt anders aus: Daten aller Formate, aus allen möglichen Quellen, werden auf verschiedenartigste Art und Weise visualisiert. Und mit zunehmender Bedeutung des Cloud Computing wird sich dieser Effekt mit recht großer Sicherheit noch verstärken: immer öfter müssen heterogene Daten aus unterschiedlichen Quellen verarbeitet werden.*

*Obwohl Oracle Application Express in der (relationalen) Datenbank läuft, lässt es sich sehr gut nutzen, um heterogene Daten ansprechend und modern zu präsentieren. Formate wie JSON und XML lassen sich mit SQL problemlos verarbeiten - und APEX kann weit mehr als nur Reports und Charts. Der Vortrag zeigt, wie JSON, XML und Tabellendaten kombiniert und mit APEX als Baum, Diagramm, Kalender, Karte oder als D3-Chart visualisiert werden.*

## *Nutzung externer Daten in Application Express*

In einer Application Express-Anwendung kann der Entwickler verschiedenste Komponenten verwenden – tabellarische Berichte, Diagramme, Bäume (Tree) oder Kalender werden aus dem Stand geliefert. Schaut man sich dazu noch in den Packaged Applications um, so finden sich zahlreiche Plug-Ins für weitere Komponententypen wie Karten, zusätzliche Diagramme und vieles mehr.

Nahezu alle APEX Komponenten haben eine wichtige Gemeinsamkeit: Die Datenquelle ist stets eine SQL-Abfrage – was logisch ist, denn APEX läuft in der Oracle-Datenbank. Solange die Daten also in normalen, relationalen Tabellen vorliegen, ist die Nutzung einfach. Mit zunehmender Bedeutung des Cloud Computing hat man es jedoch mehr und mehr mit heterogenen Datenformaten und

unterschiedlichsten Datenquellen im Internet zu tun. Sehr oft werden Daten dann über das HTTP-Protokoll im JSON oder XML-Format als RESTful Service bereitgestellt. Als Beispiel soll der Earthquake-Feed des USGS (United States Geological Survey) dienen: Die folgende URL (die man auch im Browser testen kann) liefert eine Liste aller Erdbeben mit einer Magnitude von 4,5 und mehr zurück – im JSON-Format.

```
http://earthquake.usgs.gov/earthquakes/feed/v1.0/
summary/4.5_day.geojson
```


```
{
    type: "FeatureCollection",
  + metadata: {…},
  - features: [
      - {
            type: "Feature",
          - properties: {
                mag: 4.6,
                place: "116km ESE of Phek, India",
                time: 1474550623730,
                updated: 1474551553040,
                tz: 390,
                url: "http://earthquake.usgs.gov/earthquakes/eventpage/us10006rn5",
                detail: "http://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/us10006rn5.geojson",
                felt: null,
                cdi: null,
                mmi: null,
                alert: null,
                status: "reviewed",
                tsunami: 0,
                sig: 326,
                net: "us",
                code: "10006rn5",
                ids: ",us10006rn5,",
                sources: ",us,",
                types: ",cap,geoserve,origin,phase-data,",
                nst: null,
                dmin: 1.556,
                rms: 0.8,
                gap: 56,
                magType: "mb",
                type: "earthquake",
                title: "M 4.6 - 116km ESE of Phek, India"
            },
          - geometry: {
                type: "Point",
              + coordinates: […]
            },
            id: "us10006rn5"
        },
      + {…},
      + {…},
```


*Abbildung 1: Antwort des Earthquake Feed des United States Geological Survey*

Zunächst stellt sich die Frage, wie man diese Daten abrufen kann – Application Express bietet für solche Webservices eine Unterstützung an. Zum einen können in einer APEX-Anwendung Web Service Referenzen erzeugt werden, zum anderen kann man mit PL/SQL und dem Paket APEX_WEB_SERVICE arbeiten. Der Earthquake Feed ist ein sehr einfacher REST-Service.

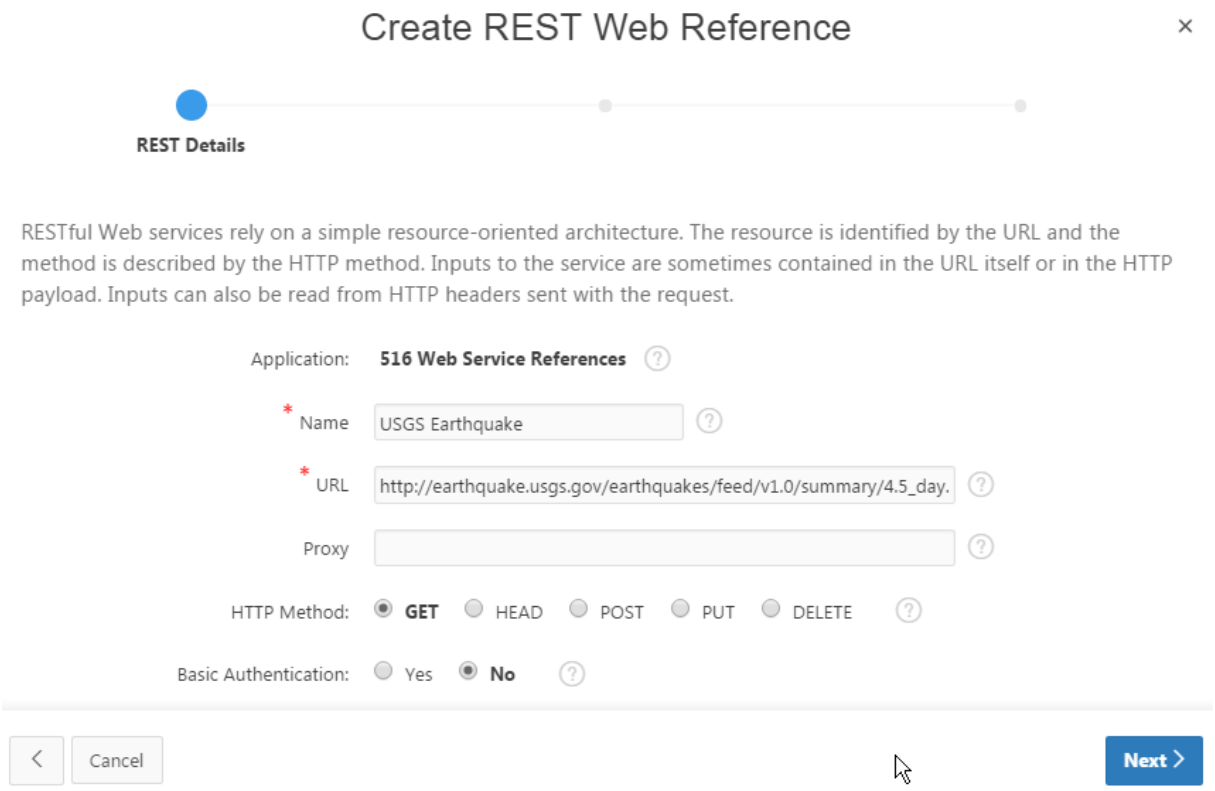



*Abbildung 2: Erzeugen einer Web Service Reference für den Earthquake-Feed*

Wenn die Webservice-Referenz erzeugt wurde, kann auf einer APEX-Seite ein Prozess angelegt werden, welcher die Antwort, also das JSON, als CLOB in eine Collection lädt. Alternativ kann man auch direkt mit PL/SQL arbeiten (Abbildung 3).

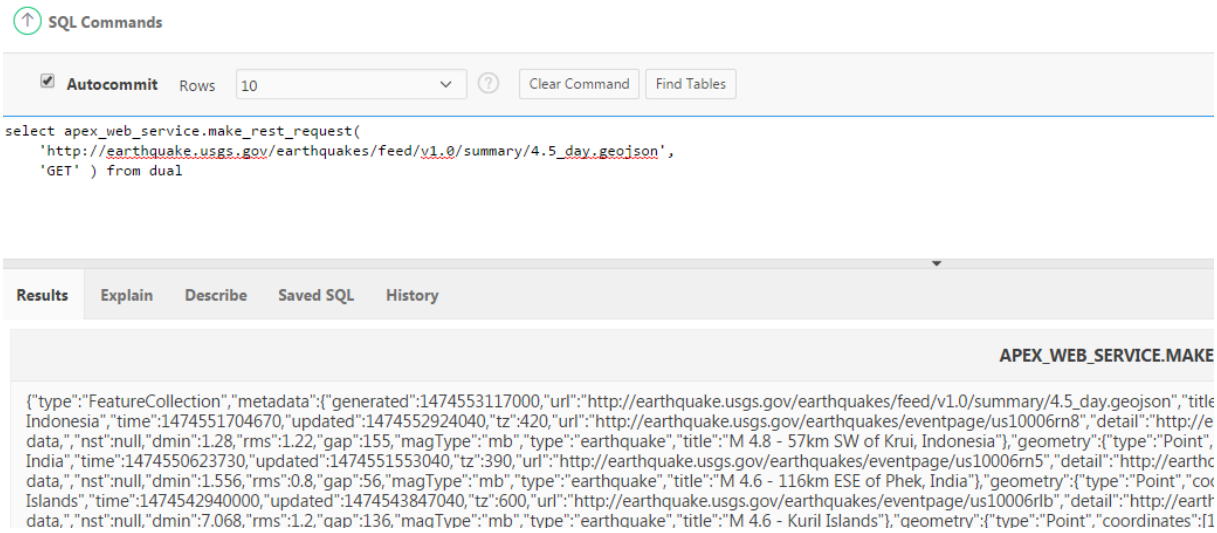



*Abbildung 3: Abrufen der externen Daten mit dem PL/SQL Paket APEX_WEB_SERVICE*

Das reine JSON-Format ist allerdings nur wenig nützlich, wenn es darum geht, die Daten für den Endanwender mit APEX-Komponenten zu visualisieren. Hier braucht es weitere Datenbankfunktionalität: Das JSON-Dokument muss geparst und in die einzelnen Datenfelder zerlegt werden. Arbeitet man mit der Datenbankversion 12.1.0.2 oder höher, so leistet die SQL-Funktion **JSON_TABLE** nützliche Dienste.

```
select
    TYPE,
    TITLE,
    MAG,
    LON,
    LAT
from json_table(
    apex_web_service.make_rest_request(
        p_url => 'http://earthquake.usgs.gov/earthquakes/feed/' ||
                 'v1.0/summary/all_day.geojson',
        p_http_method  => 'GET') format json,
    '$.features[*]'
    columns (
        TYPE    VARCHAR2(20)    path '$.properties.type',
        TITLE   VARCHAR2(100)   path '$.properties.title',
        MAG     NUMBER          path '$.properties.mag',
        LON     NUMBER          path '$.geometry.coordinates[0]',
        LAT     NUMBER          path '$.geometry.coordinates[1]'
    )
)
```

*Listing 1: Zerlegen eines JSON-Feeds in Oracle12c mit der SQL-Funktion JSON_TABLE*

Doch auch mit einer 11g-Datenbank kommt man weiter: Zwar steht JSON_TABLE hier nicht bereit; das PL/SQL-Paket APEX_JSON erlaubt es jedoch, das JSON-Dokument nach XML zu konvertieren. Darauf kann dann mit der XMLTABLE-Funktion gearbeitet werden.

```
select
    TYPE,
    TITLE,
    MAG,
    LON,
    LAT
from xmltable(
    '/json/features/row'
    passing apex_json.to_xmltype( apex_web_service.make_rest_request(
        p_url => 'http://earthquake.usgs.gov/earthquakes/feed/' ||
                 'v1.0/summary/all_day.geojson',
        p_http_method => 'GET'))
    columns
```

```
        TYPE    VARCHAR2(20)    path 'properties/type',
        TITLE   VARCHAR2(100)   path 'properties/title',
        MAG     NUMBER          path 'properties/mag',
        LON     NUMBER          path 'geometry/coordinates/row[1]',
        LAT     NUMBER          path 'geometry/coordinates/row[2]'
)
```

*Listing 2: Zerlegen eines JSON-Feeds in Oracle11g mit APEX_JSON und XMLTABLE*

Lässt man diese Abfragen im SQL Workshop laufen, so bekommt man in etwa das Ergebnis in Abbildung 3 – die Daten lassen sich betrachten, als ob sie aus einer Tabelle kämen. Man kann diese Daten nun in Tabellen oder APEX-Collections laden oder sofort weiterverarbeiten. Soll die APEX-Anwendung letztlich von der Verfügbarkeit des REST-Service unabhängig sein, so empfiehlt sich das Laden der Daten in eine Tabelle oder APEX-Collection. Ob das möglich ist, hängt allerdings auch davon ab, wie zeitnah die APEX-Anwendung die Daten des REST-Service wiedergeben soll.

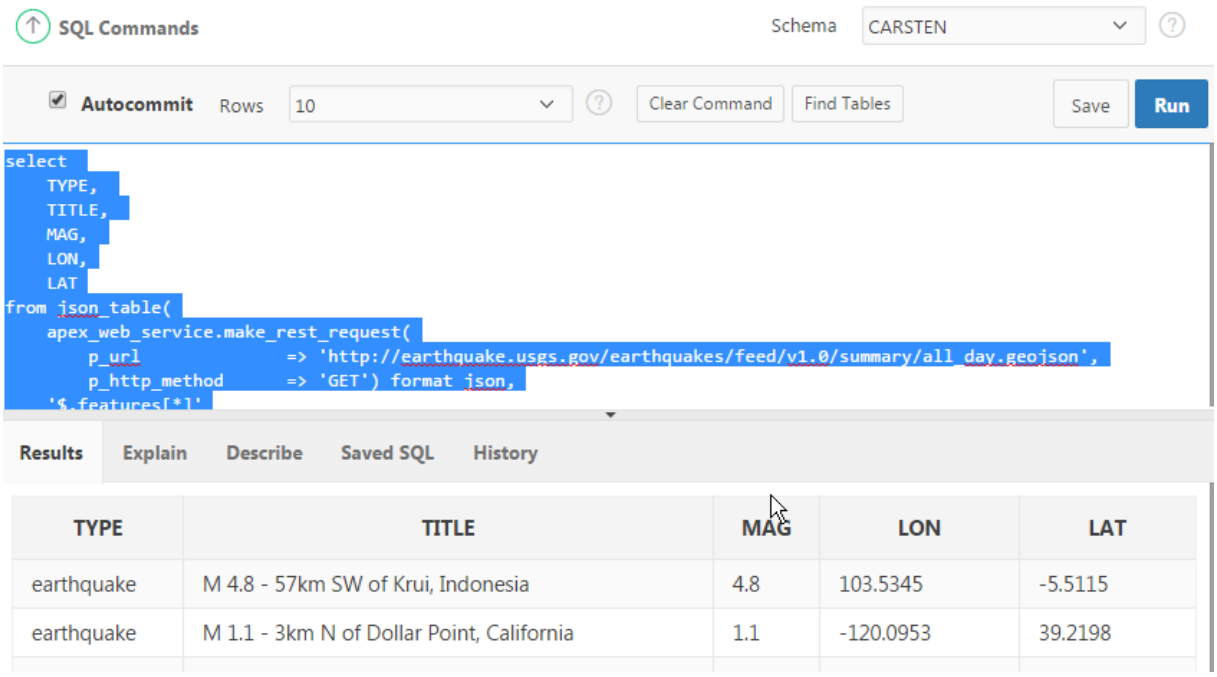
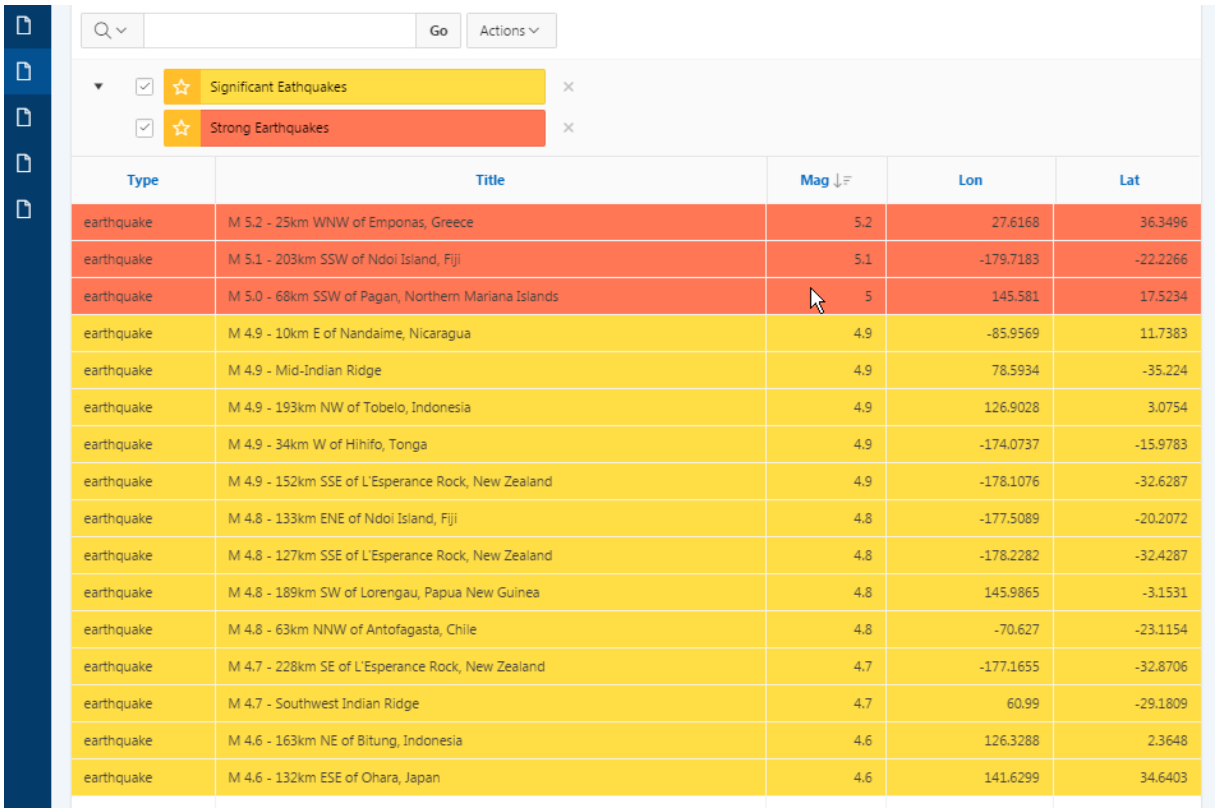



*Abbildung 3:Mit JSON-Daten umgehen wie mit einer Tabelle*

## Visualisierung der Daten mit APEX-Komponenten

Ab diesem Moment stehen alle APEX-Komponenten zur Visualisierung der Daten bereit. Das einfachste ist sicherlich der tabellarische Bericht. Die SQL-Query wird einfach als Regionsquelle eingetragen. Abbildung 4 zeigt, wie ein Application Express *Interactive Report* die Erdbebendaten visualisiert. Natürlich kann auch der Interactive Report oder ein klassischer Bericht verwendet werden.




| Type | Title | Mag | Lon | Lat |
|---|---|---|---|---|
| earthquake | M 5.2 - 25km WNW of Emponas, Greece | 5.2 | 27.6168 | 36.3496 |
| earthquake | M 5.1 - 203km SSW of Ndoi Island, Fiji | 5.1 | -179.7183 | -22.2266 |
| earthquake | M 5.0 - 68km SSW of Pagan, Northern Mariana Islands | 5 | 145.581 | 17.5234 |
| earthquake | M 4.9 - 10km E of Nandaime, Nicaragua | 4.9 | -85.9569 | 11.7383 |
| earthquake | M 4.9 - Mid-Indian Ridge | 4.9 | 78.5934 | -35.224 |
| earthquake | M 4.9 - 193km NW of Tobelo, Indonesia | 4.9 | 126.9028 | 3.0754 |
| earthquake | M 4.9 - 34km W of Hihifo, Tonga | 4.9 | -174.0737 | -15.9783 |
| earthquake | M 4.9 - 152km SSE of L'Esperance Rock, New Zealand | 4.9 | -178.1076 | -32.6287 |
| earthquake | M 4.8 - 133km ENE of Ndoi Island, Fiji | 4.8 | -177.5089 | -20.2072 |
| earthquake | M 4.8 - 127km SSE of L'Esperance Rock, New Zealand | 4.8 | -178.2282 | -32.4287 |
| earthquake | M 4.8 - 189km SW of Lorengau, Papua New Guinea | 4.8 | 145.9865 | -3.1531 |
| earthquake | M 4.8 - 63km NNW of Antofagasta, Chile | 4.8 | -70.627 | -23.1154 |
| earthquake | M 4.7 - 228km SE of L'Esperance Rock, New Zealand | 4.7 | -177.1655 | -32.8706 |
| earthquake | M 4.7 - Southwest Indian Ridge | 4.7 | 60.99 | -29.1809 |
| earthquake | M 4.6 - 163km NE of Bitung, Indonesia | 4.6 | 126.3288 | 2.3648 |
| earthquake | M 4.6 - 132km ESE of Ohara, Japan | 4.6 | 141.6299 | 34.6403 |

*Abbildung 4: JSON-Daten(!) aus dem Internet (!) formatiert mit einem APEX Interactive Report*

Auch Diagramme lassen sich so erzeugen. Aus dem Stand sind bereits viele Diagrammtypen enthalten - wem das nicht reicht, dem sei die Packaged Application **Sample Charts** empfohlen; darin finden sich Plug-Ins für weitere Diagrammtypen, welche in die eigene Anwendung kopiert werden können.

Allen Diagrammen ist gemein, dass eine SQL-Query die Daten liefert – und diese SQL-Query basiert nun auf den JSON-Daten des REST-Service. Für ein Diagramm muss die SQL-Abfrage allerdings leicht geändert werden; Abbildung 5 zeigt, wie die Anzahl der Erdbeben nach Magnitude visualisiert werden kann.

```
select
    null               as link,
    ' <' || ceil(MAG) as label,
    count(*)           as value
from xmltable( ...
```

*Listing 3:SELECT-Liste der SQL Abfrage zur Darstellung der Daten in einem Diagramm*

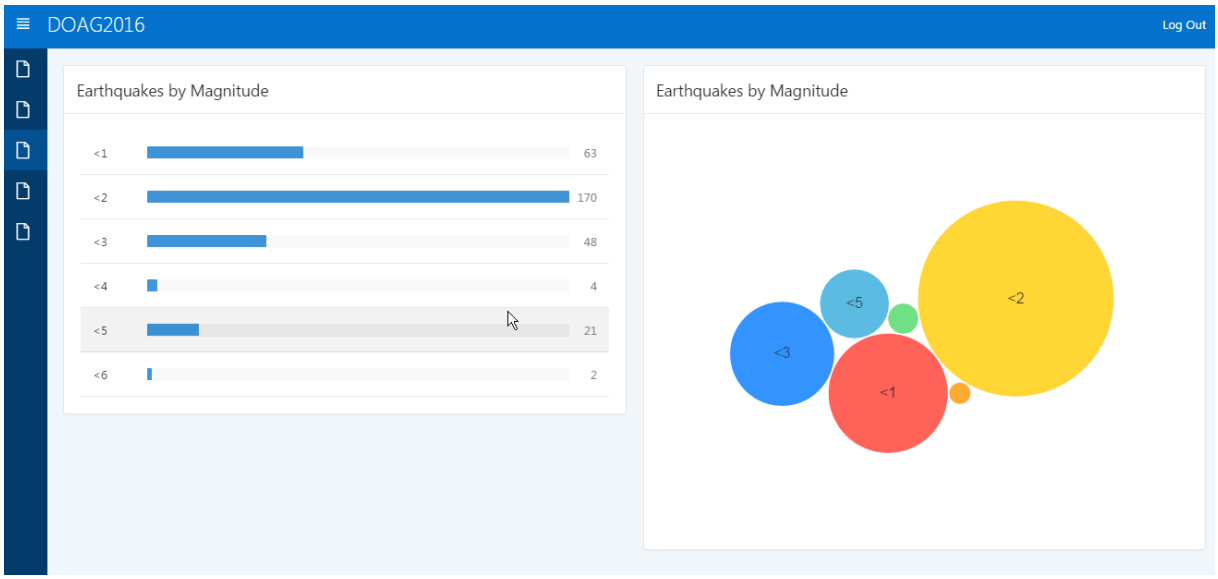



*Abbildung 5: JSON-Daten(!) aus dem Internet (!) formatiert als D3-Diagramm*

Bei Erdbebendaten ist völlig klar, dass eine Koordinate enthalten ist – die Darstellung als Karte drängt sich hier geradezu auf. Zwar bietet APEX aus dem Stand keine Kartenregion an, allerdings gibt es hierfür Plug-Ins in der Packaged Application **Sample Geolocation Showcase**. Nach deren Installation lässt sich das Plug-In **Oracle HTML5 Maps – Region** in die eigene Anwendung kopiert werden. Die Hintergrundkarte kommt vom Plug-In; die auf der Karte darzustellenden Daten wiederum von einer SQL-Abfrage – diese sieht wie folgt aus:

```
Select
    ID,
    TITLE as infotip,
    case
       when mag < 2              then 'blue'
       when mag between 2 and 4  then 'orange'
       else                           'red'
    end as style,
    sdo_geometry(
      2001, 4326, sdo_point_type( LON, LAT, null), null, null
    )   as geometry
from xmltable( ...
```

*Listing 4: SELECT-Liste für die SQL-Datenquelle für die Region vom Typ "Map"*

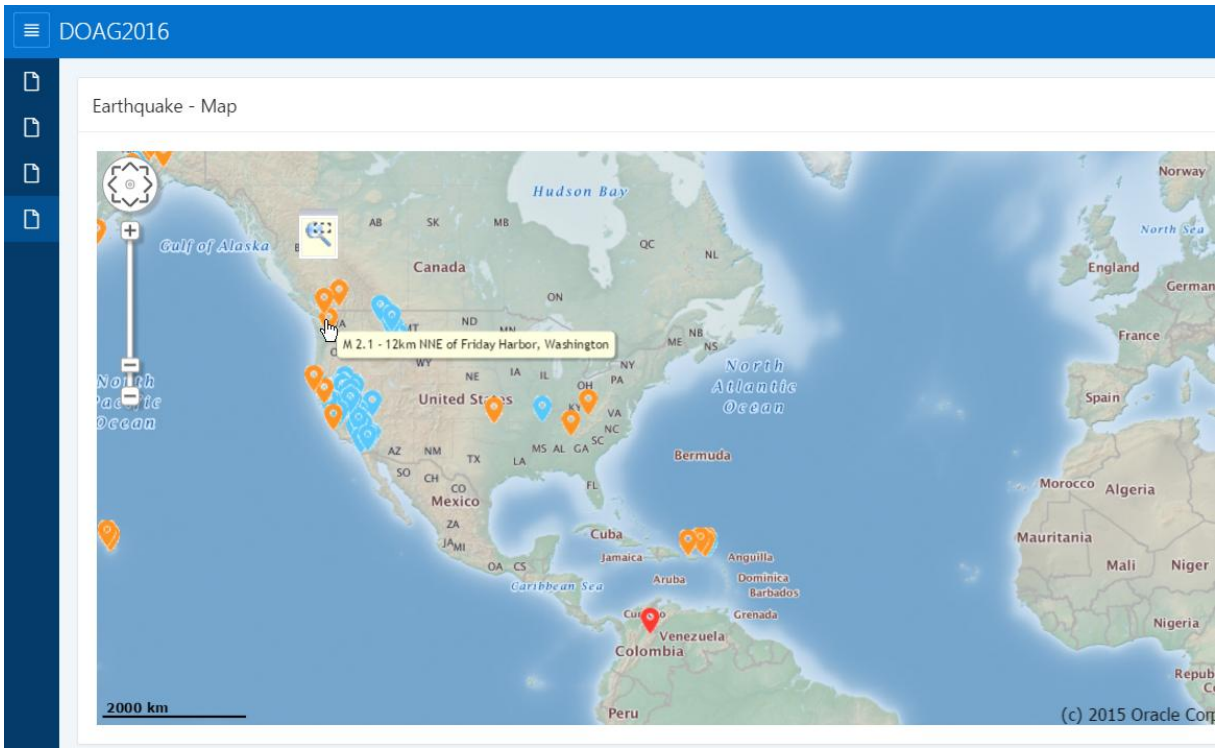



*Abbildung 6: Visualisierung der Erdbebendaten als Karte*

Zum Abschluß sei die Visualisierung als Kalender vorgestellt. Dieser ist in Application Express aus dem Stand enthalten – und wiederum dient eine SQL-Abfrage als Datenquelle. Allerdings enthält der Earthquake-Feed einen Unix-Zeitstempel, also die seit dem 1. Januar 1970 verstrichenen Millisekunden. Dies lässt sich aber leicht in ein Oracle DATE oder TIMESTAMP konvertieren, wie Listing 5 zeigt.

Damit man in der Monatssicht des Kalenders auch etwas sieht, empfiehlt es sich außerdem, einen der JSON-Feeds für die letzten 30 Tage zu nehmen.

```
select
    ID,
    TITLE,
    DATE'1970-01-01' + ( TIME / 86400000 ) as eq_date
from xmltable(
    '/json/features/row'
    passing apex_json.to_xmltype( apex_web_service.make_rest_request(
        p_url => 'http://earthquake.usgs.gov/earthquakes/feed/' ||
                 'v1.0/summary/significant_month.geojson',
        p_http_method => 'GET'))
    columns ...
```

*Listing 5: SQL-Abfrage zur Darstellung in einem Kalender*

Das Ergebnis sieht wie in Abbildung 7 aus.

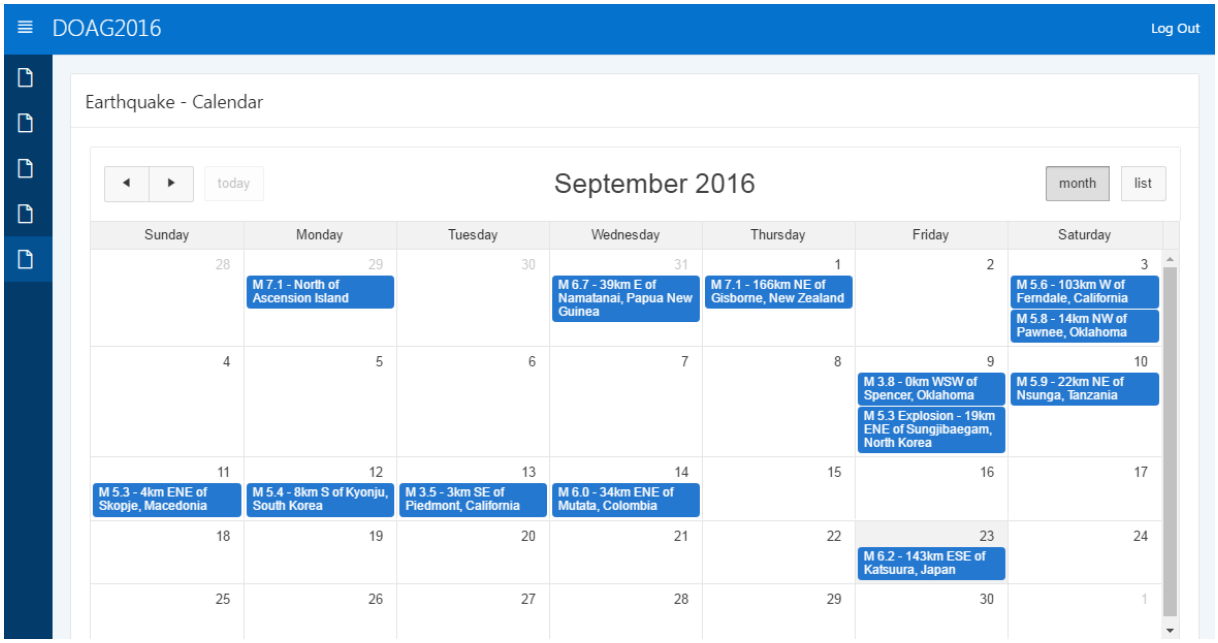



*Abbildung 7: Darstellung in einem APEX-Kalender*

# Fazit

Application Express fokussiert ganz klar auf die Arbeit mit Tabellen, SQL und PL/SQL in der Oracle-Datenbank. Das bedeutet jedoch nicht, dass andere Datenformate und -Quellen sich mit APEX nicht nutzen lassen. Sobald Daten mit SQL oder PL/SQL verarbeitet werden können, stehen sie auch dem APEX-Entwickler zur Verfügung.

Der Oracle-Datenbankkern wird seit Jahren ständig erweitert und modernen Anforderungen angepasst – das war schon 2002 mit der XML DB erkennbar. Oracle12c bringt native Unterstützung für das JSON-Formate mit. Geht es um das Ansprechen von Datenquellen im Internet (REST Services), so stehen dem Entwickler mit UTL_HTTP oder dem Paket APEX_WEB_SERVICE ebenfalls alle Möglickeiten offen. Kombiniert man das alles, so lassen sich externe Daten ebenso verarbeiten wie lokale Tabellendaten. Und die APEX-Anwendung bringt alles zum Endanwender.

## Weitere Informationen

- Application Express 5.1 Early Adopter
  http://apexea.oracle.com
- Deutschsprachige APEX und PL/SQL Community
  http://blogs.oracle.com/apexcommunity_deutsch
- Application Express im Oracle Technet
  http://otn.oracle.com/apex

## Kontakt

Carsten Czarski

Carsten.Czarski@oracle.com
http://twitter.com/cczarski
http://sql-plsql-de.blogspot.com