

Red Stack

Magazin

Apex

Schnell und einfach ans Ziel



Topaktuell

Oracle Database
Standard Edition 2

Im Interview

Martin Winkler,
Managing Director,
Oracle Austria

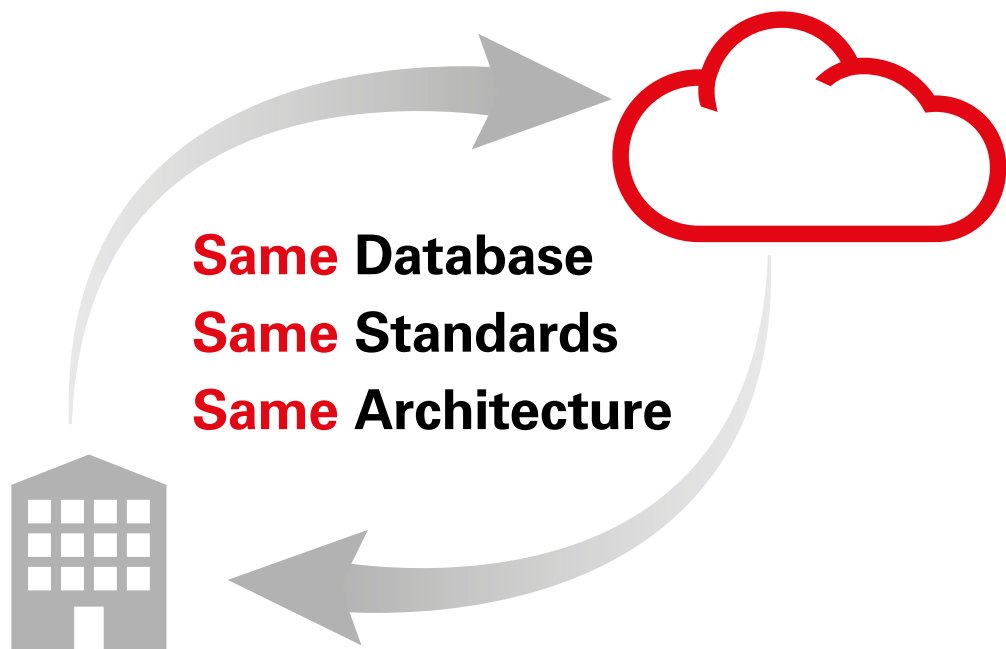


Datenbank

Heatmap und ADO
in der Praxis

Push a Button

Move Your Database to the Oracle Cloud



... or Back to Your Data Center

ORACLE®

cloud.oracle.com/database
or call 1.800.ORACLE.1



Niels De Bruijn
Verantwortlich für den
Themenbereich Apex

Liebe Mitglieder, liebe Leserinnen und Leser,

warum ist Oracle Application Express (Apex) eigentlich so beliebt? Aus meiner Sicht gibt es dazu im Wesentlichen zwei Gründe: Zum einen ist Apex Metadaten-getrieben. Web-Applikationen werden deklarativ über den Browser definiert und können darüber hinaus programmatisch beliebig erweitert werden. Diese Produktivität und zugleich Flexibilität erhöht nicht nur den Spaß bei den Entwicklern, auch die Endanwender freuen sich beispielsweise über die kürzeren Entwicklungszyklen.

Als zweiten Grund sehe ich die großartige Community, die sich über alle Kanäle austauscht. Online gibt es beispielsweise den Twitter-Hashtag „#orclapex“, die neue Community-Seite „apex.world“ und den Blog-Aggregator auf „odtug.com/apex“. Fragen sind dagegen beim Forum gut aufgehoben (siehe „forum.oracleapex.com“) oder werden auf Slack diskutiert (siehe „orclapex.slack.com“).

Neben den vielen Online-Angeboten darf der persönliche Austausch natürlich nicht fehlen. Jedes Jahr bieten die vielen Regionaltreffen, Expertenseminare und Konferenzen eine gute Gelegenheit dafür. Die wichtigste Konferenz im deutschsprachigen Raum ist Apex Connect. Für die zweite Auflage trifft sich die Community in diesem Jahr vom 26. bis 28. April in Berlin. Apex Connect bietet alles für den datenbanknahen Entwickler von Web-Applikationen – von SQL und PL/SQL über Apex bis zu modernen Technologien wie Node.js und Oracle JET. Das Programm hat es in sich: sechzig Fachvorträge, drei Keynotes, sieben Workshops, eine „Lunchtime learn“-Session und dazu jede Menge Gelegenheiten, um sich auszutauschen. Auch die Gewinner der Apex Dashboard Competition werden dort bekanntgegeben (siehe „apexcompetition.org“). Als Hauptredner erwartet uns in diesem Jahr sogar der Gründer von Apex: Mike Hichwa, Vice President Software Development bei Oracle.

Nutzen Sie als Apex-Entwickler die vielen Möglichkeiten, sich auszutauschen, und Sie werden schnell feststellen, dass Apex eine großartige Community hat.

Viel Spaß beim Lesen wünscht Ihnen

Ihr *Niels de Bruijn*

MUNIQSOFT

Schulungen

DOAG-Mitglieder erhalten 15% Rabatt.

Mit den richtigen Kursen holen Sie sich mehr **IQ** ins Haus.

+49 (0) 89 67 90 90-40

www.muniqsoft.de/schulungen

Jetzt rechtzeitig anmelden:

▶ APEX 5.0 Neuerungen	2 Tage	890 Euro	02.06. – 03.06.16
▶ APEX II Kurs	5 Tage	1.990 Euro	06.06. – 10.06.16
▶ APEX Security	2 Tage	1.190 Euro	13.06. – 14.06.16



Preise zzgl. MwSt.



Eine ausgereifte Methodik, um Mehrsprachigkeit für Apex-Anwendungen umzusetzen



Ein Überblick über die Debug-Funktionalität in Apex und ein Einstieg in die Fehlersuche



Features, die das Entwickeln von Applikationen erleichtern und verbessern können

Einleitung

- 3 Editorial
- 5 Timeline
- 8 „Auf lange Sicht hat das Cloud-Computing Auswirkungen auf unsere Organisation ...“
Interview mit Dipl.-Ing. Martin Winkler,
Managing Director von Oracle Austria

Apex

- 11 Application Express 5.1 – was ist neu?
Carsten Czarski
- 15 „Wir setzen ganz stark auf das Feedback unserer engagierten Community ...“
Fünf Fragen an Patrick Wolf, Principal Member of Technical Staff Oracle Application Express
- 16 Mehrsprachigkeit in Apex-Anwendungen
Stefan Röß
- 21 Cloud-Lösungen für Apex
Till Albert und Niels de Bruijn
- 25 Datenbank-Services für Entwickler aus der Oracle Cloud: Database Service, Schema Service & mehr
Carsten Czarski
- 30 Der Apex Session State
Jürgen Sieben
- 37 Fehlersuche in Apex – ein praktischer Einstieg
Peter Raganitsch
- 43 Wo ist der Fehler? – Debugging in Apex
Christina Funke
- 47 Apex – Tipps und Tricks
Lisa Klimesch
- 50 Apex und Workflows
Sven Böttcher
- 55 Apex Universal Theme – neue Hürden mit dem Responsive Layout
Tobias Arnhold und Richard Rieb
- 59 Apex mobile vs. Universal Theme
Steven Grzbielok

- 64 Apex on the Rocks – Hochverfügbarkeit für eine komplette Apex-Umgebung
Moritz Klein
- 69 Echtzeit-Visualisierung von Twitter & Co.
Kai Donato und Oliver Lemm
- 74 Das Tor zur apex.world
Jürgen Schuster
- 76 Apex als Tool für eine Vereins-Webseite
Rainer Willems
- 79 Vom Rechteck zur Fabrikvisualisierung – Prozessvisualisierung in Apex mit HTML5 und SVG
Frank Schubert und Angela Wobar

Datenbank

- 84 Heatmap und ADO in der Praxis
Florian Feicht

Aktuell

- 46 Starker Dollar setzt Oracle weiter zu
DOAG Online
- 68 Lizenzierung Forms 12c: Oracle bittet Stand-Alone-Kunden zur Kasse
DOAG Online
- 73 Schließung des Oracle-Software-Produkt-Supports in Deutschland steht unmittelbar bevor
DOAG Online
- 88 Oracle Database Standard Edition 2
Johannes Ahrends

Intern

- 89 Termine
- 90 Neue Mitglieder
- 90 Impressum
- 90 Inserenten

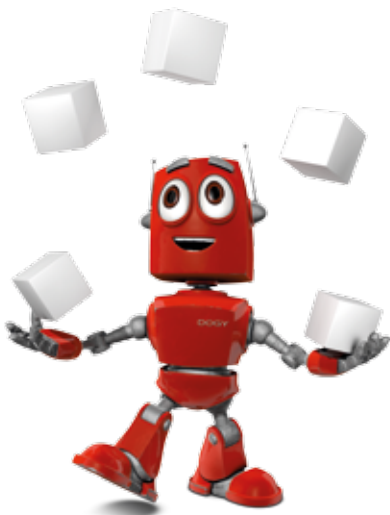
◆ Timeline

13. Januar 2016

Die IT-Dienstleister der DOAG kommen in der DOAG-Geschäftsstelle zu einem IT-Workshop zusammen. Es geht um die Planung neuer Webservices im Rahmen der Umstellung aller Seiten auf das Responsive Design der DOAG-Webseiten.

19. Januar 2016

Jörg Heckel, Designer und Zeichner der Dogy-Filme, stellt den Mitarbeiterinnen und Mitarbeitern der DOAG-Geschäftsstelle sein neues Werk vor, den Zeichentrickfilm „Duke im JavaLand“. Er wird seine Premiere auf der JavaLand 2016 haben (siehe „<http://tinyurl.com/zh-klv3v>“). Außerdem auf der Agenda: neue Episoden für den Dogy.



Dogy in Aktion

20. Januar 2016

Die Generalversammlung der AOUG in Unterpremstätten bei Graz entlastet den Vorstand für das vergangene Vereinsjahr und wählt ihn für das neue. Zudem werden die weiteren Schritte für die kommende AOUG-Anwenderkonferenz am 21. Juni 2016 in Wien beschlossen.

21./22. Januar 2016

In Würzburg findet die zweite Noon2Noon-Veranstaltung der DOAG statt, diesmal zum Thema „RAC & Dataguard“. Von 12 bis 12 Uhr, mittags bis mittags, sammeln vierzig Datenbank-Administratoren Praxiserfahrung. Bei den interaktiven Sessions heißt es: Ärmel hochkrepeln und mitmachen. Nach einer kurzen Einführung zum Thema „RAC“ beginnen die Teilnehmer im Rahmen eines RAC-Attack-Workshops, je zwei virtuelle Maschinen von Grund auf in einen RAC-Cluster zu verwandeln. Nur vom Abendessen unterbrochen, wird bis

weit nach Mitternacht installiert und konfiguriert. Natürlich werden im Anschluss daran die Failover-Fähigkeiten der funktionierenden Cluster überprüft. Am zweiten Tag steht Dataguard auf dem Programm. Das Angebot, auch nach dem offiziellen Ende am Mittag unter Anleitung weiterzuprobieren, wird vielfach wahrgenommen. Am Ende der Veranstaltung hat jeder Teilnehmer eine lauffähige RAC- und Dataguard-Installation auf dem eigenen Rechner.



Hochkonzentriert bei der Arbeit: die Teilnehmer der DOAG Noon2Noon

27. Januar 2016

Das Organisationskomitee trifft die letzten Vorbereitungen zur JavaLand 2016. Angesichts der hohen Anmeldezahlen zeichnet sich ein neuer Besucherrekord ab.

3. Februar 2016

Fried Saacke, DOAG-Vorstand und Geschäftsführer, und Carmen Al-Youssef, Leiterin des Bereichs „Veranstaltungen“ in der DOAG-Geschäftsstelle, besprechen sich mit dem Oracle-Deutschland-Marketing. Beide Seiten sind sich darüber einig, den Oracle Cloud Day wieder in die DOAG 2016 Konferenz + Ausstellung zu integrieren. Die Details werden später ausgehandelt, wenn auch der DOAG-Vorstand grünes Licht gegeben hat.

4. Februar 2016

Das Organisationsteam der diesjährigen DOAG Konferenz + Ausstellung 2016 bespricht bei einem Treffen mit Vertretern des NürnbergConvention Center organisatorische Verbesserungen für die größte DOAG-Veranstaltung, die in diesem Jahr vom 15. bis zum 18. November wieder in Nürnberg stattfinden wird.



Die DOAG 2016 Konferenz + Ausstellung verspricht wieder ein großes Programm

18. Februar 2016

Dr. Clemens Appl von der Wirtschaftsuniversität Wien präsentiert die Ergebnisse der Studie zur Rechtssituation der Oracle-Lizenzierung in VMware-Umgebungen im Rahmen des AOUG-Regionalforums in Innsbruck. Mag. Martin Löhmer vom ARZ in Tirol ergänzt das Thema mit seinem Vortrag zur Problematik der Lizenzierung in virtuellen Umgebungen und zu Strategien und Herangehensweisen im Falle eines Lizenzaudits. Am Nachmittag diskutiert Dr. Thomas Petrik von Sphinx mit den Teilnehmern über das Oracle-Container-Konzept.

23./24. Februar 2016

Zur dritten Auflage des DOAG DevCamp kommen rund 80 Teilnehmer ins Kameha Grand Hotel in Bonn und diskutieren in guter Atmosphäre aktuelle Trends und Best Practices rund um die Oracle-Entwicklerthemen. Zuvor liefern Keynote-Speaker zusätzliche Impulse für den Tag. Andreas Koop spricht über die Cloud als Treiber der digitalen Transformation. Zentrale Positionen: Die Sicherheitsaspekte relativieren sich, Basis-Technologien gewinnen an Bedeutung, die Zukunft ist Plug-in-frei. Am zweiten Tag geht Sascha Wolter auf Chancen und Herausforderungen für das Internet of Things ein. Die eigentlichen Sessions decken die gesamte Palette der Oracle-Entwicklerthemen ab: Forms, PL/SQL, Apex, ADF, SOA/BPM, JET – und natürlich sind Mobile und Cloud besonders heiß diskutierte Punkte. Außerdem stehen Themen wie „Teambuilding“, „Work/Life-Balance“ und „Kundenumgang“ auf der Agenda. In einem zusätzlichen Raum werden neue Tools vorgestellt – im Anschluss daran haben die Teilnehmer die Möglichkeit, diese in praktischen Hands-on-Sessions selbst auszuprobieren. Außerdem ist das ADF Fitness Center – bisher eine separate Oracle-Veranstaltung – in das DevCamp integriert. Networking ist dank des offenen Formats durchgehend möglich – entweder in den Sessions oder spontan auf dem Gang oder im Foyer. Ein zusätzliches Schmankerl: Am Abend des ersten Konferenztages sind die Teilnehmer zum Hütengaudi mit Bier und bayerischen Spezialitäten eingeladen.



Spannende Diskussionen und gute Laune

29. Februar 2016

In einem Expertenseminar zum Thema „Tuning im Data Warehouse: ein Survival Kit“ in der DOAG Konferenz Lounge versetzt Lothar Flatz die Teilnehmer in die Lage, die Performance einer Abfrage selbstständig zu beurteilen, Engpässe zu erkennen und zu sehen, wo sie gegebenenfalls Unterstützung brauchen. Die relativ kleine Gruppe arbeitet sehr intensiv. Am ersten Tag werden

die theoretischen Grundlagen so weit besprochen, dass die Aufgaben des zweiten Tages lösbar sind.

2. März 2016

In Baden-Dättwil findet die diesjährige Generalversammlung der SOUG statt. Präsident Peter Gübeli begrüßt etwa zwanzig Teilnehmer, die der Versammlung interessiert folgen und bei verschiedenen Punkten mit kritischen Fragen und Anmerkungen den erfolgreichen Weg der SOUG unterstützen. Ein Haupt-Tagesordnungspunkt sind die anstehenden Vorstandswahlen, die Mitglieder bestätigen alle bisherigen Vorstandsmitglieder einstimmig in ihren Ämtern. Als neues Vorstandsmitglied wird Michael Krebs, Managing Director esentri swiss AG, gewählt. Er soll sich im Vorstand um die Themen „Marketing“, „Mitgliedergewinnung“ und – ganz besonders wichtig – um die Außendarstellung der SOUG kümmern. Dabei wird er insbesondere die neue Webseite und das Branding voranbringen.

2. März 2016

Als erster großer SOUG-Event des Jahres 2016 findet in Baden-Dättwil bei der ABB im Segelhof der SOUG-Day statt. Den fast 70 Teilnehmern werden dabei drei parallele Vortrags-Streams angeboten. Im Stream I steht das Thema „Performance“ im Vordergrund; drei bekannte Oracle-Performance-Spezialisten, Lothar Flatz, Frank Pachot und Chris Antognini, teilen mit den Teilnehmern ihre Erfahrungen. Im Stream II beschäftigen sich die Vorträge mit der Infrastruktur. Hier geht es um die ODA, die neue Sparc-Architektur und darum, ob das FlashGrid der Exadata-Killer sei. Im Stream III kommen allgemeine Datenbank-Themen zur Sprache, etwa die Möglichkeiten, die Docker im Oracle-Datenbank-Umfeld bietet, was man alles beim Upgrade auf 12c beachten sollte und wie man seine On-Premise- und Cloud-Datenbanken mit dem Oracle Enterprise Manager 13c managen kann. Beim anschließenden Apéro, gesponsert von den Firmen Trivadis und dbi services, lassen die Teilnehmer die Fülle der ganzen Informationen setzen und tauschen sich untereinander und mit den Referenten aus.

7. März 2016

Am Vortrag der JavaLand 2016 läuft die JavaLand4Kids. Rund zwanzig Grundschüler im Alter von neun und zehn Jahren nutzen bei der zweiten Auflage der Veranstaltung im Phantasialand



Kinder spielerisch an das Programmieren heranführen und sie fit für den Umgang mit Computern und neuen Technologien machen

in Brühl bei Köln die Gelegenheit, viel Neues zu erforschen und auszuprobieren. Der kleine „Nao Roboter“ sorgt bei den Kindern schon gleich zu Beginn für viel Begeisterung, als er sich persönlich vorstellt. Mit seinen vielen Sensoren am Körper und einem eingebauten Sonar bietet der menschlich wirkende Roboter viele Möglichkeiten der visuellen Programmierung. Die Kinder lernen schnell, die Zusammenhänge zwischen den einzelnen Aktionen des Roboters und der zugrunde liegenden Programmierung zu verstehen, und sind schon bald in der Lage, mit dem Roboter zu interagieren und verschiedene Choreografie-Übungen zu absolvieren. Im Workshop „Jumping Sumo“ bringen die Kinder der Drohne eine Reihe von Kunststücken bei. Neben 80 Zentimeter hohen Sprüngen programmieren sie mithilfe eines Routenplans eine Abfolge von Aktionen und können so die Drohne gezielt steuern. Als inzwischen fester Bestandteil der JavaLand-Konferenz soll die Veranstaltung auch im nächsten Jahr wieder stattfinden.

8. März 2016

Zur dritten Auflage der Java-Konferenz kommen mehr als 1.200 Teilnehmer, die sich im Phantasieland Brühl bei Köln über die neuesten Trends im Java-Umfeld informieren und austauschen. Die JavaLand 2016 ist damit eine der größten Java-Konferenzen in Europa. Zum Startschuss ruft die Java-Community mit einem lauten „Jatumba“ zu einer erneuten Besiedelung des JavaLand auf. Auf dem umfangreichen Konferenzprogramm stehen mehr als 100 Vorträge, acht Streams und eine Vielzahl von Community-Aktivitäten. Top-Speaker sind die in der Java-Szene sehr umtriebigen Bruno Borges, David Blevins, Kirk Pepperdine, Hendrik Ebbers, Stephen Chin und Holly Cummins. Das Programm deckt die sieben Themenbereiche „Core Java und JVM basierte Sprachen“, „Architektur & Security“, „Enterprise Java & Cloud“, „IDEs & Tools“, „Frontend & Mobile“, „Internet der Dinge“ und „Container und Microservices“ ab. Für frischen Wind sorgt der neu geschaffene Newcomer-Stream: Zehn Entwickler, die bisher noch nie auf einer Konferenzbühne standen, feiern ihr Debüt und geben erstmals ihr Know-how zum Besten – betreut von namhaften Mentoren der Java-Szene wie Charles Nutter, Roland Huß und Anton Arhipov. Fünfzehn Community-Aktivitäten, zusammen mit zahlreichen Java User Groups gestaltet, bieten viel Spaß zum Mitmachen und Ausprobieren, ob in den Workshops „Architektur Kata“, „Coding Dojo“ oder „Code Shrink“, beim „Java Innovation Lab“ oder in der „Early Adopters' Area“ – es ist für jeden etwas dabei. Das gegenseitige Lernen voneinander und der Austausch stehen dabei an erster Stelle. Viele Gelegenheiten zum Netzwerken bietet neben der begleitenden Ausstellung mit rund 30 Ausstellern insbesondere die traditionelle Abendveranstaltung am Dienstagabend.



Die Community in Aktion

9. März 2016

Die Teilnehmer der JavaLand 2016 staunen nicht schlecht, als ihnen Pepper über den Weg läuft, ein humanoider Roboter, der in der Lage ist, Mimik und Gestik von Menschen zu analysieren und darauf entsprechend zu reagieren. Der von dem französischen Unternehmen Aldebaran Robotics SAS und dem japanischen Telekommunikations- und Medienkonzern SoftBank Mobile Corp. gemeinsam entwickelte Roboter ist in Japan bereits in verschiedenen Handelsketten in Aktion anzutreffen.

10. März 2016

Acht Workshops zu den unterschiedlichsten Themen für Anfänger, Fortgeschrittene und Experten bieten am Folgetag der JavaLand 2016 vertieftes Wissen an. Obwohl die Zahl der Schulungen im Vergleich zum Vorjahr verdoppelt wurde, kam die Kapazität an ihre Grenzen.

10. März 2016

Im Nachgang zur JavaLand 2016 stellt die Mitgliederversammlung des Interessenverbands der Java User Groups e.V. (iJUG) bereits die Weichen für die JavaLand 2017. Ein weiteres wichtiges Thema ist die Sorge der Community um den Fortbestand von Java FX. Hier wird von Oracle ein klares Signal erwartet.



Die iJUG-Mitglieder diskutieren ihre Themen

11. März 2016

Die Vorstandssitzung der DOAG bereitet das Leitungskräfte-Forum und die Delegiertenversammlung vor; die beiden wichtigsten Gremien der DOAG tagen Ende April in Berlin. Mit Unterstützung des DOAG-Syndikus Carsten J. Diercks gilt es, wichtige Anträge zu formulieren, darunter die Neugründung einer Community für die junge Generation und deren Probleme sowie die Optimierung der Satzung. Ein weiteres wichtiges Thema betrifft die Auseinandersetzungen mit Oracle zu den Lizenzbedingungen in virtuellen Umgebungen. Der DOAG-Vorstand hat hier nach wie vor große Bedenken.

16. März 2016

Dr. Dietmar Neugebauer, Vorstandsvorsitzender der DOAG, und sein Stellvertreter Stefan Kinnen treffen sich in Frankfurt mit Frank Obermeier, Country Leader der Oracle Deutschland BV. & Co. KG. Im Mittelpunkt des Gesprächs stehen die Lizenzbedingungen in virtuellen Umgebungen, die Cloud-Strategie von Oracle und deren Auswirkungen auf die Mitarbeiter in Deutschland sowie die Vorbereitung der DOAG Konferenz + Ausstellung 2016.



Martin Winkler (rechts) im Gespräch mit Klaus-Michael Hatzinger

„Auf lange Sicht hat das Cloud-Computing Auswirkungen auf unsere Organisation ...“

Der österreichische Oracle-Markt ist erfolgreich. Ing. Klaus-Michael Hatzinger, Vorstandsvorsitzender der Austrian Oracle User Group (AOUG), und Wolfgang Taschner, Chefredakteur des Red Stack Magazin, sprachen darüber mit Dipl.-Ing. Martin Winkler, Managing Director von Oracle Austria.

Oracle ist seit dreißig Jahren in Österreich vertreten. Was waren die großen Erfolge in dieser Zeit?

Winkler: Wir sind zu einem gut etablierten und respektierten Player auf dem österreichischen Markt geworden und betreuen mittlerweile rund fünftausend Kunden. Etwa neunzig Prozent der Großunternehmen in Österreich setzen unsere Produkte ein und bei vielen Unternehmen stellt Oracle die Basis für die kritischen Infrastrukturen dar.

Sie sind Country Leader von Oracle Österreich. Was sind Ihre persönlichen Ziele in dieser Funktion?

Winkler: An oberster Stelle stehen Kundenzufriedenheit und langfristige Partnerschaft mit den Kunden. Mir ist es wichtig, die Innovationen, die regelmäßig von Oracle kommen, mit meinem

Team für den Kunden nutzbar zu machen. Dies ist in den heutigen Zeiten wichtiger denn je.

Wo steht Oracle Österreich im Kontext mit den anderen europäischen Ländern, wo im internationalen Verbund von Oracle?

Winkler: Bei den Marktanteilen liegen wir im internationalen Vergleich. Organisatorisch sind wir der Region Central & Eastern Europe zugeordnet.

Österreich galt lange als wirtschaftliches Tor zu Osteuropa. Wie ist die Situation heute?

Winkler: Viele unserer Kunden haben in den 1990er Jahren wirtschaftlich stark nach Osteuropa expandiert. Es gibt zudem viele

Unternehmen, die von Wien aus den osteuropäischen Markt betreuen. Mit unserer Organisationsform stellen wir die optimale Betreuung dieser Kunden sicher. Die Dynamik in diesem Markt hat allerdings in den letzten Jahren deutlich abgenommen, was sich aber sicher wieder ändern wird.

Wie schätzen Sie die Chancen der Oracle-Applikationslösungen in Osteuropa ein? Hier ist doch eine starke Lokalisierung erforderlich.

Winkler: Die Anforderung einer Lokalisierung ist vorhanden und wird vor allem durch uns und unsere Partner auch gelöst. Von daher haben wir damit recht gute Chancen im Markt, insbesondere im Hinblick auf unsere Software-as-a-Service- und Cloud-Lösungen.

Mit welchen Geschäftsfeldern sind Sie in Österreich zufrieden, wo könnte es besser laufen?

Winkler: Derzeit geht es durch die Bank weg gut, bei den Business-Applikationen sind wir mit unserem Software-as-a-Service-Angebot sehr gut unterwegs. Gott sei dank hat sich auch die Hardware, allen voran die Engineered Systems, gut etabliert. Rund um die Security würde ich mir mehr Projekte wünschen; unsere Lösungen finden derzeit wenig Resonanz bei den Kunden. Das wird sich hoffentlich bald ändern.

Welche speziellen Entwicklungen sehen Sie rund um die IT-Bedürfnisse der österreichischen Unternehmen?

Winkler: Der österreichische Markt ist für mich nicht speziell. Wir haben die gleichen Herausforderungen wie in den anderen Ländern, derzeit vor allem die Transformationen in die Cloud-Computing-Modelle. Österreich ist sicher ein eher konservativer Markt, der nicht alle Innovationen umgehend annimmt und in dem die Kunden vor einer Entscheidung gut und genau analysieren. Im Endeffekt folgt Österreich allerdings dem internationalen Trend.

Oracle fokussiert jetzt stärker auf die Cloud. Hat das Veränderungen in der Organisation von Oracle Österreich zur Folge?

Winkler: Auf lange Sicht hat das Cloud-Computing ganz bestimmte Auswirkungen auf unsere Organisation. Momentan gehen wir noch mit den gleichen Strukturen und Teams auf den Kunden zu. Unsere Strategie ist es, den Kunden nicht in eine Zwangsposition zu bringen, sondern er kann bei uns weiterhin auf die klassischen Lizenzmodelle setzen oder mit dem gleichen Leistungsumfang in die Cloud migrieren. Für die Zukunft investiert Oracle jedoch international wie auch hier in Österreich, um auf diesen Trend gut vorbereitet zu sein.

Wie schätzen Sie die Auswirkungen der Änderungen hinsichtlich der Oracle Standard Edition 2 ein?

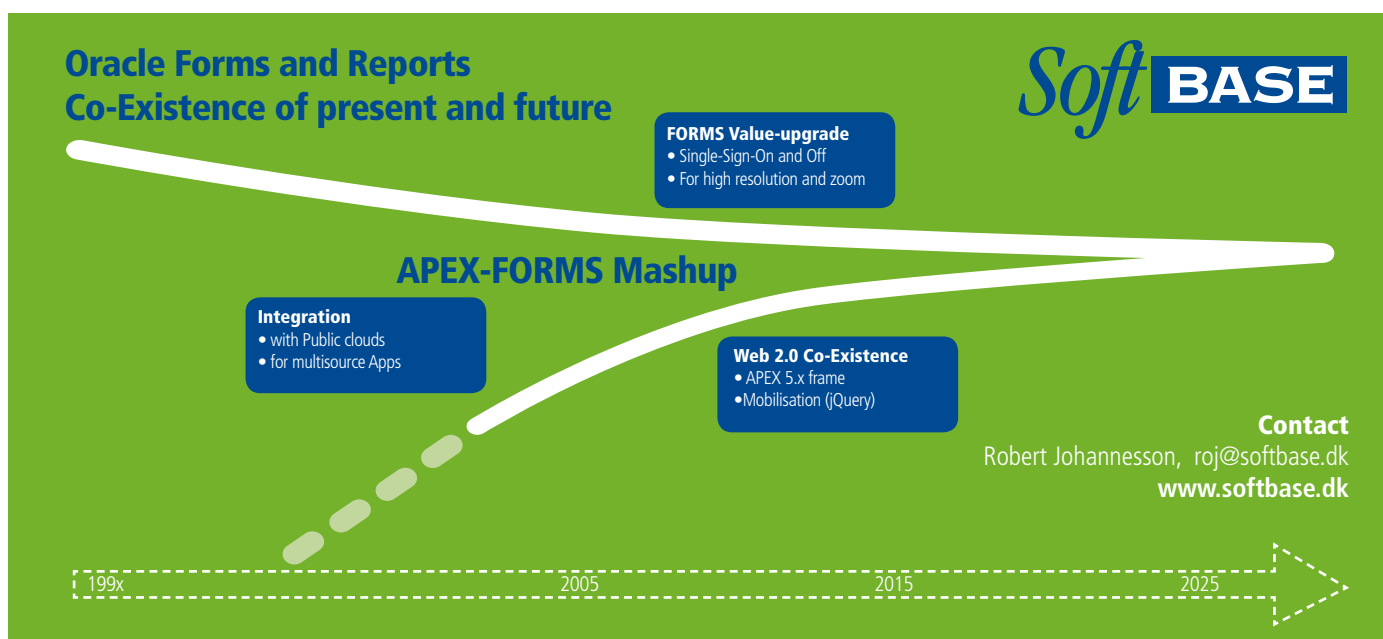
Winkler: Die Standard Edition 2 ist eine sehr gute Weiterentwicklung der Technologie mit einigen neuen und innovativen Funktionen. Ich glaube auch, dass wir einen guten Pfad für die Migration gefunden haben.

Es gab bei der Ankündigung der Oracle Standard Edition 2 Unstimmigkeiten bei den Kunden. Sind diese inzwischen ausgeräumt?

Winkler: Mit der Standard Edition 1 steht das Einstiegsmodell nicht mehr zur Verfügung. Trotzdem ist für mich auch die Standard Edition 2 eine sehr attraktive Einstiegsbasis mit allen Möglichkeiten, die die Oracle-Datenbank hinsichtlich Stabilität und Skalierbarkeit bietet.

Wie stehen Sie zur Lizenzierungs-Problematik mit VMware?

Winkler: Das ist für mich ein wichtiges Thema, wobei ich dazu sagen muss, dass Oracle seine Lizenzierungsregeln in den letzten Jahren sehr konstant und transparent gehalten hat. Darin ist genau festgehalten, wie partitionierte beziehungsweise virtualisierte Umgebungen zu lizenzieren sind. Wo immer Probleme mit VMware auftreten, bitte ich um einen entsprechenden Dialog zur Findung einer praktikablen Lösung für beide Seiten.



Warum findet man bei Oracle keine offizielle Stelle, an der die Lizenzregeln für VMware klar definiert sind?

Winkler: Unser Lizenzierungs-Guide geht nicht auf jedes Virtualisierungsprodukt ein, das auf dem Markt ist. Da diese Themen jedoch auf Corporate-Ebene entschieden werden, bin ich nicht der richtige Ansprechpartner dafür. Wir versuchen, die Situation in Österreich möglichst kundenverträglich zu handhaben.

Sie waren vor Ihrer Zeit bei Oracle bei IBM beschäftigt. Was unterscheidet diese beiden IT-Großunternehmen?

Winkler: Oracle gibt immer klar und offen seine Richtung an. Wir sind ein Technologie-getriebenes Unternehmen, bei dem Innovationen an vorderster Stelle stehen. Das hat bereits mit der Entwicklung und dem Erfolg der Datenbank begonnen. Dieses Credo steht auch noch heute in allen Produkt-Segmenten an oberster Stelle. Wir wollen, dass unsere Produkte besser sind als alles andere, was auf dem Markt verfügbar ist. Das steht nicht bei allen unserer Mitbewerber im Vordergrund. Auch jetzt beim Cloud-Computing können wir sowohl die innovative Technologie als auch das Service-Modell anbieten. Das ist ein unschätzbare Vorteil für uns.

Manche Unternehmen möchten sich nicht von der Hardware bis zur Software in die Hand eines einzigen Anbieters wie Oracle begeben. Wie reagieren Sie darauf?

Winkler: Ich setze hier auf Aufklärung, Offenheit und Transparenz. Der Weg, den wir mit unserem Stack beschreiten, hat nicht das Ziel, einen Kunden-Lock-in zu erzeugen, sondern ihm Nutzen und Vorteile in Form von Automatisierung zu bieten. Seit mehr als einem Jahrzehnt halten wir uns an offene Standards, sodass der Kunde jederzeit einen anderen Weg einschlagen kann.

Vor einigen Jahren standen die Engineered Systems an der Spitze der Technologie bei Oracle. Jetzt ist es die Cloud. Wie soll sich der Kunde hier entscheiden?

Winkler: Am besten nimmt er beides (lacht). Die Dynamik im Markt ist enorm und wir tragen dieser Dynamik Rechnung. Die Engineered Systems sind nach wie vor sehr wichtig für uns, da sie ja auch für die Rechenzentren eine Basis für das Cloud-Computing darstellen.

Wo sehen Sie für Oracle die Vorteile und wo die Nachteile einer starken Anwendervertretung wie der AOUG?

Winkler: Ich sehe hier überhaupt keine Nachteile. Wir haben in Österreich eine sehr gute Zusammenarbeit. Die AOUG bie-



Zur Person: Martin Winkler

Martin Winkler ist seit Juli 2009 Managing Director von Oracle Austria. Der IT- und Vertriebsprofi kam bereits im September 2007 zu Oracle, wo er bisher als Sales Director im Bereich Technology Solutions tätig war.

Seine Karriere begann der Absolvent der Montan-Universität Leoben bei IBM, wo er über einen Zeitraum von fünfzehn Jahren die verschiedensten Positionen innehatte. Nach einem zweijährigen Zwischenstopp bei Xerox Austria, zuletzt als Direktor Xerox Global Services, Austria + Switzerland, erschließt sich für Winkler als Geschäftsführer nun ein umfassendes Tätigkeitsfeld bei Oracle Österreich.

Der zweifache Familienvater ist leidenschaftlicher Tennisspieler.

tet eine hervorragende Plattform für den Erfahrungsaustausch zwischen den Kunden, was wir gerne mit Know-how unterstützen. Wir nehmen auch das Angebot sehr gerne wahr, unsere neuesten Produkte und Technologien auf den Veranstaltungen der AOUG zu präsentieren. Es liegt natürlich in der Natur der Sache, dass wir öfter kontrovers diskutieren. Aber solange wir fair und vertrauensvoll miteinander umgehen, kann dabei nur Gutes herauskommen.

DOAG 2016 Konferenz + Ausstellung: jetzt Vorträge einreichen!

DOAG Online

Der Call for Papers für die DOAG 2016 Konferenz + Ausstellung ist eröffnet. Gesucht werden vor allem Vorträge mit Praxisbezug – also Erfahrungsberichte, Projektberichte, Tipps und Tricks und Best Practices – rund um die Oracle-Produkte.

Jeder kann sich bewerben – Administratoren, Projektleiter, Entwickler und andere

IT-Experten. Die Schwerpunktthemen orientieren sich an der Oracle-Produktpalette sowie an aktuellen Trends und Methoden: Datenbank, MySQL, Oracle und SAP, Development, Java, SOA/BPM, Data Warehouse und BI, Infrastruktur, Middleware, Strategie und Business Practices. Wie im letzten Jahr wird es auch wieder Vorträge für Ein-

steiger sowie zu Softskills-Themen geben.

Die DOAG 2016 Konferenz + Ausstellung findet vom 15. bis 18. November 2016 statt. Vorträge können ab jetzt bis zum 8. Juni eingereicht werden. Weitere Informationen zum Call for Papers und zur DOAG 2016 Konferenz + Ausstellung stehen unter „<http://2016.doag.org/de/referenten>“.

Application Express 5.1 – was ist neu?



Carsten Czarski, ORACLE Deutschland B.V. & Co. KG

APEX 5.1

Im Sommer 2015 wurde das lang erwartete Release Apex 5.0 veröffentlicht. Die zahlreichen neuen Funktionen wie Page Designer und Universal Theme hat die Entwicklergemeinschaft sehr gut aufgenommen. Die Version 5.0 findet sich mittlerweile in den meisten Apex-Installationen. Knapp ein Jahr später steht das nächste Apex-Release vor der Tür: Apex 5.1 ist im Gegensatz zu seinem Vorgänger zwar kein Major-Release, bietet aber nicht weniger interessante Funktionen.

Bei der Auswahl der neuen Funktionen, die mit Apex 5.1 ausgeliefert werden sollen, spielen die Erfahrungen mit Apex 5.0 eine ganz besondere Rolle. Eine der wesentlichen Aufgaben des neuen Release ist es, die mit Apex 5.0 eingeführten Neuerungen abzurunden und zu vervollständigen. So wird In-

teractive Grid, das von der Entwicklergemeinschaft bereits für Apex 5.0 erwartet wurde, nun eingeführt. Es soll die Grundlage für modernisierte tabellarische und Master-Detail-Formulare sein.

Dieser Artikel gibt einen Überblick über die zu erwartenden Neuerungen in Apex 5.1. Da die Software noch nicht zur

Verfügung steht, reflektiert er mit seinen Screenshots den aktuellen Entwicklungsstand im Februar 2016. Bis zum Release der Software können sich noch zahlreiche Änderungen ergeben und es gibt keine Gewähr dafür, dass die Funktionen wirklich genau so eingeführt werden, wie hier beschrieben.

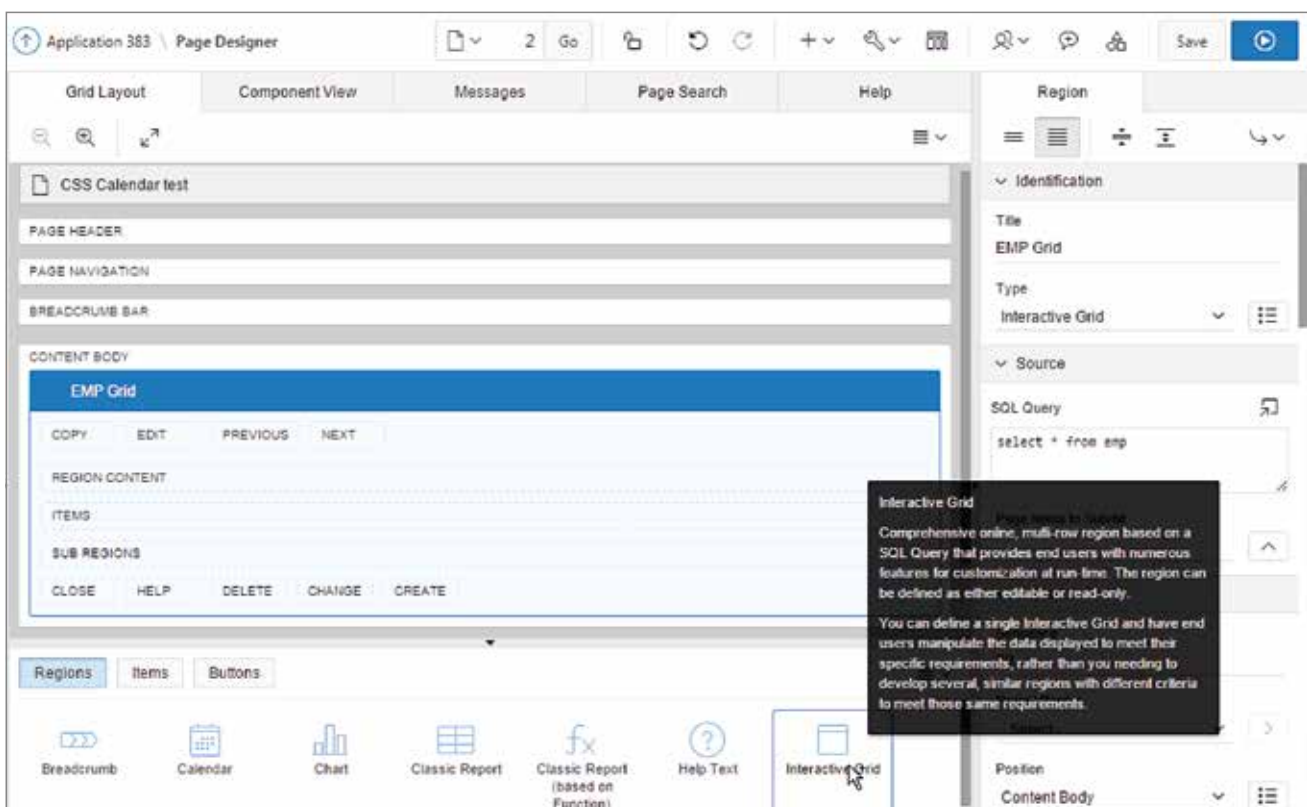


Abbildung 1: Neues interaktives Grid erstellen

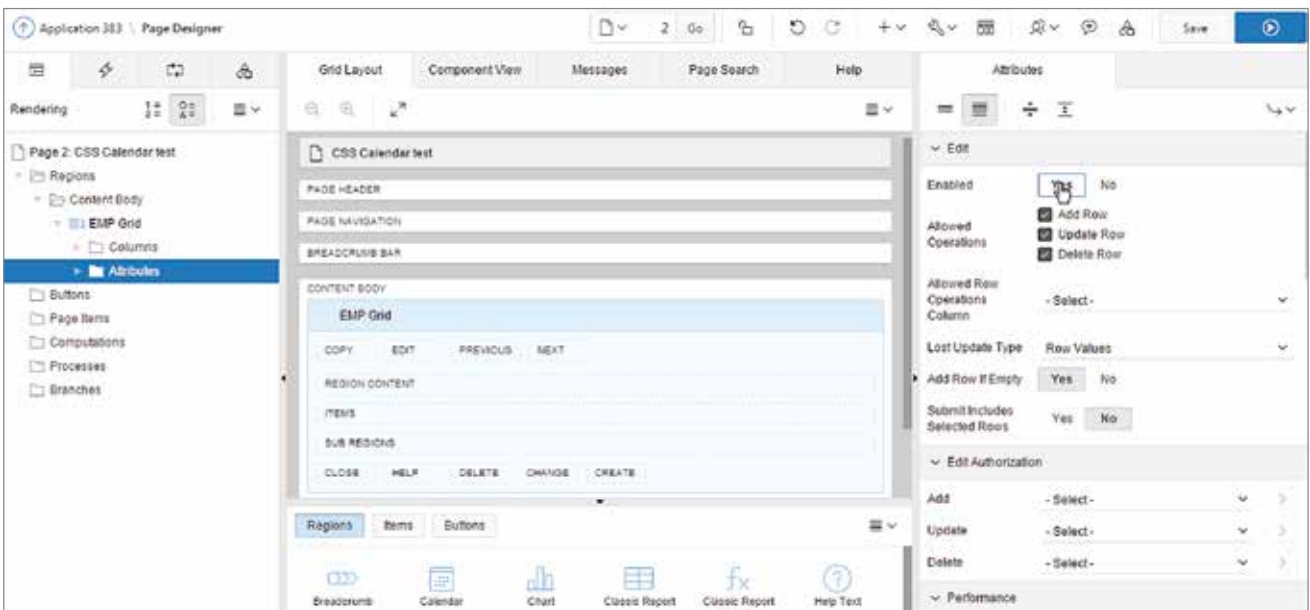


Abbildung 2: Das Interactive Grid wird editierbar gemacht

Interactive Grid

Das Interactive Grid wird sicherlich das mächtigste neue Feature in Apex 5.1 sein – und eines, auf das viele Anwender seit Langem warten. Der Bedarf nach tabellarischen Formularen ist nahezu überall vorhanden. Apex bietet zwar seit der ersten Version tabellarische Formulare an, diese unterliegen jedoch einigen Restriktionen:

- Es können nicht alle Element-Typen eines Apex-Formulars verwendet werden; Item-Plug-ins sind überhaupt nicht möglich.
- Eine Apex-Seite kann nur ein tabellarisches Formular enthalten.

- Es sind nur einfache Master-Detail-Formulare möglich: Master-Detail-Detail-Formulare werden nicht unterstützt.

Insbesondere die Anwender von Oracle Forms, die auf moderne HTML5-Anwendungen mit Apex und Geschäftslogik in der Datenbank setzen wollen, sind von diesen Einschränkungen sehr stark betroffen.

Das neue Interactive Grid hebt diese Beschränkungen auf. Es wird wie ein interaktiver Bericht mit einer SQL-Abfrage als Datenquelle erzeugt (siehe Abbildung 1). Im Gegensatz zu einem normalen interaktiven Bericht werden die Zeilen eines Interactive Grid allerdings editierbar sein (siehe Abbildung 2).

Je nach Einstellung der verschiedenen Attribute können die Tabellenzeilen nun direkt verändert werden (siehe Abbildung 3). Sie lassen sich nicht nur mit der Maus, sondern auch per Tastatur ansteuern. Mit Klick auf eine Schaltfläche werden die Änderungen in der Tabelle gespeichert. Das Interactive Grid speichert gemachte Änderungen ausdrücklich nicht sofort beim Verlassen einer Zelle, sondern alle gemachten Änderungen in einer Transaktion.

Apex 5.1 wird es erlauben, mehrere Interactive Grids auf einer Seite zu platzieren – damit werden diese zur Basis für moderne Master-Detail- und Master-Child-Grandchild-Formulare. Auf längere Sicht ist durchaus abzusehen, dass die interaktiven Berichte im Interactive Grid aufgehen werden – schließlich ist ein Interactive Grid mit abgeschalteter Editierfunktion das gleiche wie ein interaktiver Bericht. Alles im allem werden mit dem Interactive Grid und Apex 5.1 nicht nur sehr komplexe Erfassungsmasken möglich; Interactive Grids werden ein sehr zentraler Bestandteil künftiger Apex-Anwendungen werden.

Jet Charts

Während der Oracle OpenWorld 2015 wurde das JavaScript-Framework JET angekündigt. Es ist ein vollständiges Frame-

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	
7566	JONES	MANAGER	7839	02-APR-81	2975	-	
7788	SCOTT	ANALYST	7566	09-DEC-82	3000	-	
7902	FORD	ANALYST	7566	08-DEC-81	3000	-	
7369	SMITH	CLERK	7902	17-DEC-80	800	-	
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	
7876	ADAMS	CLERK	7788	13-SEP-82	1100	-	

Abbildung 3: Editieren des Interactive Grid

work zum Erstellung JavaScript-basierter Anwendungen. Teil von Oracle JET ist eine Diagramm-Engine, die vollständig auf JavaScript und HTML5 basiert. Ab Apex 5.1 kommt JET zum Generieren der Apex-Diagramme zum Einsatz. Es unterstützt vielfältige Diagrammtypen, wobei auch moderne Formen wie Bubble- oder Polar-Charts enthalten sind. JET-Charts sind außerdem „Reponsive“ und „Accessible“, können also auch von Screen-Readern verarbeitet und ohne Maus bedient werden.

Die bisher verwendete Anychart-Engine bleibt aus Gründen der Rückwärtskompatibilität zunächst noch erhalten, wird aber nicht mehr aktualisiert und die Anychart-Diagramme werden nicht mehr weiterentwickelt. Auf längere Sicht kann man davon ausgehen, dass JET in Apex Anychart ablöst.

Abbildung 4 zeigt ein Beispiel für die neuen JET-Diagramme. Für die meisten der in Apex enthaltenen Diagramm-Typen wird es eine neue JET-Implementierung geben, einige wenige, wie Gantt- oder Map-Diagramme, werden vorerst noch bei der Anychart-Implementierung bleiben. Nach dem Upgrade der Apex-Installation auf Apex 5.1 wird man bestehende Anychart-Diagramme mit dem Application Upgrade Wizard auf JET migrieren können.

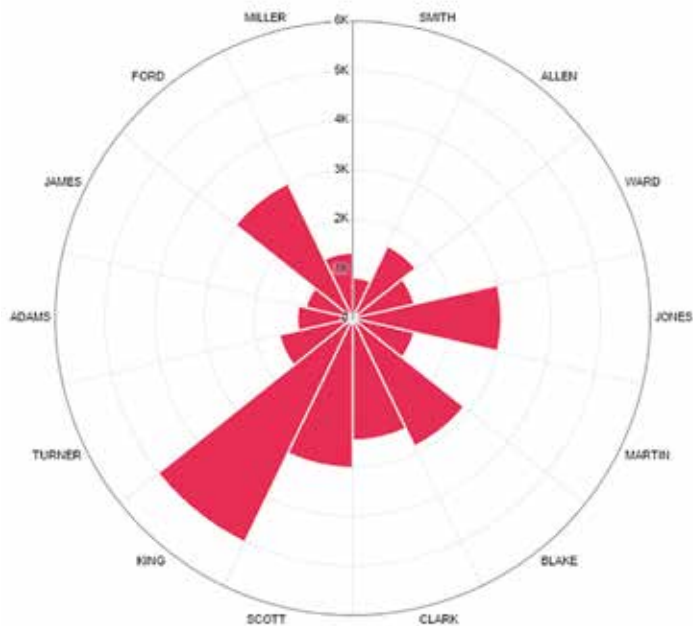


Abbildung 4: Polar Chart ist ein Beispiel für neue Diagramm-Typen mit JET Charts

Live-Template-Options

Template-Options wurden bereits mit Apex 5.0 eingeführt. Template-Options schalten die Darstellungsvarianten eines Templates deklarativ ein und aus (siehe Abbildung 5). Allerdings hat sich die Arbeit damit in der Praxis als etwas

umständlich erwiesen – oft möchte man das eine oder andere einfach nur ausprobieren und dann anschauen, wie es aussieht. Dazu sind jedoch ständige Wechsel zwischen Page Designer und der Anwendung nötig.

Die neuen Live-Template-Options werden Abhilfe schaffen: Aktiviert man in der Developer Toolbar den Quick Edit Modus, so erhalten die einzelnen Regionen auf der Seite ein Editiersymbol – klickt man es an, so öffnet sich (ganz ähnlich zum bereits seit Apex 5.0 eingeführten Theme Roller) ein Dialog zum Ändern der Template-Options (siehe Abbildung 6). Jede Änderung wird sofort in der Anwendung angezeigt, sodass man den Effekt gleich sieht. Man kann nun in Ruhe so lange ausprobieren, bis man das gewünschte Ergebnis erreicht hat, und dieses dann abspeichern.

Verschiedenes

Neben den gerade beschriebenen größeren Änderungen wird Apex 5.1, wie alle Releases bislang, eine Fülle an neuen Funktionen und Änderungen im Detail mitbringen. So wird das Universal Theme Unterstützung für die Darstellung von rechts nach links haben – das steht im deutschsprachigen Raum si-

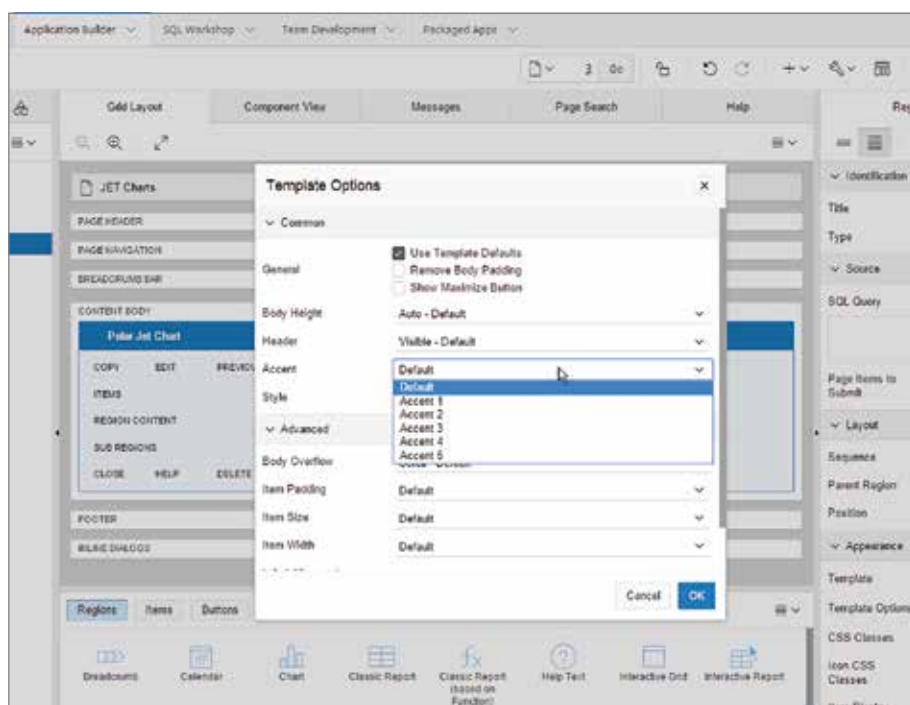


Abbildung 5: Template-Options für ein Region-Template

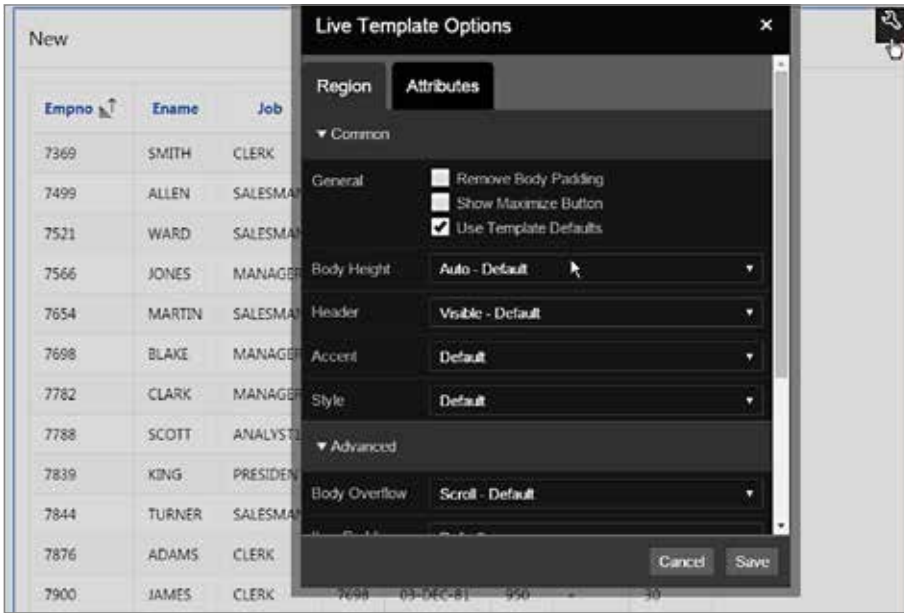


Abbildung 6: Änderungen an den Live-Template-Options sind sofort sichtbar

cherlich weniger im Fokus, ist für arabische oder hebräische Anwendungen jedoch sehr wichtig. Freunde der Component View finden diese nun als integralen Bestandteil des Page Designer vor (siehe Abbildung 7).

Es ist geplant, die alte, klassische Component View standardmäßig abzuschalten; wer ohne diese nicht auskommt, kann sie in den User-Preferences wieder aktivieren.

Fazit

Apex 5.1 bringt interessante und für die Praxis wirklich wichtige neue Funktionen mit. So wird das neue Interactive Grid, als Basis für tabellarische und Master-Detail-Formulare, sicherlich Bestandteil vieler Apex-Anwendungen werden. Einige der Neuerungen von Apex 5.0 werden mit diesem Release erst richtig abgerundet – wie man an den Live-Template-Options als Ergänzung zum



Carsten Czarski
carsten.czarski@oracle.com
<http://sql-plsql-de.blogspot.com>

Theme Roller oder den Verbesserungen im Page Designer erkennen kann.

Wie immer wird es auch hier ein öffentliches Beta-Programm für Apex geben – auf dem Demoserver kann schnell und einfach ein Workspace eingerichtet und mit dem Test der neuen Funktionen begonnen werden. Jegliches Feedback ans Apex-Entwicklerteam ist gerade in dieser Phase sehr wertvoll.

Weitere Informationen

- Öffentliche Apex-5.1-Test-Instanz: <https://apexa.oracle.com>
- Apex im Oracle Technet: <https://otn.oracle.com/apex>
- Deutschsprachige Apex- und PL/SQL-Community: https://blogs.oracle.com/apex-community_deutsch

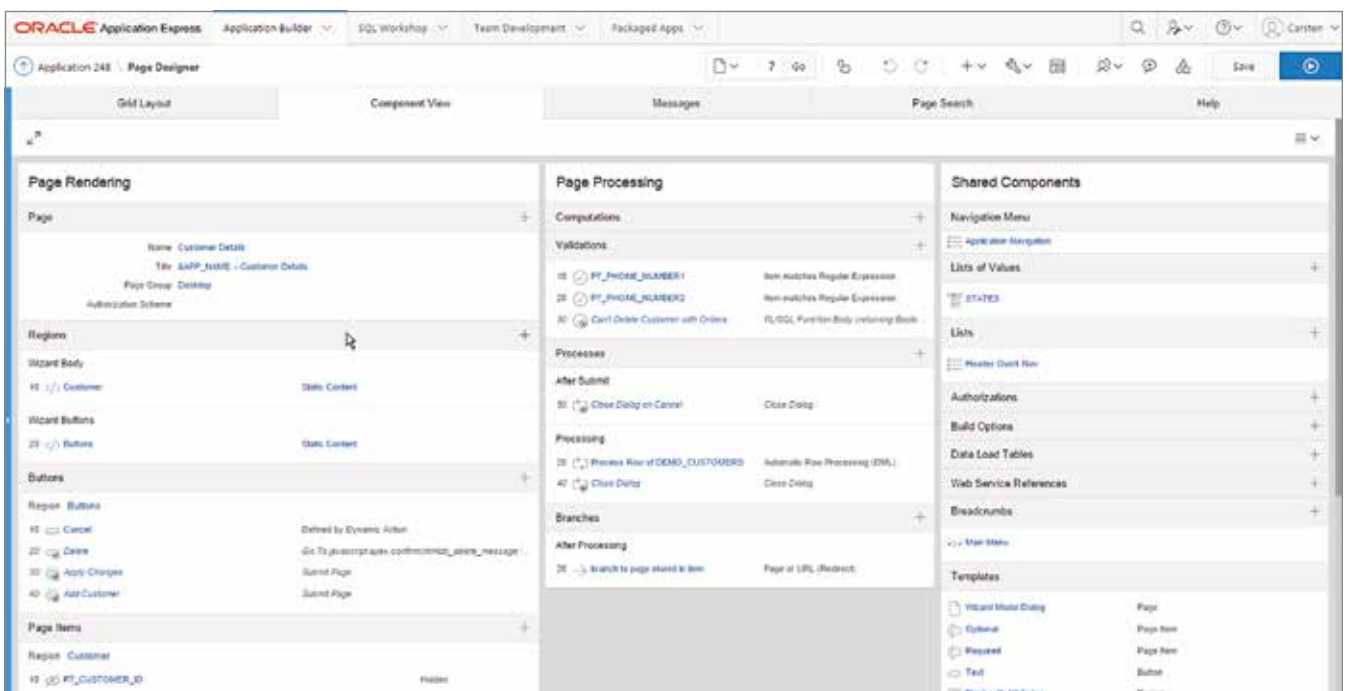


Abbildung 7: Die Component View ist nun Teil des Page Designer

„Wir setzen ganz stark auf das Feedback unserer engagierten Community ...“



Patrick Wolf

Principal Member of Technical Staff
Oracle Application Express

Was ist die Motivation von Oracle, die Version Apex 5.1 herauszubringen?

Ein Entwicklungstool wie Oracle Apex ist nie wirklich fertig, weil gerade im Web-Umfeld kontinuierlich Innovationen stattfinden, die unsere Kunden auch gerne in ihren Applikationen einsetzen möchten. Die neue Interactive Grid Region ist sicher ein gutes Beispiel dafür, wie eine zeitgemäße, sehr mächtige, aber trotzdem einfach zu verwendende Multi-Row-Eingabe-Region aussehen kann, die auch als Master-Detail (- Detail)-Region auf einer Seite verwendet werden kann. Mit Integration der Charting-Komponenten von Oracle JET tragen wir den Responsive-Design-Trends Rechnung, um ansprechende Daten-Visualisierung auf allen Geräten – angefangen vom Smartphone bis zum Desktop – ohne weitere Programmierung anzeigen zu können.

Was gilt es bei der Migration von Apex 5.0 auf 5.1 zu beachten?

Eines unserer Hauptziele bei jeder neuen Version von Apex besteht darin, eine möglichst große Abwärtskompatibilität sicherzustellen. Kunden können daher ihre alten Applikationen in die neue Version importieren und diese sollten sofort funktionieren, ohne dass Änderungen vorgenommen werden müssen.

Das ist grundsätzlich bis zurück zur Version HTMLDB 1.6 durchgängig eingehalten. Das bedeutet aber auch, dass wir es vermeiden, bei einem Upgrade das Verhalten oder das Look and Feel bestehender Applikationen zu verändern. Wenn Entwickler daher neue Features verwenden möchten, die im Universal Theme in der Version 5.1 hinzugefügt wurden, dann müssen sie das Theme in der Applikation explizit aktualisieren. Das funktioniert seit Apex 5.0 mit einem simplen Mausklick in den Theme-Einstellungen. Mit dem Application Upgrade Wizard besteht die einfache Möglichkeit, etwa bestehende Tabular Forms auf Interactive Grids und AnyChart-Regionen auf JET-Charts zu migrieren.

Wie sieht der Fahrplan bis zum Release von 5.1 aus? Welche Maßnahmen tragen dazu bei, ein qualitativ hochwertiges Release an den Start zu bringen?

Wie bei jeder neuen Version von Apex setzen wir ganz stark auf das Feedback unserer engagierten Community. Es wird daher wie früher ein oder zwei gehostete Early-Adopter-Releases geben, mit denen die Community frühzeitig die neuen Features ausprobieren, testen und Feedback geben kann. Es hat sich in der Vergangenheit gezeigt, dass diese vielen Entwickler, die einen ganz unterschiedlichen Wissenstand zu Apex haben, ganz entscheidend beim Finden von Bugs sind. Das soll natürlich nicht heißen, dass wir intern keine QA-Tests durchführen, um ein hochwertiges Release an den Start zu bringen.

Welche Informationsquellen würden Sie empfehlen, wenn jemand noch keine Erfahrung mit Apex hat?

Auf „<https://apex.oracle.com>“ befindet sich ganz unten ein „Learn More“-Bereich, in dem es Links zu Videos, Tutorials, Hands-on Labs und der Do-

kumentation gibt. Unter „http://blogs.oracle.com/apexcommunity_deutsch“ gibt es viele Tipps auch auf Deutsch. Ganz klassisch würde ich aber empfehlen, eines der vielen Bücher zu Apex zu lesen und das Gelernte dann sofort mit einem kostenlosen Workspace auf „<https://apex.oracle.com>“ auszuprobieren. Danach ist der Besuch eines Kurses oder einer Konferenz wie der Apex Connect oder der DOAG 2016 Konferenz mit ihren vielen unterschiedlichen Apex-Vorträgen sehr zu empfehlen, um das Wissen zu vertiefen.

Mike Hichwa, Vice President bei Oracle, wird auf Apex Connect einiges über die Zukunft von Apex erzählen. Was ist Ihre persönliche Vision für Apex?

Meine persönliche mittelfristige Vision ist, dass sich die Out-of-the-Box-Experience für Endanwender von Apex-Applikationen noch mehr verbessert – etwa die automatischen Client-Side-Validierungen oder die Reduktion der kompletten Seiten-Aktualisierungen („Single Page“-Apps). Ich habe auch noch einige Ideen, um den Software-Entwicklungsprozess mit Apex zu verbessern, und bei der Integration von REST-Services sehe ich auch noch viel Luft nach oben. Wir als Team sind aber meistens sehr agil bei der Umsetzung neuer Features, denn die besten Ideen entstehen oft aus der Praxis heraus, wenn uns Kunden auf einer Konferenz oder über Twitter („[#orclapex](https://twitter.com/orclapex)“) persönlich ansprechen. Darum haben wir mit „<https://apex.oracle.com/vote>“ auch eine Möglichkeit für die Community geschaffen, um aktiv die Zukunft von Apex mitzugestalten.

Patrick Wolf
patrick.wolf@oracle.com



Mehrsprachigkeit in Apex-Anwendungen

Stefan Röß, BASF-Gruppe

Um den zeitlichen und finanziellen Mehraufwand von Übersetzungen gering zu halten, einigt man sich in den Anforderungsspezifikationen zumeist darauf, die Anwendung in Englisch zu betreiben. Doch nicht nur in großen internationalen Unternehmen, auch in kleinen und mittelständischen Firmen sprechen Mitarbeiter oft unterschiedliche Sprachen. Auch müssen Anwendungen in bestimmten Ländern mehrsprachig angeboten werden, wenn mehr als eine offizielle Landessprache vorhanden ist, wie beispielsweise in der Schweiz, Belgien oder Kanada. BASF bietet eine ausgereifte Methodik an, um Mehrsprachigkeit für Apex-Anwendungen umzusetzen.

Die hier vorgestellte Lösung zeigt eine Best Practice, die erst seit der Apex-Version 4.2.3 möglich ist. Mehrsprachigkeit ist in Apex seither einfach zu haben und stellt deshalb keine nennenswerte Hürde mehr dar. Die BASF-Apex-Standard-Anwendung („BASTa“) lagert das Übersetzen als eine Funktionalität in die Anwendung

aus, man übergibt also die Verantwortung an die Personen, die die Fremdsprachen beherrschen. Dabei müssen diese noch nicht einmal fachliche Anwender der Applikation sein. Somit sind die Zeiten schwerverständlicher XLIFF-Dateien, die einen speziellen Editor benötigen, endgültig vorbei. Mithilfe des Apex-API „Apex_

LANG“ ist es gelungen, eine einfache, rollenbasierte Methodik zu entwickeln, die als Muster für jede Apex-Anwendung zur Mehrsprachigkeit dienen kann.

Was muss übersetzt werden?

Jede neue Apex-Anwendung innerhalb der BASF wird mit dem BASTa-Framework erstellt. Dabei ist es immer möglich, eine oder mehrere Sprachen nachträglich zu implementieren. Übersetzt werden alle Anwendungssseiten des Nutzer-Interface und die Status-Nachrichten wie Bestätigungen oder Fehlermeldungen (siehe Abbildung 1).

Tabellendaten, die die Inhalte einer Anwendung betreffen, werden hierbei nicht berücksichtigt. Sollen auch diese Daten übersetzt werden müssen, kann dies durch ein fachliches Datenmodell in Form einer Übersetzungstabelle geschehen. Dies ist unabhängig von Apex und von der jeweiligen Frontend-Technologie.

Übersetzungsprozess

Das Nutzer-Interface wird in BASTa mit dem integrierten Mehrsprachigkeitsmodul von Apex übersetzt. Dabei wird für jede Sprache eine Art Schatten-Applikation erzeugt.



Abbildung 1: Beispiel für ein Nutzer-Interface

Create	Anlegen der Übersetzungssprache, wird nur einmal durchgeführt
Seed	Anlage der Texte in einer Apex-Tabelle
Translate	Übersetzen der Texte
Publish	Veröffentlichen der übersetzten Texte

Tabelle 1


```

apex_lang.create_language_mapping (p_application_id => p_app_id
                                ,p_language       => p_lang
                                ,p_translation_application_id => p_trans_app_id);

```

Listing 1

Die folgenden vier Schritte müssen für jede Sprache durchgeführt werden und sind in der Anwendung für Endanwender mit der Rolle „Translator“ und „Developer“ verfügbar (siehe Tabelle 1).

Create-Prozess

Der Prozess „Create“ ordnet die Originalsprache (Primary Application Language) der übersetzten Sprache zu. Dies geschieht über einen Eintrag in der Tabelle „www_flow_language_map“ des Apex-Schemas. Ausgeführt wird der Eintrag in

die Tabelle über die API-Prozedur „create_language_mapping“ (siehe Listing 1) aus dem Parsing-Schema.

Abbildung 2 zeigt die Nutzer-Interface-Umsetzung zu Listing 1. Die Application ID „10101“ wird hier nicht angezeigt, da sie für die deutsche Sprache reserviert wurde. Wichtig bei der Parameterübergabe aus Listing 1 ist, dass man „p_language“ einen IANA-Code der Sprache zuordnet. Aktuell sind 133 IANA-Codes unterstützt (siehe „https://docs.oracle.com/cd/B28359_01/appdev.111/b32258/global.htm“). Der Parameter „p_translation_application_id“ kann eine frei wählbare ID sein. Es hat sich jedoch er-

wiesen, dafür die APPLICATION_ID mit einer angefügten 01 bis 09 zu verwenden.

So wurde in Abbildung 3 der APPLICATION_ID 101 und dem IANA-Code der deutschen Sprache „de“ die „01“ zugeordnet; somit ergibt sich die ID 10101.

Benötigt man mehr als zehn unterschiedliche Sprachen, werden die IDs 11 bis 19, 21 bis 29, 31 bis 39 etc. angefügt. Die ID darf nicht mit der Zahl „NULL (0)“ enden, da dies zu einem Fehler beim Ausführen von Listing 1 führt. Listing 2 zeigt die wichtigsten Parameter nach dem Ausführen von Listing 1 an.

Abbildung 4 zeigt den Eintrag zu „Application Language Derived From“ = „Item Preference ...“. Um diesen Eintrag verwenden zu können, muss eine Applikationsvariable „FSP_LANGUAGE_PREFERENCE“ erstellt werden. Ein Applikationsprozess, der beim Login der Anwendung ausgeführt wird, beziehungsweise ein Page-Prozess, der im Nachgang vom Endanwender aktiv angestoßen werden kann, setzt diese Variable auf den gewünschten IANA-Code und damit auf die Zielsprache des Endanwenders. Ist die Übersetzung eines Textes nicht vorhanden, wird zunächst in der darüber liegenden Sprache der Eintrag gesucht, beispielsweise von Frankokanadisch (fr-ca) zu Französisch (fr). Ist diese Übersetzung auch nicht vorhanden, wird die Originalsprache der Anwendung („Application Primary Language“) verwendet.

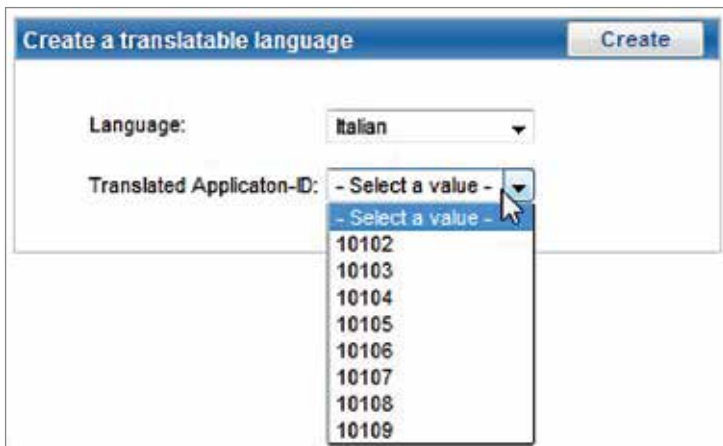


Abbildung 2: Anlegen einer Translation Application ID

```

SELECT a.workspace
       ,a.application_id
       ,m.translated_application_id
       ,a.application_primary_language
       ,m.translated_app_language
       ,a.language_derived_from
FROM apex_applications a, apex_application_trans_map m
WHERE 1 = 1
AND m.primary_application_id = a.application_id;

```

Listing 2

Seed-Prozess

Im Prozess-Schritt „Seed“ werden Texte der Originalsprache des Nutzer-Interface in der Apex-Tabelle „www_flow_translatable_text\$“ angelegt. Die folgenden Apex-Schema-Objekte sind davon betroffen. Dabei wird die Tabelle über die Prozedur „www_flow_lang.

WORKSPACE	APPLICATION_ID	TRANSLATED_APPLICATION_ID	APPLICATION_PRIMARY_LANGUAGE	TRANSLATED_APP_LANGUAGE	LANGUAGE_DERIVED_FROM
BASTA	101	10101	en-us	de	ITEM_PREFERENCE
BASTA	101	10102	en-us	it	ITEM_PREFERENCE

Abbildung 3: Die Ergebnisse aus Listing 2 – man sieht den IANA Code „de“ und „it“

seed_translations“ und mit Bezug auf die Meta-Tabelle „www_flow_translatable_cols\$“ mit allen übersetzbaren Werten befüllt. Man spricht hier von Apex-internen Vorgängen, die über die API-Prozedur „seed_translations“ aus dem „Parsing-Schema“ heraus ausgeführt werden (siehe Listing 3).

Translation-Prozess

Jetzt werden über das Parsing-Schema der API-Prozedur „update_translated_string“ die Werte der Spalte „translate_to_text“ aus der Tabelle „www_flow_translatable_text\$“ des Apex-Schemas geändert (siehe Listing 4).

Texte können auf zwei Arten übersetzt werden: entweder über eine Einzeldatensatz-Verarbeitung, etwa für nachträgliche, punktuelle Anpassungen, oder über einen Excel-Download zur Massenverarbeitung und anschließenden Excel-Upload, etwa für das initiale, erstmalige Übersetzen (siehe Listing 5 sowie Abbildungen 5).

Publish-Prozess

Die Übersetzungen werden durch „publish“ live geschaltet. Dies geschieht zunächst über einen internen Kopiervorgang, das Erzeugen der sogenannten „Schatten-Applikation“. Dabei sind unter anderem die folgenden Apex-Schema-Objekte betroffen: Zunächst wird über die Prozedur „www_flow_translation_util_api.flow_copy“ die neue Sprachen-ID im Language-Repository angelegt. Dann werden die übersetzten Texte mithilfe der Prozedur „www_flow_translation_util_api.sync_translations“ aus der Tabelle „www_flow_translatable_text\$“ auf die jeweiligen Tabellen des Language-Repository übertragen. Als Metadaten-Referenz gilt hier ebenfalls „www_flow_translatable_cols\$“. Dies sind interne Vorgänge im Apex-Schema, die über die API-Prozedur „publish_application“ aus dem Parsing-Schema heraus gestartet werden (siehe Listing 6). Abbildung 5 zeigt oben in der Leiste den Publish-Button, der das API „apex_lang.publish_application“ ausführt.

Löschen einer Sprache

Im Löschvorgang werden alle notwendigen Daten aus den Tabellen des Sprach-



Abbildung 4: Globalisierungsparameter der Anwendung 101

```
apex_lang.seed_translations (p_application_id => p_app_id
                           ,p_language       => lv_trans_lang);
```

Listing 3: Seeding der Sprache. Dem Parameter „p_language“ wird der IANA-Code der zu übersetzenden Sprache übergeben („lv_trans_lang“)

```
apex_lang.update_translated_string (p_id       => p_id
                                   ,p_language => lv_trans_lang
                                   ,p_string   => p_to_string);
```

Listing 4

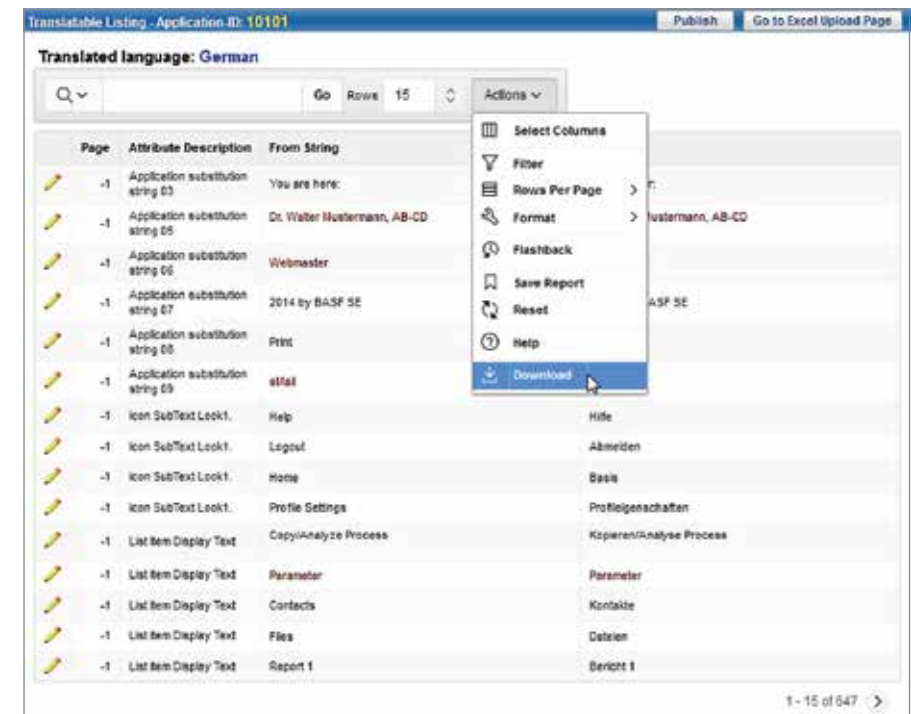


Abbildung 5: Report auf die Apex-View „apex_application_trans_repos“ mit der Möglichkeit zum Download für Massenverarbeitung oder über das Stift-Icon zur Einzeldatensatz-Verarbeitung

chen-Repository mit der Prozedur „delete_language_mapping“ gelöscht. Listing

7 zeigt den Aufruf aus dem Parsing-Schema.

Anlegen von Translatable Messages

Mit den Translatable Messages sind Nachrichten wie Bestätigungen oder Fehlermeldungen gemeint. Sie können aber auch für alle anderen Arten von Text Verwendung finden. Nachrichten werden in die Tabelle „`wwv_flow_messages$`“ des Apex-Schemas eingefügt und können über die View „APEX_APPLICATION_TRANSLATIONS“ des Parsing-

Schemas ausgewählt werden. Mit dem Aufruf der Prozedur „`create_message`“ werden Nachrichten befüllt (siehe Listing 8).

Aktualisieren von Nachrichten

Um eine bestimmte Nachricht zu verändern, ruft man die Prozedur „`update_message`“ aus dem Parsing-Schema auf (siehe Listing 9).

```
FOR c
  IN (SELECT seq_id
      ,replace(c002, '') c002
      ,c003
      ,c004
      ,c005
      ,c006
      FROM apex_collections
      WHERE collection_name = p_coll_name AND c001 = p_excel_name AND
      seq_id != 1)
LOOP
  Apex_lang.update_translated_string (p_id          => c.c002
                                     ,p_language   => lv_trans_lang
                                     ,p_string     => c.c006);
END LOOP;
```

Listing 5: Massenverarbeitung erfolgt über eine „Apex_COLLECTION“

```
apex_lang.publish_application (p_application_id=> p_app_id
                              ,p_language     => lv_trans_lang);
```

Listing 6: Live-Schalten einer bestimmten Mapping-Language

```
apex_lang.delete_language_mapping (p_application_id => p_app_id
                                   ,p_language     => lv_trans_lang);
```

Listing 7: Löschen einer bestimmten Mapping Language

```
wwv_flow_api.create_message (p_flow_id      => p_app_id
                             ,p_name        => upper (p_message_name)
                             ,p_message_language=> lower (p_target_lang)
                             ,p_message_text => p_message_text);
```

Listing 8: Call von „`create_message`“ vom Parsing-Schema. Der Parameter „`p_message_name`“ legt den Message-Identifizier für eine bestimmte Sprache „`p_target_lang`“ fest. In „`p_message_text`“ steht der übersetzte Text

Löschen von Nachrichten

Interessanterweise werden Nachrichten mit der gleichen Prozedur gelöscht wie hinzugefügt. Der entscheidende Unterschied ist der Modus („Global Mode“). Man setzt diesen Modus auf „REMOVE“ und übergibt der „`create_message`“-Prozedur die „`translated_entry_id` (lv_trans_id)“, das ist der eindeutige Schlüssel für den Datensatz (siehe Listing 10).

Darstellung der Nachricht

Eine erstellte Nachricht wird im PL/SQL-Code über die Funktion „`apex_lang.message`“ aufgerufen und in der Applikation dargestellt. Dabei können bis zu zehn verschiedene Parameter dem Identifier mitgegeben werden. In Listing 11 wurden dem Message-Identifizier „CREATE_LANG_MAP_SUCCESS“ zwei Parameter übergeben. Diese Parameter flexibilisieren die zu erzeugende Nachricht (siehe Abbildung 6).

Ist der Application Builder in weiteren Sprachen installiert, was über die SQL-Skripte „`load_language.sql`“ wie „`load_de.sql`“ oder „`load_it.sql`“ möglich ist, sind die übersetzten Texte für den Interactive Report und die Paginierung der Reports enthalten. Ist dies nicht der Fall, sind die „APEXIR_*“-Texte wie „APEXIR_SELECT_COLUMNS“, „APEXIR_FILTER“ und „APEXIR_ROWS_PER_PAGE“ sowie die Pagination-Texte wie „PAGINATION.PREVIOUS_SET“ in den Messages für die jeweilige Sprache zu übersetzen (siehe „https://docs.oracle.com/cd/B28359_01/appdev.111/b32258/global.htm“ und Abbildung 7).

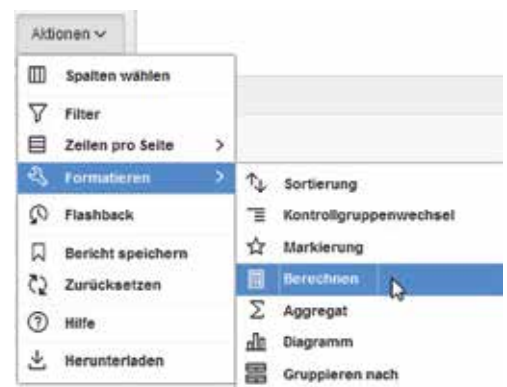


Abbildung 7: IR-Report-Action-Button und dessen Einträge



Abbildung 6: Der „CREATE_LANG_MAP_SUCCESS“-Message-Identifizier

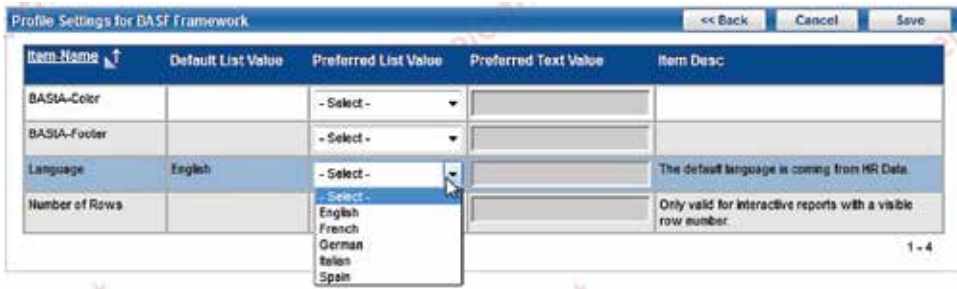


Abbildung 8: Umschalten in eine andere Sprache

```
apex_lang.update_message (p_id          => lv_trans_id
                        ,p message text => p_upd_message_text);
```

Listing 9

```
wwv_flow_api.g_mode := 'REMOVE';
wwv_flow_api.create_message (p_id => lv_trans_id);
```

Listing 10

```
apex_lang.message ('CREATE_LANG_MAP_SUCCESS'
                 ,lv_lang
                 ,p-trans_app_id);
```

Listing 11

Auswahl der gewünschten Nutzer-Interface-Sprache

Für jeden neuen Anwender wird die jeweilige Sprache in einer Userprofile-Tabelle festgehalten. Meldet sich ein Anwender an, erfolgt ein „Post Authentication Process“, der das Application-Item „FSP_LANGUAGE_PREFERENCE“ setzt. In den Globalisierungsparametern wurde deshalb unter „Shared Components“ die „Application Derived Language“ auf „Item Preference ...“ eingestellt (siehe Abbildung 6). Nachdem alle Sprachkomponenten übersetzt sind, kann jeder sein eigenes Sprachenprofil auch verändern. Dazu wird über einen Page-Prozess „FSP_LANGUAGE_PREFERENCE“ verändert (siehe Abbildung 8).

Auf die API-Methode „apex_lang.lang“ („Klavierspieler-Methode“) hat BASF ganz verzichtet. Damit könnten Werte aus dynamischen Lists of Values (LOV) und Werte aus anderen Tabellen übersetzt werden. Da dies nicht zu den typischen Eigenschaften eines Nutzer-Interface zählt, werden Übersetzungen in einem solchen Fall von einem fachlichen Datenmodell berücksichtigt.

Applikationserweiterungen

Im Laufe der Zeit ändern sich Anwendungen aufgrund neuer Anforderungen. In diesem Fall wird die Applikation ganz normal über den Application Builder weiterentwickelt, dies geschieht immer in der „Application Primary Language“.

Zu jeder Zeit können die Prozesse „Seed“, „Translate“ und „Publish“ durchgeführt werden, sodass neue Items, Regionen und Pages in der zu übersetzenden Sprache dargestellt beziehungsweise gelöscht werden, wenn sie obsolet sind. Das interne Mehrsprachigkeitsmodul von Apex übernimmt dafür die nötigen Schritte.

Export und Import übersetzter Applikationen

Die übersetzten Nachrichten („Translated Messages“) werden bei einem Export immer einbezogen und stehen bei einem Import der Anwendung sofort zur Verfügung.

Dabei spielt die Einstellung der Export-Einstellungen keine Rolle. Das Export-Translation-Flag ist lediglich für das Sprachen-Repository gültig, das mit dem internen Mehrsprachigkeitsmodul („Create“, „Seed“, „Translate“, „Publish“) angelegt wurde. Damit lässt sich steuern, ob alle Übersetzungen exportiert werden sollen. Hat man beim Export das Flag auf „Yes“ gesetzt, erhält man beim anschließenden Import in eine andere Datenbank eine neue Schatten-Applikations-ID („Translated Application ID“). Dabei wird die nächste freie Nummer gezogen. Das ist nicht weiter tragisch, nur sollte man es wissen. Aber auch hier gibt es Möglichkeiten, die gleiche ID zu behalten.

Fazit

Das Mehrsprachigkeitsmodul von Apex ist mit seinen bestehenden Bibliotheken einfach in eine Applikation einzubinden, um daraus ein eigenes Framework aufzubauen. Der Entwickler kann sich daraufhin voll und ganz um die Umsetzung der fachlichen Anforderungen kümmern. Übersetzungen werden von den Kunden beziehungsweise der Fachseite selbst vorgenommen. Mit den bestehenden API-Methoden kann man effizient Mehrsprachigkeit umsetzen. Das Apex-Framework der BASF ist somit mehr als konkurrenzfähig für multilinguale Anwendungen gegenüber anderen Software-Produkten.



Stefan Röß
stefan.roess@basf.com

Cloud-Lösungen für Apex

Till Albert und Niels de Bruijn, MT AG

„Cloud“ ist aktuell in aller Munde, dabei ist dieses Thema für Apex nichts Neues. Seit seiner Entstehung ist Apex nämlich per se Cloud-fähig. Es wird nur einmal installiert und kann dann von sämtlichen Abteilungen im Unternehmen verwendet werden. Dabei ist ausgeschlossen, dass die Anwendungen wechselseitig einsehbar sind.

In der letzten Zeit gibt es jedoch immer mehr Angebote, um Apex als „Platform as a Service“ (PaaS) in der Cloud zu betreiben. Die Auswahl am Markt ist groß, doch welche Lösung passt am besten zu den Anforderungen des Unternehmens? Einen Server mit CentOS mieten und die Wartung selbst in die Hand nehmen oder doch lieber direkt einen Cloud-Anbieter für Apex wählen und die Wartung übernehmen lassen? Was gibt es für Einsteiger in Sachen „Apex in der Cloud“?

Kostenlose Hosting-Angebote

Apex in der Cloud hat sicher fast jeder Apex-Entwickler schon einmal eingesetzt. Name, E-Mail-Adresse, Name des Workspace und das Datenbank-Schema – diese Angaben reichen auf „apex.oracle.com“, um nach wenigen Minuten eine sofort einsetzbare Apex-Umgebung benutzen zu können (siehe Abbildung 1). Dieser Service von Oracle ist komplett kostenfrei, jedoch ausdrücklich nur für Testzwecke und nicht für den Produktivbetrieb gedacht. Daher werden die Workspaces automatisch nach einer gewissen Zeit ohne Aktivität gelöscht. Der Zugriff auf die Umgebung, etwa über SQL*Plus oder den SQL Developer, ist darüber hinaus nicht möglich. Auch der Zugriff auf externe Quellen ist vom Server heraus gesperrt.

Für den produktiven Betrieb von Apex-Anwendungen bietet Oracle bereits „cloud.oracle.com“ an. Hier ist für das Jahr 2016 eine abgespeckte und kostenfreie Variante des Angebots geplant, die nach

aktuellem Stand eine Begrenzung von 100 MB Speicherplatz haben wird [1].

Für den schmalen Geldbeutel

Hat man seine erste Anwendung erstellt und möchte diese in den produktiven Betrieb nehmen, kann man für ein geringes Budget zwischen zehn und fünfzig Euro pro Monat je nach Ausprägung eine gehostete Apex-Umgebung innerhalb kurzer Zeit bereitgestellt bekommen [2]. Diese bietet eine vorinstallierte Oracle-Datenbank mit der aktuellen Version von Apex und Oracle REST Data Services

(ORDS). Je nach Konfiguration werden tägliche Backups und die Überwachung des Systems übernommen (siehe Abbildung 2). Bei einigen Anbietern sind auch verschiedene Reporting-Tools wie Jasper-Reports bereits integriert.

Die Kontrolle behalten

Wer das Know-how für die Installation und den Betrieb hat und wenig investieren will, kann Apex & Co. auch selbst installieren. Bei der Auswahl eines Anbieters sollte man sich dabei die Frage stellen, was neben der Bereitstellung der Ka-

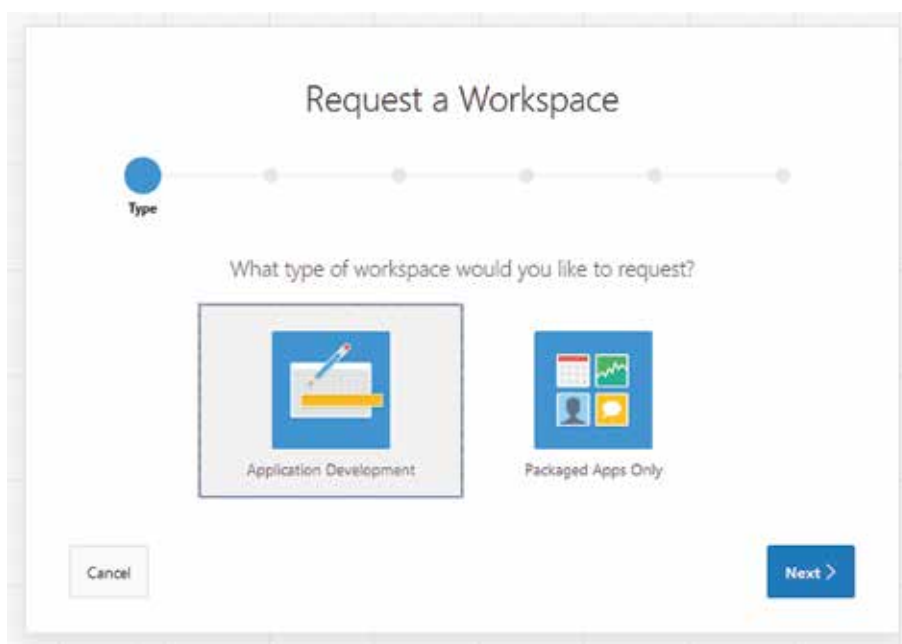


Abbildung 1: Beantragung eines kostenlosen Workspace auf „apex.oracle.com“

pazitäten eine Rolle spielt. Möchte man schnell einsteigen, ohne von einer Flut an Möglichkeiten überwältigt zu werden, und nur eine Apex-Instanz nutzen, bieten sich Anbieter wie „digitalocean.com“ (US) oder „domainfactory.eu“ (DE) an. Die Anzahl an Möglichkeiten ist überschaubar; man erkennt schnell, welche Infrastruktur benötigt wird, und kann loslegen.

Bei reinen Cloud-Anbietern ist es üblich, dass auch wirklich nur dann Kosten anfallen, wenn der Server genutzt wird. Dabei werden monatlich die genutzten Stunden in Rechnung gestellt, maximal aber eine Obergrenze von einem fixen Betrag pro Monat – abhängig von der jeweiligen Konfiguration des Systems. Wenn wirklich keine Kosten anfallen sollen, ist die Instanz des Servers zu deaktivieren, wobei ein Snapshot erstellt wird, der jederzeit wieder eingespielt werden kann. Ist mehr Leistung für das System erforderlich, reicht eine Änderung der Konfiguration mit anschließendem Neustart der Maschine aus.

Es ist zu beachten, dass man die Installation der Software selbst vornehmen muss. Meist ist CentOS als Betriebssystem bereits vorinstalliert. In diesem Fall kann man innerhalb weniger Minuten auf Oracle Linux umsteigen [3]. Die Installation der Apex-Software erfolgt anschließend bequem mit dem Build-Script „Oxar“ vom OraOpenSource-Projekt (Martin Giffy D'Souza, [4]). Dieses enthält mit einer XE-11g-R2-Datenbank, Apex 5 und ORDS alles Notwendige, um sofort mit der Entwicklung der Apex-Anwendungen loslegen zu können.

Enterprise-Ready-Lösungen

Ist die Anzahl der Zugriffe auf die Anwendung über das Jahr sehr unterschiedlich, etwa zu Spitzenzeiten während des Weihnachtsgeschäfts für E-Commerce-Anwendungen, muss die Architektur schnell und ohne Downtime skalierbar sein. Die Infrastruktur hat sich dem Bedarf anzupassen, eventuell sogar automatisch. Dies ist bei den kleineren Anbietern nicht ohne Weiteres möglich. Zudem sind die Vertragslaufzeiten bei diesen oft nicht flexibel genug. Große Anbieter wie AWS, Oracle und Microsoft bieten Lösungen an, mit denen die komplette Kontrol-

Managed Private Clouds Plans (Dedicated)			
Item	Options		Price (€/month)
Database	SE1 <input checked="" type="radio"/>	SE <input type="radio"/>	
Service Level	STANDARD <input checked="" type="radio"/>	PREMIUM <input type="radio"/>	+ € 0,-
VM	2 vCPU <input checked="" type="radio"/>	4 vCPU <input type="radio"/>	+ € 0,-
Tablespace (GB)	<input type="range" value="5"/>		5 GB + € 0,-
Webpace (GB)	<input type="range" value="1"/>		1 GB + € 0,-
Oracle Database License	BYOL <input checked="" type="radio"/>	LICENSE INCLUDED <input type="radio"/>	+ € 0,-
Oracle license and prices are subject to approval by Oracle and may differ.		TOTAL	€ 155,-
			<input type="button" value="Request"/>
Contract Period: 12 months prices may change after the contract period ends			

Abbildung 2: Konfigurieren einer Instanz, hier beispielhaft auf apex-cloud.com.

le behalten und die Services nur für einen gewissen Zeitraum gemietet werden können.

Im Vergleich zu „Easy-Entry“-Anbietern wie „digitalocean.com“ bieten sie viel mehr Möglichkeiten, sodass das komplette Angebot nicht mal eben überblickt werden kann. Nehmen wir beispielsweise die Amazon Web Services (AWS), die je nach Konfiguration auch schon für kleines Budget genutzt werden können. Sie lassen sich sogar bis zu einem bestimmten Limit zwölf Monate lang kostenlos nutzen. Dabei fallen in diesem Zeitraum nur dann Kosten an, wenn die kostenlosen Limits überschritten werden. Diese lassen sich bequem überwachen; bei Überschreitung des Limits wird, wenn gewünscht, eine Benachrichtigungs-E-Mail versendet.

AWS bietet mehr als fünfzig verschiedene Cloud-Services an. Für den Betrieb von Apex sind hauptsächlich die Services Elastic Cloud Compute (EC2) und Relational Database Service (RDS) relevant. Ersterer ermöglicht es, mit einem riesigen Pool von Software-Images schnell

einen Server in der Cloud verfügbar zu machen. So ist beispielsweise innerhalb von wenigen Minuten ein Linux-Server bereit zum Einsatz.

Der RDS-Service ist, wie der Name es vermuten lässt, für den Betrieb von Datenbanken gedacht. Dabei stehen neben der Oracle-Datenbank auch andere Datenbank-Systeme wie die Amazon-eigene Aurora-Datenbank oder Microsofts SQL-Server zur Verfügung. Für Oracle-Datenbanken sind dabei verschiedene Versionen von 11 und 12 in unterschiedlichen Editionen vorhanden. Zu Produktionszwecken können bei der Installation zusätzlich Optionen für Hochverfügbarkeit und eine erhöhte Leistung gewählt werden.

Die Infrastruktur für den Aufbau der Apex-Umgebung kann also beliebig gewählt werden. So können Oracle-Datenbank, ORDS und Apache zum einen auf einer oder mehreren (wenn Datenbank und ORDS getrennt laufen) EC2-Instanzen betrieben werden. Als Alternative dazu kann die Datenbank auch direkt über den RDS-Service laufen, der

eine flexible Skalierung ermöglicht. In diesem Fall muss für diesen jedoch zusätzlich auf einer EC2-Instanz ein ORDS als Listener betrieben werden.

Leider gibt es aktuell beim RDS-Service einige gravierende Nachteile für die Entwicklung mit Apex. Denn bei diesem handelt es sich um einen Managed Service. Dies bringt zwar den Vorteil, dass die Wartung und Backups der Datenbank erledigt werden. Es bedeutet zugleich aber auch, dass zum einen keine SYSDBA-Rechte für den Kunden vorhanden sind. Zum anderen können auch keine TCP-Verbindungen aus der Datenbank aufgebaut werden, somit fallen UTL_TCP, UTL_SMTP, Apex_MAIL etc. weg.

Darüber hinaus kann Apex mangels Privilegien nicht selbst installiert und gewartet werden, dies wird auf Wunsch automatisch erledigt. Aktuell ist für die 11g-Datenbank nur Apex 4.1.1 und für die 12c-Datenbank nur Apex 4.2.6 verfügbar.

Der RDS-Service ist deshalb erst empfehlenswert, wenn auch Apex 5 für ihn zur Verfügung steht und die erwähnten Einschränkungen gelockert werden.

Um vorhandene Datenbanken aus dem heimischen Rechenzentrum in die AWS-Cloud migrieren zu können, bietet AWS in Zukunft den Database Migration Service an. Dieser ist derzeit nur als Preview verfügbar. Dabei muss sich entweder die Quell- oder Ziel-Datenbank in EC2 oder RDS befinden. Der Service stellt während der Datenbank-Migration automatisch sicher, dass Datenänderungen in der Quell-Datenbank, die während der Migration durchgeführt werden, in die Ziel-Datenbank mit repliziert werden. Darüber hinaus können Quell- und Ziel-Datenbank auch aus unterschiedlichen Engines bestehen, was beispielsweise eine Migration von MySQL zu Oracle sehr leicht ermöglicht und so vielleicht sogar den Weg zu Apex erleichtert, wenn man

bisher keine Oracle-Umgebung genutzt hat (siehe Abbildung 3).

Die Stärke von AWS liegt in der Menge der Möglichkeiten auf Seiten der Infrastruktur. So kann beispielsweise einfach ein virtuelles Netzwerk für die eigenen Instanzen angelegt werden, das einen logisch isolierten Bereich darstellt. Dieses bietet unter anderem die volle Kontrolle über die virtuelle Netzwerk-Umgebung, den IP-Adressbereich sowie die Konfiguration von Routing-Tabellen und Subnetzen.

Der Nachteil von AWS liegt bei so zahlreichen Möglichkeiten natürlich darin, dass für die Apex-Instanz (auf EC2) alles selbst konfiguriert werden muss. Es gibt keine Option, um eine Oracle-Datenbank mit Apex direkt als Image zu erstellen. Auch der ORDS muss selbst installiert und konfiguriert werden, dies gilt übrigens auch für die RDS-Variante.

Flexible SLAs. Weil Ihre IT einmalig ist.

dbi FlexService
ISO 20000

dbi FlexService für Datenbanken und Middleware: Ihr flexibles, kosteneffizientes und ISO 20000-zertifiziertes Service Management. Profitieren Sie insbesondere von unserem dedizierten Service Desk für validierte Systeme im Pharma-Bereich.

Phone +41 32 422 96 00 · Basel · Nyon · Zürich · dbi-services.com



Infrastructure at your Service.

dbi services

The screenshot shows the AWS Management Console interface for an Oracle RDS instance. At the top, there are navigation tabs for 'Launch DB Instance', 'Show Monitoring', and 'Instance Actions'. Below this is a search bar and a filter set to 'All Instances'. The instance details are displayed in a grid format:

- Configuration Details:** Engine: Oracle SE 11.2.0.4 v6; License Model: Bring Your Own License; Created Time: February 1, 2016 at 4:44:59 PM UTC+1; DB Name: ORCL; Username: Till; Character Set: AL32UTF8; Option Group: default:oracle-se-11.2 (in-sync); Parameter Group: default:oracle-se-11.2 (in-sync); Copy Tags To Snapshots: No.
- Security and Network:** Availability Zone: us-west-2b; VPC: [redacted]; Subnet Group: default (Complete); Subnets: subnet-ea734b7d, subnet-ea16f28e, subnet-63683e3a; Security Groups: default (sg-c54c8ba2) (active); Publicly Accessible: Yes; Endpoint: [redacted].amazonaws.com; Port: 1521; Certificate Authority: rds-ca-2015 (Mar 5, 2020).
- Instance and IOPS:** Instance Class: db.t2.micro; Storage Type: General Purpose (SSD); IOPS: disabled; Storage: 10 GB.
- Encryption Details:** Encryption Enabled: No.
- Availability and Durability:** DB Instance Status: available; Multi AZ: No; Automated Backups: Enabled (7 Days); Latest Restore Time: February 2, 2016 at 3:08:23 PM UTC+1.
- Maintenance Details:** Auto Minor Version Upgrade: Yes; Maintenance Window: sun:08:11-sun:08:41; Backup Window: 08:43-09:13; Pending Maintenance: None.

Abbildung 3: Übersicht einer Oracle-Datenbank auf einer RDS-Instanz in der Management-Console von AWS

Bei der enormen Menge an Optionen ist es schwierig, auch bei der Planung der Kosten einen Überblick zu behalten. Daher bietet AWS einen Kalkulator an, mit dem sich die Kosten anhand von festzulegenden Kennzahlen grob abschätzen lassen [5].

Fazit

Generell muss jedes Unternehmen für sich die Frage beantworten, welche Anforderungen an eine Apex-Instanz in der Cloud gestellt werden. Um mal eben schnell den Kollegen eine Demo zu zeigen, reichen die kostenlosen Varianten vollkommen aus.

Wer mit seiner ersten kleinen Applikation produktiv gehen will, aber kein Personal für den Betrieb der Server und Datenbanken hat, für den kommt eine Cloud-Lösung wie von „apex-cloud.com“, „maxapex.com“ oder „revion.com“ infrage, bei denen Backups und Wartung in fremde Hände gegeben werden.

Arbeiten Server-Admins und DBAs in der Firma, so könnte ein günstiger CentOS-Server als Alternative interessant sein.

Auch bei einer EC2-Instanz von AWS müssen Installation und Wartung der Oracle-Datenbank sowie von ORDS und Apex von den eigenen Mitarbeitern oder von Partnern durchgeführt werden. Lediglich die Wartung und Überwachung des Servers übernimmt AWS.

Dafür bietet AWS jede Menge Optionen zum Thema „Skalierbarkeit“. Eine RDS-Instanz von AWS ist aufgrund der vielen Einschränkungen dagegen nicht zu empfehlen. Möchte man sich jedoch rein auf die Entwicklung in einer produktiven und skalierbaren Apex-5-Umgebung konzentrieren, so sind die von Oracle [6] angebotenen Cloud-Services empfehlenswert.

Weiterführende Links

- [1] https://www.youtube.com/watch?v=fh_AeFMjFmo (02:54)
- [2] apex-cloud.com, maxapex.com, revion.com
- [3] <http://linux.oracle.com/switch/centos/>
- [4] <http://www.oraopensource.com/oxar/>
- [5] <http://calculator.s3.amazonaws.com/index.html>
- [6] <http://cloud.oracle.com> und <https://apex.oracle.com/pls/apex/germancommunities/apexcommunity/tipp/4541/index.html>



Till Albert
till.albert@mt-ag.com



Niels de Bruijn
niels.de.bruijn@mt-ag.com

Datenbank-Services für Entwickler aus der Oracle Cloud: Database Service, Schema Service & mehr



Carsten Czarski, ORACLE Deutschland B.V. & Co. KG

Ob im Unternehmen, auf Konferenzen oder im Web 2.0: Cloud Computing wird überall diskutiert. Auch die Oracle-Datenbank steht seit einiger Zeit als Service in der Oracle Cloud zur Verfügung.

Für Apex- und PL/SQL-Entwickler ist die Oracle Cloud eine sehr interessante Möglichkeit, schnell und einfach an eine laufende Datenbank oder einen laufenden Apex-Workspace zu kommen. Die Cloud kann dies in der Regel innerhalb kurzer Zeit bereitstellen. Vergleicht man dies mit dem Alltag im Unternehmen, wo für eine Datenbank oder ein Schema nicht selten Tage oder Wochen vergehen, so lässt sich erahnen, wie flexibel und schnell man mit der Cloud werden kann.

Im Rahmen des Platform-as-a-Service-Angebots (PaaS) der Oracle Cloud kommt die Datenbank als Service in unterschiedlichen Ausprägungen vor. So bietet Oracle den Database Schema Service, Database as a Service und einen Exadata Service an. Dieser Artikel fokussiert sich auf Database Schema Service und Database as a Service. *Abbildung 1* zeigt den Unterschied.

Der wichtigste Unterschied wird schon aus dem Namen des Angebots deutlich: Während man mit dem Schema Service im Wesentlichen ein Datenbank-Schema (und vor allem einen Apex-Workspace) abonniert, bekommt man bei Database as a Service eine ganze Datenbank. Folgerichtig fallen beim Schema Service so gut wie keine administrativen Arbeiten mehr an: Um Installation, Betrieb, Backup und Recovery der Datenbank und des Apex-Webservers kümmert sich die Oracle Cloud.

Abonniert man die ganze Datenbank, so fallen logischerweise auch administrative Aufgaben wie Benutzer- und Rechteverwaltung an; dafür hat man mehr Freiheiten und kann auch die ganze Datenbank voll ausnutzen. In allen Fällen wird der grundlegende Betrieb von der Oracle Cloud übernommen.

Database Schema Service

Mit dem Database Schema Service steht das Arbeiten mit Apex im Mittelpunkt des Angebots; nach Anmeldung an der Oracle Cloud bekommt man eine Service-Übersicht und von dieser aus gelangt man direkt und ohne Umwege zu

seinem Apex-Workspace (*siehe Abbildung 2*).

Was die Arbeit mit Apex angeht, so fühlt sich der Database Schema Service genauso an wie jede andere Apex-Umgebung (*siehe Abbildung 3*). Im Application Builder erstellt man Apex-Anwendungen, der SQL Workshop erlaubt das Arbeiten mit Datenbank-Objekten wie Tabellen, Views oder PL/SQL-Prozeduren und Team Development bietet einfache Werkzeuge zur Koordination kleinerer Entwicklungsprojekte an. Auch alle Packaged Applications stehen zur Verfügung; der Database Schema Service lässt sich also auch nutzen, um Packaged Applications wie den Survey Builder, das Bug Tracking, den Application Standards Tracker oder andere bereitzustellen.

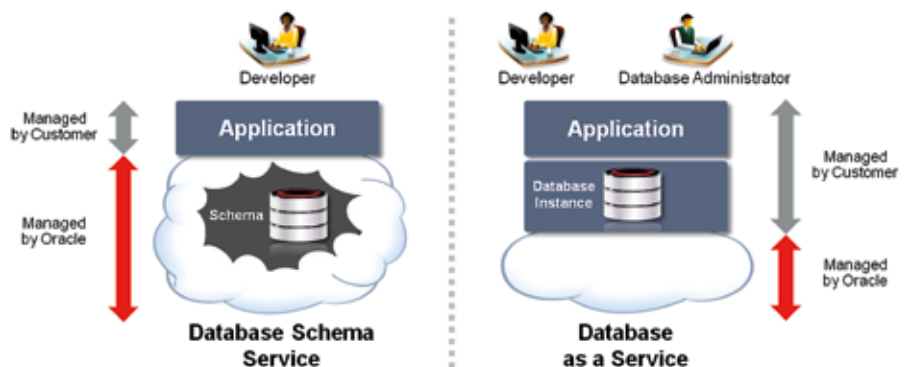


Abbildung 1: Database Schema Service vs. Database as a Service

Die Apex-Umgebung selbst, bestehend aus Betriebssystem, Datenbank und Webserver, wird von der Oracle Cloud bereitgestellt und betrieben. Neben Apex-Anwendungen können auch

REST Services sehr einfach aus dem Database Schema Service heraus bereitgestellt werden – die Definition erfolgt im Apex SQL Workshop (siehe Abbildung 4 und 5).

Zugriff auf den Dienst kann man mit Application Express, dem SQL Developer oder mit REST Webservices nehmen. SQL*Net ist abgeschaltet; Zugriffe mit SQL*Plus, der Oracle Data Pump oder anderen Client-

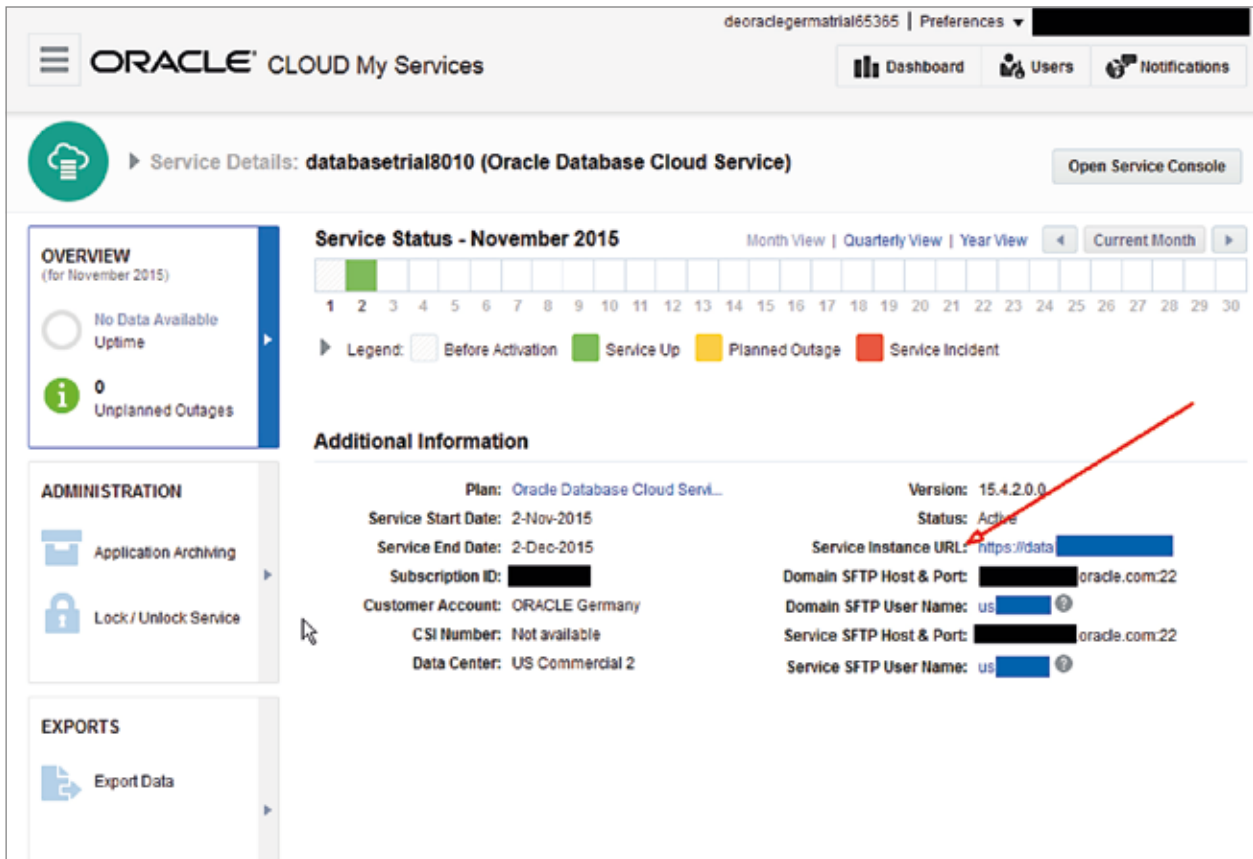


Abbildung 2: Übersichtsseite zum Database Schema Service

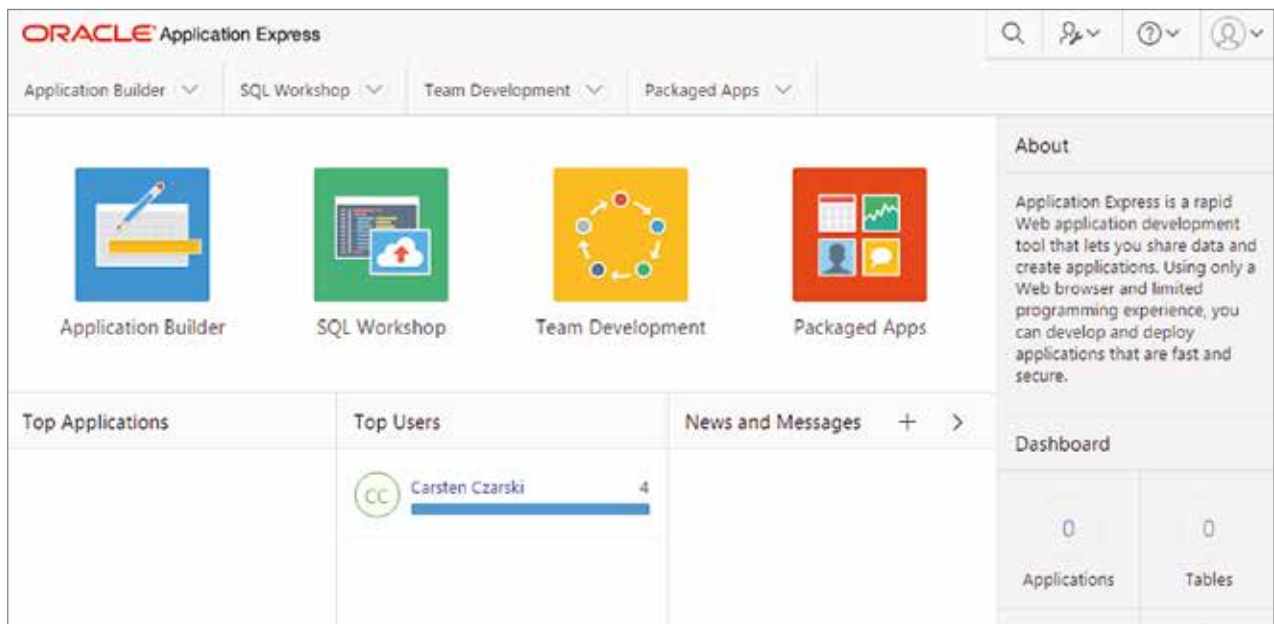


Abbildung 3: Apex-Workspace-Homepage im Database Schema Service

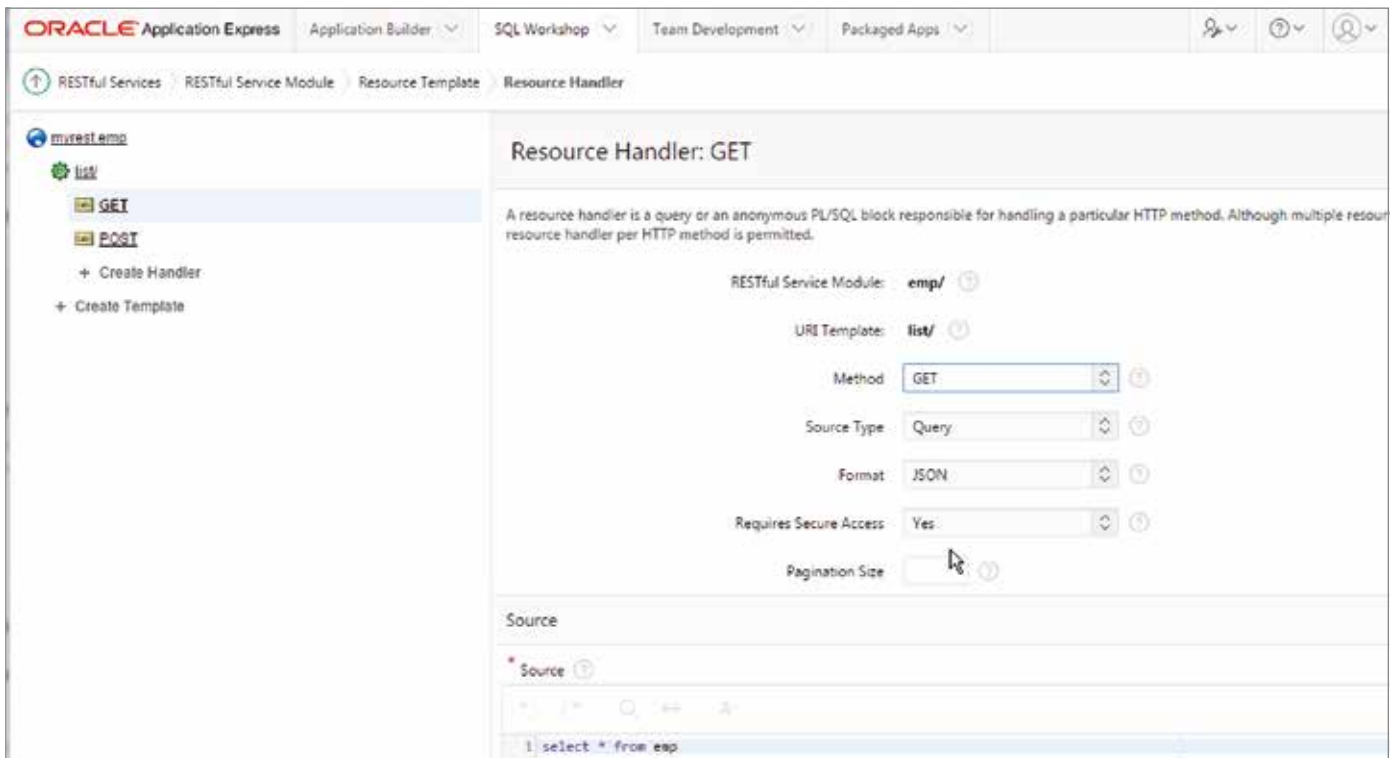


Abbildung 4: Definition eines REST Service im Oracle Database Schema Service

Werkzeugen sind bei diesem Service also nicht möglich. Auch können native Treiber wie JDBC, ODP.NET oder node-oracledb (für Node.js) mit dem Database Schema Service nicht verwendet werden.

Der Schema Service bietet nicht alle Funktionen der Oracle-Datenbank an; die Dokumentation enthält eine Liste der nicht unterstützten Datenbank-Features. Dazu zählen Oracle Spatial, Oracle

Text, Java in der Datenbank und andere. Werden diese benötigt, wird der nachfolgend beschriebene Database Service interessant.

Der Database Schema Service wird monatlich abonniert und abgerechnet. Der Preis richtet sich nach der Größe des abonnierten Datenbankschemas. Oracle bietet hier die Wahl zwischen 5, 20 und 50 GB.

Ganze Datenbanken aus der Cloud: Database as a Service

Nutzt man Database as a Service, abonniert man eine ganze Datenbank. Im Gegensatz zum Database Schema Service kann man darin so viele Schemata anlegen, wie man möchte. Man bekommt nicht nur einen DBA-, sondern auch einen Betriebssystem-Zugriff und hat so wirklich vollen Zugriff auf den (virtuellen) Server.

Natürlich sind auch der Betrieb von Apex oder das Bereitstellen von REST Services in einer solchen Datenbank möglich – das fühlt sich dann genauso an wie auf dem heimischen Datenbankserver: Man administriert Datenbank-

```
{
  - next: {
    $ref: "http://scccloud001:8080/ords/pdb01/doagnews/emp/list/?page=1"
  },
  - items: [
    - {
      empno: 7369,
      ename: "SMITH",
      job: "CLERK",
      mgr: 7902,
      hiredate: "1980-12-16T23:00:00Z",
      sal: 800,
      deptno: 20
    },
    - {
      empno: 7499,
      ename: "ALLEN",
      job: "SALESMAN",
      mgr: 7698,
      hiredate: "1981-02-19T23:00:00Z",
      sal: 1600,
      comm: 300,
      deptno: 30
    },
    - {
      empno: 7521,
      ename: "WARD",

```

Abbildung 5: Ausführen eines REST Service

User, Tablespaces, die Apex-Umgebung und die Apex-Workspaces selbst.

Neben der reinen Software und der virtuellen Maschine als Ablauf-Umgebung bietet Database as a Service zusätzliche Werkzeuge an: So kann die Cloud-Instanz direkt aus der Oracle-Cloud-Web-Oberfläche erstellt, gestartet, gestoppt oder gelöscht werden. Eine solche Datenbank enthält ebenfalls die fertig eingerichteten Oracle REST Data Services (ORDS) als Apex-Webserver. Auch Backups, Recovery und das Einspielen wichtiger Patches können über die Weboberfläche erfolgen (siehe Abbildung 7).

Diese Werkzeuge („Cloud Tools“) machen den Umgang mit der Cloud-Datenbank wesentlich einfacher, denn ein Unix-Login und der Aufruf von Executables sind für diese Aufgaben dann schlicht nicht mehr nötig – der Browser reicht aus. Wer diese Werkzeuge dennoch nicht braucht oder nicht möchte, kann Database as a Service auch als „Virtual Image“ abonnieren. Dies ist tatsächlich nichts weiter als ein virtueller Server mit vorinstallierter Oracle-Software. Zum Erstellen einer konkre-

ten Datenbank oder zum Einspielen eines Patches loggt man sich (wie auf dem heimischen Server) am Betriebssystem ein und führt die Operationen wie gewohnt durch.

Database as a Service kann stundenweise („metered Service“) oder monatlich („non-metered Service“) abonniert werden. Abonniert man ihn als Virtual Image, so ist er etwas preisgünstiger als mit den erwähnten Cloud-Werkzeugen. Die Datenbank selbst wird in vier Ausprägungen angeboten:

- Standard Edition Service
- Enterprise Edition Service
- High Performance Service
- Extreme Performance Service

Bei den Services „Standard Edition“ und „Enterprise Edition“ sagt der Name bereits, welche Datenbank-Edition enthalten ist. Im High Performance Service ist eine Reihe von Datenbank-Optionen enthalten – unter anderem sind dies Advanced Compression, Advanced Security, Diagnostic und Tuning Pack und

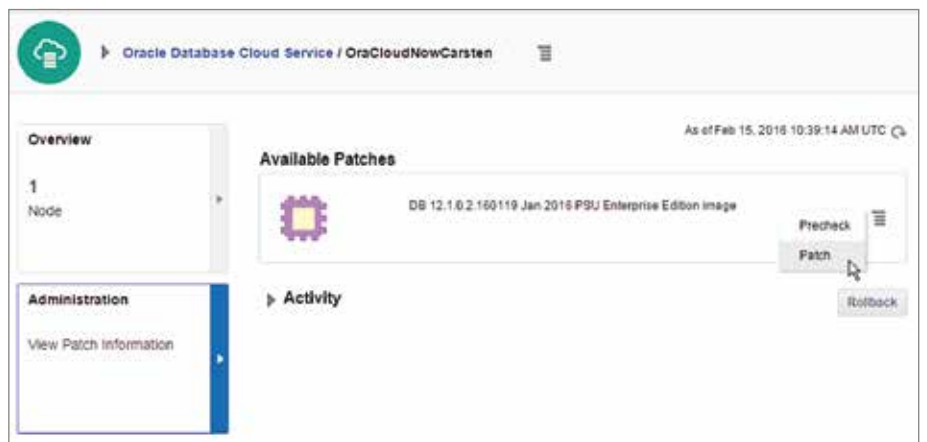


Abbildung 7: Einspielen eines Patches

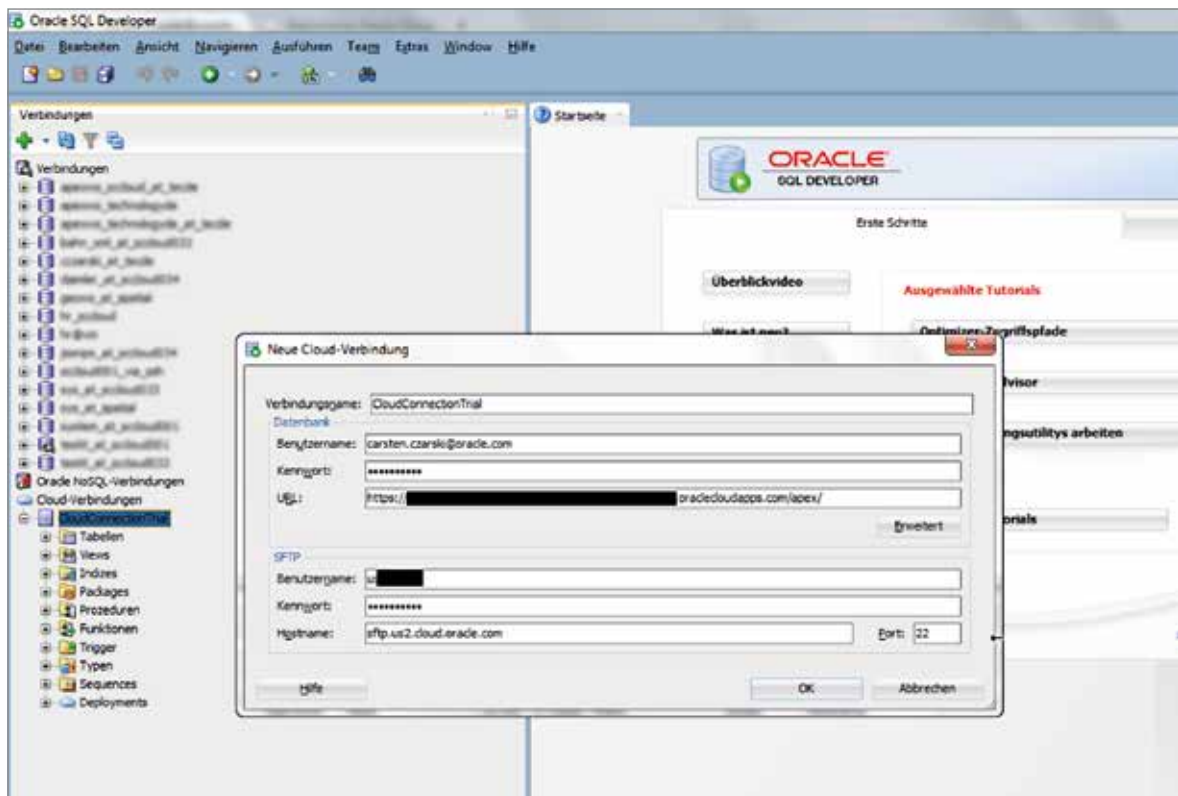


Abbildung 6: Mit dem Oracle SQL Developer auf den Schema Service zugreifen – Cloud Connections

einiges mehr. Die Variante „Extreme Performance“ bietet darüber hinaus die In-Memory Database und Active Data Guard an.

Arbeiten mit Database as a Service

Als Apex- beziehungsweise PL/SQL-Entwickler arbeitet man mit Database as a Service letztlich wie mit einer Datenbank im eigenen Rechenzentrum. Am auffälligsten ist sicherlich die Tatsache, dass die Datenbank nun über das Internet erreichbar ist. Aus Sicherheitsgründen sind per Default alle Netzwerk-Zugriffe bis auf den SSH-Dienst gesperrt. Diese Firewall-Einstellungen sind konfigurierbar, sodass man einzelne Ports durchaus freigeben kann – speziell für eine Apex-Umgebung ist es wichtig, die HTTP- beziehungsweise HTTPS-Ports freizugeben, damit Apex für den Browser erreichbar wird.

Das „SQL*Net“-Protokoll sollte nur dann freigegeben werden, wenn man es mit SSL verschlüsselt, was SQL*Net ja bereits seit Jahren anbietet. Ohne SSL-Verschlüsselung ist es besser, die „SQL*Net“-Verbindung durch das SSH-Protokoll zu tunneln. Im SQL Developer, aber auch im Kommandozeilen-Utility „sqlcl“, ist diese Unterstützung bereits fest eingebaut (siehe Abbildung 8). Oft ist dieser SSH-Tunnel ohnehin nötig, um die Oracle Cloud überhaupt durch die Unternehmens-Firewall erreichen zu können.

Wie bereits erwähnt, ist die Cloud-Datenbank mit einer aktuellen Apex-Version ausgestattet – zur Zeit der Drucklegung dieses Artikels ist das Apex 5.0. Apex-Anwendungen, REST Services, aber auch alle anderen Formen datenbankgestützter Anwendungen, lassen sich aus der Cloud-Datenbank genauso betreiben wie aus einer On-Premise-Datenbank im eigenen Rechenzentrum. Da die ganze Datenbank abonniert wurde und auf einem virtuellen Linux-Server bereitsteht, hat man volle Kontrolle über seinen Datenbankserver.

Fazit

Mit Database Schema Service und Database as a Service bietet Oracle seine Datenbank in unterschiedlichen Ausprägun-

gen in der Cloud an. Der Schema Service richtet sich vor allem an Apex-Entwickler, die eine unkomplizierte Umgebung zum Erstellen oder zum Betrieb von Apex-Anwendungen suchen. Mit der Administration der Datenbank oder des Webservers wird man nicht behelligt; nach dem Login geht es direkt zum Apex-Workspace, in dem man mit seiner Arbeit beginnen kann. Allerdings sind die Möglichkeiten eingeschränkt: So steht kein „SQL*Net“-Zugang zur Verfügung, mit Oracle-Client-Werkzeugen kann man also nicht oder nur eingeschränkt arbeiten.

Anders sieht das bei Database as a Service aus: Hier abonniert man eine ganze Datenbank, bekommt Zugriff auf einen virtuellen Linux-Server und kann diesen genauso bedienen wie den Server im heimischen Rechenzentrum. Man ist selbst DBA, kann so alle Funktionen der Datenbank nutzen und diese so gestalten, wie man möchte. Je nachdem, welche Datenbankfunktionen man benötigt, lässt sich der Service als Standard Edition, Enterprise Edition, High Performance oder Extreme Performance buchen. Zusätzlich ist man frei in der Entscheidung darüber, ob man den Service stündlich oder monatlich abrechnen möchte.

Weitere Informationen

- Oracle Cloud: <https://cloud.oracle.com/>
- Oracle Cloud Database Services: https://cloud.oracle.com/de_DE/database
- Oracle Cloud Database Services – FAQ: https://cloud.oracle.com/de_DE/database?mResID=1385151585800&tabID=1385151586115
- YouTube-Videoserie „Oracle Cloud Now“: <http://tinyurl.com/oracloudnow>



Carsten Czarski
carsten.czarski@oracle.com
<http://sql-plsql-de.blogspot.com>



Exzellente Baupläne für die Digitale Ökonomie!

Dafür steht PROMATIS als Geschäftsprozess-Spezialist mit mehr als 20 Jahren Erfahrung im Markt. Gepaart mit profundem Oracle Know-how schaffen wir für unsere Kunden die Digitale Transformation:

- Oracle SaaS für ERP, SCM, EPM, CX, HCM
- Oracle E-Business Suite und Hyperion
- Oracle Fusion Middleware (PaaS)
- Internet of Things und Industrie 4.0

Vertrauen Sie unserer Expertise als einer der erfahrensten Oracle Platinum Partner – ausgezeichnet mit dem Oracle Partner Excellence Award 2015.

PROMATIS



PROMATIS Gruppe
Tel.: +49 7243 2179-0
www.promatis.de · info@promatis.de
Ettlingen/Baden · Hamburg
Graz · Zürich

Der Apex Session State

Jürgen Sieben

Der Session State ist eine zentrale Komponente jeder Web-Anwendung. Nur durch ihn ist es möglich, seitenübergreifend Informationen für eine Anwendung bereitzustellen.

HTTP ist bekanntermaßen ein zustandsloses Protokoll, sodass nur durch einen externen Session State in Verbindung mit einer Session-ID, die durch die Anwendung verwaltet wird, dieses Protokoll überhaupt zur Anwendungsentwicklung genutzt werden kann. Apex implementiert den Session State - naheliegenderweise als Tabelle des Benutzers „Apex_nnnnnn“. Der Zugriff auf den Session State ist daher leicht aus SQL und PL/SQL möglich. Dennoch sind der aktuelle Zustand des Session State und die verschiedenen Möglichkeiten des Zugriffs auf ihn ein Thema, das Einsteiger in Apex regelmäßig zur Verzweiflung bringt. Der Artikel bringt Licht ins Dunkel.

Der Session State ist ganz trivial eine Tabelle im Schema „Apex_nnnnnn“ mit

dem Namen „WWV_FLOW_DATA“. Darin wird für jede Session (die man an ihrer Session-ID erkennt und die in dieser Tabelle „FLOW_INSTANCE“ heißt) und für jedes Element der Anwendung der aktuell gültige Wert hinterlegt. Eine Ausnahme bilden die Kollektionen, die innerhalb von Apex verwendet werden, um zum Beispiel Tabular Forms im Session State zu halten. Diese sind in der Tabelle „WWV_FLOW_COLLECTIONS\$“ gespeichert.

Wozu benötigen wir den Session State eigentlich? Denn wenn der Session State letztlich eine einfache Tabelle ist und die Daten der Anwendung ja ohnehin aus Tabellen des Workspace-Schemas kommen, wozu sind diese dann in einer Tabellenlandschaft von Apex gespeichert und nicht direkt in den Tabellen, aus denen sie

kommen? Der Grund liegt in der Implementierung der Transaktionen in Apex.

Ein Seitenaufruf (ohne Zwischenwege wie dynamische Aktionen etc.) besteht aus einer Anfrage und einer Antwort, also „Request“ und „Response“. Interessant ist hier insbesondere der Request, also die Anfrage eines Benutzers, denn in dieser Anfrage sind häufig Daten enthalten, die in der Datenbank gespeichert werden müssen.

Ein Request ist immer eine Transaktion, sie wird also im Erfolgsfall mit „commit“ abgeschlossen. Dies muss so sein, denn Apex verfügt über einen Connection Pool (genauer: ORDS verfügt über einen solchen Pool), sodass nicht sichergestellt ist, dass ein in der gleichen Apex-Anwendung folgender Request des gleichen Benutzers über die gleiche Da-

The screenshot shows two side-by-side windows. The left window is an Oracle APEX application page titled 'Mitarbeiter bearbeiten' (Employee Edit). It contains a form with fields for 'Emp Id', 'Emp First Name', 'Emp Last Name', 'Emp Email', 'Emp Phone Number', 'Emp Hire Date', 'Emp Job Id', 'Emp Salary', 'Emp Commission Pct', 'Emp Emp Id', and 'Emp Dep Id'. The right window is a browser window titled 'Session State - Mozilla Firefox' showing the Oracle APEX Session State table. The table has columns: Application, Page, Item Name, Display, Item Value, Status, and Encrypted. The table contains one row of data.

Application	Page	Item Name	Display	Item Value	Status	Encrypted
105	5	PS_ROWID	Hidden	AAAW5IQAKAAAADOAAG	Inserted	No

Abbildung 1: Der Session State ist leer

tenbank-Verbindung abgewickelt wird. Eine Transaktion kann also technisch nur in einem Request vereinbart werden und nicht mehrere Requests umfassen.

Ist auf der Seite, die den Request stellt, ein Formular enthalten (bei Apex ist das immer so), werden alle Formular-Elemente der Seite an Apex gesendet. Der Prozess der Verarbeitung dieses Request ist komplex und umfasst eine Reihe von Einzelschritten, die jeder für sich auch scheitern könnten. Ist dies irgendwann der Fall, muss Apex in der Lage sein, die Seite mit den Eingaben des Benutzers erneut aufzubauen, um ihm die Möglichkeit zur Fehlerkorrektur zu geben. Natürlich sollten diese fehlerhaften Daten sich dann noch nicht in den eigentlichen Workspace-Tabellen befinden (je nach Fehler können die Daten auch gar nicht in die Tabellen eingefügt werden, weil zum Beispiel ein „Not Null“-Constraint verletzt

sein könnte), sondern irgendwo anders, und das ist eben im Session State.

Ein weiteres Anwendungsszenario ist ein Assistent, der über mehrere Seiten hinweg Daten zusammenträgt, um abschließend eine oder mehrere Tabellen des Workspace-Schemas damit zu füllen. Jede einzelne Seite des Assistenten ist ein abgeschlossener Request-Response-Zyklus, der alle eingegebenen Daten speichern, die Ursprungstabellen aber nicht ändern soll, bis schließlich der Assistent als Ganzes bestätigt oder auch verworfen wird. Auch dies wäre ohne einen generischen Session State nicht möglich. Letztlich gibt es auch Daten, die nicht in Tabellen des Workspace-Schemas gespeichert, aber in der Apex-Anwendung benötigt werden, etwa der Name des angemeldeten Benutzers oder andere Angaben. Auch hierfür steht der generische Session State zur Verfügung.

Den Session State beschreiben

Der Session State in der Datenbank kann (und wird auch sehr häufig) andere Daten enthalten als die Apex-Seite im Browser. Dies ist zum Beispiel immer der Fall, wenn man ein normales Formular für eine Tabellenzeile einrichtet. Auf der Apex-Seite werden alle Daten einer Tabellenzeile in Eingabefeldern dargestellt, der Session State zeigt aber, dass alle Elemente dieser Seite leer sind, vielleicht mit Ausnahme einer Primärschlüsselspalte, die benutzt wird, um der Seite mitzuteilen, welche Zeile der Tabelle dargestellt werden soll. *Abbildung 1* zeigt, dass auf der Seite Daten zu sehen sind, der Session State aber leer ist.

Ist das nicht widersinnig? Wenn man eine Formularseite anzeigt, um eine Tabellenzeile zu editieren, müssten dann nicht initial im Session State und auf der

Alles, was die SAP-COMMUNITY wissen muss, finden Sie monatlich im E-3 MAGAZIN.
Ihr WISSENSVORSPRUNG im Web, auf iOS und Android sowie PDF und Print:
e-3.de/abo

Wer nichts weiß,
muss alles glauben!

Marie von Ebner-Eschenbach



SAP® ist eine eingetragene Marke der SAP AG in Deutschland und in den anderen Ländern weltweit.

www.e-3.de

0.00000	0.00000	...Session State: Save "P5_ROWID" - saving same value: "AAANbQAAKAAAAD00AAG"	4
0.00000	0.00000	...Session State: Saved Item "P5_EMP_ID" New Value="106"	4
0.00000	0.00000	...Session State: Saved Item "P5_EMP_FIRST_NAME" New Value="Valli"	4
0.00000	0.00000	...Session State: Saved Item "P5_EMP_LAST_NAME" New Value="Pataballa"	4
0.00000	0.00000	...Session State: Saved Item "P5_EMP_EMAIL" New Value="VPATABAL"	4
0.00000	0.00000	...Session State: Saved Item "P5_EMP_PHONE_NUMBER" New Value="590.423.4560"	4
0.00000	0.00000	...Session State: Saved Item "P5_EMP_HIRE_DATE" New Value="05.02.06"	4
0.00000	0.00000	...Session State: Saved Item "P5_EMP_JOB_ID" New Value="IT_PROG"	4
0.00000	0.00000	...Session State: Saved Item "P5_EMP_SALARY" New Value="4900"	4
0.00000	0.00000	...Session State: Saved Item "P5_EMP_COMMISSION_PCT" New Value=""	4
0.00000	0.00000	...Session State: Saved Item "P5_EMP_EMP_ID" New Value="103"	4
0.00000	0.00000	...Session State: Saved Item "P5_EMP_DEP_ID" New Value="60"	4
0.00000	0.00000	Processes - point: ON_SUBMIT_BEFORE_COMPUTATION	4

Abbildung 2: Der Ablauf

Seite die gleichen Werte enthalten sein? Verständlich wäre eine Diskrepanz, wenn ein Feld auf der Seite verändert würde; wenn das jedoch nicht geschieht, warum sind dann Session State und Anwendungsseite nicht synchron?

Die Antwort lautet, dass Apex den Session State in diesem Fall nicht pflegen muss. Apex könnte dies tun, hätte aber nichts davon, denn die Daten werden aller Wahrscheinlichkeit nach durch das Formular verändert, daher ist der Session State sowieso sehr bald nicht mehr synchron. Und wenn die Seite abgeschickt wird, müssen die Eingaben des Benutzers ohnehin in den Session State übernommen werden; die vorher geleistete Arbeit wäre vergeblich gewesen. Daher wird beim normalen Darstellen einer Seite (im Response-Zyklus) der aktuelle Wert eines Formularfeldes aus der Workspace-Tabelle gelesen und im HTML der Seite integriert, aber nicht in den Session State kopiert. Eine Ausnahme ist, wenn die Daten eines Eingabefeldes über eine Computation in das Element geschrieben werden.

Umgekehrt besteht die – so ziemlich – erste Aktivität beim Übermitteln einer Seite an Apex darin, die eingegebenen Werte der Formular-Elemente in den Session State zu kopieren. Man kann das beim Debuggen der Verarbeitung einer Seite sehen: Zuerst werden die „National Language Support“-Settings (NLS) gesetzt, anschließend die Session-ID auf Gültigkeit geprüft, und dann geht es los: Alle Elemente der Seite werden in den Session State kopiert. Die meisten Werte werden neu gesetzt (die Meldung lautet: „Saved Item "P5_EMP_ID" New Va-

lue="105" anstatt „saving same value: "...“). Weil erst nach dieser Aktion die Computations, Validations, Processes und Branches ausgeführt werden, die für diese Seite definiert wurden, können diese auf dem Server auf den aktuellen Stand der Eingabefelder über den Session State zugreifen (siehe Abbildung 2).

Probleme mit dem Session State

Es gehört zu den großen Vorteilen von Apex, dass die Interna der Session-State-Verwaltung nur ganz selten direkt programmiert werden müssen. Ein prominentes Beispiel, das einen in direkten Kontakt damit bringt, ist die Verwendung des Collection-API, das man zum Beispiel verwendet, um innerhalb eines Assistenten ein Tabular-Form einzusetzen. Das ist schon etwas fortgeschrittenes Arbeiten mit Apex, kommt aber natürlich vor.

Unabhängig davon sind Probleme in der Entwicklung von Apex-Seiten, die durch den Session State verursacht werden, allerdings an der Tagesordnung und normalerweise darin begründet, dass man nicht durchblickt, wann welche Daten im Session State enthalten sind und wann nicht.

Diese Fehler tauchen auf, wenn man den einfachen Request-Response-Pfad, der für Apex typisch ist, verlässt, zum Beispiel wenn ein Partial Refresh auf einen Bericht oder eine Region ausgeführt werden soll. Das Problem: Nach dem Response der Datenbank zeigt der Browser die aktuelle Seite. Für den Server ist „nach dem Response“ nicht „vor dem Request“, denn der Server erwartet nichts vom Client, sondern reagiert erst, wenn der nächste Request kommt. Nur im Zuge eines Request schreibt Apex automatisiert die Werte der Oberfläche in den Session State. Ansonsten hält der Session State still und enthält, je nachdem, entweder

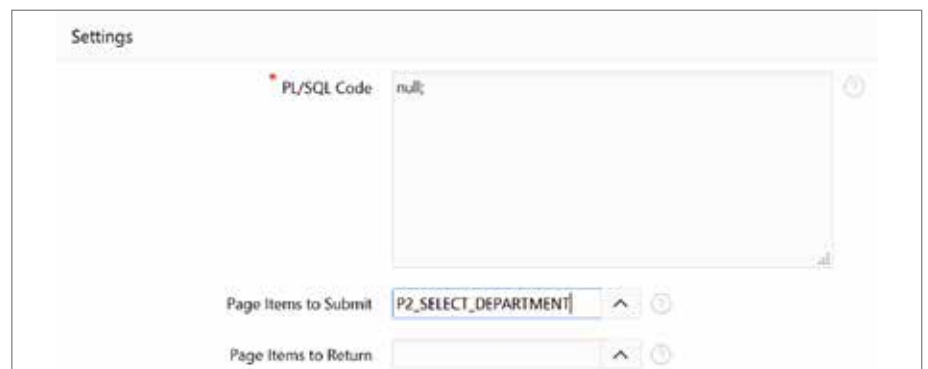


Abbildung 3: PL/SQL-Code „NULL;“

den initial gültigen Wert des Zeitpunkts, an dem die Seite ausgeliefert wurde, oder einfach „NULL“. Ein Partial Refresh durchbricht nun das Request-Response-Spiel, denn es wird nicht die Seite als Ganzes abgeschickt, sondern lediglich ein AJAX-Request abgesetzt. Dieser Request ist eine einfache Anfrage an den Server (die normalerweise nicht die Elementwerte auf der Seite umfasst), die vom Server auch beantwortet wird, ohne dass der Browser die Antwort als neue Seite aufbaut. Die Antwort wird vom AJAX-Request entgegengenommen und dynamisch in die bestehende Seite integriert. So lässt sich also zum Beispiel ein Bericht einer Seite aktualisieren, ohne alle anderen Seitenelemente erneut vom Server anfordern und durch den Browser darstellen lassen zu müssen.

Ein Szenario

Problematisch ist es, wenn ein Bericht auf einer SQL-Abfrage beruht, die durch einen Wert des Session State gefiltert wird. Man stelle sich eine Seite mit einer Auswahlliste und einen zugehörigen Bericht vor. Die Auswahlliste wählt die Abteilung, für die ein Bericht angezeigt werden soll. Als die Seite geladen wurde, war noch keine Abteilungsauswahl vorhanden; der Elementwert der Auswahlliste ist im Session State „NULL“. Damit durch die Auswahl einer Abteilung nicht die gesamte Seite neu geladen wird, wurde festgelegt, dass im Fall einer Änderung der Auswahlliste nichts weiter geschehen soll, damit man durch eine dynamische Aktion nur den Bericht aktualisieren kann. Diese dynamische Aktion ist als Refresh-Aktion auf die Berichtsregion angelegt, die immer dann feuert, wenn sich die Auswahlliste ändert.

Nun wird ein Eintrag der Auswahlliste gewählt. Dies geschieht im Browser, der Server erfährt davon nichts – der Session State bleibt daher unverändert. Der Bericht wird durch die dynamische Aktion aktualisiert, zeigt jedoch keine Daten. Im Licht des bisher Erklärten wird klar, warum: Der Session State enthält für das angegebene Element immer noch keinen Wert, die Filterung über einen fehlenden Wert hat zur Folge, dass eben auch keine Ergebnisse gezeigt werden können.

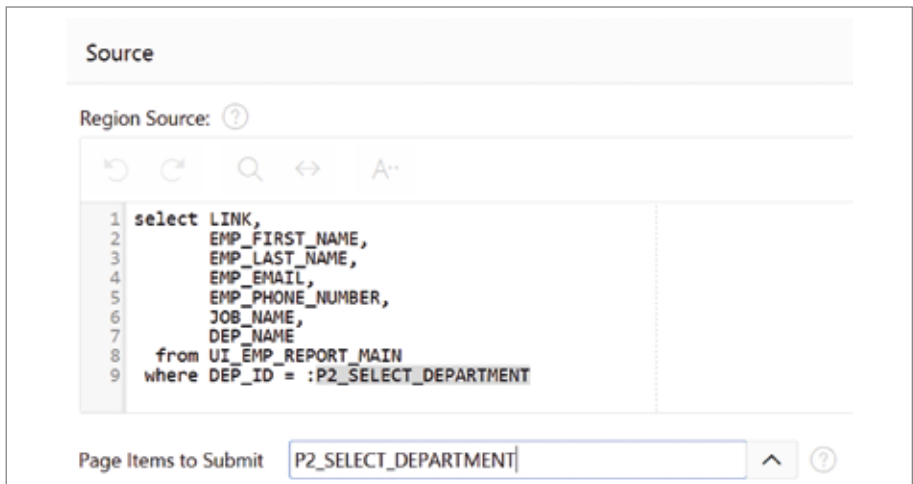


Abbildung 4: Eintrag des Seitenelements

Lösung des Problems

Eine erste Lösung des Problems besteht darin, der dynamischen Aktion eine zweite Aktion vom Typ „PL/SQL-Code ausführen“ hinzuzufügen. Der Prozess selbst soll in der Datenbank gar nichts tun, daher ist der Code, der ausgeführt wird, schlicht „NULL;“, aber der Dialog bietet die Möglichkeit, Elementwerte der Seite in den Session State zu übernehmen oder aus diesem zu lesen. Daher wird hier der Name der Auswahlliste in die Liste der Elementnamen, die an die Datenbank übermittelt werden sollen, eingetragen (siehe Abbildung 3).

Es gilt jetzt noch darauf zu achten, dass diese zweite dynamische Aktion vor der Refresh-Aktion des Berichts ausgeführt wird (über die Sequenz-Nummer der Aktivitäten), sonst ist der Session State zu spät aktualisiert worden, als dass die SQL-Abfrage auf den Wert hätte zugreifen können.

Einfacher ist es, in den Einstellungen der Region einzutragen, welche Seitenelemente im Fall eines Refresh der Region übermittelt werden sollen. Diese Option verbirgt sich unterhalb der „select“-Anweisung für den Bericht (siehe Abbildung 4).

Regionstypen, die keine dynamischen Aktualisierungen unterstützen, verfügen auch nicht über ein entsprechendes Eingabefeld. In diesem Fall werden die hier eingetragenen Seitenelemente automatisiert vor dem Aktualisieren des Berichts im Session State eingetragen, eine zusätzliche PL/SQL-Aktion ist daher nicht

nötig. Das Problem lässt sich also dadurch lösen, dass dem Session State ein Wert übergeben wird, bevor eine Aktivität in der Datenbank ausgeführt wird.

Diese Problemlösung kann als Blaupause für ähnliche Probleme verwendet werden, beinahe immer ist die fehlende Harmonisierung des Session State Grund für unvorhergesehenes Verhalten von Apex bei dynamischen Aktivitäten. Nun gibt es den Einwand, dass diese Implementierung sehr unschön sei, sie zwei AJAX-Requests erfordere und immer auf diese Weise programmiert werden müsse. Das stimmt, allerdings mag sich dies in einer neuen Version von Apex verbessern (hoffen wir auf Version 5.1) und zum anderen ändert es nichts daran: Wesentlich ist zu verstehen, worin das Problem besteht. Die eleganteste Lösungsstrategie kann sich ändern, aber das Problem muss gelöst werden.

Zugriffsvarianten

Um auf die Werte des Session State zuzugreifen, könnte man theoretisch SQL einsetzen, allerdings ist dieser Weg nicht möglich, weil Sicherheitsfeatures von Apex dies nicht zulassen. Stattdessen bietet Apex verschiedene Möglichkeiten an, auf den aktuellen Session State zuzugreifen:

- Über eine Bindevariable mit dem gleichen Namen des Elements
- Über eine Funktion mit dem Namen „V“, der im gleichen Schema definiert ist und öffentlich über ein gleichnamiges Alias genutzt werden kann

Natürlich liefern alle Zugriffspfade das gleiche Ergebnis und alle liefern nur im Kontext einer Apex-Session überhaupt einen Wert. Ist ein Wert nicht vorhanden, wird übrigens kein Fehler, sondern lediglich „NULL“ geliefert, was einen dazu zwingt, akribisch auf korrekte Schreibweise zu achten. Hier könnte der Hinweis auf den Advisor in den Utilities hilfreich sein, denn dort werden Zugriffe auf Seitenelemente erkannt, die nicht existieren.

Zugriff auf die Bindevariablen

Gemäß den beiden oben geschilderten Zugriffspfaden stehen mehrere syntaktische Formen zur Verfügung, um den Session State abzufragen. Zunächst einmal lässt sich überall dort, wo PL/SQL eingesetzt wird, die Bindevariable über die Notation „:P1_ITEM“ ansprechen. Man kennt dieses Verfahren von Zeilen-Triggern, die ähnliche Umgebungsvariablen unter den Namen „NEW“ und „OLD“ anbieten. Immer dann, wenn der PL/SQL-Code selbst eine Variable nicht deklariert, sondern aus der Umgebung zur Verfügung gestellt bekommt, ist der vorangestellte Doppelpunkt zu verwenden.

Wird kein PL/SQL-Code ausgeführt, kann der Wert der Bindevariablen mit der Schreibweise „&P1_ITEM.“ angesprochen werden. Wichtig ist der abschließende Punkt hinter dem Namen des Elements. Er ist erforderlich, weil Apex dadurch erfährt, wo eine Ersetzungszeichenfolge endet. Nicht immer ist es möglich, das Ende durch ein Leerzeichen anzuzeigen, daher muss der Punkt diese Aufgabe übernehmen.

Auch diese Variante ist außerhalb von Apex bekannt, denn auf diese Weise wird in SQL*Plus eine Variable, die dort definiert wurde, in SQL oder PL/SQL als Ersetzungszeichenfolge eingefügt. Dieses Verfahren beschreibt bereits einen wesentlichen Unterschied zwischen den beiden syntaktischen Varianten: Die erste Variante („:P1_ITEM“) gilt als immun gegen SQL-Injection-Angriffe, weil die Variable typsicher ist und kein dynamisches SQL oder PL/SQL erfordert.

Im Gegensatz dazu ist die Verwendung der zweiten Variante einem Einsetzen einer Zeichenkette in eine dyna-

mische SQL-Anweisung gleichzusetzen. Weil hierdurch potenziell die SQL-Anweisung selbst geändert werden kann, gilt dieser Ansatz als unsicher. Da aber nicht überall in Apex PL/SQL im Hintergrund verwendet wird, um eine Funktion auszuführen, kommt man manchmal nicht umhin, die Notation „&P1_ITEM.“ zu verwenden. Diese Schreibweise ist zum Beispiel erforderlich, wenn die Bezeichnung einer Region aus dem Wert eines Seitenelements abgeleitet werden soll, oder bei der Übergabe von Elementwerten als Parameter in Links.

Zugriff über die Funktion „V“

Wer möchte, kann immer dann, wenn PL/SQL im Einsatz ist, auch explizit die Funktion „V“ aufrufen und dadurch den Wert des Session State erfragen. Diese Variante bedeutet zunächst einmal mehr Schreibaufwand, schwerer wiegt jedoch die in Blogs häufig zu findende Empfehlung, diese Schreibweise nicht zu verwenden, weil sie langsam sei. Bevor darauf eingegangen wird, zunächst der Hinweis, dass es Situationen gibt, in denen die beiden anderen Schreibwei-

```
SQL> declare
  2     l_user_name emp.ename%type := 'BLAKE';
  3     l_deptno emp.deptno%type;
  4  begin
  5     select deptno
  6         into l_deptno
  7         from emp
  8         where ename = l_user_name;
  9  end;
 10  /
```

PL/SQL-Prozedur erfolgreich abgeschlossen.

Listing 1

```
SQL> select ename, dbms_random.value zufallszahl
  2  from emp;
```

ENAME	ZUFALLSZAHL
SMITH	,689783172
ALLEN	,512884812
WARD	,710363434
...	

14 Zeilen ausgewählt

Listing 2

```
SQL> select ename,
  2         (select dbms_random.value
  3            from dual) zufallszahl
  4  from emp;
```

ENAME	ZUFALLSZAHL
SMITH	,664207243
ALLEN	,664207243
WARD	,664207243
JONES	,664207243
...	

14 Zeilen ausgewählt

Listing 3

sen nicht funktionieren und von daher nur dieser Weg bleibt, um mit dem Session State zu interagieren: in PL/SQL-Packages und Views zum Beispiel, also immer außerhalb der Apex-Seiten selbst. Nun liest man genauso häufig die Emp-

fehlung, kein ausuferndes PL/SQL oder SQL auf den Apex-Seiten selbst zu integrieren, sondern stattdessen diesen Code als Package oder View in die Datenbank auszulagern. Ein Widerspruch also, der nach einer Lösung verlangt.

Zunächst eine Erläuterung des Problems: Warum sollte der Zugriff auf „V“ langsamer sein als der Zugriff auf eine Binde-Variable in der „:“-Notation? Die Implementierung von PL/SQL sorgt dafür, dass Variablen, die in SQL eingesetzt werden, vorab als konstante Zeichenfolgen in die SQL-Anweisung integriert werden, bevor diese ausgeführt wird.

In *Listing 1* wird der Wert „BLAKE“ als Konstante in die „select“-Anweisung integriert und diese erst dann ausgeführt. Dadurch kann SQL effizient arbeiten. Anders ist das, wenn eine PL/SQL-Funktion ins Spiel kommt, wie etwa die Funktion „V“. PL/SQL (und SQL schon gleich gar nicht) kann nicht voraussehen, welches Ergebnis „V“ für einen Parameter liefern wird. Das Ergebnis hängt ja vom Session State ab, der für jede Session und auch innerhalb einer Session über die Zeit verschieden sein kann. Daher kann der Compiler das Ergebnis der Funktion nicht einfach durch sein Ergebnis ersetzen, auch wenn das in diesem konkreten Fall logisch möglich wäre.

In *Listing 2* erwartet man nun für jede Zeile der Ergebnismenge eine neue Zufallszahl. Hätte die Datenbank einfach einmal eine Zufallszahl berechnet und diese dann ausgegeben, wäre der Sinn der Funktion verfehlt. Daher muss also eine Funktion wie „V“ in SQL stets berechnet werden. Diese Berechnung erfolgt aber in PL/SQL, sodass SQL für jede betroffene Zeile einen Umgebungswechsel zwischen SQL und PL/SQL durchführen muss. Schlimmer noch: Die Funktion „V“ ermittelt den Wert des Session State durch eine SQL-Abfrage gegen die Tabelle „WWW_FLOW_DATA“, macht also seinerseits einen Umgebungswechsel nach SQL. Diese Umgebungswechsel sind sehr aufwändig und summieren sich auf erhebliche Werte, wenn sie sehr oft erfolgen.

Die Verwendung der Funktion „V“ in der „where“-Klausel einer SQL-Anweisung ist also fatal, denn dann muss für jede Zeile der Tabelle die Funktion berechnet werden, selbst wenn nur eine Zeile als Ergebnis der Filterung übrig bliebe. Dies ist im Zusammenhang mit der Funktion „V“ besonders ärgerlich, da sie ihren Wert gar nicht ändern kann, denn dazu müsste der Anwender in seiner Session ja den Session State ändern,

```
SQL> with params as
  2         (select dbms_random.value zufallszahl
  3             from dual)
  4 select ename,
  5         zufallszahl
  6         from emp
  7         cross join params;

ENAME          ZUFALLSZAHL
-----
SMITH          ,481366804
ALLEN          ,692801788
WARD           ,740467425
JONES          ,701446446
...
14 Zeilen ausgewählt
```

Listing 4

```
SQL> with params as
  2         (select /*+ outline */
  3             dbms_random.value zufallszahl
  4             from dual)
  5 select ename,
  6         zufallszahl
  7         from emp
  8         cross join params;

ENAME          ZUFALLSZAHL
-----
SMITH          ,212091558
ALLEN          ,672092931
WARD           ,672092931
JONES          ,672092931
...
14 Zeilen ausgewählt
```

Listing 5

```
SQL> declare
  2     cursor emp_cur is
  3         select ename, job, sal
  4             from emp;
  5 begin
  6     for emp in emp_cur loop
  7         dbms_output.put_line(
  8             emp.ename || ' erhaelt die ' ||
  9             v('P1_AUSZEICHNUNG') ||
  10            ' fuer vorbildlichen PL/SQL-Code');
  11     end loop;
  12 end;
  13 /
```

Listing 6

und das kann er während einer „select“-Abfrage nicht.

Problemlösung und Empfehlungen

So lässt sich beides erreichen: Code in die Datenbank verlegen und dennoch eine performante Anwendung erhalten. Die Lösung des Problems, der Umgebungswechsel in SQL, ist immer die gleiche, obschon sie zunächst überraschend aussieht: Man schreibt den Funktionsaufruf in eine skalare Unterabfrage der Form „where ename = (select v('P1_ENAME') from dual)“. Der Effekt lässt sich schön durch den Aufruf der Funktion „dbms_random“ aus dem vorigen Beispiel erkennen (siehe Listing 3).

SQL wird eine skalare Unterabfrage vorab einmal berechnen und das Ergebnis als Konstante für die restliche „select“-Anweisung betrachten: Die negativen Folgen des Umgebungswechsels bestehen nicht mehr. Hat man mehrere Seitenelemente, die man für eine SQL-Anweisung benötigt, könnte man geneigt sein, diese in einer Unterabfrage zu sammeln und anschließend gegen diese Unterabfrage zu joinen. Das ist grundsätzlich eine gute Idee, allerdings ist der Optimizer hier ein wenig übereifrig: Er meint zu erkennen, dass diese Unterabfrage gar nicht nötig ist, und löst sie wieder auf, was dazu führt, dass alle Funktionen wieder für jede Zeile aufgerufen werden.

Zumindest in Datenbank-Version 12c lässt sich dies durch den Optimizer-Hint „/*+ outline */“ in der Unterabfrage verhindern (interessant, dass die Datenbank in diesem Fall die Funktion zweimal aufruft), ansonsten bleibt nur der Weg, jeweils eine separate skalare Unterabfrage für jeden Session-State-Wert zu verwenden (siehe Listing 4 und 5).

```
SQL> declare
  2     cursor emp_cur is
  3         select ename, job, sal
  4             from emp;
  5     l_auszeichnung varchar2(200);
  6 begin
  7     l_auszeichnung := v('P1_AUSZEICHNUNG');
  8     for emp in emp_cur loop
  9         dbms_output.put_line(
10             emp.ename || ' erhaelt die ' ||
11             l_auszeichnung ||
12             ' fuer vorbildlichen PL/SQL-Code');
13     end loop;
14 end;
15 /
```

Listing 7

Die Funktion „V“ in PL/SQL verwenden

Analog lautet die Empfehlung für PL/SQL, den Aufruf der Funktion „V“ nicht in einer Schleife in PL/SQL einzubauen, sondern die benötigten Werte aus dem Session State vor dem Aufruf der Schleife in lokale Variablen umzukopieren, denn auch in einer Schleife wären sonst unnötige Umgebungswechsel zwischen PL/SQL und SQL unvermeidlich. Listing 6 würde man also besser wie in Listing 7 formulieren.

Aber auch dieser Rat gilt generell: Liefert ein Funktionsaufruf ein konstantes Ergebnis, ist er wie eine konstante Variablen-Zuordnung vor einer Schleife und nicht in ihr durchzuführen. Besondere Beachtung sollten hier auch die Funktionen „SYSDATE“ und vor allem „USER“ erfahren, da diese auch Umgebungswechsel zur Folge haben können beziehungsweise bei „USER“ in jedem Fall haben werden.

Empfehlung

Apex erfordert, sich in die Denkweise des Session State einzufinden und sich genau

klarzumachen, was in welcher Reihenfolge bei der Verarbeitung einer Anfrage in Apex geschieht. Nur, wer sich in dieser Arbeitsweise auskennt, wird Fehler vermeiden können. Der Autor erinnert sich an diese Fehler aus seiner eigenen Einarbeitungsphase nur zu gut und ist fest davon überzeugt, dass die Kenntnis des Session State und dessen Arbeitsweise zu den Kern-Kompetenzen gehört, um mit Apex erfolgreich arbeiten zu können. Wer sich die Vor- und Nachteile der unterschiedlichen Zugriffsstrategien auf den Session State zu eigen macht, ist in der Lage, Sicherheitsprobleme zu vermeiden und gleichzeitig aufgeräumten, lesbaren und schnellen Code zu erstellen.



Jürgen Sieben
j.sieben@condes.de

Kritische Lücke in Java SE: Oracle veröffentlicht außerplanmäßiges Update

DOAG Online

Oracle hat ein Patch für Java SE bereitgestellt, das eine kritische Sicherheitslücke schließen soll. Der Fehler erlaubt es Angreifern, ein System vollständig zu übernehmen. Betroffen sind Oracles Java SE 7 Update 97 und Version 8 Update 73 und 74 für Windows, Mac OS X, Linux und Solaris.

Nach Angaben von Oracle können Angreifer die Lücke dazu nutzen, um über eine speziell präparierte Webseite Schadcode aus der Ferne und ohne Authentifizierung einzuschleusen und auszuführen. Die Schwachstelle ist auf der Skala des umfassenden „Common Vul-

nerability Scoring System“ mit 9,3 Punkten bewertet, wobei 10 die höchste Stufe darstellt.

Da die technischen Details der Schwachstelle bereits veröffentlicht wurden, empfiehlt Oracle allen Nutzern dringend zu einem Update.



Fehlersuche in Apex – ein praktischer Einstieg

Peter Raganitsch, FOEX GmbH

Fehler zu vermeiden, ist ein guter Plan, in der Realität werden sich in Apex-Anwendungen aber immer mal wieder kleine Probleme einschleichen. Dieser Artikel gibt einen Überblick über die Debug-Funktionalität in Apex und bietet somit einen Einstieg in die Fehlersuche.

Etwas läuft hier falsch, oder nicht? Was passiert hier eigentlich und wieso läuft meine Apex-Applikation nicht so, wie ich will? Das sind typische Fragen, die sich wohl jeder Entwickler immer mal wieder stellt. Um herauszufinden, was passiert, bietet Apex ein paar Werkzeuge an, allen

voran den sogenannten „Debug“. Streng genommen handelt es sich dabei um ein Logging, das – sobald aktiviert – in einer Log-Tabelle mitprotokolliert, was beim Laden oder Speichern einer Seite alles passiert und welche einzelnen Verarbeitungsschritte Apex durchführt.

Der Begriff „Debugging“ ist in Wikipedia recht schön so definiert: „Debugging is the process of finding and resolving bugs or defects that prevent correct operation of computer software or a system. Debugging tends to be harder when various subsystems are tightly cou-

pled, as changes in one may cause bugs to emerge in another" (siehe „<https://en.wikipedia.org/wiki/Debugging>").

Die Apex-Architektur

Ein Blick auf die Architektur von Apex ruft uns das vorangegangene Zitat über Debugging gleich nochmals in Erinnerung, denn Apex ist tatsächlich ein Sys-

tem, bestehend aus mehreren Subsystemen (siehe Abbildung 1). Ein Fehler in Apex kann also seine Ursache in jedem der verwendeten Subsysteme haben – also entweder in der Datenbank, am Webserver, im Netzwerk oder im Browser ausgelöst worden sein. Das Apex Debugging kann nicht alle Subsysteme mitprotokollieren, aber mit seiner Hilfe kann man die Ursache zumindest lokalisieren.

Apex Debug

Der Apex-Debug-Modus lässt sich über die Developer Toolbar ganz einfach aktivieren (siehe Abbildung 2). Anschließend wird die aktuelle Seite neu geladen und alle Verarbeitungsschritte werden im Hintergrund mitprotokolliert. In der Browser-Adressleiste kann man nun den aktivierten Debug durch das Schlüsselwort „YES“ an der fünften Position der Apex-



Abbildung 1: Apex-Architektur, Quelle: <http://www.oracle.com/technetwork/developer-tools/apex/overview/apex-dba-1867077.pptx>

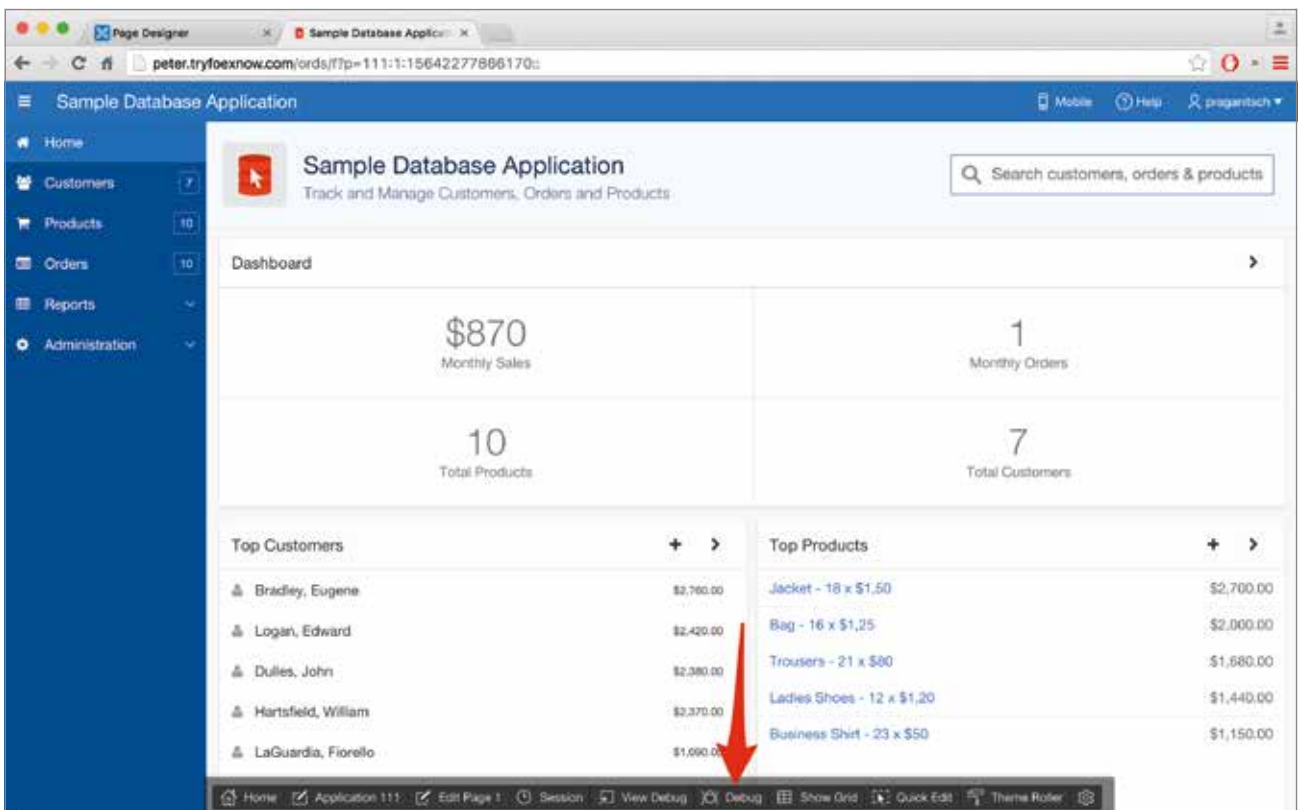


Abbildung 2: Debug über Developer Toolbar aktivieren

Level	Beschreibung
1 - Error	Kritische Fehler
2 - Warning	Weniger kritische Fehler
4 - Info	Default Level (=YES), wenn Debug ohne Angabe von Levels aktiviert wird. Zeigt alle normalen Debug-Informationen
5 - App Enter	Applikation: Meldungen zum Aufruf von Prozeduren
6 - App Trace	Applikation: Andere Meldungen innerhalb von aufgerufenen Prozeduren
8 - Engine Enter	Apex Engine: Meldungen zum Aufruf von internen Prozeduren
9 - Engine Trace	Apex Engine: Andere interne Meldungen

Tabelle 1

URL erkennen: „f?p=111:1:15642277866170::YES“. Neben der einfachen Aktivierung des Debug ist es seit Version 4.2 möglich, die Granularität des Debug zu definieren. Dafür stellt Apex neun Debug-Level zur Verfügung (siehe Tabelle 1). Weitere Informationen dazu stehen in der Apex-Dokumentation unter „API-Reference“ im Kapitel „Apex Debug“ (siehe „https://docs.oracle.com/cd/E59726_01/doc.50/e39149/apex_debug.htm#AEAPI29184“).

An dieser Stelle sei angemerkt, dass durch die Aktivierung von Debug und mit jedem höheren Level etwas mehr Zeit für die Verarbeitung erforderlich ist als mit ausgeschaltetem Debug. Das hängt schlicht von der Anzahl der zu schreibenden Log-Einträge ab und kann auf komplexeren Seiten in die tausende Datensätze gehen. Was genau mitgeloggt wird, kann man sich wieder über die Developer Toolbar, diesmal über den Button „View Debug“ ansehen (siehe Abbildung 3).

In dieser Übersicht sehen wir sogenannte „Debug-Vorgänge“. In dem Beispiel geht es um die Startseite der „Sample Database Application“ und man sieht, dass gleich vier Vorgänge protokolliert wurden. In der Spalte „Path Info“ ist als Hinweis vermerkt, worum es sich handelt: „show“ ist das eigentliche Laden einer Seite, „ajax plugin“ ist der AJAX-Callback eines Plug-ins, das auf der Seite verwendet wird. Im Beispiel werden also erstmal die Seite selbst geladen („show“) und danach drei Plug-ins mit AJAX-Callbacks eingesetzt. Im Falle eines Speichervorgangs („submit“) würde in der „Path Info“ übrigens „accept“ stehen, gefolgt von dem verwendeten Request.

Klickt man nun auf einen der Debug-Vorgänge, so werden dessen Details angezeigt. Unter dem Eintrag „show“ zeigt schon die Übersicht, dass jetzt 748 Detail-Einträge folgen (siehe Abbildung 4).

Die Detail-Ansicht bietet eine Reihe interessanter Informationen. In dem Block ganz oben steht als „Elapsed Time“ die gesamte Dauer der Verarbeitung (in diesem Fall das Laden der Seite), die Apex in der Datenbank aufwendet. Das ist also die Netto-Zeit der Datenbank, ganz ohne Netzwerk, Webserver und den Browser

des Endanwenders. Die „Maximum Execution Time“ ist die längste Dauer eines Einzelschritts in der aktuellen Verarbeitung. Wenn hier eine große Zahl steht, dann findet man diesen Einzelschritt weiter unten in der Liste und kann ihn gezielt optimieren. Die Balkengrafik darunter gibt einen Überblick über die Dauer der jeweiligen

View Identifier	Session Id	User	Application	Page	Path Info	Entries	Timestamp	Seconds
4483574	15642277866170	PRAGANITSCH	111	1	ajax plugin	621	12 minutes ago	0.4260
4483571	15642277866170	PRAGANITSCH	111	1	ajax plugin	31	12 minutes ago	0.2645
4483569	15642277866170	PRAGANITSCH	111	1	ajax plugin	29	12 minutes ago	0.4194
4483565	15642277866170	PRAGANITSCH	111	1	show	748	12 minutes ago	0.6754

Abbildung 3: Debug-Übersicht

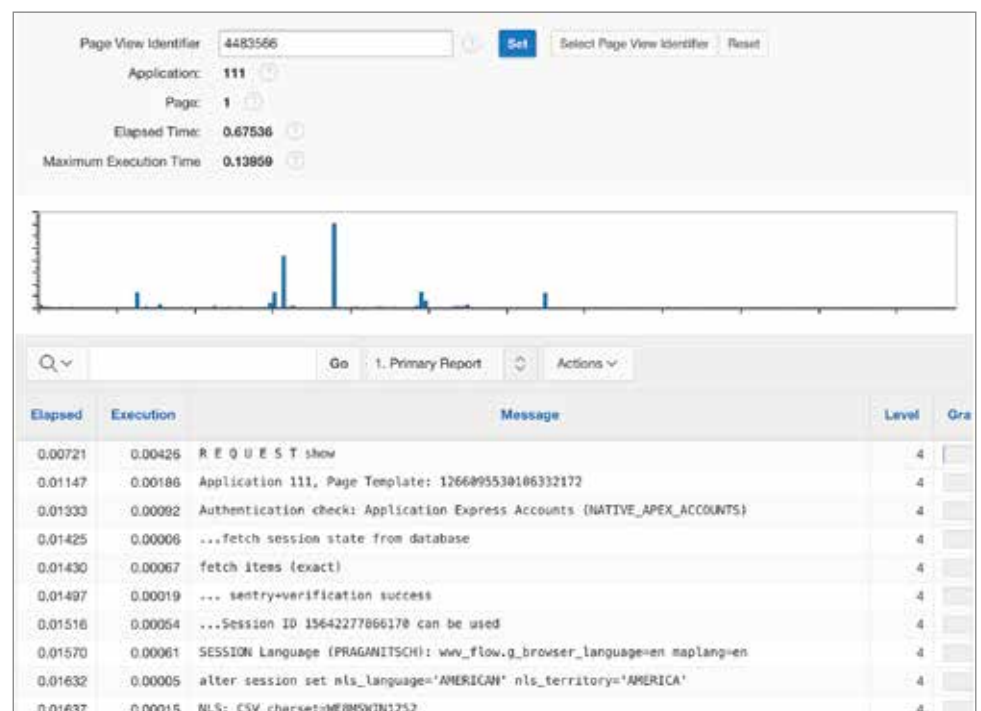


Abbildung 4: Debug Detail

Einzelschritte und vermittelt somit einen Eindruck davon, ob die Verarbeitungszeit gleichmäßig aufgeteilt ist oder ob es ein paar Ressourcen-Fresser gibt.

Danach folgt ein Interactive Report mit einer sortierten Auflistung aller Einzelschritte. In diesem Report kann nun Schritt für Schritt nachvollzogen werden, was Apex alles macht, bevor das Ergebnis (die fertige Seite) angezeigt wird. Wichtig für das Verständnis ist, dass die Einträge in der Reihenfolge ihrer Bearbeitung vorliegen. Dies kann einige Probleme aufklären, wenn beispielsweise durch den Debug klar wird, dass ein Element eventuell schon früher bearbeitet wird, als man das angenommen hatte. Beispiele für Detail-Einträge sind:

- Durchführung von Computations und Prozessen
- Das Nicht-Durchführen von Bearbeitungen, wenn dies von einer Condition, einer Authorization oder Ähnlichem verhindert wird
- Gesetzter Session State durch URL-Parameter, Computations/Prozesse etc.
- Ausgeführte Statements (Prozesse, Computations, Regionen)
- NLS-Settings der Datenbank-Session
- Aufgetretene Fehler
- Verarbeitete Regionen und Items

Den Debug Output lesen

Wie der Debug Output zu lesen ist, hängt in erster Linie von der Fragestellung ab, also davon, was man herausfinden möchte. Um nachzuvollziehen, was Apex genau macht, liest man den Debug am besten sequenziell von oben nach unten, in der zeitlichen Reihenfolge der Ereignisse. Ist man auf der Suche nach Performance-Hotspots, dann interessieren einen die Ausschläge in der Balken grafik beziehungsweise man kann den Interactive Report so sortieren oder fil-

tern, dass die Einträge mit der höchsten „Execution“-Zeit oben stehen.

Genau so entscheidend ist auch die Wahl des richtigen Debug-Levels. Mit dem Default „YES“ (=Level 4) sind eventuell wichtige Einträge gar nicht vorhanden. Immer Level 9 zu aktivieren, ist aber auch nicht zielführend, da man hier von der Menge der Einträge erschlagen wird. Mit etwas Ausprobieren findet man für die jeweilige Situation bald die richtigen Einstellungen sowie die Balance zwischen der Anzahl der Log-Einträge und dem Informationsgehalt.

```

BEGIN
  Apex_DEBUG.ENTER('My Procedure');
  MY_PACKAGE.FUNCTION1;
  Apex_DEBUG.TRACE('Returned from Function1');
  MY_PACKAGE.FUNCTION2;
  Apex_DEBUG.TRACE('Returned from Function2');

  IF :P1_MY_ITEM = 'ERROR'
  THEN
    Apex_DEBUG.ERROR('Something went wrong');
  END IF;

  Apex_DEBUG.TRACE('Done, leaving My Procedure');
END;
    
```

Listing 1

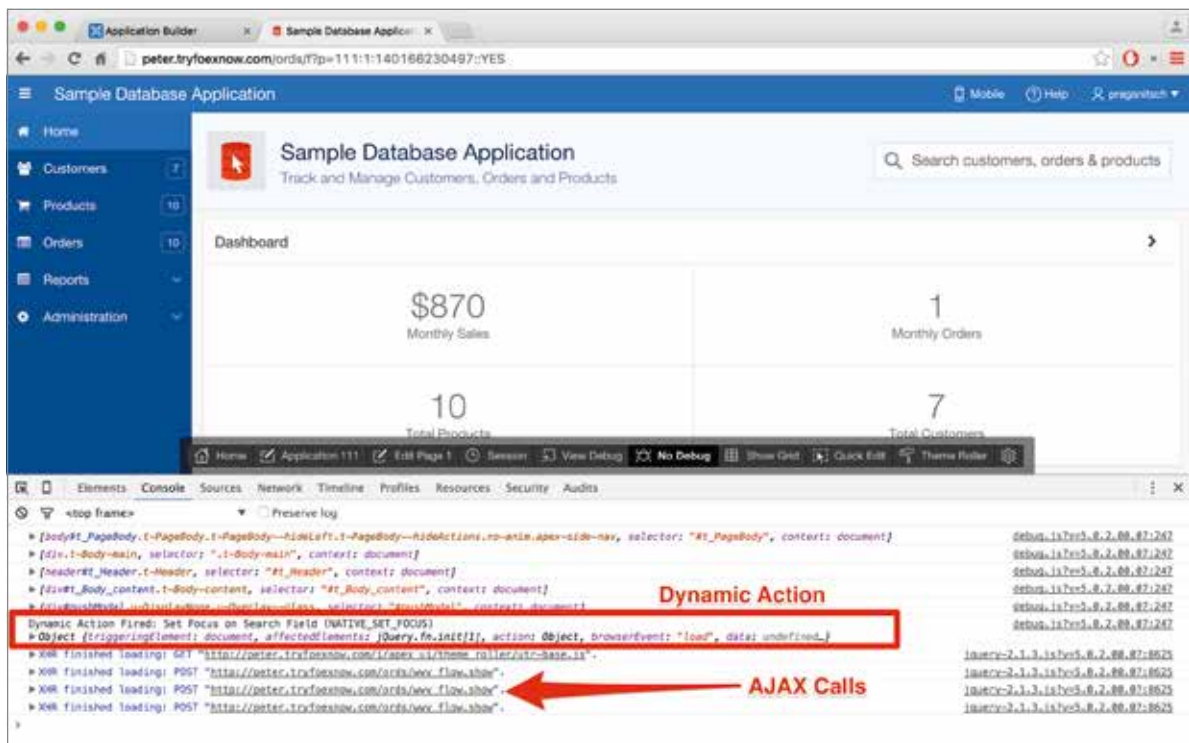


Abbildung 5: Debug-Meldungen in der Browser Console

Da der Debug Output als Interactive Report präsentiert wird, lassen sich hier auch verschiedene Ansichten als Private Reports speichern, in denen je nach Art der Nachforschung andere Spalten oder Sortierungen verwendet werden. Wer noch mehr Flexibilität will, der kann auch direkt die View „APEX_DEBUG_MESSAGES“ abfragen und eigene Reports darauf aufsetzen. Diese View ist auf „Runtime only“-Produktionssystemen auch der einzige Weg, um an die Debug-Information zu gelangen.

Zeitliche Begrenzung der Debug-Information

Wie alle internen Apex-Log-Tabellen wird auch Debug periodisch gelöscht. Per Default ist das Lösch-Intervall auf 14 Tage gestellt, es kann aber über die Apex-Instanz-Administration auf maximal 180

Tage erhöht werden. Für die Speicherung werden intern zwei Tabellen verwendet, die bei Erreichen des Lösch-Intervalls rollieren: Erst wird in Tabelle1 geschrieben, nach 14 Tagen in Tabelle2. Weitere 14 Tage später wird Tabelle1 gelöscht und dann neu beschrieben. Das soll bedeuten, dass die Daten normalerweise „Löschintervall*2-1“ Tage vorhanden sind. Die View „APEX_DEBUG_MESSAGES“ fragt automatisch immer beide Tabellen ab.

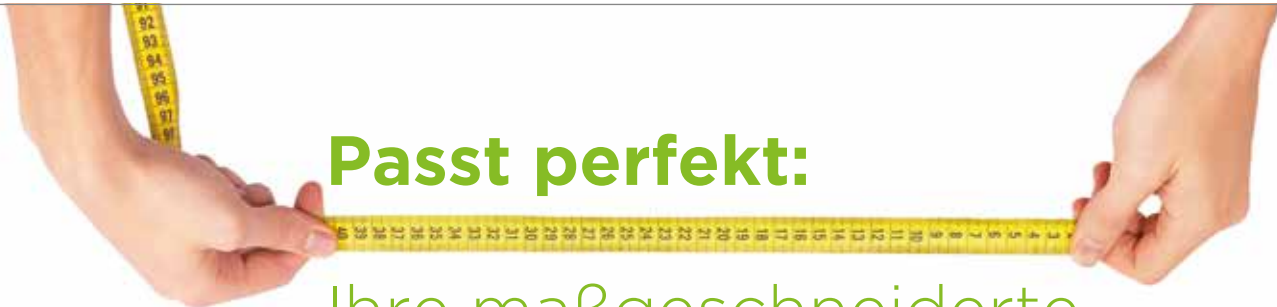
Eigene Einträge hinzufügen

Mit den Debug-Einträgen von Apex lässt sich schon viel herausfinden. Sobald man aber eigene Business-Logik einsetzt, will man diese vielleicht auch im Debug mitprotokollieren. Dazu steht das „APEX_DEBUG_API“ bereit, das mit diversen Log-Funktionen das Protokollieren in verschiedenen Levels ermöglicht: „ERROR“, „WARN“, „INFO“,

„TRACE“, „ENTER“ etc. Damit kann nun nicht nur in Apex-PL/SQL-Prozessen etwas in den Debug geschrieben werden, auch in aufgerufenen Packages und Funktionen lassen sich Log-Einträge erstellen (siehe Listing 1).

Gut zu wissen ist, dass Debug-Einträge mit der Funktion „ERROR“ immer geschrieben werden, unabhängig davon, ob Debug aktiviert ist. Das ist sehr praktisch, da man durch Selektieren von „APEX_DEBUG_MESSAGES“ gezielt nach solchen Error-Einträgen suchen und damit eine automatisierte QA-Maßnahme aufsetzen kann.

Wird ein bestehendes Package aufgerufen, das Meldungen auf „DBMS_OUTPUT“ schreibt, lassen sich diese mit „APEX_DEBUG.LOG_DBMS_OUTPUT“ auf den Apex Debug umleiten. Ein sehr nützliches Feature in Zusammenhang mit der Verwendung älterer Bestands-Packages. „ENABLE“ beziehungsweise „DISABLE“ steuern den Debug-Modus auch programmatisch; so



Passt perfekt:

Ihre maßgeschneiderte Business Applikation

Unsere Oracle APEX Spezialisten analysieren gemeinsam mit Ihnen die spezifischen Anforderungen und Abläufe Ihrer Organisation und entwickeln mit Oracle APEX in erstaunlich kurzer Zeit Ihre individuelle, maßgeschneiderte Business Applikation.

Nutzen Sie die Vorteile einer optimal passenden Software Lösung für Ihr Unternehmen, um Ihren Vorsprung gegenüber den Marktbegleitern auszubauen.

Wir sind nur einen Anruf entfernt: +43 1 890 89 990

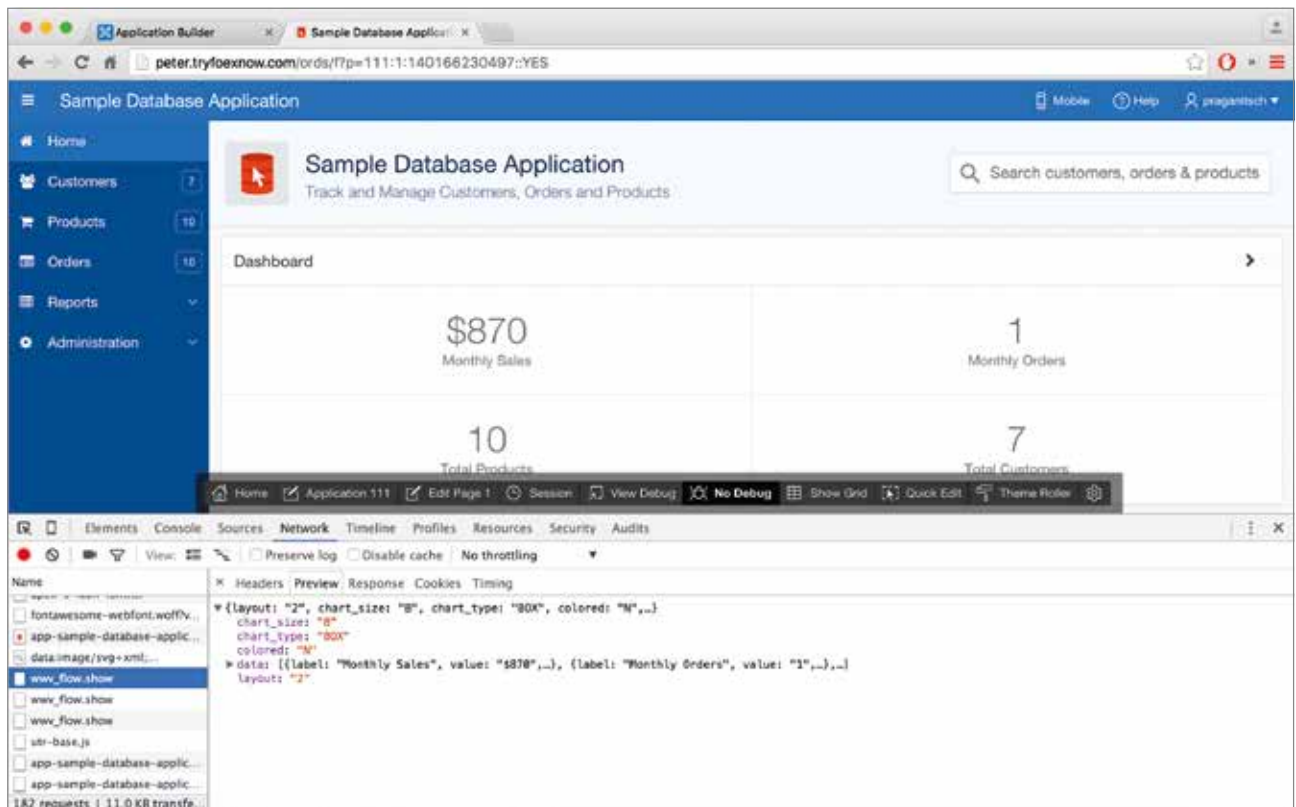


Abbildung 6: Das Ergebnis eines AJAX-Calls

kann man seinen Anwendern beispielsweise einen Button zum Aktivieren des Debug anbieten.

Debug auf dem Browser

Die bisher betrachtete Verwendung des Debug beschränkt sich auf die Verarbeitung der Apex-Seiten und AJAX-Calls innerhalb der Datenbank. In moderneren Anwendungen geschieht durch Dynamic Actions allerdings auch sehr viel im Browser. Nun ist es aber so, dass der Browser keine durchgehende Verbindung in die Datenbank hat und deswegen dort auch nicht in die Log-Tabelle schreiben kann. Dafür extra immer einen AJAX-Call abzusetzen, ist auch ausgeschlossen, das würde die Performance und Usability zerstören. Stattdessen bedient sich Apex hier der Browser-eigenen Console und deren Logging-Möglichkeiten.

Alle modernen Browser bieten inzwischen sogenannte „Developer Tools“ an. Darin enthalten ist eine Console, die JavaScript-Meldungen protokolliert und sonstige Fehler und Warnungen anzeigt. Beim Aktivieren des Apex-Debug-Mode wird

auch das JavaScript Logging aktiviert und man sieht Apex-Debug-Meldungen in der Console. Dort werden außerdem alle Dynamic Actions mitgeschrieben; so hat man auch im Browser einen guten Überblick über den Ablauf der Geschehnisse (siehe Abbildung 5).

Die Abbildung zeigt wiederum die erste Seite der Sample Database Application, in der bei Page Load eine Dynamic Action ausgeführt wird. Gut sichtbar sind dort auch die drei AJAX-Aufrufe der Plug-ins. Klickt man auf einen dieser AJAX-Calls, kann man sich die Details und das Ergebnis anzeigen lassen, um so die empfangenen Daten zu kontrollieren (siehe Abbildung 6).

Auch im JavaScript-Bereich ist es möglich, eigene Debug-Meldungen zu schreiben, das API dafür lautet „apex.debug“ und funktioniert ganz ähnlich wie sein PL/SQL-Äquivalent.

Fazit

Sobald man alle Werkzeuge kennt, kann auch die Fehlersuche beginnen. Als Grundregel ist es dabei immer ratsam, die Browser Console zu öffnen und den De-

bug Mode zu aktivieren. Je nach Fehler beziehungsweise Problem, das untersucht werden soll, wird man sich dann mehr auf die Browser Console oder die Log-Einträge in der Datenbank konzentrieren. Für einen gesamthaften Überblick sollte aber beides betrachtet werden. Unter „<https://apex.oracle.com/pls/apex/f?p=25355>“ gibt es eine Beispiel-Anwendung mit ein paar typischen Fehlern, die mithilfe von Debug einfach entdeckt werden können.



Peter Raganitsch
peter.raganitsch@tryfoexnow.com



Wo ist der Fehler? – Debugging in Apex

Christina Funke, Apps Associates GmbH

Oracle Application Express ist ein Tool, um relativ schnell umfangreiche Web-Applikationen zu programmieren. Leider schleicht sich dabei schnell mal der eine oder andere Fehler ein. Die Fehlersuche ist dann nicht immer ganz einfach. Dieser Artikel zeigt die verschiedenen Möglichkeiten des Debuggings und der Fehlersuche in Apex. Außerdem werden unterschiedliche Varianten zum Erstellen benutzerfreundlicher und aussagekräftiger Fehlermeldungen vorgestellt, um die Fehlersuche ebenfalls erheblich zu erleichtern.

Jeder, der schon mal eine Apex-Anwendung geschrieben hat, weiß: Die Fehlersuche kann sehr aufwändig sein. Oft weiß man gar nicht, welches Objekt gerade fehlerhaft ist. Ist es ein Prozess,

eine Validierung oder eine Region? Wird die Prozedur auch mit den richtigen Werten aufgerufen und welche Daten sind eigentlich in meiner Apex-Collection gespeichert?

Apex Debug Mode

Ein nützliches Werkzeug zum Debuggen in Apex ist der Apex Debug Mode. Er hilft dem Entwickler zu verstehen, wel-

View Identifier	Session Id	User	Application	Page	Path Info	Entries	Timestamp	Seconds
732326236	2426994693213	CHRISTINA	11002	6	show	83	Now	0.1973
732326186	2426994693213	CHRISTINA	11002	6	-	1	6 minutes ago	0.0977

Abbildung 1: Die Developer Toolbar

che Aktion wann in Apex ausgeführt wird und bei welcher Aktion es überhaupt zum Fehler kommt. Um den Apex Debug Mode vollständig nutzen zu können, muss dieser erst unter „Edit Application Properties“ aktiviert sein. Wird die Applikation nun gestartet, kann der Debug Mode geöffnet werden, indem man den Button „Debug“ und anschließend den Button „View Debug“ in der Developer Toolbar betätigt. *Abbildung 1* zeigt das sich öffnende Fenster.

Zunächst werden einige allgemeine Informationen zu den einzelnen Debug-Durchläufen angezeigt. Hinter der Spalte „View Identifier“ verbirgt sich ein Link, der auf ein neues Fenster verweist. Dort erscheint eine Liste, in der genau steht, welche Computations, Validierungen und Prozesse in welcher Reihenfolge mit welchen Werten ausgeführt werden. Mithilfe dieser Liste lassen sich schnell Aktionen identifizieren, die vielleicht nicht oder mit den falschen Werten ausgeführt wurden. Außerdem ist die verbrauchte Zeit der einzelnen Aktionen ablesbar. So können Ausreißer identifiziert und gegebenenfalls die Performance verbessert werden.

Auch andere Punkte in der Developer Toolbar können bei der Fehlersuche hilfreich sein. Zum Beispiel werden unter dem Punkt „Session“ die aktuellen Werte der einzelnen Page-Items dargestellt. Auch die Werte von Application Items (globale Variablen) können hier angezeigt und kontrolliert werden. Es werden sogar einige Collection-Werte angezeigt. Unter dem Punkt „Error“ ist aufgelistet, auf welcher Page welche Fehler wann vorgekommen sind.

Debug Messages

Unter Umständen kann es auch sehr nützlich sein, selbst Meldungen in den Debug Mode zu schreiben, um besser nachvollziehen zu können, wo es innerhalb eines PL/SQL-Codes zum Fehler kommt. Mit dem Befehl „apex_debug.message“ ist dies möglich. Dazu müssen die Meldungen, die später im Debug Mode sichtbar sein sollen, als „DBMS_Output“-Befehl in den PL/SQL-Code (etwa einer Funktion, Prozedur oder Package) geschrieben werden. Anschließend muss man nur noch den Prozess anpassen, der den PL/SQL-Code aufruft. *Listing 1* zeigt ein Beispiel. Es ist natürlich auch möglich, die selbst erstellten Meldungen in einer Datenbank-Tabelle oder in einer Apex-Collection zu speichern und sie anschließend als Report in Apex sichtbar zu machen.

Error Handling und benutzerfreundliche Fehlermeldungen

Aussagekräftige Fehlermeldungen können die Fehlersuche ebenfalls erheblich vereinfachen. Die typische Prozess-Error-Message „Die Daten konnten nicht gespeichert werden“ ist viel zu allgemein, als dass sie einem Entwickler bei der Fehlersuche helfen könnte. Warum die Daten nicht gespeichert werden konnten, geht aus so einer Fehlermeldung nicht hervor. Mit der Fehlermeldung „ORA-01722: invalid number“ dagegen ist einem Entwickler schon sehr geholfen. Diese Fehlermeldung ist allerdings für einen Endanwender der Applikation absolut unverständlich und frustrierend. Besser wäre hier eine Fehlermeldung wie in *Abbildung 2*.

```

DECLARE
    v_zeilen dbms_output.chararr;
    v_anzahl number := 100;
BEGIN
    --DBMS_OUTPUT aktivieren
    dbms_output.enable;

    --eigentliche Procedure ausführen
    my_procedure(parameter1, parameter 2);

    --Meldungszeilen aus dem Puffer abrufen
    dbms_output.get_lines(v_zeilen, v_anzahl);

    --Meldungszeilen als Apex Debugging Meldung ausgeben
    FOR i IN 1..v_anzahl LOOP
        apex_debug.message(v_zeilen(i));
    END LOOP;
END;

```

Listing 1

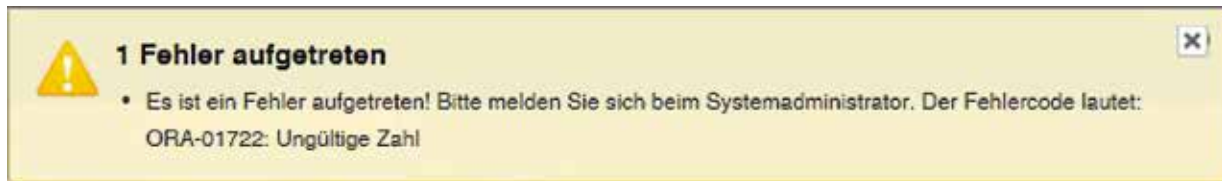


Abbildung 2: Aussagekräftige Fehlermeldung

Diese ist für den Endanwender verständlich und gibt dem Entwickler der Anwendung gleichzeitig einen Hinweis darauf, warum die Daten nicht gespeichert werden konnten. Natürlich gibt es mehrere Alternativen, eine solche Fehlermeldung zu erstellen. Eine Möglichkeit wäre, ein Exception Handling in dem PL/SQL-Code des Prozesses zu definieren. Listing 2 zeigt auch dazu ein Beispiel.

Bei diesem Vorgehen darf keine Process Error Message definiert sein, da in dem Fall nur diese angezeigt wird und der User die Meldung, die im Exception Handling definiert wurde, nicht sehen wird. Außerdem hat diese Variante den Nachteil, dass das gleiche oder ein sehr ähnliches Exception Handling für jeden Prozess der Applikation geschrieben werden müsste.

Diese redundante Arbeit lässt sich vermeiden, indem eine Funktion erstellt wird,

in der für verschiedene Fehler-Typen benutzerfreundliche und aussagekräftige Messages definiert sind. Ein Beispiel für eine solche Funktion findet sich auf der Oracle-Seite „http://docs.oracle.com/cd/E23903_01/doc/doc.41/e21676/apex_error.htm“. Der Name dieser Funktion ist anschließend unter „Edit Application Properties“, „Error Handling“ und „Error Handling Function“ hinterlegt.

Debugging mit Apex Collections

Apex Collections werden verwendet, um mehrere Datensätze in einer Session zwischenspeichern. In dieser spezifischen Session lassen sich die Daten ansprechen, ändern und verarbeiten. Aber eben nur in dieser spezifischen Session, in einer anderen Session sind die Inhal-

te schon nicht mehr sichtbar. Das macht das Debugging von Applikationen, die Collections verwenden, so schwierig. Um sich die Inhalte einer Collection doch über SQL*Plus, den SQL Developer oder aus einer anderen Apex-Sitzung heraus ansehen zu können, ist es hilfreich, sich die Definition der View „Apex_COLLECTIONS“ anzusehen (siehe Listing 3).

Die „WHERE“-Bedingung verhindert, dass die Inhalte der Collection beispielsweise im SQL Developer angezeigt werden. Um sich die Inhalte trotzdem anzeigen zu lassen, müssen die Werte für die Session-ID („www_flow.g_instance“), die Workspace-ID („www_flow_api.set_security_group_id“) und die Application-ID („www_flow.g_flow_id“) richtig gesetzt sein. Die Werte der Session-ID und der Applikation-ID können der URL im Browser entnommen und die Workspace-ID aus der View „Apex_WORKSPACES“ ausgelesen werden. Anschließend lassen sich die Inhalte der Collection mit einem Select auf die View „Apex_COLLECTIONS“ anzeigen.

Eine andere Möglichkeit, den Inhalt einer Apex-Collection zu betrachten, ist, eine View zu erzeugen, die nicht auf die Session-ID, die Workspace-ID und die Application-ID begrenzt ist. In diesem Fall sollte die View einen anderen Namen bekommen und vor unberechtigten Zugriffen geschützt werden.

Remote Debugging mit dem SQL Developer

Mit Remote Debugging lässt sich PL/SQL-Code, der in einer Apex-Anwendung ausgeführt wird, im SQL Developer testen. Dabei wird der Quellcode aus Apex heraus gestartet. Das Debugging findet dann im SQL Developer statt. Damit das Remote Debugging mit dem SQL Developer auch funktioniert, muss dem Parsing-Schema, in dem die Apex-Applikation läuft, zunächst noch das System-

```
...
EXCEPTION WHEN OTHERS THEN
  raise_application_error(-20101,'Es ist ein Fehler aufgetreten.
  Bitte melden Sie sich beim Systemadministrator.
  Der Fehlercode lautet: ' || substr(SQLERRM, 1,3500));
```

Listing 2

```
select
  c.collection_name,
  m.seq_id, m.c001, m.c002, m.c003, m.c004, m.c005, m.c006,
  m.c007, m.c008, m.c009, m.c010, m.c011, m.c012, m.c013,
  m.c014, m.c015, m.c016, m.c017, m.c018, m.c019, m.c020,
  m.c021, m.c022, m.c023, m.c024, m.c025, m.c026, m.c027,
  m.c028, m.c029, m.c030, m.c031, m.c032, m.c033, m.c034,
  m.c035, m.c036, m.c037, m.c038, m.c039, m.c040, m.c041,
  m.c042, m.c043, m.c044, m.c045, m.c046, m.c047, m.c048,
  m.c049, m.c050, m.clob001, m.md5_original
from wwv_flow_collections$ c, wwv_flow_collection_members$ m
where c.id = m.collection_id
  and c.session_id = (select v('SESSION') from dual)
  and c.security_group_id = (select wwv_flow.get_sg_id from dual)
  and c.flow_id = (select nv('FLOW_ID') from dual)
```

Listing 3

Privileg „DEBUG CONNECT SESSION“ erteilt werden. Das Schema, unter dem die Apex-Sessions ablaufen (in der Regel „Apex_PUBLIC_USER“), benötigt das Privileg „DEBUG ANY PROCEDURE“.

Das „DEBUG“-Privileg kann natürlich auch an nur einzelne Prozeduren vergeben werden („GRANT DEBUG ON COMMUNITY.MY_PROCEDURE to Apex_PUBLIC_USER“). Außerdem ist zu beachten, dass PL/SQL-Code, der direkt als Prozess-Text in Apex hinterlegt wurde, nicht mit dem SQL Developer überprüft werden kann. Der Code muss dazu als PL/SQL-Funktion, -Prozedur oder -Package hinterlegt sein. Dieses Vorgehen ist sowieso zu empfehlen.

Wurden die Rechte an die entsprechenden Schemata erteilt, kann die PL/SQL-Funktion, -Prozedur oder -Package ganz normal erstellt und kompiliert werden. Anschließend ist ein Breakpoint zu definieren und die Logik noch einmal für Debug zu kompilieren.

Nun wird der Remote Debugger eingerichtet. Dazu braucht es einen Rechtsklick auf die Verbindung und einen Klick auf „Remote Debugging ...“. Es öffnet sich ein Dialog-Fenster. Hier werden ein freier TCP/IP-Port und die IP-Adresse des Rechners, auf dem der SQL Developer läuft, eingetragen. Damit Apex den SQL Developer über das Netzwerk kontaktieren kann, muss der PL/SQL-Aufruf definiert sein (siehe Listing 4).

Wird die Apex-Seite nun gestartet und gegebenenfalls der Prozess angestoßen, wartet der Browser. Das ist auch richtig

```
BEGIN
  dbms_debug_jdwp.connect_tcp('#IP-Adresse', #TCP/IP-Port);
  my_procedure(parameter);
  dbms_debug_jdwp.disconnect;
END;
```

Listing 4

so, da der Debugger im SQL Developer am oben definierten Breakpoint stehen geblieben ist. Nun kann der Entwickler den PL/SQL-Code schrittweise durchgehen und dabei alle Variablen im Auge behalten. Die Werte der Variablen können auch geändert werden. Das erleichtert und beschleunigt die Fehlersuche massiv.

Fazit

Die Fehlersuche ist bei allen Entwicklungsarbeiten eine lästige und oft auch schwierige Aufgabe. Allerdings bietet Apex einige Möglichkeiten, die Fehler innerhalb einer Applikation einzugrenzen und ihnen auf die Schliche zu kommen. Sehr hilfreich ist dabei auch die Verwendung von aussagekräftigen Fehlermeldungen. Selbst wenn nur die Beispiel-Funktion von Oracle in Apex eingebunden wird, können spezielle Fehler mit aussagekräftigen Fehlermeldungen versehen werden. Dies erleichtert die Fehlersuche erheblich.

Besonders in Verbindung mit dem SQL Developer bietet der Remote Debugger

eine gute Möglichkeit, genau nachzuvollziehen, mit welchen Parametern der PL/SQL-Code aufgerufen wird und was genau innerhalb des Codes mit den einzelnen Variablen passiert. So kann relativ einfach herausgefunden werden, an welcher Stelle des Codes genau der Fehler steckt.

Mit diesen hier vorgestellten Werkzeugen und Vorgehensweisen fällt die lästige Aufgabe der Fehlersuche und des Debuggings zwar nicht ganz weg, aber immerhin kann man sie doch sehr vereinfachen und so auch stark verkürzen.



Christina Funke
christina.funke@appsassociates.com

Starker Dollar setzt Oracle weiter zu

DOAG Online

Im dritten Geschäftsquartal hat Oracle erneut Schwächen im traditionellen Geschäft mit dem Verkauf von Lizenzen und Wartung gezeigt, die auch ein Umsatzanstieg im Cloud-Geschäft nicht überwinden konnte. Wie bereits in den vorherigen Quartalen belastete der starke Dollar die Zahlen zusätzlich.

Wie Oracle zum Ende seines dritten Geschäftsquartals mitgeteilt hat, fiel der Umsatz des Unternehmens um 3,4 Prozent auf neun Milliarden US-Dollar. Auch der Nettogewinn ging um 14 Prozent auf 2,14

Milliarden US-Dollar zurück. Grund dafür sind neben des zurzeit starken Dollars vor allem die stark rückläufigen Verkäufe von Software-Lizenzen und Wartung: Die Umsätze im Kerngeschäft sanken um vier Prozent auf 6,3 Milliarden US-Dollar, während sich die Umsatzeinbußen in der Hardware- und Servicesparte auf jeweils 13 Prozent (1,1 Milliarden US-Dollar) beziehungsweise sieben Prozent (793 Millionen US-Dollar) beliefen.

Oracle versucht weiterhin, die Einbußen im traditionellen Geschäft durch Zu-

wächse im Cloud-Bereich wettzumachen. Trotz des allgemeinen Umsatz- und Gewinnrückgangs zeigen sich die Oracle CEOs Safrat Catz und Mark Hurd zuversichtlich. Letzterer betont insbesondere den enorm gewachsenen Bruttoumsatz im SaaS- und PaaS-Bereich um 96 Prozent. Außerdem hätte man im dritten Quartal 942 neue SaaS-Kunden gewonnen. Laut Larry Ellison, Oracle-Vorsitzender und CTO, wachse das Unternehmen in diesem Bereich außerdem weit schneller als die Konkurrenz.



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

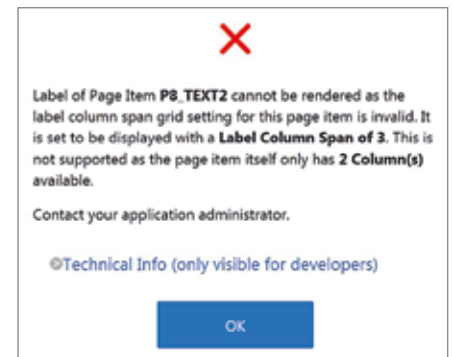


Abbildung 1: Fehlermeldung zum „Invalid Grid Setting“



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



Apex und Workflows

Sven Böttcher, Apps Associates

Als Apex-Berater ist man in Kundenprojekten oft mit der Frage konfrontiert, welche Möglichkeiten Apex für die Entwicklung komplexer Workflows bietet. Die Antwort lautet: Leider keine, es muss alles in PL/SQL implementiert werden! Dies führt häufig zu einer sehr komplexen und nur schwer wartbaren Applikationslogik. Grund genug, sich mit Alternativen der Workflow-Implementierung in Apex auseinanderzusetzen.

Apex erfreut sich als Rapid-Application-Development-Tool für die Entwicklung datenbankzentrischer Web-Anwendungen großer Beliebtheit. Gründe sind unter anderem, dass Apex lizenzkostenfrei verwendet werden kann und die Entwicklung von Eingabemasken durch einen GUI-Editor schnell und einfach erfolgt. Die GUI-Editor-gestützte Anwendungs-

entwicklung macht vor allem ein prototypisches Vorgehen möglich, wodurch sich die späteren Benutzer sehr gut in den Software-Entwicklungsprozess einbeziehen lassen. Auf diese Weise kann auf Missverständnisse oder gegebenenfalls auftretende Änderungswünsche bereits in einem frühen Stadium des Entwicklungsprozesses reagiert werden.

Es kommt jedoch häufig vor, dass für eine komplexe Applikationslogik entsprechend aufwändige Workflows umgesetzt werden müssen. An dieser Stelle bietet Apex keinerlei Unterstützung. Die einzige Möglichkeit, Workflows umzusetzen, besteht in der manuellen Programmierung, im Allgemeinen in der Programmiersprache PL/SQL. Dies führt nicht selten zu einem

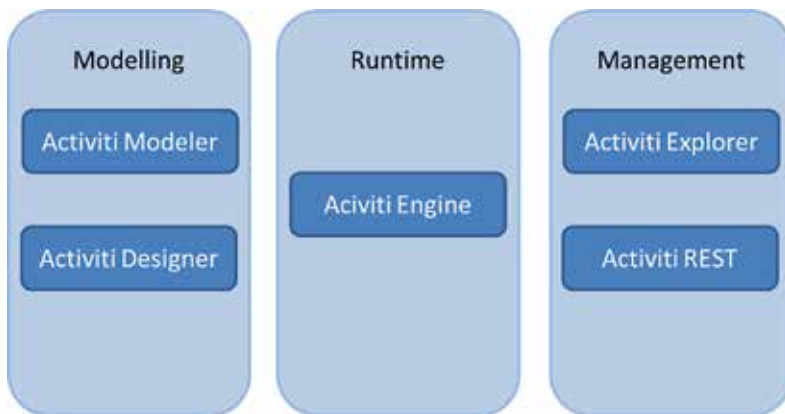


Abbildung 1: Die Activiti-Komponenten (Quelle: Activiti.org)

sehr langen und komplizierten Programmcode, der fehleranfällig und nur schwer wartbar ist. Zudem kann der Programmcode in der Regel nicht mehr von den späteren Benutzern der Anwendung verstanden werden. Dies und der zumeist hohe Aufwand der Workflowprogrammierung erschweren wiederum die prototypische Anwendungsentwicklung und das Einbeziehen der Benutzer in den Entwicklungsprozess. Eine Lösung dieses Dilemmas besteht in der Integration von Apex und einer Workflow-Engine, die idealerweise in einer grafischen Beschreibungssprache (wie Business Process Model and Notation – BPMN) beschriebene Workflows ausführen kann.

Integration von Apex und Workflow Engines

Die Vorteile der Integration von Workflow Engines beziehungsweise BPM-Suiten und Apex liegen auf der Hand. Prozess-Analysten

können die zugrunde liegenden Workflows in ihnen bekannten Modellierungssprachen modellieren und die so entstehenden Workflows lassen sich ohne eine Re-Implementierung (in PL/SQL) direkt ausführen. Da die verwendeten Modellierungssprachen zumeist grafischer Natur sind, werden die beschriebenen Workflows auch von den Benutzern der Anwendung verstanden. Zudem wird die Applikation wartungsärmer und die Applikationslogik verständlicher, da sich die Menge des Quellcodes reduziert.

Die grundlegende Idee der Integration besteht darin, dass die Workflow Engine den Programmablauf beziehungsweise die Programmlogik steuert. Apex hingegen sollte Eingabemasken für die Nutzer-Interaktion bereitstellen. Die einzelnen Tasks eines Workflows müssen daher mit entsprechenden Apex-Seiten für die Bearbeitung der Tasks verknüpft sein. Diese Verknüpfung kann dabei über das Anreichern der Tasks eines Workflows mit Meta-Informationen erfolgen. Für die Abfol-

ge von Eingabemasken beziehungsweise Apex-Seiten würde das zum Beispiel bedeuten, dass zu jedem Task des Workflows festgelegt wird, mit welcher Apex-Seite dieser Task zu bearbeiten ist.

Die eigentliche Integration beziehungsweise Kommunikation zwischen Apex und der Workflow Engine kann über den Aufruf von Webservices erfolgen. Diese werden durch Apex initialisiert und dienen beispielsweise dem Starten von Prozessen, der Abfrage von als Nächstes auszuführenden Tasks, der Zuweisung von Tasks zu Benutzern und dem Beenden von Tasks.

Da jeder Task über Meta-Informationen mit einer Apex-Seite verknüpft ist, kann einem Benutzer nach der Übernahme eines Tasks die entsprechende Apex-Seite für die Bearbeitung dieses Tasks angezeigt werden. Den Entwicklern einer Apex-Anwendung kommt nun die Aufgabe zu, die Tasks des Workflows mit entsprechenden Meta-Informationen anzureichern sowie die Implementierung der eigentlichen Integration von Apex und der Workflow Engine durchzuführen. Auch wenn Letzteres aufwändig erscheint, entsteht der Vorteil, dass die Webservice-Aufrufe im Gegensatz zu in PL/SQL implementierten Workflows nicht projektspezifisch sind und sich als eine Art Framework in den verschiedensten Projekten wiederverwenden lassen.

Activiti

Activiti ist eine quelloffene Workflow- und Business-Process-Management-Plattform (BPM), die im Rahmen der Apache-Lizenz kostenlos verwendet werden kann. *Abbil-*

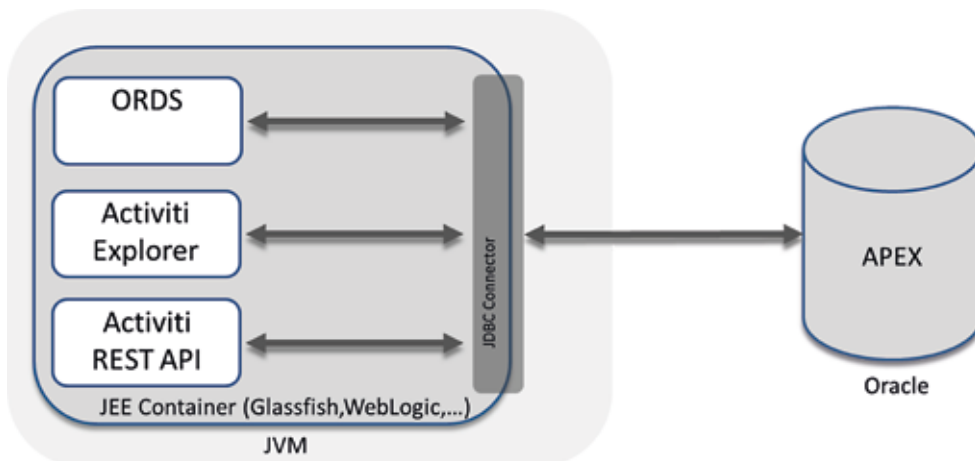


Abbildung 2: Apex- und Activiti-Infrastruktur

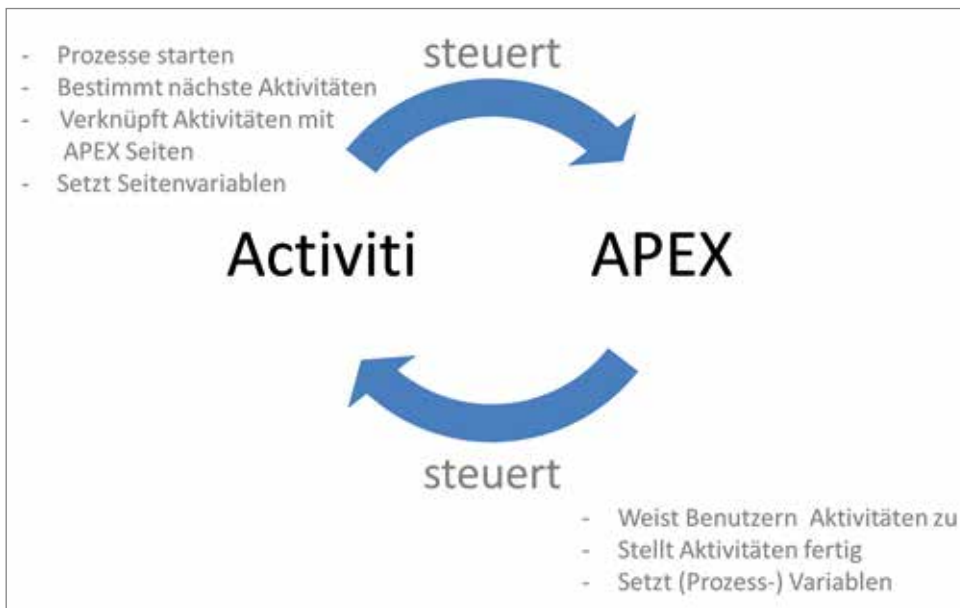


Abbildung 3: Wechselseitige Steuerung zwischen Apex und Activiti

Abbildung 1 zeigt die zentralen Komponenten von Activiti. Sie lassen sich grob in die Bereiche „Modellierung“, „Laufzeit“ und „Management“ gliedern.

Während der Activiti Modeler eine sehr einfache und webbasierte Möglichkeit zur Modellierung von BPMN-2.0-Prozessen bietet, besteht der eigentliche Kern der Activiti-BPM-Suite aus der Activiti Engine. Diese führt BPMN-2.0-Prozesse aus und kann entweder standalone quasi als Java-Programm oder auf einem JEE-konformen Application Server (wie GlassFish, WebLogic etc.) betrieben werden. Die komplette Steuerung der Activiti Engine kann entweder Java-basiert oder über ein REST-API (Activiti REST) erfolgen. Für Letzteres ist ein JEE-konformer Application Server erforderlich. Dieses Feature ist Grundlage der hier beschriebenen Apex-Integration.

Apex und Activiti

Activiti verfügt über ein sehr umfangreiches REST-API, mit dem Activiti komplett gesteuert werden kann. Darüber hinaus lassen sich Tasks über den Activiti Modeler (mit dem man auch die Workflows modelliert) mit den benötigten Meta-Informationen anreichern. Die beschriebenen Anforderungen für die Apex-Integration sind demzufolge von Activiti erfüllt.

Apex und Activiti passen aber auch hinsichtlich der benötigten Infrastruktur bes-

tens zusammen. Da Apex in der Oracle-Datenbank läuft und sich diese problemlos als Activiti-Repository verwenden lässt, ist keine zusätzliche Datenbank notwendig. Wird auf Apex (wie von Oracle empfohlen) mithilfe der Oracle REST Data Services (ORDS) zugegriffen, kann auch für das Activiti-REST-API und die Activiti Modeler die Apex-Infrastruktur mitverwendet werden. Je nach Deployment ist keine zusätzliche Infrastruktur nötig, was insbesondere die Be-

triebskosten hinsichtlich der Administration und Wartung gering hält. *Abbildung 2* zeigt schemenhaft die gemeinsame Nutzung der Infrastruktur von Apex und Activiti.

Für die Integration beider Tools existiert eine PL/SQL-Bibliothek, die im Wesentlichen eine Menge von Webservice-Aufrufen darstellt. Diese steuern die Activiti Engine wie beschrieben und liefern als Ergebnis verschiedene Informationen zurück, etwa die Apex-Seite, die zur Bearbeitung eines Tasks angezeigt werden soll. Dafür müssen aber zunächst die Tasks eines Workflows mit entsprechenden Meta-Informationen angereichert sein. Dazu dient ein Activiti-spezifisches Dokumentations-Tag; die Meta-Informationen werden innerhalb dieses Tags durch Kommas getrennt in die Zeichen „/{...}/“ gefasst und so als Meta-Informationen kenntlich gemacht.

Zum jetzigen Entwicklungsstand können mithilfe dieser Meta-Informationen die zur Bearbeitung des Tasks aufzurufende Apex-Seite, allgemeine Informationen zu dem Task und zu setzende Apex-Page-Items angegeben werden. Durch „apex_page:page_number“ wird zum Beispiel die Seite „page_number“ aufgerufen und durch „set_page_item(page_item:activiti_process_variable)“ das Page-Item „page_item“ auf den Wert der Activiti-Prozess-Variablen „activiti_process_variable“ gesetzt. Über „get_info(activiti_process_variable)“

```
--Request erstellen
l_http_request := utl_http.begin_request(
  'http://192.168.56.101:8080/activiti-
  rest/service/runtime/tasks?candidateGroup=it_helpdesk
  &unassigned=true',
  'GET',
  'HTTP/1.1');

--Activiti Benutzerdaten setzen
utl_http.set_authentication(
  l_http_request,
  'admin',
  'admin');

--Request absetzen
l_http_response := utl_http.get_response(l_http_request);
utl_http.read_text(l_http_response, l_response_text);
utl_http.end_response(l_http_response);

--Response parsen
l_jsonObj := json(l_response_text);
l_jsonList := json_ext.get_json_list(l_jsonObj, 'data');
```

Listing 1

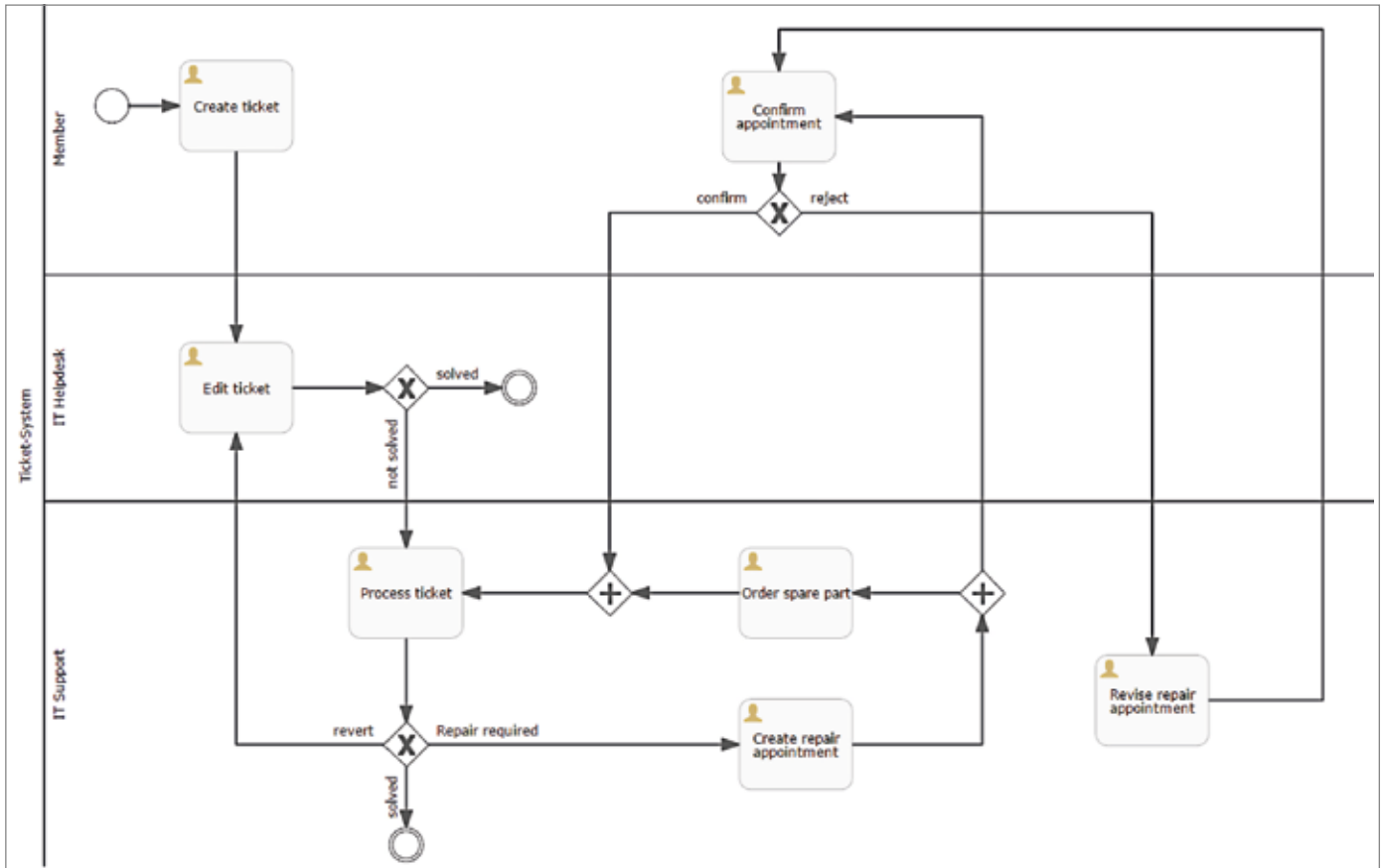


Abbildung 4: Workflow der Beispiel-Anwendung

lassen sich allgemeine Informationen zu dem Task abfragen, die zuvor durch eine Activiti-Prozess-Variablen über einen Webservice-Aufruf gesetzt wurden.

Eine vollständige Meta-Information besitzt das Format „/{apex_page:page_number,set_page_item(item:value) ,get_info(info)}/“. Auf diese Weise erfolgt die wechselseitige Steuerung zwischen Apex und Activiti, wie in *Abbildung 3* dargestellt. Durch Apex beziehungsweise entsprechende Webservice-Aufrufe werden Activiti-Prozesse gestartet, Benutzern Aktivitäten zugewiesen, Aktivitäten abgeschlossen sowie benötigte Activiti-Prozess-Variablen gesetzt. Activiti steuert dagegen über die beschriebenen Meta-Informationen den Ablauf der anzuzeigenden Apex-Seiten, setzt Apex-Page-Items und bestimmt, welche Aktivitäten als Nächstes ausgeführt werden können.

Über einen Webservice-Aufruf lassen sich beispielsweise alle Tasks abfragen, die noch keinem Benutzer zugeordnet sind und als Nächstes ausgeführt werden können (*siehe Listing 1*). Die Tasks werden in Form einer JSON-Liste zurückgeliefert und

können einfach geparkt werden, zum Beispiel mithilfe der PL/JSON-Bibliothek. Alternativ kann dies natürlich auch über das seit Apex 5.0 verfügbare Paket „Apex_JSON“ erfolgen.

Beispiel-Anwendung

Als Beispiel wurde ein einfaches Ticket-system implementiert. Der Workflow

wurde mit dem Activiti Modeler modelliert und ist in *Abbildung 4* dargestellt. Alle Tasks, die ein Benutzer ausführen kann, werden durch den oben beschriebenen Webservice-Aufruf abgefragt und dem Benutzer auf einer Übersichtseite zur Auswahl dargestellt. Die ausführbaren Tasks hängen dabei von der Rolle des Benutzers ab. Ein Benutzer mit der Rolle „IT-Helpdesk“ könnte beispielsweise die in *Abbildung 5* dargestellten Tasks sehen.



Abbildung 5: Auswahlseite der als Nächstes ausführbaren Tasks

Wird die Bearbeitung des Tasks abgeschlossen, geht der entsprechende Prozess in den nächsten Zustand. Der abgeschlossene Task verschwindet aus der Liste der als Nächstes ausführbaren Tasks des Benutzers mit der Rolle „IT-Helpdesk“ und die Benutzer mit der Rolle „IT-Support“ sehen einen neuen Task „Process Ticket“ in ihrer Liste der nachfolgend ausführbaren Tasks.

Die Apex-Seite, auf die ein Benutzer nach der Auswahl eines Tasks geleitet wird, hängt ausschließlich von den mit dem Task verknüpften Meta-Informationen ab. Der Task „Edit ticket“ wurde beispielsweise mit der Meta-Information „/{apex_page:301,set_page_item(P301_TICKET_ID:ticket_id)}“ versehen. Dadurch wird der Benutzer bei der Auswahl eines Tickets auf die Apex-Seite mit der Nummer „301“ geleitet und das Page-Item „P301_TICKET_ID“ auf den Wert der Activiti-Variablen „ticket_id“ gesetzt. Durch das Setzen der Variable „ticket_id“ beim Starten eines Activiti-Prozesses, der hier die Bearbeitung eines Tickets beschreibt, wird der Prozess mit einem in der Datenbank gespeicherten Ticket verknüpft.

Durch die Activiti-Integration müssen keinerlei Workflow-Informationen wie zum Beispiel darüber, welcher Task durch wel-

chen Benutzer ausgeführt wird, durch die Applikation gespeichert beziehungsweise verwaltet werden. Apex wird ausschließlich zum Bearbeiten von in der Datenbank gespeicherten Informationen (hier Tickets) verwendet, während der Workflow der Anwendung komplett durch Activiti gesteuert wird und ausschließlich in der in *Abbildung 4* dargestellten grafischen Beschreibung vorliegt.

Fazit und Ausblick

Durch die vorgestellte Methode lässt sich Apex grundsätzlich mit diversen Workflow-Engines beziehungsweise BPM-Plattformen integrieren. Die Voraussetzungen dafür sind, dass sich zum einen die zu integrierende Workflow-Engine über Webservices steuern lässt und sich zum anderen Tasks um Meta-Informationen anreichern lassen. Während beispielsweise eine Integration mit der Activiti-BPM-Plattform prototypisch durchgeführt wurde, besteht das Ziel in der Entwicklung eines Frameworks, über das sich verschiedene BPM-Plattformen integrieren lassen. Auch wenn die Integration zunächst einen recht hohen Entwicklungsaufwand fordert, ergibt sich der Vorteil, dass ein solches Framework in ver-

schiedenen Projekten wiederverwendet werden kann.

Konkrete Workflows, die in PL/SQL programmiert werden, sind dagegen sehr spezifisch und werden von Prozess-Analysten und den Anwendern der Applikation nicht verstanden. Durch die toolgestützte, grafische Entwicklung und die anschließende Einbindung dieser Workflows in Apex lässt sich der Rapid-Prototyping-Ansatz zumindest auf Teile der Programmlogik ausweiten. Zudem lassen sich Fehler in der Programmlogik minimieren, da die zeitaufwändige Re-Implementierung der Workflows in PL/SQL entfällt.



Sven Böttcher
sven.boettcher@appassociates.com

DOAG 2016 Datenbank

10. - 11. Mai 2016 | Düsseldorf

Oracle-Betrieb, Datenbank-Editionen, Hochverfügbarkeit



Apex Universal Theme – neue Hürden mit dem Responsive Layout

Tobias Arnhold, IT Consulting und Richard Rieb, DB System

Seit April letzten Jahres ist Apex 5 verfügbar. Vor allem das Universal Theme und das Responsive Design lösten große Begeisterung aus. Die ersten Entwicklungen haben jedoch gezeigt, dass das resultierende Layout doch nicht immer optimal ist. Dieser Artikel zeigt, wie man mit einfachsten Lösungen die durch das Responsive Design hervorgerufenen Layout-Probleme im Universal Theme beheben kann.

Das Universal Theme (UT) ist das neue Nutzer-Interface für Apex-Anwendungen. Es wurde erstmalig mit Apex 5 ausgeliefert. Eines der Kernelemente im UT ist das Responsive Design und dessen Responsive Layout (RL). Die darin enthaltene wichtigste Änderung im Design ist die Umstellung der bis Apex 4.2 verwendeten tabellenbasierten Darstellung auf einen festen Zwölf-Spalten-DIV-Element-basierten Aufbau der Seite. Diese und weitere technische Neuerungen sorgen dafür, dass mit UT erstellte Apex-Anwendungen sofort auf jedem Bildschirmformat optimal

dargestellt werden. Diese Anpassungsfähigkeit bietet eine sehr leichte Integration von Tablet- und Handy-Nutzern.

Desktop-Nutzer haben jedoch mit einigen Hürden zu kämpfen. So werden Interactive Reports und Forms-Masken verzerrt dargestellt. Zusätzlich erscheinen einzelne Seiten-Elemente wie Breadcrumbs übermäßig platzintensiv. Nachfolgend ist anhand einzelner Fallbeispiele erklärt, wie diese Probleme umgangen werden und sich in Reports- und Forms-Masken die gewohnte Desktop-Darstellung wiederherstellen lässt.

Interactive Report

Der Interactive Report ist eines der mächtigsten Werkzeuge von Apex und Kern der meisten Anwendungen. Er ermöglicht dem Endanwender, eigenständig individuelle Berichte zu definieren, und bietet eine Vielzahl von Funktionalitäten. Im Responsive Layout wird die Breite des Interactive Reports mindestens auf die Breite seines Containers ausgedehnt. Dabei werden alle angezeigten Spalten gleichmäßig auf diese Breite verteilt. Dies führt jedoch dazu, dass viele Spalten we-

Abbildung 1: Breite von Icon-Spalten im Interactive Report

sentlich breiter als ihr Inhalt angezeigt werden.

Insbesondere Spalten mit einem Icon sind davon betroffen, da diese nur eine vordefinierte, verhältnismäßig kleine Größe benötigen (siehe Abbildung 1).

Das Problem lässt sich mit zwei kleinen CSS-Erweiterungen lösen. Im Falle des „Bearbeiten“-Icons muss ein CSS-Snippet als Datei oder auf der betroffenen Seite „Page>CSS>Inline“ hinzugefügt werden (siehe Listing 1).

Anschließend muss dem Interactive Report noch die entsprechende CSS-Klasse durch „IR Region>Appearance>CSS Classes“: minimize-first-table-col mitgegeben werden. Bei allen anderen Spalten ist dem Report und der entsprechenden Spalte eine

```
.minimize-first-table-col td:first-child{
width: 1%;
overflow: visible;
}
```

Listing 1

```
"IR Region>Advanced>Static ID": reportid
"IR Region>Column>Advanced>Static ID": colid
```

Listing 2

```
#reportid th#colid, #reportid td[headers=colid]{
width: 1%;
overflow: visible;
}
```

Listing 3

Abbildung 2: Forms-Maske, visualisiert auf einem 24-Zoll-Bildschirm

ID zu erteilen (siehe Listing 2). Die Formatierung erfolgt durch ein CSS (siehe Listing 3).

Forms-Masken

Forms-Masken dienen in erster Linie der Bearbeitung von Datensätzen. Damit sind sie Bestandteil nahezu jeder Anwendung. Das UT hat den grundsätzlichen Aufbau einer Apex-Forms-Maske nicht

verändert. Durch das RL wurde die Visualisierung der Seitenelemente und Regionen jedoch insbesondere bei der Nutzung von großen Bildschirmen negativ beeinflusst. So sind auch Forms-Masken durch die automatisierte Anpassung an die Bildschirmgröße verzerrt. Dies resultiert in Leerräumen innerhalb der Seite.

Der Grund für dieses Verhalten ist, dass das UT Objekte, wie Seiten und Re-

gionen, in zwölf Spalten unterteilt. Standardmäßig nehmen diese zwölf Spalten zusammen die gesamte Bildschirmbreite ein, wobei einzelne Objekte gleichmäßig auf die Spalten aufgeteilt sind. Bei zwei Regionen nimmt damit jede Region sechs Spalten und fünfzig Prozent der Bildschirmbreite ein (siehe Abbildung 2).

Um dieses Verhalten zu umgehen, muss das responsive Verhalten reglementiert werden. Eine Lösung besteht darin, den Regionen mitzugeben, in welcher Spalte sie beginnen sollen. Dadurch

```
@media (min-width:550px) {
  .col-600 {
    min-width: 600px !important;
    max-width: 600px !important;
  }
}
```

Listing 4

```
.t-Breadcrumb-item{
  font-size: 1.2rem !important;
}
.t-Breadcrumb-item.is-active{
  font-size: 2rem !important;
  line-height: 1.2em !important;
}
```

Listing 5

```
.t-BreadcrumbRegion-body {
  padding-top: 10px;
  padding-bottom: 10px;
}
```

Listing 6



Abbildung 3: Breadcrumb im UT

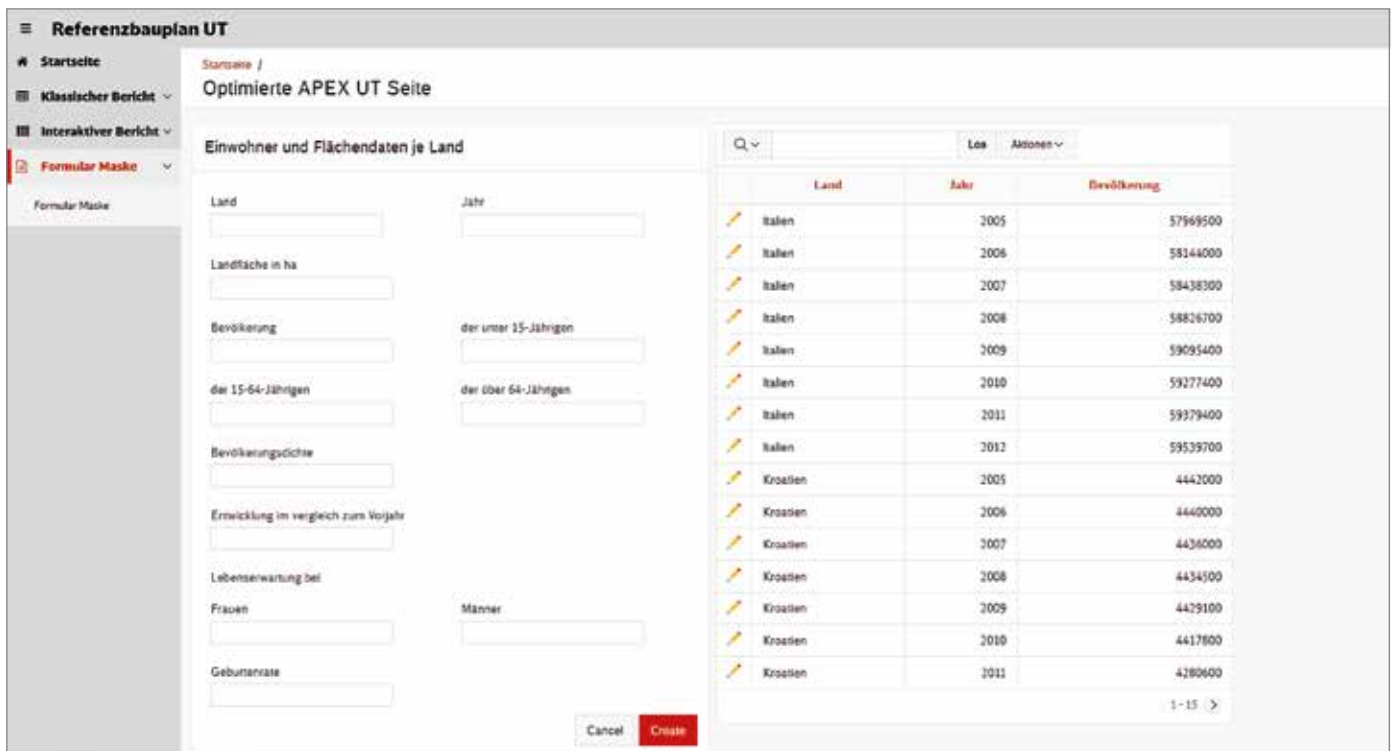


Abbildung 4: Ergebnis mit allen Design-Anpassungen

werden die Breitenverhältnisse entsprechend den Definitionen berechnet. Bei den links ausgerichteten Regionen ist dies nicht notwendig, da diese automatisiert in der ersten Spalte beginnen. Um beispielsweise zu definieren, dass die erste Region lediglich fünfundzwanzig Prozent der Breite einnimmt, ist die zweite Region mit „Region Source>Grid>Column“:4“ auf Spalte 4 zu definieren. Dies führt jedoch auch bei kleineren Bildschirmgrößen zu einer Verkürzung der Region, die in vielen Fällen nicht mehr optimal ist. Daher empfiehlt es sich, die Breite der Regionen über CSS-MEDIA-Queries zu definieren (siehe Listing 4).

Im Anschluss ist beiden Regionen noch mit „Region Source>Appearance>CSS Classes“: col-600“ die entsprechende CSS-Klasse zuzuweisen. Dadurch werden die Regionen untereinander angezeigt, sobald eine kleine Bildschirmgröße dafür sorgt, dass die Mindestbreite im RL unterschritten ist.

Breadcrumbs

Breadcrumbs zeigen Nutzern, in welcher Verzweigung der Anwendung sie sich gerade befinden, und ermöglichen es, direkt zu einer übergeordneten Seite zu springen. Sie werden üblicherweise oben auf der Seite in einer Textzeile angezeigt. Das UT nutzt Breadcrumbs auch als Seitenüberschriften. Dabei wird ihnen übermäßig viel Platz auf der Seite zugewiesen, der durch andere Bereiche effektiver genutzt werden könnte (siehe Abbildung 3).

Dies liegt daran, dass Breadcrumbs eine Schriftgröße von 2,4 beziehungsweise 1,4 rem im Fall des Navigationspfads haben und einen Innenabstand von 24 px definieren. Um Breadcrumbs optimaler darzustellen, muss zunächst die Schriftgröße mittels CSS angepasst werden (siehe Listing 5). Ein CSS verringert den Innenabstand (siehe Listing 6). Abbildung 4 zeigt die beschriebenen Beispiele noch einmal im Zusammenspiel.



Tobias Arnhold
tobias-arnhold@hotmail.de



Richard Rieb
richard.rieb@deutschebahn.com

DOAG 2016 BI

8. - 9. Juni 2016 in Bonn



Apex mobile vs. Universal Theme

Steven Grzbielok, MT AG

Die größte Neuerung und Erleichterung durch Apex 5 ist für jeden Apex-Entwickler wohl das Universal Theme: Es ermöglicht die Entwicklung einer Responsive-Design-fähigen Anwendung, die sowohl auf dem Desktop als auch auf mobilen Endgeräten eine komfortable Nutzung erlauben soll. Doch was ist aus Apex mobile geworden? Wo stößt das Universal Theme an seine Grenzen und wo sollte man weiterhin auf Apex mobile zurückgreifen? Dieser Artikel gibt einen Überblick über die Unterschiede und jeweiligen Vorteile, um die Entscheidung zwischen den beiden Themes zu erleichtern.

Für die in *Abbildung 1* aufgeführte Gegenüberstellung wurde die Sample Database Application verwendet. Wer beide Themes testen möchte, kann auf „apex.oracle.com“ einen kostenfreien Workspace erstellen und die Packaged Application „Sample Database Application“ installieren. Sie wird direkt mit beiden Themes installiert. Ein Button schaltet zwischen Apex mobile und dem Universal Theme um.

Universal Theme – one size fits all?

Während frühere Apex-Themes das Responsive Design lediglich unterstützten, wurde dies durch die Einführung des Universal Theme in Apex 5 grundlegend erweitert: Bisher hatte der Entwickler bei der Erstellung einer Apex-Anwendung die Möglichkeit, aus vielen verschiedenen Themes zu wählen. Diese Entscheidung brachte dementsprechend unterschiedliche Features mit sich, besonders im Hinblick auf die mobile Nutzung. Manche Themes waren geeigneter als andere und je nach Anforderung an die Templates war hier einiges bei der Auswahl zu bedenken.

Oracle hat nun für Apex das Universal Theme eingeführt, wobei der Name schon die Intention dahinter vermittelt: Ein Theme, das allen gängigen Anwendungsfällen und Geräten gerecht werden soll. Oracle hat gezielt hierfür die ID nicht fortlaufend weitergeführt, sondern die ID 42 (die Antwort auf

die Frage nach dem Leben, dem Universum und allem Übrigen aus „Per Anhalter durch die Galaxis“) gewählt.

Das Universal Theme ist komplett auf die Nutzung auf sowohl mobilen als auch Desktop-Geräten ausgerichtet. Es werden die neuesten Standards hinsichtlich Webentwicklung wie HTML5 und CSS3 genutzt. Während früher die Navigation als Tabs verwendet wurde, wurde sie nun links in einer Baumstruktur implementiert (siehe *Abbildung 2*). Diese Navigationsleiste lässt sich mit dem weitverbreiteten „Hamburger“-Icon (dargestellt als drei

waagerechte Balken) ein- und ausklappen. Auf dem Desktop wird diese Navigation im Standard komplett mit Icons und Menüeinträgen eingeblendet, auf dem Tablet sind nur Icons für die Navigation zu sehen und auf dem Smartphone ist die Navigation im Standard komplett ausgeblendet.

Dieses Verhalten variiert jedoch immer in Abhängigkeit von der Bildschirmgröße, sodass etwa bei einem sehr großen Tablet die komplette Navigation angezeigt oder bei einem kleinen Netbook die Navigation auf die Icons beschränkt wird. Bei Bedarf kann der Nutzer jederzeit die Navigation

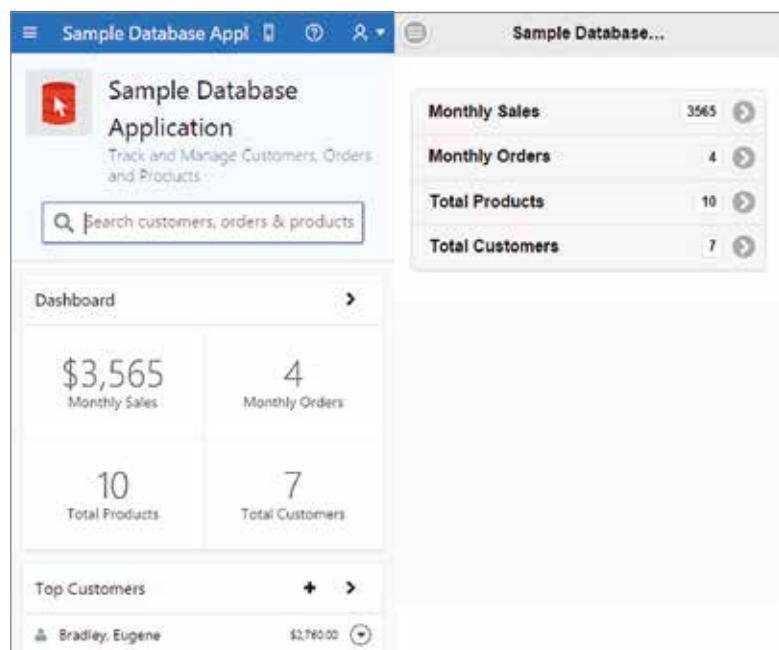


Abbildung 1: Sample Database Application im Universal Theme (links) und Apex mobile (rechts)

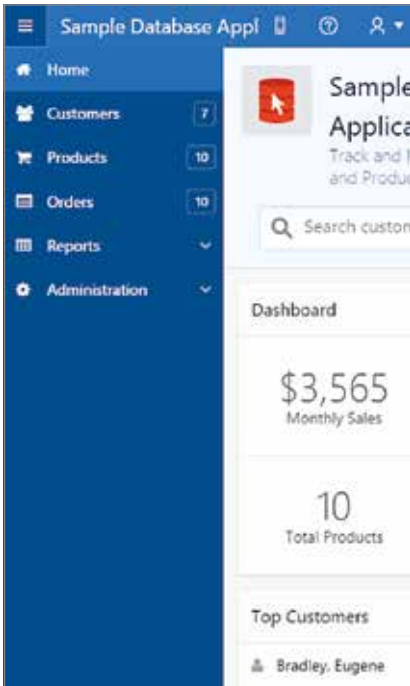


Abbildung 2: Die neue Navigation in einer Baumstruktur mit „FontAwesome“-Icons

ausklappen, um bequem durch die Seiten zu navigieren.

Außerdem fällt auf, dass die Schriftgröße der Texte sich ebenfalls an die Bildschirmgröße anpasst: Um nicht weniger Informationen auf kleinen Bildschirmen darzustellen, werden die Überschriften und Textkörper bei einem kleineren Bildschirm automatisch verkleinert.

Die Komponenten wurden in Apex 5 ebenfalls so überarbeitet, dass auch Reports so komfortabel wie möglich auf allen Bildschirmgrößen nutzbar sind. Zu beachten ist jedoch, dass vor allem Interactive Reports und Tabular Forms mit besonders vielen Spalten problematisch auf Smartphones werden und zu horizontalem Scrollen führen können oder lediglich im Querformat eine komfortable Nutzung ermöglichen (siehe Abbildung 3).

Regionen auf einer Seite werden nun automatisch in einem überarbeiteten Grid-Layout angeordnet: Eine Seite besteht aus zwölf Spalten. Man kann wählen, wie viele Spalten eine Region umfassen soll, oder Apex diese Entscheidung überlassen. Bei den unterschiedlichen Bildschirmgrößen vergibt Apex nun für die jeweiligen Regionen eine Breite in Abhängigkeit von den gewählten Spalten. Sobald der Bildschirm jedoch eine gewisse Größe unterschreitet, sodass diese

Spalten zu schmal werden und somit eine Region ebenfalls nicht mehr komfortabel nutzbar ist, werden die Regionen untereinander dargestellt. So kann auf einem Smartphone dieselbe Anwendung wie auf dem Desktop genutzt werden, trotz der unterschiedlichen Bildschirmverhältnisse.

Die meisten der Komponenten im Universal Theme wurden ebenfalls so entwickelt, dass Responsive Design ermöglicht wird. Es werden in den meisten Fällen keine manuellen CSS-Anpassungen benötigt: Listen-Regionen wie Cards werden zunächst nebeneinander dargestellt, wobei die Größe der Karten sowie deren Anordnung komplett abhängig von der Bildschirmgröße dynamisch ermittelt werden.

Die neue Kalender-Region bietet zunächst auf dem Desktop eine Monatsansicht, die auf Smartphones (hochkant) viel zu breit wäre. Hier wechselt die Region automatisch in eine Listen-Ansicht, sodass alle Termine übersichtlich untereinander dargestellt und überflüssige Tage nicht angezeigt werden.

Zusätzlich wurde auch Tap Event Handling in dem Universal Theme eingeführt: Bei der Nutzung von Anwendungen auf einem Smartphone werden Berührungen mit einer Verzögerung von 300 ms interpretiert. Dies ist bei vielen nativen Apps eine Standard-Eigenschaft, da eine Berührung nicht zwangsläufig ein absichtlicher Klick ist. Die Verzögerung ermöglicht es der Anwendung, zwischen einem Klick durch Tippen oder dem Scrollen durch eine ziehende Berührung auf dem Bildschirm zu differenzieren.

Das Universal Theme wurde dahingehend optimiert, um derartige mobile Standards out of the box bei der Nutzung eines Touchscreens zu ermöglichen. Zusätzlich bietet das Universal Theme durch die Einführung von Template Options noch mehr Möglichkeiten, um ohne aufwändige CSS-Modifizierungen eine Anwendung „mobile friendly“ zu gestalten. Der Entwickler kann viele Einstellungen deklarativ definieren und so CSS-Klassen via Mausclick setzen. Sollten Tests auf mobilen Geräten nun Probleme in der Darstellung offenbaren, lassen sich durch die Template-Options verschiedene Einstellungsmöglichkeiten ausprobieren, um so die beste Option zu finden.

Die in Apex 5 eingeführten modalen Dialoge sind ebenfalls mobil verfügbar und bieten auch hier Vorteile: Zum Beispiel lässt sich ein Datensatz eines Reports bearbeiten, ohne dass die Seite gewechselt werden muss. Das Formular zur Bearbeitung wird einfach als modaler Dialog (vergleichbar mit einem Popup) eingeblendet und sobald die Bearbeitung beendet ist, lässt sich die Berichtsregion neu laden. Insgesamt hat das Universal Theme einen deutlichen Fortschritt hinsichtlich Responsive Design gemacht, sodass ein Apex-Entwickler ohne viel Aufwand eine Anwendung für verschiedene Endgeräte und Bildschirmgrößen entwickeln kann.

Mehraufwand für Apex mobile

Während das Universal Theme noch sehr neu ist, ist Apex mobile bereits seit der

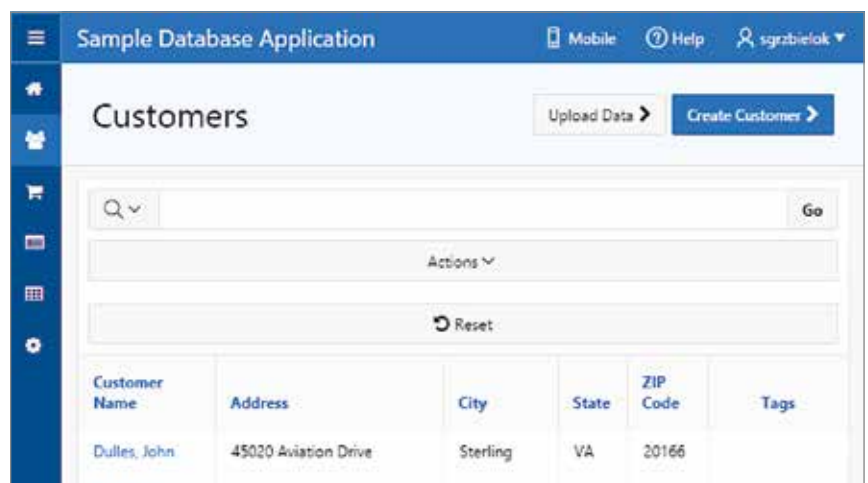


Abbildung 3: Im „Landscape“-Modus lassen sich auch Interactive Reports noch bequem auf dem Smartphone nutzen

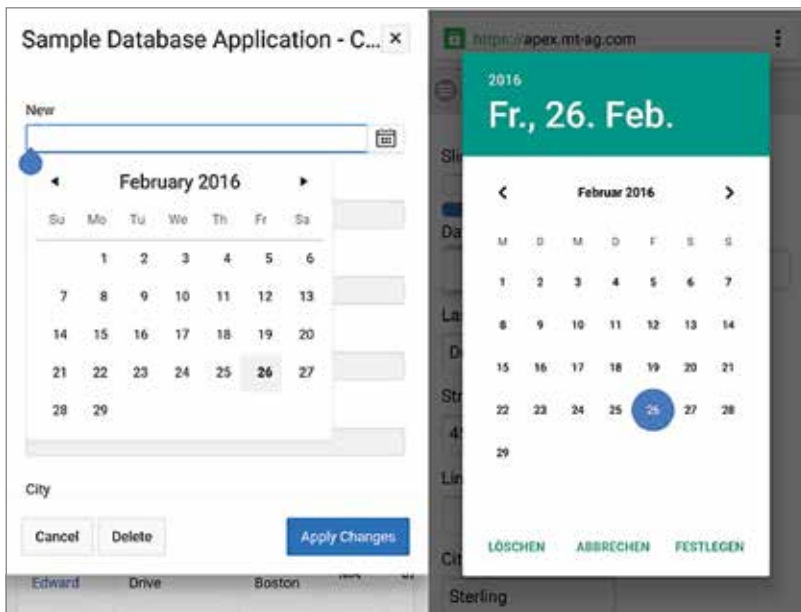


Abbildung 4: Apex mobile (rechts) unterstützt im Gegensatz zum Universal Theme (links) native Form-Elemente

Einführung von Apex 4.2 (im Jahr 2012) verfügbar. Apex mobile wurde ausschließlich für die Nutzung auf mobilen Endgeräten entwickelt und basiert auf dem jQuery-Mobile-Framework. Eine Nutzung der Anwendung ist zwar ebenfalls auf dem Desktop möglich, jedoch sollte ein separates User-Interface mit einem anderen Theme für den Desktop genutzt werden, da das Design von Apex mobile nicht für den Desktop-Monitor optimiert ist. Es wurde hier auf eine besonders hohe Kompatibilität mit verschiedenen mobilen Plattformen geachtet, damit möglichst unabhängig vom Endgerät die Funktionalität und der Komfort garantiert werden können.

Apex mobile ist im Gegensatz zum Universal Theme deutlich leichter konstruiert, es müssen also weniger HTML, CSS und JavaScript geladen werden, um eine Seite aufzubauen. Da auf mobilen Endgeräten oft die Netzwerk-Verbindung limitierend für Anwendungen wirken kann, wird die Seite auch bei schlechtem Empfang schnell aufgebaut und verbraucht weniger Datenvolumen. In Apex mobile werden dem Entwickler spezielle Form-Elemente zur Verfügung gestellt, die besonders komfortabel auf einem Touchscreen nutzbar und nach Möglichkeit auch mit den nativen Elementen des jeweiligen mobilen Betriebssystems verknüpft sind (siehe Abbildung 4). So wer-

den zum Beispiel Datums-Eingaben mit den Android- beziehungsweise Apple-Date-Picker verwendet, damit der Nutzer wie gewohnt sein Gerät nutzen kann. Slider und Kippschalter können in Formulare eingebunden werden, um die Bearbeitung der Daten auch mit Touch-Gesten zu ermöglichen.

In Apex mobile wurden ebenfalls einige Regionen integriert, die ausschließlich auf mobilen Geräten verfügbar sind: Die Reflow Table dreht die Ansicht, sobald der Bildschirm zu schmal für einen herkömmlichen Report wird. Es werden die Spalten horizontal dargestellt und dann untereinander die einzelnen Datenreihen angezeigt. So muss der Nutzer ausschließlich vertikal scrollen und behält auch auf einem kleineren Bildschirm den Überblick über komplexere Datensätze.

Der Column Toggle Report hingegen ist ein einfacher Report, der lediglich so viele Spalten darstellt, wie auf dem Bildschirm ausreichend Platz finden. Der Nutzer kann dann über Checkboxes auswählen, welche Spalten er angezeigt haben möchte.

Als Best Practice werden zudem häufig in Reports lediglich die ID und der Name (beziehungsweise eine fachliche Bezeichnung) eines Datensatzes in Listenform angezeigt. Der Nutzer kann dann über einen Klick auf den Datensatz zu einer Detailansicht gelangen.

Die intuitive Nutzung von Touch-Gesten ist mittlerweile ein Standard auf mobilen Endgeräten. In Apex mobile lassen sich diese Touch-Events auch für Dynamic Actions verwenden. Beispielsweise wird ein Wischen über den Bildschirm so von Apex erkannt und kann eine entsprechende Aktion auslösen. Mit Apex mobile wird versucht, dem Nutzer das Look & Feel einer nativen Anwendung im Browser zu vermitteln.

Der Header und der Footer sind im Standard fixiert und bilden den Rahmen der Anwendung. Der Seitenaufruf findet durch AJAX statt, es muss also nicht die komplette Seite neu aufgebaut, sondern per JavaScript lediglich der Body geladen werden. Dies führt zu einer schnelleren Reaktionszeit und verstärkt das App-Gefühl. Apex mobile bietet somit eine Vielzahl spezifischer mobiler Features, die aktuell im Universal Theme nicht deklarativ nutzbar sind. Weniger manuelle CSS- und JavaScript-Anpassungen sind hier notwendig.

Was ist besser?

Das Universal Theme bietet eine schnelle Möglichkeit, Anwendungen für Desktop und mobile Endgeräte zu entwickeln. Die Nutzung von Apex mobile hingegen bedeutet einen erheblichen Mehraufwand, da hier eine doppelte Entwicklung und Anwendungspflege nötig ist. Dafür ist Apex mobile jedoch deutlich optimierter für mobile Geräte und bringt einige Features ausschließlich für mobile Geräte mit.

Um zu entscheiden, welches Theme in einem speziellen Use Case am sinnvollsten ist, sollte man sich zunächst selbst einige Fragen stellen: Mit welcher Priorität wird die Anwendung mobil genutzt? Wird die Anwendung primär auf einem Desktop eingesetzt und sollen nur sekundär mobile Geräte unterstützt werden? Oder liegt der Fokus klar auf mobilen Geräten? Was ist wichtiger: Design oder Funktionalität?

Das Universal Theme hat deutlich mehr in Sachen Optik zu bieten als Apex mobile: Es wirkt wesentlich schicker, wohingegen Apex mobile eher einen nüchternen Eindruck macht.

Die integrierte Nutzung des Theme Roller sowie die ansprechende, an Googles Material Design orientierte Gestaltung des Theme machen einen sehr moder-

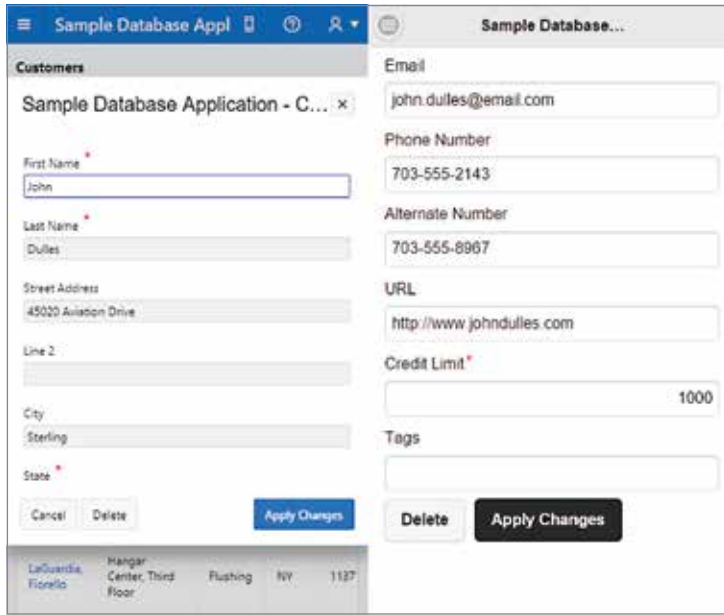


Abbildung 5: Die Nutzung von Formularen ist bei Apex mobile (rechts) deutlich bequemer, da sich die Elemente intuitiver gestalten und sich die Größe präziser am Bildschirm ausrichtet

nen Eindruck. Dem Nutzer wird direkt der Eindruck vermittelt, dass er mit einer hochaktuellen Anwendung arbeitet. Jedoch hat eben diese Optik seinen Preis: In einem Test bei regulärer 3G-Verbindung dauert der Seitenaufbau mit dem Universal Theme mehr als eine Sekunde länger. Die benötigte Datenmenge, die vom Server auf das Endgerät übertragen wird, ist nahezu doppelt so groß. Dies mag zwar bei gelegentlicher Nutzung nicht besonders gravierend sein, jedoch macht es gerade bei schlechter Netzwerkverbindung und sehr intensiver Nutzung der Applikation auf Dauer einen Unterschied.

Durch den Seitenaufbau per AJAX vermittelt Apex mobile in den Tests einen noch zügigeren Eindruck beim Seitenwechsel. Der Nutzer verwendet zwar auf seinem mobilen Gerät einen Browser, jedoch wird durch die AJAX-Nutzung und den fixierten Header und Footer das Look & Feel einer nativen App vermittelt (siehe Abbildung 5).

Ein weiterer entscheidender Faktor ist die Auswahl der Regionen. Das Universal Theme bietet einige Standard-Regionen, die in Apex mobile nicht verfügbar sind: Classic Report, Interactive Report, Tabular Form, Map Chart, Region Display Selector und Tree. Besonders das Map Chart und der Interactive Report bieten Funktionalitäten, auf die man oft nicht verzichten kann. Diese Regionen wer-

den jedoch bewusst nicht in Apex mobile angeboten, da auf einem kleinen Bildschirm ein zu komplexes User-Interface nicht dargestellt werden kann. Hier muss entschieden werden, ob eine korrekte Darstellung oder die gebotenen Funktionalitäten der Regionen wichtiger sind.

Apex mobile bietet jedoch ebenfalls einige Regionen an, die wiederum nicht im Universal Theme nutzbar sind: Column Toggle Report, List View und Reflow Re-

port. Diese sind speziell für mobile Geräte entwickelt worden und bieten viele Vorteile hinsichtlich Usability, die von den Regionen des Universal Theme nicht erfüllt werden können. Entsprechend den Anforderungen der Nutzer sollte nun beurteilt werden, welche Regionen eher die benötigten Funktionen liefern können (siehe Abbildung 6).

Letztendlich kommt es bei der Entwicklung mit den verschiedenen Themes darauf an, wie wichtig in dem jeweiligen Anwendungsfall die Nutzung auf mobilen Endgeräten ist. Liegt der Fokus auf mobilen Geräten, sollte weiterhin Apex mobile zum Einsatz kommen, da durch Touch-Events eine bessere Performance und ein natives Look & Feel eine komfortablere und zuverlässigere Nutzung möglich ist. Auch die speziell für mobile Geräte entwickelten Regionen und Komponenten bieten hier viele Vorteile.

Wird die Anwendung jedoch primär für Desktops entwickelt und soll lediglich auch auf mobilen Geräten lauffähig sein, so ist dem Universal Theme der Vorzug zu geben. Apex nimmt dem Entwickler hier in den meisten Fällen schon die Arbeit ab und stellt die Anwendung im responsiven Design dar. Ebenfalls sind die gewohnten Regionen auf dem Desktop und auf dem mobilen Gerät verfügbar, wodurch an keiner Stelle Einbußen bezüglich der Funktionalitäten in Kauf genommen werden müssen.

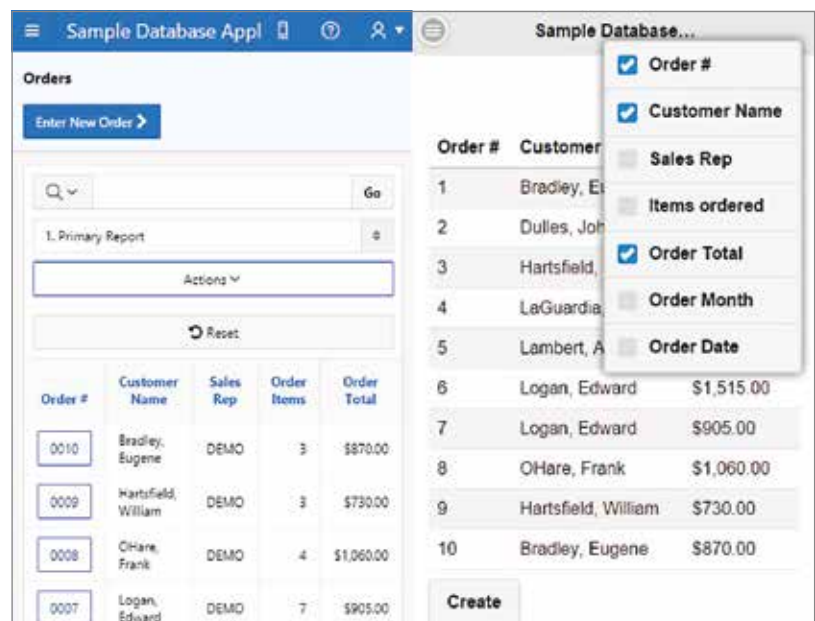


Abbildung 6: Interactive Reports (links) bieten mehr Funktionen zum Suchen und Filtern, Column Toggle Reports (rechts) sind jedoch übersichtlicher

Da komplexe Regionen jedoch auf mobilen Geräten unter Umständen nicht korrekt dargestellt werden, sollte die Anwendung besonders auch auf kleinen Bildschirmgrößen getestet werden. Zu komplexe Regionen oder Seiten sollte man gegebenenfalls bewusst ausblenden, damit überladene Seiten und zu viel Scrollen vermieden werden. Um eine komfortable Nutzung auf mobilen Endgeräten zu garantieren, ist das ausführliche Testen auf verschiedenen Geräten bei beiden Themes essenziell.

Was die Zukunft bringt

Das Universal Theme wird in den kommenden Updates viele Verbesserungen erhalten. Vorrangig für den Desktop interessant, werden die neue Region „Interactive Grid“ als Ablösung der Tabular Forms eingeführt und die Master-Detail-Ansicht optimiert. Zusätzlich wird eine neue Charting-Engine eingeführt: Oracle-JET-Charts werden künftig als Standard-Charting-Engine für Apex eingesetzt. Es werden neue

und verbesserte deklarativ erstellbare Diagramme verfügbar sein. Da Oracle JET jedoch noch nicht alle Diagramm-Typen unterstützt, die aktuell durch AnyCharts nutzbar sind, werden diese zunächst weiterhin deklarativ mit AnyCharts verfügbar sein. Außerdem soll Tap-Event-Handling ab Apex 5.1 ebenfalls deklarativ für Dynamic Actions im Universal Theme genutzt werden können.

Für Apex 5.1 sind aktuell keine speziellen neuen Features für Apex mobile geplant, jedoch wird Apex mobile ebenfalls Features wie Oracle-JET-Charts nutzen können. Der integrierte Theme Roller, der bereits für das Universal Theme eingesetzt werden kann, soll ebenfalls für Apex mobile verfügbar sein. Aktuell kann für Apex mobile nur der externe jQuery Mobile Theme Roller verwendet werden. Außerdem wird daran gearbeitet, dass auch neue Updates von jQuery Mobile in Apex integriert werden.

Nach einer langen Pause werden aktuell die Arbeiten an jQuery Mobile 1.5 weitergeführt und frühestens in Apex 5.2 können Apex-Entwickler ebenfalls mit

der Einbindung dieser Neuerungen rechnen. Damit Apex 5.1 vor dem Release getestet werden kann, arbeitet Oracle wieder an einem Early Adopter.

Eine erweiterte Gegenüberstellung von Apex mobile und dem Universal Theme in Verbindung mit Oracles Mobile Application Framework (MAF) wird auf der Apex connect 2016 von Davide Groppuso und Steven Grzbielok vorgestellt. Weitere Informationen hierzu auf „apex.doag.org“.



Steven Grzbielok
steven.grzbielok@mt-ag.com

Die aktuellen Trends der Logistik-Branche und deren IT-Herausforderungen



DOAG 2016 Logistik + IT

@ CeMAT Hannover | 2. Juni 2016



Apex on the Rocks – Hochverfügbarkeit für eine komplette Apex-Umgebung

Moritz Klein, MT AG

Häufig starten Oracle-Application-Express-Anwendungen als kleine, von Fachabteilungen gestartete Initiativen, entwickeln sich dann aber schnell zu geschäftskritischen Anwendungen. Um diesem Umstand gerecht zu werden, reicht der Rechner unter dem Schreibtisch nicht mehr aus. Dieser Artikel zeigt eine mögliche Architektur, um Apex zukunftssicher zu betreiben.

Oracle Application Express bietet als Plattform für die Entwicklung von Web-Anwendungen einen sehr schnellen Einstieg. Zu Beginn machen sich viele deshalb meist wenig Gedanken über die notwendige Infrastruktur. Häufig wird eine vorhandene Datenbank-Installation oder auch die kostenfreie Express Edition genutzt und in der Regel werden diese Umgebungen dann durch einen versierten Mitarbeiter aus der Fachabteilung betrieben. Die Einfachheit, mit der man eine zunächst völlig ausreichende Umgebung aufbauen kann, ist eine der Stärken von Apex. Weitere Stärken, wie große Flexibilität und hohe Entwicklungsgeschwindigkeit, führen dann zu einer Art viralem Marketing. Begeisterte Nutzer erzählen ihren Kollegen von erreichten Arbeitserleichterungen, dies führt dann zu einem größeren Nutzerkreis und zusätzlichen Applikationen.

Apex ist aber bei Weitem nicht auf kleine Applikationen beschränkt, sondern auch – unter anderem durch die Verbindung mit der Oracle-Datenbank – absolut für geschäftskritische Anwendungen tauglich. Sobald man sich an geschäftskritische Anwendungen heranwagt, sollte man sich allerdings umfangreiche Gedanken über die notwendige Umgebung für solche Anwendungen machen.

Die vom Autor betreute Apex-Umgebung stand im September 2014 genau

an dieser Stelle und es musste dringend eine neue, zukunftssichere Umgebung her. Bis dahin lief Apex auf einem gemeinsam genutzten Datenbank-Server; der Web-Server war als virtuelle Maschine aufgesetzt. Durch die hohe Menge an unterschiedlichen Datenbanken auf dem gleichen Server beeinträchtigten sich diese zum Teil gegenseitig und es wurde zunehmend schwieriger, Wartungsfenster zu finden. Glücklicherweise hatten die Anwendungen schon einen so hohen Stellenwert erreicht, dass man aus dem Vollen schöpfen konnte. Schnell war klar, dass eine Cluster-Lösung zum Einsatz kommen sollte – vor allem, um Hoch-

verfügbarkeit zu gewährleisten. Aber zunächst eine kurze Bestandsaufnahme, um die abzusichernden Komponenten zu identifizieren.

Apex-Architektur

Apex als solches besteht nur aus Metadaten-Tabellen und PL/SQL-Paketen. Lässt man die statischen Dateien wie Bilder, Stylesheets und JavaScript-Dateien außen vor, liegen alle benötigten Teile in der Datenbank. Zur Auslieferung der erzeugten Webseiten benötigt man nur noch einen Web Listener. Dies kann prinzipiell

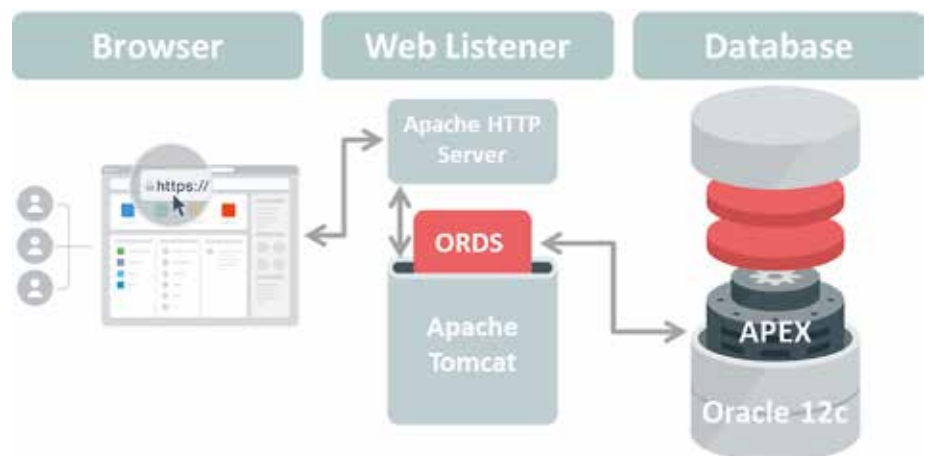


Abbildung 1: Apex-Architektur

Oracle-XDB („Embedded PL/SQL Gateway“) sein, davon sollte man allerdings in Produktions-Umgebungen Abstand nehmen. Eine heute übliche Umgebung besteht aus den folgenden Komponenten (siehe Abbildung 1):

- Oracle Database Server 12c
- Oracle REST Data Services (ORDS)
- Apache Tomcat
- Apache HTTP Server

Basierend auf der generellen Architektur kann man die Analyse von Cluster-Lösungen auf zwei Gesichtspunkte aufteilen: Absicherung der Datenbank (inklusive Apex) und Absicherung des Web-Tier.

Absicherung der Datenbank

Um einen Überblick über von Oracle zur Verfügung gestellte Produkte und Datenbank-Optionen zu erhalten, lohnt sich der Blick in die Dokumentation, in diesem Fall „Database High Availability Overview“ (siehe „<https://docs.oracle.com/database/121/HAOVW/toc.htm>“). Wer direkt die empfohlenen Architekturen für die verschiedenen Service-Level sehen will, kann dabei direkt zu Kapitel 7 springen. Oracle unterscheidet hier in Bronze, Silber, Gold und Platinum mit immer höheren Anforderungen an die maximal erlaubte Ausfallzeit.

Die Entscheidung im Projekt fiel hierbei auf die Silber-Architektur, dabei handelt es sich um die Nutzung von Oracle Real Application Clusters (RAC). Eine Lösung mittels Data Guard wurde alternativ untersucht, da auch hiermit die generel-

len Ansprüche an Hochverfügbarkeit erfüllt worden wären. Allerdings bietet RAC die Möglichkeit, durch Hinzufügen von weiteren Servern flexibel auf geänderte Anforderungen zu reagieren, um zum Beispiel später zusätzliche Apex-Installationen im Unternehmen auf einer zentralen Plattform zu konsolidieren.

Real Application Clusters

Real Application Clusters (RAC) erlaubt den Zugriff mehrerer Datenbank-Instanzen von verschiedenen Servern im Cluster auf gemeinsame Daten-Dateien, auch als Datenbank bezeichnet. Diese Datenbank erstreckt sich also über mehrere Server, stellt sich der Applikation aber als eine einheitliche Datenbank dar. Bei RAC handelt es sich um eine „shared disk“-Lösung, alle beteiligten Server benötigen also den Zugriff auf die gleichen Daten-Dateien. Die einzelnen Instanzen besitzen jeweils einen eigenen lokalen Buffer Cache, können jedoch auch auf Inhalte aus dem Buffer Cache einer anderen Instanz zugreifen, da alle Buffer Caches mittels Cache-Fusion logisch zusammengefasst sind.

RAC setzt auf Komponenten der Grid Infrastructure auf. Der Datenspeicher wird meist durch Automatic Storage Management (ASM) verwaltet und den Datenbank-Instanzen zur Verfügung gestellt. Die Oracle Clusterware übernimmt die Steuerung und Verwaltung aller Ressourcen im Cluster. Folgt man der von Oracle zur Verfügung gestellten Dokumentation und hält sich an die beschriebenen Hard- und Software-Anforderungen, ist die Installation nicht viel komplizierter als die einer Nicht-

RAC-Datenbank. Die Installations-Assistenten von Oracle sind mittlerweile auch wesentlich besser verständlich und damit auch besser nutzbar geworden.

Apex und RAC

Um Apex für den effizienten Einsatz mit RAC fit zu machen, bedarf es der Ausführung eines Skripts. Dieses findet sich unter dem Namen „apxpart.sql“ im Unterordner „utilities“ der Apex-Installations-Dateien. Es führt einige Datenmodell-Optimierungen für eine RAC-Umgebung aus. Dabei werden einige Tabellen mit Range-Partitionierung basierend auf der Session-ID versehen. Die führenden Zahlen der Session-ID orientieren sich dabei an der jeweiligen Instanz-Nummer. Kombiniert man das Ganze mit Instanz-Affinität auf dem Web Server, müssen Daten, die von der Session abhängen, seltener zwischen den Caches der RAC-Instanzen transferiert werden.

Möchte man sich einen Überblick über die durchgeführten Änderungen machen, verbindet man sich als „SYS“ und führt die Befehle aus Listing 1 aus. Steht keine RAC-Datenbank zum Testen zur Verfügung, kann man im Listing „partitions=auto“ durch „partitions=2“ ersetzen, um ein Zwei-Knoten-RAC zu simulieren. Die Optionen „print“ und „noexec“ sorgen dafür, dass die Befehle für die Datenmodell-Änderungen mittels „DBMS_OUTPUT“ ausgegeben, aber nicht ausgeführt werden. Trotzdem sollte man das natürlich nicht einfach auf einer Produktions-Datenbank ausprobieren.

Clusterware kann mehr als nur RAC

Die Oracle Clusterware kennen viele nur als Basis für RAC, es handelt sich jedoch um eine vollwertige Clusterware. Demnach lassen sich auch andere Applikationen und Prozesse überwachen. Bei den Überlegungen zur Absicherung der Web-Tier-Komponenten spielte dies eine maßgebliche Rolle und war einer der Gründe, RAC statt Data Guard zu benutzen. Schließlich ist es für den Betrieb der Umgebung von Vorteil, wenn man sich nur mit einer Cluster-Software vertraut machen muss.

```
SET SERVEROUTPUT ON
SPOOL apxpart.log
@apxpart partitions=auto,print,noexec
SPOOL OFF
```

Listing 1

```
yum install httpd #Apache installieren#
yum install tomcat tomcat-admin-webapps #Tomcat installieren#
systemctl start httpd #Apache starten#
systemctl stop httpd #Apache stoppen#
systemctl is-active httpd #Läuft Apache?#
```

Listing 2

Oracle liefert die Oracle Grid Infrastructure Agents aus, um ausgewählte Applikationen als Clusterware-Ressourcen zu verwalten. Hierzu kann man sich in der Ausgabe von „bin/agctl“ aus der Grid-Infrastructure-Installation anschauen, welche Applikationen konkret unterstützt werden. Für weitere Informationen sollte man auf jeden Fall auch den Oracle Reference Guide „Oracle Grid Infrastructure Agents“ nutzen (siehe „<http://www.oracle.com/technetwork/database/database-technologies/clusterware/downloads/ogiba-2189738.pdf>“).

Will oder kann man nicht auf die vorgefertigten Agents setzen, lohnt sich ein Blick in das Verzeichnis „crs/demo“. Weitere Informationen hierzu liefert der „Oracle Clusterware Administration and Deployment Guide“, speziell Kapitel 9, „Making Applications Highly Available Using Oracle Clusterware“ (siehe „<http://docs.oracle.com/database/121/CWADD/crschp.htm#CHDGHGJA>“).

Apache HTTP Server und Apache Tomcat installieren

Die Komponenten des Web-Tier sollten möglichst mit Betriebssystem-Mitteln installiert und betrieben werden. Solange man mit den vom Betriebssystem-Hersteller angebotenen Versionen zufrieden ist, ist diese Vorgehensweise auch generell zu bevorzugen. Die Integration in vorgegebene Verzeichnis-Strukturen und Betriebssystem-Werkzeuge ist dann einfach besser als bei einer Installation von Hand.

Im Projekt werden Red Hat Enterprise Server 7 eingesetzt und demnach per „yum“ die Web-Komponenten installiert. Als Werkzeug zur Dienste-Verwaltung kommt dann „systemctl“ zur Verwendung, das vom Betriebssystem mitgeliefert wird und entsprechend gut integriert ist. *Listing 2* zeigt hier die Befehle zur Installation und einige Beispiele zur Verwaltung der Dienste.

Der Apache HTTP Server wird als Proxy vor den Tomcat geschaltet und liefert statische Dateien direkt aus. Die generelle Konfiguration beider Komponenten kann in der jeweiligen Dokumentation nachgelesen werden und unterscheidet sich prinzipiell nicht von einer Konfiguration ohne Cluster. Bei den Oracle REST Data Services ist nur zu beachten, dass man den sogenannten „SCAN Listener“ und nicht etwa

```
appvipcfg create -network=1 -ip=192.168.56.250 -vipname=webVIP
-user=root
```

Listing 3

```
crsctl add type systemctl_agent_type
-basetype cluster_resource
-attr "ATTRIBUTE=SVC_NAME, TYPE=string, FLAG=REQUIRED,
ATTRIBUTE=ACTION_SCRIPT, TYPE=string,
DEFAULT_VALUE=systemctl_agent.sh"
```

Listing 4

```
case $1 in
...
'check')
/usr/bin/systemctl is-active $_CRS_SVC_NAME
RET=$?
...
esac
if [ $RET -eq 0 ]
then exit 0
else exit 1
fi
```

Listing 5

einen lokalen Listener adressiert, dies gilt aber generell, wenn man auf eine RAC-Datenbank zugreift.

Die Web-Komponenten absichern

Der Apache HTTP Server benötigt zuerst eine IP-Adresse, unter der er von den Clients erreichbar ist. Diese sollte immer die gleiche sein, egal auf welchem Knoten der Web-Server gerade läuft. Weiterhin muss die Clusterware erkennen können, ob die Komponenten laufen, und auch wissen, wie diese gestartet und gestoppt werden. Hängen Komponenten voneinander ab, sollte das natürlich auch konfiguriert werden. Schließlich muss noch entschieden werden, wie man für konsistente Konfiguration auf allen Knoten sorgt.

Virtuelle IP für den Web-Server

In der Clusterware gibt es einen Ressourcen-Typ „Virtual IP“ (VIP). Dieser wird auch für den SCAN-Listener benutzt, um die Zuordnung der IP-Adressen zu den einzel-

nen Knoten dynamisch verwalten zu können. Für den Web-Server kommt hier ein spezieller Typ „Application Virtual IP“ in Betracht, Ressourcen dieses Typs werden mittels appvipcfg verwaltet. Das Skript in *Listing 3* zeigt den Aufruf von „appvipcfg“ in einer Test-Umgebung, um eine entsprechende Ressource anzulegen. Es werden das Standard-Netzwerk des Clusters verwendet, die gewünschte IP bestimmt, der Ressourcen-Name festgelegt und „root“ als Besitzer spezifiziert.

Eigenen Ressourcen-Typ anlegen

Neben den von der Clusterware vordefinierten Ressourcen-Typen kann man auch eigene anlegen, um sich Vorlagen für ähnliche Ressourcen zu erstellen. Diese werden von einem Basis-Typ abgeleitet, um bereits definierte Attribute nicht erneut definieren zu müssen. Hier bietet sich der generische Typ „cluster_resource“ an.

Listing 4 zeigt den entsprechenden Aufruf, um einen eigenen Ressourcen-Typ „systemctl_agent_type“ anzulegen sowie die beiden Attribute „SVC_NAME“ und „ACTION_SCRIPT“ zu definieren. At-

```

crsctl add resource tomcat01 -type systemctl_agent_type -attr "SVC_NAME=tomcat, PLACEMENT=restricted,
CARDINALITY=1, SERVER_POOLS=*,
START_DEPENDENCIES='hard(webVIP,ora.data.asm_shared.acfs)
pullup(webVIP,ora.data.asm_shared.acfs)',
STOP_DEPENDENCIES='hard(webVIP,ora.data.asm_shared.acfs)'"

```

Listing 6

tribute werden durch die Clusterware als Umgebungsvariablen mit dem Präfix „_CRS_“ durchgereicht, das Attribut „SVC_NAME“ ist also über eine Umgebungsvariable namens „_CRS_SVC_NAME“ verfügbar. Das Attribut wird hier als erforderlich markiert, da das unter „ACTION_SCRIPT“ als Standard definierte Skript diese Variable benötigt.

Action-Skript erstellen

Die Komponenten des Web-Tier werden wie beschrieben mittels „systemctl“ gesteuert und der angelegte Ressourcentyp legt das vom Clusterware Agent aufzurufende Skript fest. Die Anforderungen an das Skript sind in der entsprechenden Dokumentation vermerkt, hier die wichtigsten: Das Skript muss auf die vier Kommandos „start“, „stop“, „check“ und „clean“ reagieren können und im Erfolgsfall mit Status „0“ enden. Unter Unix/Linux-Betriebssystemen gehört es zum

guten Ton, dass Skripte im Erfolgsfall „0“ und bei einem Fehler einen Wert zwischen „1“ und „255“ als Status-Code zurückgeben, „systemctl“ verhält sich hier auch entsprechend.

Ein eigenes Action-Skript als Clusterware-tauglicher Wrapper für „systemctl“ kann unter „<https://github.com/commi235/ocw-systemctl>“ inklusive einer kurzen Anleitung zur Integration heruntergeladen werden. Im Prinzip werden nur das Kommando („\$1“) vom Clusterware-Agent entgegengenommen, der entsprechende „systemctl“-Aufruf ausgeführt und der erhaltene Status an den Agent zurückgegeben. Listing 5 zeigt eine gekürzte Fassung, in der nur der Zweig für das Kommando „check“ sowie die Rückgabe an den Clusterware Agent detailliert dargestellt sind.

Die Konfigurations-Dateien

Mit dem Ressourcen-Typ und dem Action-Skript hat man die grundlegenden Bau-

steine, um eine eigene Ressource anzulegen. Allerdings ist es doch aufwändig und fehleranfällig, wenn man selbst für eine Synchronisation der Konfigurationsdateien und anderer von allen Knoten benötigten Dateien sorgen muss. Auch das Action-Skript muss natürlich auf allen Knoten, auf denen die Komponenten des Web-Tier laufen sollen, verfügbar sein. Hier gibt es verschiedene Möglichkeiten, wie zum Beispiel ein per NFS oder CIFS gemountetes Verzeichnis oder auch „rsync“.

Setzt man bereits ASM ein, kann man mithilfe von Automatic Storage Management Cluster File System (ACFS) auch einen Teil des von ASM verwalteten Speichers auf Betriebssystem-Ebene zur Verfügung stellen, wenn das verwendete Betriebssystem unterstützt wird. Zu Beginn des Projekts war ACFS noch nicht einsetzbar, da der notwendige Patch einen Konflikt mit dem Januar-PSU hatte, dies ist jedoch seit dem April-PSU behoben.

Hat man diese Hürde überwunden, kann man mit dem ASM-Konfigurations-Assistent (ASMCA) auf einer vorhandenen Disk Group ein Volume anlegen und im Betriebssystem mounten lassen. Abschließend kopiert man dann die benötigten Dateien auf den neu angelegten Speicherort, löscht die lokalen Kopien und zeigt per symbolischen Link auf den neuen Speicherort. Im Projekt wurden die Konfigurationsdateien des Apache HTTP Server, Apache Tomcat und ORDS, das Action-Skript und das Apex-Images-Verzeichnis so für alle Knoten im Cluster verfügbar gemacht.

Cluster-Ressourcen für Web-Tier anlegen

Nachdem jetzt alle Bausteine beisammen sind, können wir die einzelnen Ressourcen für die Komponenten des Web-Tier anlegen. Bevor man jedoch einfach eine Ressource einrichtet, sollte man sich

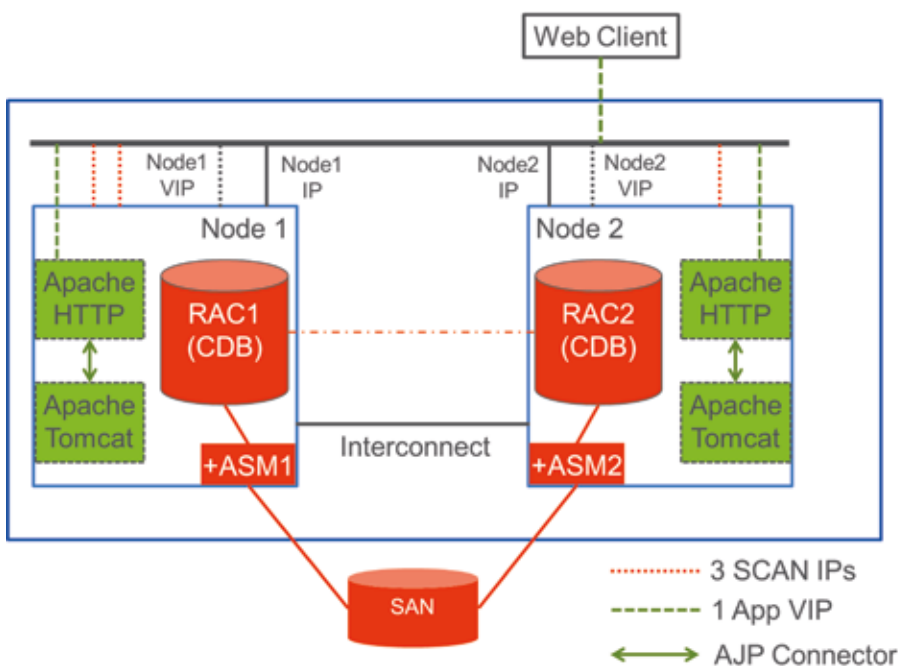


Abbildung 2: Fertige Cluster-Architektur

Gedanken um deren Abhängigkeiten machen. Diese sollte man der Clusterware mitteilen, sonst kann sie die Ressource nicht korrekt kontrollieren und verwalten. Liegen die Konfigurationsdateien für den Apache Tomcat auf einem ACFS-Speicherort, kann dieser auf einem Knoten ohne diesen Speicherort nicht starten. Ist die angelegte virtuelle IP auf dem Knoten nicht verfügbar, kann auch der Apache HTTP Server nicht richtig starten.

In der Clusterware werden diese Abhängigkeiten über die Attribute „START_DEPENDENCIES“ und „STOP_DEPENDENCIES“ der Ressource definiert. *Listing 6* zeigt das Anlegen der Ressource für Apache Tomcat exemplarisch, hierbei werden auch der vorher angelegte Ressourcen-Typ verwendet und das dort definierte Attribut „SVC_NAME“ mit dem entsprechenden Service-Namen belegt.

Einen Knoten-Ausfall simulieren

Nachdem alle benötigten Komponenten installiert und konfiguriert sind, sollte man natürlich einige Tests fahren. Als initiale Minimal-Tests kann man mittels „crsctl“ oder „srvctl“ Ausfälle der einzelnen Komponenten simulieren. Die Unterscheidung, wann man welches Skript benutzt, ist relativ einfach; beginnt eine Ressource mit „ora.“, sollte man prinzipiell „srvctl“ benutzen, ansonsten landet

man auch mal bei einer nicht mehr von Oracle unterstützten Konfiguration.

Zuerst verschafft man sich mit „crsctl status resource -t“ einen Überblick darüber, in welchem Zustand die Ressourcen sind und auf welchem Knoten sie laufen. Im angenommenen Ausgangszustand laufen die Ressourcen „apache01“, „tomcat01“ und „webVIP“ auf Knoten „1“, aufgrund der definierten Abhängigkeiten liegen alle drei auf dem gleichen Knoten. Das Kommando „crsctl eval relocate resource webVIP -f“ zeigt an, was passiert, wenn die Ressource „webVIP“ auf einen anderen Knoten umziehen muss. Als Erstes gehen „apache01“ und „tomcat01“ auf Knoten „1“ in den Zustand „offline“. Sobald die IP-Adresse auf Knoten „2“ „online“ ist, starten dort auch die beiden Web-Server.

Um den Ausfall von ACFS auf einem Knoten zu simulieren, gibt man „srvctl predict filesystem -device /dev/asm/asm_shared-331“ ein, wobei der Pfad nach „-device“ an die eigene Konfiguration angepasst werden muss. Abschließend lässt man zumindest einmal die Web-Komponenten von einem auf den anderen Knoten wandern, indem man „crsctl relocate resource apache01 -f“ aufruft.

Fazit

Es bedeutet schon einigen Aufwand, eine Umgebung, wie im Artikel aufgezeigt, aufzubauen (*siehe Abbildung 2*). Speziell

die notwendige Recherche um die Clusterware Agents zu verstehen und auch ein grundsätzliches Verständnis von RAC zu bekommen, benötigt doch einige Zeit.

Dieser Initial-Aufwand wurde allerdings auch belohnt, die Umgebung ist seit März 2015 in Betrieb und läuft seitdem völlig fehlerlos. Ganz klarer Vorteil der Umgebung ist auch, dass wirklich alle benötigten Komponenten innerhalb eines Clusters gemeinsam verwaltet werden und man sich möglichst wenige zusätzliche Abhängigkeiten einhandelt. Die Entscheidung für Oracle 12c und RAC erlaubt es zukünftig auch, zum Beispiel „Apex as a Service“ anzubieten. Durch die Multi-Tenant-Option können schnell zusätzliche Datenbanken angelegt werden, ohne zusätzliche Datenbank-Instanzen erstellen zu müssen, und mit RAC ist man in der Lage, flexibel auf gestiegene Anforderungen zu reagieren.



Moritz Klein
moritz.klein@mt-ag.com

Lizenzierung Forms 12c: Oracle bittet Stand-Alone-Kunden zur Kasse

DOAG Online

Kunden, die Forms mit einer „Forms and Reports“-Lizenz nutzen, müssen bei einem Upgrade auf Forms 12c die Repository-Datenbank nachlizenzieren. Als Stand-Alone-Version war zuvor keine Repository-Datenbank notwendig. Erst mit dem Release des Weblogic Server 12c wurde der Oracle Platform Security Service (OPSS), eine Layerschicht für Rollen und Authentifizierungen, in die Datenbank eingebettet. Da Forms den Enterprise Manager nutzt, der wie alle Anwendungen auf ADF-Basis auf die OPSS-Verwaltung zugreift, wird mit der Version 12c auch für

Forms die Datenbank als Repository benötigt.

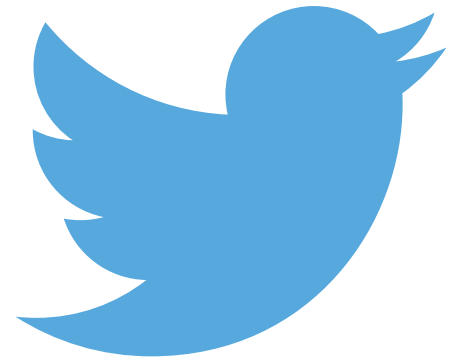
„Hier muss schleunigst nachgebessert werden“, fordert Michael Paege, Leiter Competence Center Lizenzierung bei der DOAG. Kunden, die Forms and Reports als Teil der Internet Application Server Enterprise Edition oder Weblogic Server Suite erworben haben, benötigen keine weitere Lizenz, sondern können die Repository-Datenbank als kostenfreie „Restricted Use“-Version nutzen. „Oracle sollte Kunden, die eine ‚Forms and Reports‘-Lizenz erworben haben, die benötigte Reposi-

tory-Datenbank ebenfalls als ‚Restricted Use‘-Lizenz bereitstellen“, so Paege.

Auch Jan-Peter Timmermann, Leiter der Infrastruktur & Middleware Community, zeigt sich bestürzt: „Viele Kunden haben beim Umstieg von Forms 6i auf 10g aus gutem Grund die Stand-Alone-Version erworben – Oracle Portal und Discoverer wurden nicht benötigt, damit war dies der einfachste und kostengünstigste Weg, Forms im Web zu betreiben.“ Lediglich als Teil der Internet Applikation Server Enterprise Edition wurde damals für Forms eine Datenbank als Repository benötigt.

Echtzeit-Visualisierung von Twitter & Co.

Kai Donato und Oliver Lemm, MT AG



Dieser Artikel zeigt, wie der Bedarf an Visualisierungen im Datenbank-Umfeld mehr und mehr zunimmt. Viele Daten zu erheben und zu analysieren, ist das Eine – sie für den Menschen anschaulich und verständlich zu machen, das Andere.

Viele Unternehmen erheben Metadaten, um diese für ihre Zwecke zu analysieren und zu ihrem Vorteil zu nutzen. So sammeln Onlineshops Suchbegriffe und Zugriffszahlen, um gewisse Trends frühzeitig zu erkennen und darauf zu reagieren. Soziale Netzwerke verwenden diese gesammelten Daten, um passende Werbung und interessante Artikel zustellen zu können. Staatliche Institutionen nutzen große Datenmengen, um Straftaten aufzudecken oder sogar präventiv zu verhindern.

All diese Einsatzzwecke stimmen zumindest in dem Punkt überein, dass sie für den jeweiligen Einsatzzweck aufbereitet dargestellt werden müssen. Sobald die menschliche Komponente in diesem Analyse-Prozess auftritt, sind Daten in aggregierter Form und nicht selten auch visualisiert aufzubereiten.

Echtzeit-Visualisierung

Häufig sind nicht nur die Daten selbst von hohem Wert, sondern auch deren Aktualität. Was nützt ein Dashboard, das kritische Prozesse überwacht, wenn es nur die alten Daten vom Vortag anzeigt? Um den vollen Nutzen eines Dashboards auszureizen, müssen die Daten in Echtzeit übermittelt und verarbeitet werden.

Dieses Ziel haben in der Vergangenheit viele Entwickler gelöst, indem sie mit dem sogenannten „Polling“ arbeiteten. Dies ist nicht von Vorteil und es gibt eine wesentlich bessere Lösung für diese Pro-

blematik, wie das Beispiel der Echtzeit-Visualisierung zeigt.

Für dieses Beispiel wurde ein Online-Dienst ausgesucht, der seine Existenzberechtigung gerade aus der schnellen Interaktion von vielen Benutzern bezieht. Die Mikroblogging-Plattform Twitter ist bestens dafür geeignet, den Nutzen der Verarbeitung in Echtzeit darzustellen.

Um einen konkreten Use-Case zu generieren, wurde für das Projekt ein Rahmen gewählt, der sich auf eine bestimmte Themengruppe einschränkt. Für die anstehende Konferenz „APEXConnect 2016“ in Berlin existiert bereits ein sogenannter „Hashtag“, der eine Kurznachrichte einem Thema zuordnet und sich von allen, die sich für dieses Thema interessieren, finden und abonnieren lässt (siehe Abbildung 1).

Da Twitter die Tweets und auch die Antworten auf bestimmte Tweets standardmäßig in Listenform darstellt, ist es naheliegend, diese Tweets und Beziehungen zueinander in einer grafischen Form auszuwerten. Hierzu ist es notwendig, die Tweets und Beziehungen anhand der erhobenen Daten zu ermitteln und an eine geeignete Visualisierungs-Engine weiter-

zugeben. Anschließend werden die Daten aufbereitet und dargestellt.

Woher die Daten kommen

Um die Daten zu beziehen, ist von vornherein der Aspekt zu berücksichtigen, dass sie möglichst schnell nach der Veröffentlichung vorliegen. Dies lässt sich anhand des Streaming-API von Twitter optimal und vor allem in Echtzeit realisieren. Der Vorteil dabei ist, dass die Daten, sobald sie veröffentlicht wurden, direkt an einen lauschenden Server weitergeleitet werden und dort bereits wenige Sekunden später verarbeitet werden können. Sofern ein Server aufgesetzt und konfiguriert wurde, liefert Twitter die Daten in JSON-Form. Listing 1 zeigt einen Auszug aus einer solchen Datenübertragung.

Anhand dieser Daten wird die weitere Verarbeitung vorgenommen. Beim Twitter-API ist zu beachten, dass es Einschränkungen bezüglich der Frequenz und Anzahl der Anfragen gibt. Bei dem von den Autoren hauptsächlich genutzten Streaming-API gibt es lediglich die Einschränkung der Anzahl an aktiven Verbindungen. Für die Visu-



Abbildung 1: Ansicht zweier Tweets zum Thema „#apexconn16“

alisierung von spezifischen Hashtags haben sie sich für die Variante „User-Streams“ entschieden. Diese ist nach der Anmeldung als Entwickler und der Einrichtung einer Applikation kostenfrei. Im Beispiel ist ausschließlich auf die Anzahl der aktiven Streaming-Verbindungen zu achten. Um mit dem genannten NodeJS-Server nur eine aktive Verbindung zu benötigen, wird dieses Limit nicht überschritten.

Für die manuell initiierten Anfragen, die nach dem REST-Prinzip funktionieren, muss jedoch auf deren Anzahl geachtet werden. Für die Suche nach konkreten Begriffen werden 180 Anfragen pro 15 Minuten zugelassen und bei allen anderen Interaktionen mit dem REST-API dürfen 15 Anfragen pro 15 Minuten durchgeführt werden. Damit der NodeJS-Server sich als Stream-Endpoint anmelden darf, benötigt dieser eine Kombination aus mehreren Sicherheitstoken:

- Consumer Key
- Consumer Secret
- Access Token Key
- Access Token Secret

Diese Sicherheitstoken werden beim Erstellen einer Applikation im Entwicklerbereich generiert und anschließend benötigt, um die Daten von Twitter abzurufen.

```
[{
  [...]
  "created_at": "Tue Jan 12 07:09:54 +0000 2016",
  "id": 686807297725698000,
  "id_str": "686807297725698048",
  "text": "Ich freue mich auf die #apexconn16 !",
  "source": "<a href=\"http://tapbots.com/software/tweetbot/mac\"
    rel=\"nofollow\">Tweetbot for Mac</a>",
  "truncated": false,
  "in_reply_to_status_id": null,
  "in_reply_to_status_id_str": null,
  "in_reply_to_user_id": null,
  "in_reply_to_user_id_str": null,
  "in_reply_to_screen_name": null,
  [...]
}]
```

Listing 1

Mit den genannten Schlüsselpaaren authentifiziert sich der NodeJS-Server beim Twitter-API und hat fortan die Möglichkeit, auf diverse Daten zugreifen zu können (siehe Abbildung 2).

Die Daten ohne merkliche Verzögerung verarbeiten

Nun profitiert man davon, das Twitter die Daten ohne Verzögerung liefert, steht jedoch vor dem Problem der verzögerungsfreien Weiterleitung an die Visualisierungs-Engine.

An dieser Stelle haben sich die Autoren für ein serverseitiges JavaScript entschieden und einen NodeJS-Server eingerichtet, der permanent auf Meldungen von Twitter wartet und diese anschließend weiterleitet. Der Vorteil dabei ist, dass die Einrichtung des Servers nur einen minimalen Arbeitsaufwand verursacht. Mithilfe eines bereits entwickelten NPM-Pakets ist auch die Anbindung an Twitter innerhalb von kurzer Zeit und wenigen Code-Zeilen realisiert. Für die Weiterleitung an den Browser und somit an die Visualisierungs-Engine muss

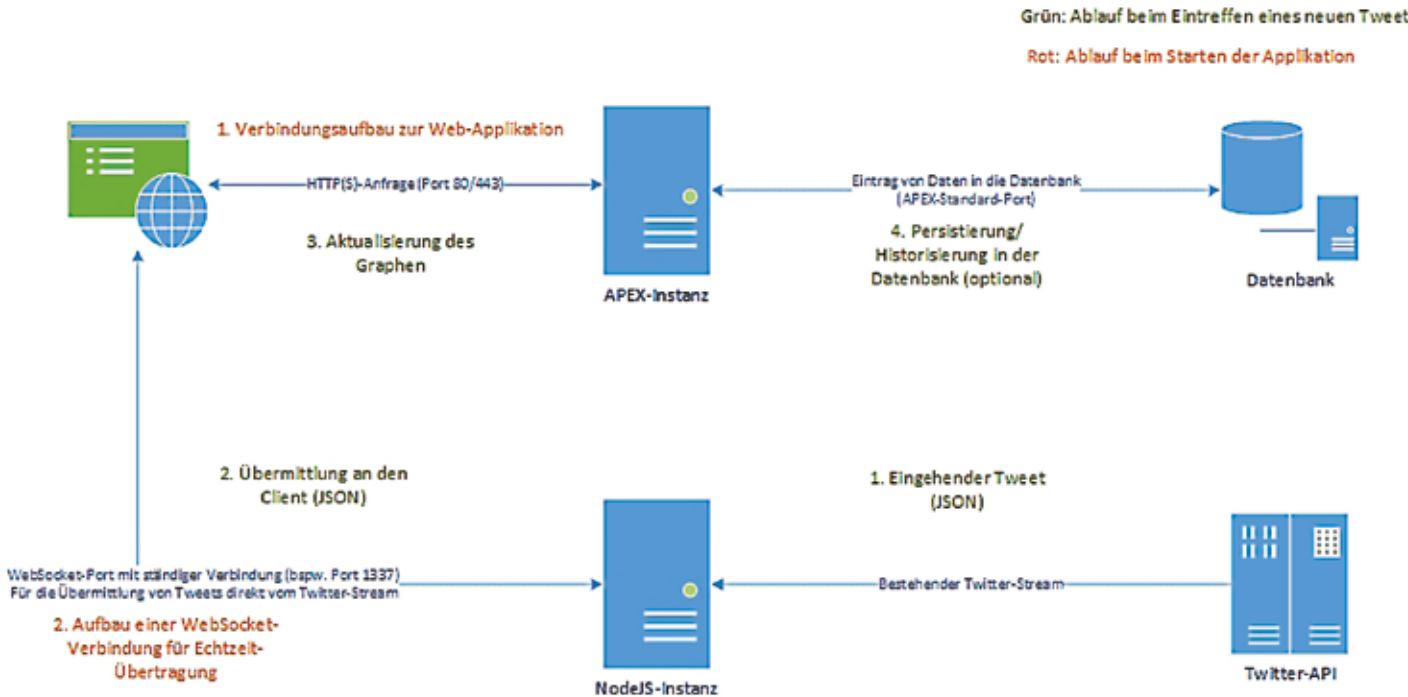


Abbildung 2: Schematische Darstellung des Ablaufs

hinsichtlich der unerwünschten Verzögerung auch auf eine Technologie gesetzt werden, die sich von dem genannten Polling-Verfahren unterscheidet.

WebSockets

Um die permanenten Anfragen des Client-Browsers zu unterbinden, stellt es sich als äußerst sinnvoll dar, die Datenübertragung von der Seite des Servers zu initiieren. Dies lässt sich mithilfe der in HTML5 eingeführten WebSocket-Technologie erreichen.

Der einmalige Aufbau einer WebSocket-Verbindung zwischen Server und Client ermöglicht es, jederzeit Daten in beide Richtungen zu transferieren und im Anschluss definierte Aktionen auszuführen. So gelangen die von Twitter übermittelten und von NodeJS aufbereiteten Daten über die WebSocket-Verbindung an den Client-Browser.

Um diesen Prozess zu realisieren, wurde im NodeJS-Server ein Stream-Endpunkt für das Twitter-API implementiert und in Form von Callback-Funktionen auf eingehende Daten reagiert. Durch die Non-Blocking-Eigenschaft von NodeJS ist es möglich, neben dem ständig aktiven Streaming-Endpunkt noch weitere Funktionalitäten zu realisieren. Dazu gehört unter anderem auch die

manuelle Abfrage nach historischen Daten, die für die grafische Aufbereitung verwendet werden.

Graphen-Theorie

Um große Mengen an Daten aggregiert und verständlich darzustellen, eignen sich Graphen besonders gut. Im Vergleich zu tabellarisch komprimierten Daten, die sich meist auf die Darstellung von Zahlen beschränken, bieten Graphen mehrere Dimensionen zur Visualisierung. Selbst in der einfachsten Form des Graphen, bestehend aus Kanten und Knoten, ist eine Verbindung von zwei Punkten selbsterklärend als Beziehung zu erkennen. Nimmt man nun die Möglichkeit einer gerichteten Kante hinzu, kann man dadurch auch die Flussrichtung der Information erkennen. Weitere Eigenschaften können in der Kantenlänge beziehungsweise im Abstand zwischen den Knoten, in der Größe der Knoten sowie in unterschiedlichen Farben von Knoten abgebildet werden.

Die Transformation

Der maßgebliche Faktor in einer Visualisierung durch Graphen liegt in der Transformation der Grund-Daten. Diese stam-

men aus Twitter und sind in Form einer Liste mit Metadaten vorhanden, die zum Teil in Beziehung zueinander stehen. Um diese in einer Graphen-Struktur darzustellen, werden die Tweets zu einem bestimmten Thema (in Twitter als „Hashtag“ bezeichnet) gefiltert. Im Beispiel werden die Postings zur nächsten Apex-Connect-Konferenz mit dem Hashtag „#APEXConn16“ verwendet.

Neben dem Hashtag selber stellen die Eigenschaften des Benutzernamens, der Posting-ID und des Profilbilds die relevanten Daten dar, die für eine spätere Visualisierung benötigt werden. Dazu werden alle JSON-Objekte, die durch den WebSocket empfangen werden, hinsichtlich des Benutzernamens überprüft. Falls dieser nicht vorhanden ist, wird er als neues Objekt wiederum im JSON-Format für den Graphen gespeichert, wobei das Profilbild beziehungsweise die URL des Bilds später für die Darstellung des Knoten verwendet werden soll. Die Posting-ID wird dabei genutzt, um zu gewährleisten, dass ein bereits verarbeiteter Post nicht nochmals benutzt wird. *Listing 2* zeigt, wie das im JSON-Format aussieht.

Die Anforderungen

Für die Darstellung eines Graphen kommt ein auf JavaScript basierendes Framework zum Einsatz, um einen nahtlosen Ansatz im Browser zu gewährleisten. Es soll die Möglichkeit bieten, einen vorhandenen Graphen zur Laufzeit zu erweitern, ohne dass man die Bezugspunkte verliert. Im Vergleich zu einer Visualisierung von statischen Daten wird ein Graph nicht direkt mit allen Daten dargestellt, sondern es existieren anfangs keine oder nur eine kleine Menge an Daten.

Über die Zeit werden mithilfe des Streaming-API immer neue Postings übermittelt und der vorhandene Graph wächst so Stück für Stück. Bei diesen Anpassungen soll sich der Graph möglichst nur gering in der gesamten Darstellung verändern, damit ein Benutzer schnell erkennen kann, wo sich eine Veränderung ergeben hat, ohne den Rest der schon vorhandenen Darstellung aus den Augen zu verlieren.

```
{
  "nodes": [{
    "user_id": 689815827
    "image_url": "https://pbs.twimg.com/profile/images/..."
    "tweets": [{
      "tweet_id": 689815827479203800
    }, {
      "tweet_id": 689815827479203801
    }
  ]
}]
}
```

Listing 2

```
// if user is running mozilla then use it's built-in WebSocket
window.WebSocket = window.WebSocket || window.MozWebSocket;
var connection = new WebSocket('wss://mywebsocketserver.de:11111');
connection.onopen = function() {
  console.log('Connection established');
};
```

Listing 3



Abbildung 3: Darstellung des Graphen

Die Frameworks

Als ein mögliches Framework zur Graphen-Visualisierung bot sich D3 an, das eine Vielzahl von verschiedenen Graphen-Darstellungen unterstützt. Dabei stehen aber meist statisch zu visualisierende Daten zur Verfügung, die nicht noch zur Laufzeit angepasst werden müssen.

Nach weiteren Recherchen ist man auf das Visualisierungsframework VivaGraphJS gestoßen. Der technische Ansatz ist dem D3-Framework sehr ähnlich, da beide SVGs im Browser zur Darstellung der Graphen-Strukturen nutzen. VivaGraphJS wurde dabei aber speziell zur Visualisierung von Netzstrukturen entwickelt und bietet neben dem rein grafischen Ansatz auch Unterstützung für Gravitations-Effekte. Dies ist vor allem deshalb interessant, da ein Hinzufügen eines Knotens nicht eine komplett andere Struktur mit sich bringt, sondern der Graph sich beim Hinzufügen leicht anpasst und diese Anpassung durch den Gravitations-Effekt sichtbar ist. Eine Auflistung verschiedener Graph Libraries, die unter GitHub zur Verfügung stehen, kann man unterhalb des VivaGraphJS-Projekts auf GitHub einsehen.

Die Umsetzung

Für die Umsetzung wurde eine Apex-5-Anwendung erstellt, in der das VivaGraphJS-Framework als ZIP-Datei in die statischen Anwendungsdateien importiert wurde. In der Anwendung wurde neben der Login-Seite eine einzelne Seite erstellt, die die komplette Logik enthält. Durch eine Dynamic Action entsteht beim Seitenladen eine Verbindung zum NodeJS-Server per WebSocket-Verbindung (siehe Listing 3).

Bei jeder Information (Tweet), die man über die Verbindung empfängt, wird eine Funktion „add_node_to_graph“ aufgerufen, wodurch sich der Graph jeweils um einen Knoten erweitert oder ein bestehender Knoten in der Größe wächst. Neben den Knoten, die eine Person darstellen, werden zentrale Inhalte, die als Hashtags referenziert sind, auch als Knoten dargestellt. In unserem Fall „#apexconn16“ oder „#orclapex“.

Um neben der Anpassung des Graphen einen neuen Tweet und die Person hervorzuheben, kommt zusätzlich eine kleine Bibliothek namens „PNotify“ zum Einsatz, die am oberen rechten Bildschirmrand eine Benachrichtigung ein-

blendet, sobald neue Daten eintreffen (siehe Abbildung 3).

Best Practices

Als eines der ersten Probleme kristallisierte sich die Verbindung zwischen Apex und NodeJS heraus, da der Apex-Server per HTTPS betrieben wird und die Kommunikation mit dem NodeJS-Server anfangs nur unverschlüsselt lief. Der ausschlaggebende Punkt war, dass die Applikation funktionierte, wenn alle Komponenten ohne SSL verbunden sind. Wenn der Benutzer die Applikation im Browser per HTTP aufruft und die Verbindung des WebSocket ebenfalls ohne SSL aufgenommen wurde, gab es keine Probleme. Erst wenn eine der genannten Verbindungen auf SSL-Verschlüsselung umgestellt wurde, verweigerte der Browser die Verbindung aufgrund von Sicherheitsrichtlinien. Um dieses Verhalten und somit auch die Sicherheit der gesamten Applikation zu gewährleisten, wurden der NodeJS-Server und somit auch die WebSocket-Verbindungen mit validen SSL-Zertifikaten versehen.

Bei Testläufen bestand ein weiteres Problem darin, dass oft genug Daten benötigt wurden, es aber aktuell keine Tweets gab. Hier wird man also fleißig in Twitter aktiv, um dies voranzutreiben. Um auch ohne Livedaten eine Visualisierung zu testen, wurde neben dem Streaming-API auch eine Abfrage über das REST-API implementiert.

Nach einigen Testläufen stellte man fest, dass hin und wieder einige Knoten dem Graphen hinzugefügt wurden, die mit keinem zentralen Knoten verknüpft waren und sich dabei vom Graph entfernten. Der Grund war, dass in den übermittelten JSON-Daten des Streaming-API die Groß- und Kleinschreibung mit übernommen wurde, aber im Graph die Ausdrücke „#APEXCONN16“ oder „#ORCLAPEX“ nur in Großbuchstaben als Knoten existierten. Dieses Problem wurde durch eine Anpassung des Verarbeitungsschritts auf Seiten des NodeJS-Servers behoben. Alle Hashtags werden nur noch in Großbuchstaben verarbeitet.

Ein weiteres Problem bei solchen dynamischen Graphen ist die Größe beziehungsweise die Skalierung des Graphen.

Da anfangs nicht bekannt ist, wie groß der Graph insgesamt wird, und ein automatisches Anpassen der Größe nicht vorhanden ist, sind die Graphen nicht ohne händisches Anpassen perfekt zu nutzen.

Alleinstellungsmerkmal

Das Thema bietet eine Menge interessanter Technologien, aber somit auch eine Menge von Problemen, die dabei auftreten können. Vor allem der Einstieg in die Visualisierung von Echtzeitdaten ist mit wesentlich mehr Anforderungen verbunden als eine Visualisierung von statischen Daten. Der Aufwand lohnt sich jedoch, gerade bei großen Datenmengen bietet sich eine Graphen-Struktur durch die Aggregation der Daten an, um den Überblick zu behalten. Zusätzlich wird durch die Echtzeit-Verarbeitung der Twitter-Daten und die Ak-

tualisierung des Graphen durch das performante VivaGraphJS-Framework eine Livedatensvisualisierung ohne Polling-Ansatz erreicht.

Vergleichbare Ansätze basierend auf dem Twitter-Streaming-API, die eine Graphen-Struktur ohne Beschränkungen in der Anzahl der verknüpften Personen darstellen, sind bisher nicht vorhanden. Alles Weitere zu dem Thema wird es auf der diesjährigen Apex Connect in Berlin geben.

Weitere Informationen

- D3, JavaScript Graph Library: <https://d3js.org>
- VivaGraphJS, JavaScript Graph Library: <https://github.com/anvaka/VivaGraphJS>
- Graph Libraries, diverse unter GitHub vorhandene JavaScript-Frameworks zur Visualisierung von Graphen: <https://github.com/anvaka/graph-drawing-libraries>
- Pnotify, JavaScript Framework zur Darstellung von Benachrichtigungen: <https://github.com/sciactive/pnotify>



Kai Donato
kai.donato@mt-ag.com



Oliver Lemm
oliver.lemm@mt-ag.com

Schließung des Oracle-Software-Produkt-Supports in Deutschland steht unmittelbar bevor

DOAG Online

Die bisherigen Gerüchte um den Stellenabbau im europäischen Support sollen sich schon bald bewahrheiten. Mario Kowalski, Customer Support Country Leader bei Oracle Deutschland, bestätigte im Gespräch mit der DOAG die bevorstehende Schließung des Oracle-Software-Produkt-Supports in Deutschland zum 31. März 2016. Er versprach: Sollten DOAG-Mitgliedern dadurch Probleme entstehen, stünde er als Ansprechpartner zur Verfügung.

In Deutschland sind rund 150 Mitarbeiter von der Stellenstreichung betroffen. Wie Oracle bestätigte, betrifft der Stellenabbau im Oracle-Support jedoch ganz Europa. Neben Deutschland trifft es auch die Supportzentren in Großbritannien, den Niederlanden und Frankreich. Insgesamt sollen 450 Stellen wegfallen. Ziel der Maßnahme ist die Verlegung des Supports nach Rumänien. Dort soll er laut Oracle quantitativ größer aufgestellt werden, genaue Mitarbeiterzahlen nannte das Unterneh-

men jedoch bisher nicht. Die Kunden stellt diese Entwicklung vor viele offene Fragen hinsichtlich der Gestaltung der Übergangszeit und der zu erwartenden Auswirkungen auf die Support-Qualität.

Die DOAG befürchtet hier einen zumindest kurzfristigen Qualitätsverlust in der Übergangsphase und initiierte daher ein Treffen mit Oracle, um über die Entscheidung und deren Auswirkungen zu sprechen. „Wichtig für die Kunden ist vor allem, ob Oracle in der Lage sein wird, die Qualität des Supports sowohl qualitativ als auch quantitativ aufrecht zu erhalten“, so Dr. Dietmar Neugebauer, Vorstandsvorsitzender der DOAG. „Entscheidend wird dabei sein, wie der Übergang Ende März reibungslos durchgeführt werden kann. Nichtsdestotrotz bedauert die DOAG es sehr, dass Oracle Deutschland sich von vielen langjährigen Mitarbeitern verabschiedet und dadurch sehr viel Know-how in Deutschland verloren geht.

Im Gespräch sicherte der Oracle-Supportverantwortliche Kowalski der DOAG zu, bei Support-Problemen den DOAG-Mitgliedern als Ansprechpartner zur Verfügung zu stehen. Christian Trieb, Leiter des DOAG Competence Center Support und der Datenbank Community, begrüßte im Namen der DOAG den angestrebten Dialog und hofft nun auf eine möglichst komplikationsfreie Übergangszeit. „Bezüglich der Support-Thematik wird die DOAG auch weiterhin eng mit Oracle in Kontakt stehen, denn erst langfristig wird sich herausstellen, wie sich die Servicequalität entwickelt und welche Rolle mögliche Schwierigkeiten in der sprachlichen Verständigung dabei spielen“, so Trieb weiter. Wie die Kunden mit dem neuen Software-Produkt-Support in Rumänien zurechtkommen, wird auch die im Herbst dieses Jahres erneut geplante DOAG-Umfrage zur Qualität des Oracle-Supports zeigen.



Das Tor zur apex.world

Jürgen Schuster, IT-Beratung

Was hat es mit diesem Apex auf sich, dass sich Tausende Entwickler in der Welt „ā'pěks“-Sticker (inoffizielles Logo) auf ihre Laptops und Autos kleben? Dass es einen Fan-Shop gibt (siehe „<https://apex.world/ords/f?p=100:430>“) mit T-Shirts, Tassen, Regenschirmen und neuerdings auch ein 3D-Modell des Logos? Sogar einen eigenen Podcast (siehe „<http://apex.press/talkshow>“) – und alles ohne das geringste Zutun von Oracle? Dass bei jeder Oracle-Konferenz die Mehrzahl der Vorträge von Apex handeln und auch rege besucht werden? Dass es mittlerweile sogar mehrtägige „Apex only“-Konferenzen gibt („siehe apex.doag.org“), die sich immer größerer Beliebtheit erfreuen? Das letzte Kind dieser anscheinend verrückten Apex Community ist das Portal „apex.world“. Entwickelt von der Community für die Community, mit einem kleinen internationalen Team, das in seiner Freizeit daran arbeitet, Apex-News mit dem Rest der Community weltweit zu teilen. Wie kam es dazu?

Vor etwa elf Jahren – ich war damals bei Oracle in München als Consultant angestellt – nutzte ich eine Unterbrechung meines neuen Program-Manager-Jobs, um mir den Nachfolger von Web-DB anzuschauen. Marc Sewtz und Jason Straub aus dem Development-Team präsentierten diesen in der Münchner Geschäftsstelle. Ich schenkte den beiden – vertieft in die Abarbeitung meiner Admin-Mails – kaum Aufmerksamkeit. Zu gut kannte ich ja schon die webbasierte Software-Entwicklung mit Web-DB, die ich bereits in einem Projekt eingesetzt hatte.

Am Ende der Präsentation verpasste ich dann doch offensichtlich den Schritt der Generierung der PL/SQL-Packages, wie von Web-DB gewohnt, und das war dann auch meine einzige Frage. Marc merkte gleich, dass er mich nur schwer mit Worten überzeugen konnte und demonstrierte, dass sich in dem Schema kein einziges Package, keine Prozedur oder Funktion befand. Nun hatten die beiden meine volle Aufmerksamkeit.

Vielleicht ist es jetzt Zeit zu erwähnen, dass ich nie gerne programmiert habe. Ich hatte ursprünglich BWL studiert und bin nur deshalb widerwillig zur IT gekommen, weil ich es leid war, mit Computer-Nerds sprechen zu müssen, die nie begriffen, was ich eigentlich wollte, und am Ende immer das taten, was ihnen Spaß machte, nämlich komplizierte Sourcecode-Probleme zu lösen. Ich war da eher störend mit meinen praktischen Anforderungen.

Auch in den dreizehn Jahren als Oracle-Consultant hat sich das nicht geändert. Ich habe neben der Datenbank verschiedene Systeme gelernt und auch eingesetzt – Java mit verschiedenen Frameworks, BPEL, Web-DB, Oracle Portal, etc.

Jetzt sah ich jedoch zum ersten Mal Licht am Ende des langen dunklen Software-Tunnels. Wenn dieses Apex am Ende keinen Sourcecode generiert, in dem ich dann mühsam wühlen muss, um individuelle Anpassungen vorzunehmen, dann scheinen wir endlich ein neues Zeitalter der Software-Entwicklung erreicht zu haben, in dem auch ich leben möchte. Rein deklarative Entwicklung, mit der ich keine einzige Zeile Sourcecode schreiben muss, um vollständig funktionsfähige Applikationen zu generieren, die sich genauso spielend leicht ändern lassen, wie sie erzeugt wurden. Und das Beste: Im Gegensatz zu anderen deklarativen Entwicklungssystemen kann ich in Apex alles zu jeder Zeit und mit den einfachsten Technologien, wie PL/SQL, HTML und JavaScript, individuell anpassen und dem Kunden damit jeden Wunsch erfüllen.

Nach der Präsentation rief ich sofort meinen erstaunten Manager an und kündigte erst mal meinen Projektleiter-Job mit den Worten: „Ich mache ab jetzt nur noch Apex (damals HTML DB).“ Endlich konnte ich mit meinem Laptopkoffer zum Kunden gehen, mit dem Fachbereich direkt sprechen, mir dessen Wünsche anhören, meinen Laptop

auspacken, vor seinen Augen die Anforderungen live webbasiert umsetzen und nebenbei noch einen technikaffinen Mitarbeiter schulen, sodass sie in Zukunft die Lösung auch selbst erweitern konnten, zur großen Freude auf beiden Seiten.

Zehn Jahre später

Ich bin meinem Versprechen treu geblieben und habe wirklich seitdem nur noch Apex-Projekte. Basierend auf konkreten Download-Statistiken, schätzt Oracle derzeit etwa eine halbe Million Apex-Entwickler weltweit und mehr als 120 Consulting Companies. Ich scheine also nicht der Einzige zu sein, der begriffen hat, dass man mit Apex vor allem Lösungen anstelle von Software-Projekten bekommt und dass dies für beide Seiten sehr befriedigend ist.

In diesen zehn Jahren hat sich auch eine sehr lebendige und hilfsbereite Community gebildet. Nicht zuletzt, weil das Development Team rund um Joel Kallman auf allen Konferenzen zu finden ist, viele Vorträge hält, mit den Apex-Entwicklern direkt spricht und sich Fragen und Wünschen stellt. Änderungswünsche an Apex kann man auch offline unter „apex.oracle.com/vote“ anmelden. In dieser Community gibt es täglich viele interessante Lösungsbeispiele, Blog-Posts, Präsentationen, Plugins, Videos, Demo-Applikationen etc.

apex.world

Genau dies ist aber auch das Problem. Es gibt sehr, sehr viele verschiedene interessante Stellen, an denen man etwas Neues über Apex erfahren kann. Fast alles wird auf Twitter gepostet, doch man sollte schon mindestens hundert Apexianern dort folgen, um alles mitzukriegen. Allerdings stellt sich auf den verschiedenen Apex-Konferenzen immer wieder heraus, dass nicht mehr als zehn bis zwanzig Prozent überhaupt Twitter aktiv nutzen. Ein Einsteiger, der mich über die Apex-Facebook-Gruppe kontaktiert hatte, äußerte seinen Frust, dass er zwar sehr gerne Apex lernen würde – da er so viel Begeisterndes darüber höre – es ihm jedoch sehr schwerfiele, in den vielen verschiedenen Quellen die Informationen zu finden, die für ihn relevant seien.

Apex zu lernen, ist kinderleicht. Es zu verstehen, auch nicht schwer, aber auf dem aktuellen Stand darüber zu bleiben, welche neuen Lösungen es gibt, war bisher sehr aufwendig. Ich investiere täglich eine Stunde, um all diese Neuigkeiten zu finden und abzuspeichern. Denn als Freelancer kann ich, mit dem Wissen um nur einen Blog-Post zu einem bestimmten Thema, dem Kunden sehr viel Geld sparen. Ich bin überzeugt, dass man sich einen großen Teil der Eigenentwicklungen in Apex schenken könnte, wenn man wissen würde, dass es bereits eine Lösung dafür gibt und von wem.

Ich selbst war mit dieser Wissens-Diversifikation sehr unzufrieden, deshalb suchte ich seit Langem nach einer besseren Lösung, an der alle partizipieren können und nicht jeder täglich dieselbe Arbeit machen muss wie ich. Auf der Heimfahrt von der DOAG 2014 Konferenz hatte ich die Gelegenheit, lange mit Joel über diese Idee zu sprechen. Er sicherte mir seine volle Unterstützung zu, doch ich war damals viel zu sehr in Projekte eingebunden, als dass ich das wirklich hätte starten können.

Vor der Kscope 2015 besuchte ich Marc Sewtz in der New Yorker Geschäftsstelle zur Aufzeichnung eines Podcasts und traf dort auch auf meinen Mentor Dan McGhan („The Professor“), der selbst erst seit ein paar Wochen bei Oracle im Advocate Team von Steven Feuerstein angefangen hatte.

Ich erzählte ihm von meiner Idee und Dan sicherte mir ebenfalls sofort seine volle Unterstützung zu, sollte ich auf technische Probleme stoßen, ebenso Marc.

Mit Dan im Rücken, der jedes technische Problem lösen und vor allem auch einfach erklären kann, sah ich keine technischen Hürden mehr. Die letztendliche Entscheidung, es dann umzusetzen und dies auch auf der Kscope anzukündigen, fiel nach der legendären „#LetsWreckThisTogether“-Brandrede von Joel in Miami.

Danach ging alles sehr schnell – jeder, dem ich davon erzählte, war von der Idee begeistert und so wuchs das Team von Tag zu Tag. Am Ende hatte ich ein großes internationales Team von Entwicklern und Unterstützern zur Verfügung, die meine Ideen aufgriffen und perfektionierten – und das völlig unabhängig davon, ob das nun Kollegen von Oracle waren, solche aus dem Apex-Development-Team oder der Apex-Kollege gegenüber. Zurzeit haben wir folgende Unterstützer:

- Michael Dürbeck, <https://twitter.com/mduerbeck>
- Dan McGhan, <https://twitter.com/dmcghan>
- Daniel Hochleitner, <https://twitter.com/Dani3lSun>
- Carsten Czarski, <https://twitter.com/cczarski>
- Adrian Png, <https://twitter.com/fuzziebrain>
- Christian Neumüller, <https://twitter.com/chrisneumueller>
- Shakeeb Rahman, <https://twitter.com/shakeeb>
- Nathan Catlow and Tim Austwick from Recx, <https://twitter.com/RecxLtd>
- Marc Sewtz, <https://twitter.com/msewtz>
- Martin Giffy D'Souza, <https://twitter.com/martindsouza>
- Peter Raganitsch, <https://twitter.com/PeterRaganitsch>
- Nick Buytaert, <https://twitter.com/nbuytaert1>

In nur drei Monaten gelang es uns, ein Portal auf die Beine zu stellen und am 19. Oktober 2015 in einem Live Webcast vorzustellen. Es hat folgende Funktionen:

- News
- Twitter Feed
- Slack Chat (dort kann man Fragen stellen oder über Themen diskutieren)
- Job Market
- Plug-ins
- Konferenzen
- Meetups
- Beginners
- Fan-Shop

- Community Members
- Links
- YouTube Channel

Es gibt einen Videocast über den Launch von apex.world, den wir einige Wochen später aufgenommen haben (siehe „<https://www.youtube.com/watch?v=uMnCjrMlVUE&index=2&list=PL3S33P01ea06WdgNtuFvGd-f8PvP54-g0>“). Am 14. März 2016 war der erste Live-Webcast „apex.world On Air“ geplant mit den ständigen Mitgliedern Joel Kallman, Scott Spendolini, Dan McGhan und Martin Giffy D'Souza, in dem wir Neuigkeiten austauschen, Gäste einladen, Apex-Lösungen testen und vor allem live Fragen der Community beantworten.

Außerdem ist geplant, das Oracle Apex Forum mit einzubinden, auch Trainings, Newsletter etc. In unseren regelmäßigen Projekt-Meetings fallen uns immer neue Ideen ein und es macht noch immer so viel Spaß wie am Anfang, denn wir tun das, was uns Apex-Entwicklern am meisten Spaß macht, wir können uns ganz auf Lösungen konzentrieren – ohne Einschränkungen und dann einfach mit Apex umsetzen.

Für mich ist „apex.world“ ein klassisches Beispiel einer Win-Win-Situation, mit der ich grundsätzlich alle meine Vorhaben bewerte. Durch „apex.world“ bin ich selbst geradezu gezwungen, immer auf dem neuesten Stand zu sein, und ich kann auch von meinen professionellen Kollegen viel dazulernen. Außerdem ermöglicht es mir den Kontakt zu vielen interessanten Community-Mitgliedern. Dies hilft mir sehr bei der Arbeit als Apex-Freelancer, vom Spaß ganz zu schweigen. Und obwohl es nicht notwendig ist, sich bei „apex.world“ zu registrieren, haben wir dennoch bereits 1.350 Mitglieder, die darüber ihr Apex-Skill-Set präsentieren und sich auf Slack mit der Community austauschen können.



Jürgen Schuster
j_schuster@me.com
<http://apex.world>

Apex als Tool für eine Vereins-Webseite

Rainer Willems, ORACLE Deutschland B.V. & Co. KG

Als Training für Apex 5.0 wurde eine Vereins-Webseite mit Apex gebaut. Ziel war eine überschaubare Webseite, die optisch die bisherige Seite aufgreift und die bisherigen Informationen übernimmt, aber zusätzliche Funktionalitäten bietet, insbesondere die Möglichkeit des einfachen Änderns der Seite ohne IT-Kenntnisse. Dieser Artikel zeigt, welche Apex-Funktionalitäten zum Einsatz kamen, wo Apex allein nicht ausreichte, wo das Ganze betrieben wird und welches Resümee aus dem Test gezogen werden kann.

Die Stärke von Apex liegt sicher in der Datennähe. Formulare, Berichte und andere datennahe Anforderungen lassen sich mit Apex schnell und einfach erledigen. Genau diese Funktionalitäten werden auch in Vereinen gebraucht und wurden im Beispiel auch bereits verwendet. Die Verwaltung der Mitglieder und insbesondere der Spielplan samt Spielbetreuung (wer fährt/betreut wann welches Team) wurde bereits vorher mit Apex gesteuert. Warum also nicht auch den Web-Auftritt des Vereins damit kombinieren und eine integrierte Anlaufstelle für externe und interne Zugriffe bekommen. Apex 5.0 mit dem Universal Theme schien gut geeignet.

Die Ausgangsbasis

Die seit dem Jahr 2006 in Betrieb befindliche Webseite war funktional, aber nicht sonderlich hübsch (siehe *Abbildung 1*). Die Anforderung an eine neue Seite

lautete: etwas schicker, aber wiedererkennbar.

Die grundsätzliche Struktur und Optik sollte in etwa erhalten bleiben. Nichts leichter als das. Das Universal Theme und die darin enthaltene Navigationsstruktur sowie der Theme Roller können dies einfach bewerkstelligen. Mit dem Desktop Navigation Menu für die bisherigen, „klassischen“ Menüpunkte und der Desktop Navigation Bar insbesondere für die internen Punkte konnte die Navigation schnell umgesetzt und durch die Verwendung von Font Awesome Icons und der List-basierten Möglichkeiten hübsch und gut nutzbar gestaltet werden (siehe *Abbildung 2*).

Die Icons des Font Awesome werden einfach im Auswahlfeld „Image/Class“ in den Property eines Menüpunkts gesetzt. Die farbliche Gestaltung mittels Theme Roller ist nicht nur simpel, sondern hat auch den Charme einer schnellen Änderungsmöglichkeit. Da mehrere Stile hin-

terlegt werden können, ist die Änderung der Farbkombinationen quasi auf Knopfdruck möglich, was auf der Webseite für aktuelle Trauerfälle verwendet wird.

Das in *Abbildung 2* abgebildete Tischtennis-Piktogramm im Menü wurde im List Template hinterlegt. Der Eintrag List Template Before Rows wurde hierzu von *Listing 1* in *Listing 2* geändert.

Zudem gibt es hier ein Template für interne und eines für externe Seiten, damit der (interne) User direkt erkennt, ob er auf einer internen oder einer „public“ Seite ist. Folgende weitere neue Möglichkeiten für Regionen aus Apex 5.0 wurden verwendet, um die Seite zu gestalten:

- Carousel Container
- Tabs Container
- Card Reports
- CSS Kalender Region

Carousel Container können mehrere Subregions haben, die automatisch oder

```
<div class="t-Body-nav t-Body-nav--dark" id="t_Body_nav" role="navigation" aria-label="&APP_TITLE!ATTR.">
<div class="t-TreeNav #COMPONENT_CSS_CLASSES#" id="t_TreeNav" data-id="#PARENT_STATIC_ID#_tree" aria-label="&APP_TITLE!ATTR."><ul style="display:none">
```

Listing 1

```
<div class="t-Body-nav t-Body-nav--dark" id="t_Body_nav" role="navigation" aria-label="&APP_TITLE!ATTR.">
<div><center></center></div>
<div class="t-TreeNav #COMPONENT_CSS_CLASSES#" id="t_TreeNav" data-id="#PARENT_STATIC_ID#_tree" aria-label="&APP_TITLE!ATTR." style="min-height: 85%; max-height: 85%"><ul style="display:none">
```

Listing 2

```

select
  ID,
  SORTIERUNG,
  '<center><b>'||name||'</b></center>' as CARD_TITLE,
  '<center><img src=""||foto||' width="100%" /></center>' as CARD_TEXT,
  '<center><b>'||klasse||'</b></center>' as CARD_SUBTEXT,
  Initialen as CARD_INITIALS,
  apex_util.prepare_url('f?p=&APP_ID.:25:&APP_SESSION.:::P25_TEAM:'||ID) as CARD_LINK
from VFR_TEAMS
order by SORTIERUNG

```

Listing 3

manuell wechselweise zur Ansicht gebracht werden (siehe Abbildung 3). Im Gegensatz dazu werden bei Tab Containern für alle Subregionen Tabs angezeigt und es findet kein automatischer Wechsel statt (siehe Abbildung 4).

Card Reports ermöglichen auf Basis einer Query, eine Karteikarte pro Datensatz zu erzeugen, die klickbar ist, um beispielsweise Details anzuschauen. Schlüsselwörter für die Ausgabespalten der Query bestimmen hierbei die Verwendung (siehe Listing 3 und Abbildung 5).

Basierend auf dem FullCalendar (siehe „<http://fullcalendar.io>“), können nun optisch ansprechende Kalender in Apex verwendet werden (siehe Abbildung 6). Dies kann per JavaScript erfolgen (siehe Listing 4). Natürlich ist als Grundlage auch eine Query möglich, mit der deklarativen Angabe, welche Ausgabe-Spalten wie verwendet werden sollen (siehe Abbildung 7).

Responsive Design

Ein Vorteil des Universal Theme und der angesprochenen Oberflächen-Komponenten ist ohne Frage die Tatsache, dass das Layout responsive ist. Mit Apex 5.0 wird ein dediziertes Entwickeln für mobile Devices nicht gänzlich überflüssig, aber in vielen Fällen kann eine Apex-Anwendung auch ohne eine solche auskommen. Insbesondere die oben erwähnten Card Reports sind ideal hierfür geeignet. Die Rückmeldung zur Seite auf mobilen Geräten war äußerst positiv und hat gezeigt, wie viele User mit mobilen Devices auf klassischen Webseiten unterwegs sind. Diesen Anwendungsfall bekommt man quasi mit dem Universal Theme geschenkt.

Den alten Content übernehmen

In fast zehn Jahren hatte sich einiger Content auf der Seite angesammelt, der nicht verloren gehen sollte. Der relevante HTML-Code der alten Seite wurde einfach per Copy & Paste in statische Regionen übertragen, die sich nun in den Bereichen „Chronik“ und „Archiv“ befinden. Im Archiv gibt es quasi einen dynamischen Übergang zwischen dem alten HTML-Content und den neuen, datenbankbasierten Informationen.

Neue redaktionelle Funktionen

Die Anwendung verfügt über ein einfaches Rechtekonzept mit sechs Berechtigungsrollen, die über Authorization Schemes abgebildet sind. So können nun mehrere User die Webseite bearbeiten und insbesondere auch weniger IT-erfahrene Vereinsmitglieder Inhalte beisteuern, da alle Informationen in Tabellen liegen und über dementsprechende Masken deklarativ geändert werden können. Wie in der Einleitung erwähnt, war dies schon vorher für Interna so implementiert, mit Apex wird dies nun auch für den Webseiten-Content verwendet. Neben veröffentlichten Mannschaftsaufstellungen oder den Spielplänen handelt es sich hierbei insbesondere um die „News“ und die „Hot News“.

Die Nachrichten und Berichte auf der Seite werden über eine klassische Form in die Datenbank geschrieben. Somit kann man eben auch ohne HTML-Kenntnisse Inhalte zur Verfügung stellen – wenn es sein muss, auch gerne mit SQL*Plus. Das notwendige HTML wird im

Hintergrund mithilfe eines PL/SQL-Packages gebaut. Dieses Package erzeugt nach einer vorgegebenen Struktur aus den Eingabefeldern (zu denen auch Text Areas mit Editor gehören) und Parametern jeweils ein Ziel-HTML, das mit in der Datenbank abgelegt wird (siehe Abbildung 8). In dem Bericht, der dann die Nachrichten ausgibt, wird diese Spalte mit dem HTML verwendet und durch das Setzen von „Escape special characters=No“ das gewünschte Ergebnis ausgegeben. Wichtige Informationen können mit Ablaufdatum in eine Maske eingegeben werden und erzeugen eine (bedingte) Region mit einem Laufband mit einer oder mehreren Nachrichten (siehe Abbildung 9).

PL/SQL

Neben dem Package zum Generieren der News im HTML-Format sind im Hintergrund noch zwei weitere PL/SQL-Packages im Einsatz. Im Tischtennis ist es verpflichtend, recht schnell das Spielergebnis in eine zentrale Anwendung einzupflegen, damit diese dann auch auf den offiziellen Seiten erscheinen. Warum sollten also die Spielergebnisse extra auf der Webseite eingepflegt werden, wenn sie woanders schon stehen.

Da es zu der zentralen Datenbank der Ergebnisse kein API gibt, wurde mit PL/SQL ein kleiner Crawler geschrieben, der regelmäßig die dementsprechende Webseite ausliest. Somit ist die Vereinsseite stets aktuell inklusive der Links zu den detaillierten Spielergebnissen. Das Package liest mit der Funktion „`httpuritype('myurl')`“ die Seite in eine „CLOB“-Variable, um diese dann mithilfe des Packages „DBMS_LOB“ nach den relevanten Informationen zu durchsuchen.

```

$( '#TCAL' ).fullCalendar({
  header: {
    left: '',
    center: 'title',
    right: ''
  },
  columnFormat: {
    month: '',
    week: 'dddd',
    day: ''
  },
  minTime: '14:00:00', maxTime: '22:00:00',
  allDaySlot: false, weekends: false,
  contentHeight: 350,
  slotEventOverlap: true,
  defaultView: 'agendaWeek',
  defaultDate: moment('2015-05-05'),
  titleFormat: '[Trainingswoche]',
  eventSources: [
    {
      events: [ {
        title : 'Erwachsene',
        start : '2015-05-04T19:00:00',
        end   : '2015-05-04T22:00:00',
        allDay : false
      },
      {
        title : 'Erwachsene',
        start : '2015-05-07T19:30:00',
        end   : '2015-05-07T22:00:00',
        allDay : false
      }
    ],
    color: 'black',
    textColor: 'yellow'
  },
  .....
});

```

Listing 4

Ein weiteres Package stellt FTP-Funktionalitäten zur Verfügung. Auf der Apex-Umgebung, die in der Cloud läuft, steht nur begrenzter Speicherplatz in der Datenbank sowie im Filesystem zur Verfüg-

ung. Daher werden die Bilder der Webseite an einer anderen Stelle abgelegt. Um nicht zusätzlich manuell mit einem FTP-Client die Bilder zu transferieren, wurde mithilfe des Packages „UTL_TCP“

ein schlanker FTP-Client geschrieben, der den Transfer der Bilder aus den Apex-Masken durchführt.

Apex in der Cloud

Durch die Architektur von Apex ist es für den Einsatz in der Cloud prädestiniert. Für die Vereins-Webseite wird ein kostengünstiger „Schema as a Service“ verwendet, der auf der Express Edition, also der Version 11g basiert. Das Einspielen neuer Patch Sets muss also nicht selbst durchgeführt werden, sondern erfolgt nach Ankündigung automatisch und sorgt lediglich für maximal eine Stunde Nichtverfügbarkeit der Seite.

Man mag ein paar Funktionen der anderen Editionen beziehungsweise des aktuellen Releases vermissen, aber selbst XE 11g stellt für solch ein Projekt nicht nur genügend Fähigkeiten, sondern auch eine sehr zufriedenstellende Performance und Stabilität zur Verfügung. Mehr zum Thema „Apex in der Cloud“ steht in den Artikeln von Niels de Bruijn und Carsten Czariski in dieser Ausgabe.

Fazit

Die Anwendung ist inzwischen produktiv gesetzt, der Test war also erfolgreich. Apex ist auch außerhalb des Business sinnvoll einsetzbar und kostengünstig betreibbar. Das Ergebnis ist unter „<http://vfr-fehlheim-tischtennis.de>“ zu sehen.

Hinweis: Die Abbildungen 1 bis 8 finden sie online unter: www.doag.org/go/redstack/willems_abb



Abbildung 9: Hot News



Rainer Willems
rainer.willems@oracle.com



Vom Rechteck zur Fabrikvisualisierung – Prozessvisualisierung in Apex mit HTML5 und SVG

Frank Schubert und Angela Wobar, AIS Automation Dresden GmbH

Für die Entwicklung von datenbasierten Web-Anwendungen stellt Apex ein ausgereiftes Entwicklungswerkzeug dar. Es bietet umfassende Möglichkeiten zur Präsentation von Daten in tabellarischer und grafischer Form. Mithilfe von HTML-Tags und CSS lassen sich vorgegebene Layouts einfach anpassen. HTML5 mit seinen erweiterten Sprach-Elementen bietet ganz neue Möglichkeiten, innovative und interaktive Web-Seiten zu gestalten.

Im Rahmen eines Projekts bestand die Aufgabe darin, die bestehende „C++“-Applikation FabEagle TFM auf eine mobile Plattform zu portieren. Augenmerk sollte dabei auf Plattform-Unabhängigkeit und ein flexibles Layout zur Umsetzung eines

„Responsive Webdesign“ gelegt werden. Die Visualisierung, die einen schnellen Überblick über den Zustand der Fertigung bietet, sollte dabei erhalten bleiben. Da die vorhandene Applikation auf einer Oracle-Datenbank basiert, stand

der Verwendung von Apex als Framework nichts im Wege. Aber wie können die Fabrik- und Prozessvisualisierungen migriert werden? Dieser Artikel soll dabei keine Anleitung für eine Implementierung sein, sondern mit möglichst ein-

fachen Beispielen Anregungen für eine Umsetzung liefern.

SVG und seine Möglichkeiten

Mit der Einführung von HTML5 rendern die modernen Browser Scalable Vector Graphics (SVG) ohne Plug-in in HTML-Seiten. SVG stellt für das Zeichnen beliebiger Formen das Pfad-Element zur Verfügung. Zur Vereinfachung gibt es für die Grundformen Rechteck, Kreis, Ellipse, Linie und Polygon ebenfalls entsprechende Elemente. Darüber hinaus bietet SVG Sprachelemente für Text, Animation und zum Einfügen externer Grafiken. SVG basiert auf XML, lässt sich mit CSS formatieren und mit JavaScript dynamisch ändern. Eine vollständige Abhandlung aller Sprachelemente würde an dieser Stelle den Rahmen sprengen. *Listing 1* zeigt ein kleines Beispiel, *Abbildung 1* die entstehende Grafik.

Aufgrund der einfachen Umsetzbarkeit und der in der Datenbank bereits vorhandenen Layout- und Equipment-Koordinaten lag es nahe, das SVG selbst zu generieren. Beispielhaft werden hier zwei Objekte, eine Produktionslinie („line 1“) und ein Equipment („equipment 1“), in einem Fertigungsbereich („shop-floor“) platziert. *Abbildung 2* zeigt das zugrunde liegende Koordinatensystem. In *Listing 2* ist die beispielhafte Umsetzung in SVG dargestellt; Fertigungsbereich, Equipment und Produktionslinie werden vereinfacht durch Rechtecke symbolisiert (siehe *Abbildung 3*).

Schöner geht immer ...

Die grundsätzlichen Anforderungen an die Fabrikvisualisierung sind nun erfüllt. Anlagen lassen sich positionsgenau im Fertigungsbereich platzieren. Ein Farbschema dient zur Visualisierung der Anlagenzustände. Werden nun aber mehrere Anlagen dargestellt, wird es schnell unübersichtlich. Zusätzliche räumliche Merkmale fehlen.

Natürlich könnte man die Gruppe „shop-floor“ erweitern und zusätzliche Informationen einzeichnen. Die Autoren sind jedoch einen anderen Weg gegangen. Zu jeder Fertigung existieren CAD-Zeichnungen, die als SVG exportiert werden können. Werden diese SVG-Dateien unter den „Shared Components“ der Applikation ab-

```
<svg width="640" height="480">
  <g stroke="#000000" stroke-width="2">
    <rect height="50" width="100" y="50" x="50"
      fill="#7fff00"/>
    <circle r="30" cy="100" cx="150" fill="#aad4ff"/>
  </g>
</svg>
```

Listing 1: Einfaches SVG-Beispiel

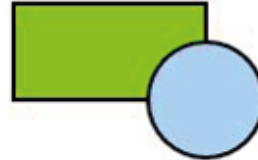


Abbildung 1: Das Ergebnis von Listing 1 - Rechteck und Kreis

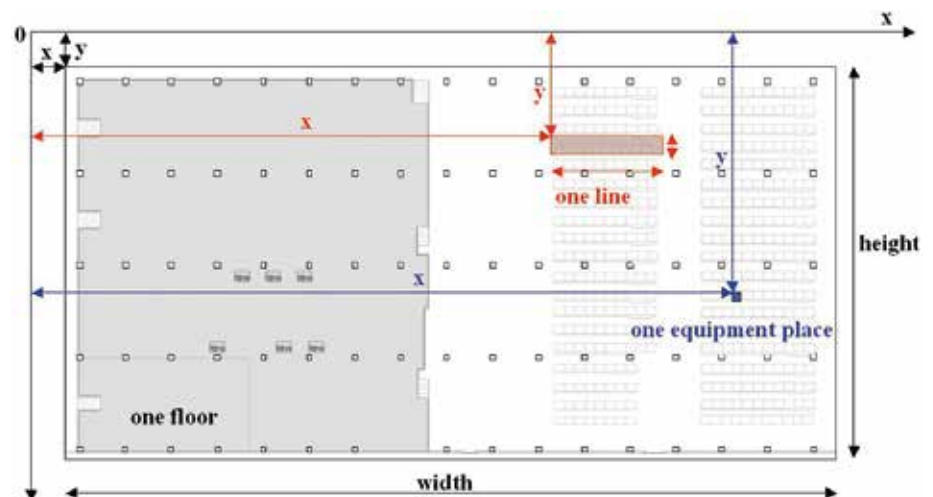


Abbildung 2: Fab-Koordinatensystem

```
<svg width="1000" height="400">
  <g stroke="#000000" stroke-width="1">
    <!-- shop-floor -->
    <rect x="0" y="0" width="1000" height="400"
      fill="#ffffff" />
    <!-- line 1 -->
    <rect x="600" y="100" width="100" height="10"
      fill="orange" />
    <!-- equipment 2 -->
    <rect x="800" y="230" width="20" height="10"
      fill="blue" />
  </g>
</svg>
```

Listing 2: Implementierung als SVG anhand der existierenden Datenpunkte

gespeichert, können sie über ein Image-Tag eingebunden und so als Hintergrund genutzt werden.

Ein Beispiel dafür zeigen *Listing 3* und die daraus resultierende *Abbildung 4*. So-

mit haben wir mit wenigen Zeilen Code eine sehr ansprechende Visualisierung implementiert. Wem die Equipments noch zu rechteckig sind, kann auch deren Darstellung durch SVG-Dateien ersetzen.

Generieren von SVG in PL/SQL

Bei komplexeren Visualisierungen empfiehlt es sich, die Generierung des SVG nicht direkt in Apex durchzuführen, son-

dern in die Datenbank auszulagern und entsprechende Funktionen beziehungsweise Prozeduren zu erstellen. Das ermöglicht einfacheres Testen, Debuggen und Fehlerhandling. Außerdem sollten die in Oracle vorhandenen XML-Funktionen ge-

nutzt werden. Das garantiert Typsicherheit und eine bessere Performanz. Ein Anwendungsbeispiel ist in *Listing 4* zu sehen. Zudem bietet PL/SQL viele Funktionen zum Modifizieren von existierendem XML (wie „insertchildxml“, „updateXML“ etc.).



Abbildung 3: Resultierende Darstellung

```
<svg width="1010" height="710">
  <!--shop-floor -->
  <image x="0" y="0" height="710" width="1040"
    xlink:href="#APP_IMAGES#FAB.svg" />
  <!-- equipments -->
  <g stroke="#000000" stroke-width="1">
    <!-- equipment 1 -->
    <rect x="555" y="127" width="100" height="10"
      fill="orange" />
    <!-- equipment 2 -->
    <rect x="800" y="230" width="10" height="20"
      fill="blue" />
  </g>
</svg>
```

Listing 3: Integration von externen SVG-Dateien über ein Image-Tag

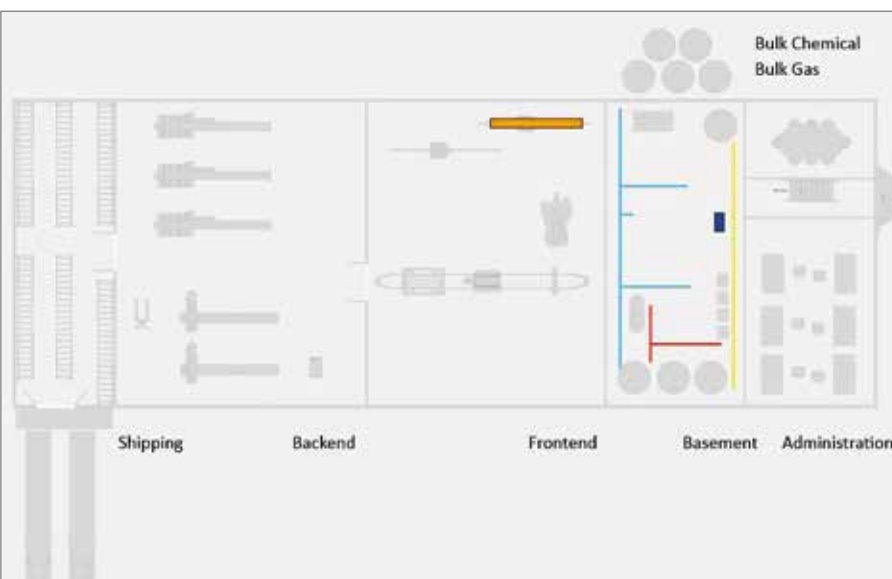


Abbildung 4: Resultierende Darstellung – externes SVG als Fabriklayout und 2 Equipments

Benutzer-Interaktionen

Von innovativen Benutzeroberflächen erwarten Anwender heutzutage dennoch mehr Funktionalität als nur eine statische Grafik. Tooltips, Links und Zoom gehören zu modernen Anwendungen einfach dazu. Die Kombination von JavaScript und SVG macht auch das möglich.

Den SVG-Elementen können Maus-Events („onmouseout“, „onmousemove“, „onclick“) hinzugefügt werden. So lassen sich mit JavaScript-Funktionen Tooltips ein- beziehungsweise ausblenden und Links öffnen (siehe *Listing 5* und *Abbildung 5*). Im Internet lassen sich viele Beispiele für die Umsetzung von Tooltips in JavaScript finden. So lässt sich für jedes Equipment ein eigener Tooltip-div-Container erstellen, der mit Informationen über das Equipment gefüllt ist. Über CSS werden das Aussehen des Tooltip sowie dessen Position geregelt.

Neben dem im Browser vorhandenen Zoom möchte mancher Anwender eventuell auch die Visualisierung vergrößern, ohne die restliche Seite zu beeinflussen. Um diese Forderung umzusetzen, kann das ViewBox-Attribut des SVG-Elements genutzt werden. Die ViewBox definiert einen Anzeigebereich. Sie enthält vier Zahlenwerte, die den x- und y-Versatz sowie die Breite und Höhe des anzuzeigenden Koordinatensystems beschreiben. Alle Kind-Elemente des SVG werden in dem in der ViewBox beschriebenen Koordinatensystem dargestellt.

Wie groß die Grafik auf der Seite dargestellt ist, bestimmt sich aus den Attributen „height“ und „width“ des SVG-Elements (siehe *Listing 6* und *Abbildung 6*). Zum Zoomen wird dem SVG-Element beispielsweise ein Mousewheel-Event hinzugefügt, das eine Neuberechnung der ViewBox auslöst.

Heutzutage kann nicht davon ausgegangen werden, dass auf eine Website nur von einem Computer-Desktop aus zugegriffen wird. Der steigende Markt-

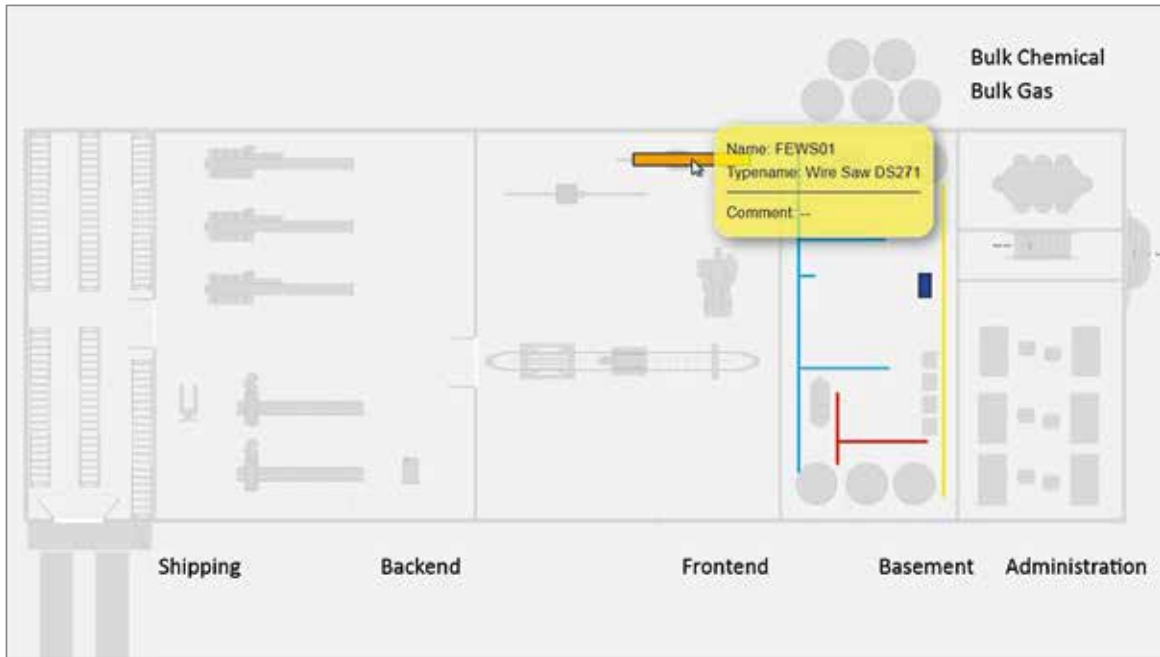


Abbildung 5: Resultierende Darstellung – Equipment mit Tooltip

teil von Smartphones und Tablets führt zu der Anforderung, die Website so flexibel zu gestalten, dass die Benutzerfreundlichkeit unabhängig vom verwendeten Endgerät erhalten bleibt. „Responsive Webdesign“ lautet das Stichwort.

Für ein SVG ist diese Technik besonders einfach umzusetzen. Anstatt einen festen Wert für die Breite und Höhe des SVG anzugeben, kann eine flexible Breite in Prozent zugewiesen werden (siehe Listing 7). So passt sich das SVG jeder Auflösung an.

Dynamische Aktualisierung

Nun bringt eine Prozessvisualisierung reichlich wenig, wenn man nur statische Daten anzeigt. Daher ist es notwendig, eine Aktualisierung umzusetzen. Der einfachste Weg wäre, eine Funktion zur Aktualisierung der gesamten Seite, auf der sich die Visualisierung befindet, zu implementieren und diese in einem bestimmten Zeitintervall aufzurufen. Dies führt jedoch zu Performance-Einbußen, da auch Daten nachgeladen werden, die sich nicht verändert haben. Eine performantere Lösung stellt das dynamische Aktualisieren der Seite dar, bei der nur die DOM-Elemente bearbeitet werden, deren zugrunde liegende Daten sich auch tatsächlich verändert haben.

```
<svg id="svg_zoom" width="100" height="100" viewBox="0 0 100 100">
[...]
```

```
<svg id="svg_zoom2" width="100" height="100" viewBox="-100 -100 90 90">
[...]
```

Listing 6: Zwei SVGs mit gleicher Höhe und Breite, aber unterschiedlichem ViewBox-Attribut

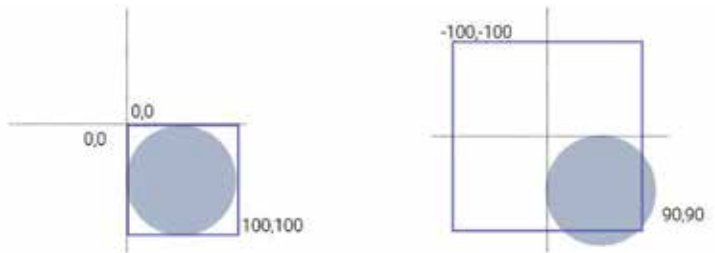


Abbildung 6: Resultierende Abbildung – Der blau umrandete Bereich würde jeweils mit einer Größe von 100x100 Pixeln dargestellt werden.

```
<svg style="width:100%" viewBox="0 0 100 100">
[...]
```

Listing 7: Dieses SVG passt sich flexibel an die Bildschirmgröße an.

Dazu kann beispielsweise ein Prozess erstellt werden, der bei einem AJAX-Aufruf gestartet wird und ein JSON-Objekt zurückliefert. *Abbildung 7* zeigt die Konfiguration eines solchen Prozesses. Der Source-Typ ist PL/SQL-Code, in dem das JSON-Objekt in den HTP-Buffer geschrieben wird. Die

Daten kann der Apex-Entwickler manuell einfügen. Dazu können PL/SQL-Funktionen wie „htp.prn()“ zum Einsatz kommen.

Um ein valides JSON-Format zu erhalten, ist jedoch genau darauf zu achten, an den richtigen Stellen entsprechend den Datentypen Anführungszeichen und alle erforder-

lichen Klammern und Kommata zu setzen. Um Formatfehler im JSON zu vermeiden, empfiehlt sich die Verwendung des mit Apex 5.0 eingeführten „Apex_JSON“-Package. *Abbildung 8* zeigt das JSON-Objekt, das unser Beispiel-Prozess zurückliefert.

Der AJAX-Aufruf wird clientseitig gestartet. In Apex lässt sich der „apex.server“-Namespace für eine solche Kommunikation mit der Datenbank nutzen. Durch die asynchrone Datenübertragung bleibt die Seite bedienbar, während auf die Server-Antwort gewartet wird. Außerdem ist es möglich, mehrere asynchrone Anfragen gleichzeitig zu stellen.

Die vom Server erhaltenen Daten lassen sich im JavaScript-Code weiterverarbeiten. Über die zugrunde liegende DOM-Struktur können SVG-Elemente direkt adressiert werden. Dabei wird das „id“-Attribut verwendet. *Listing 8* zeigt einen beispielhaften Serveraufruf zur Daten-Aktualisierung.

Prozessvisualisierung

Nach dem gleichen Prinzip kann man auch Prozessvisualisierungen beziehungsweise Anlagen-Fließbilder (*siehe Abbildung 9*) in eine HTML-Seite integrieren. Das Fließbild wird dabei komplett als SVG-Grafik erstellt und die Aktualisierung der Prozesswerte und Zustände erfolgt über JavaScript. Die Objekte werden über das „id“-Attribut der SVG-Elemente angesprochen und aktualisiert.

Wiederwendbarkeit

Um die realisierte Fabrikvisualisierung auch in anderen Projekten verwenden zu können, ist ein Apex-Plug-in entstanden. Um dieses nutzen zu können, sind nur ein Select-Statement für das Fabrik-Layout und ein weiteres Statement für die anzuzeigenden Zustände und Detail-Informationen der einzelnen Anlagen zu übergeben.

Fazit

Mit HTML5 und SVG lassen sich komplexe Visualisierungen einfach in Apex integrieren. Die resultierenden Grafiken sind skalierbar und auch auf mobilen Endgeräten gut lesbar. Auch Benutzer-Interaktionen lassen sich einfach umsetzen. Trotz komplexer

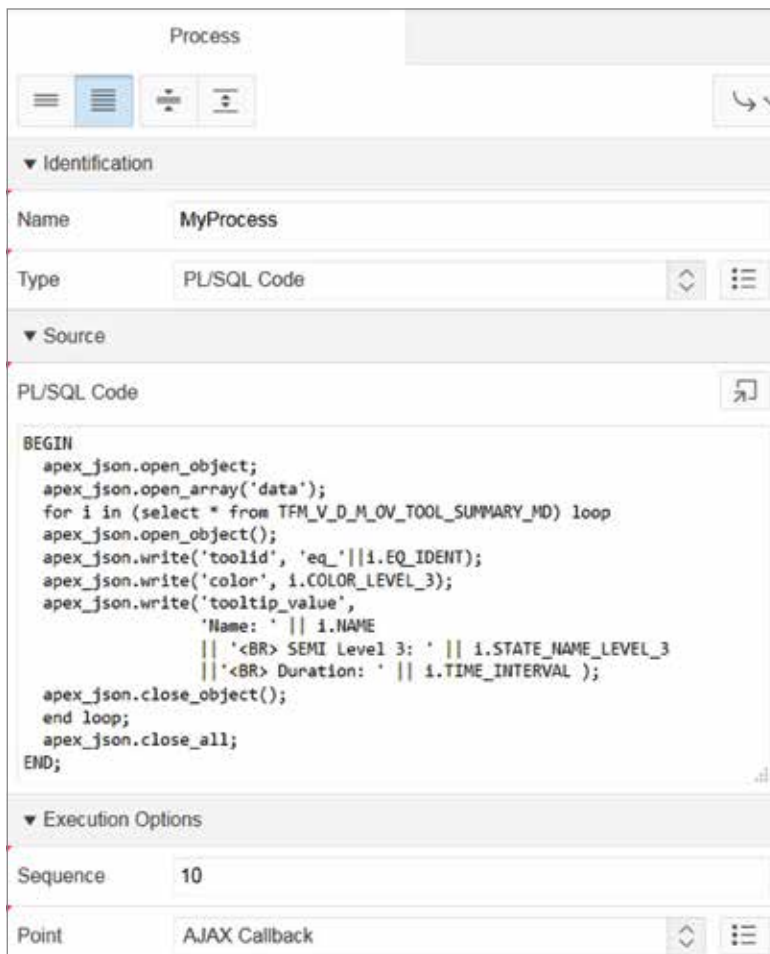


Abbildung 7: Beispiel-Konfiguration eines AJAX-Prozesses zur Aktualisierung

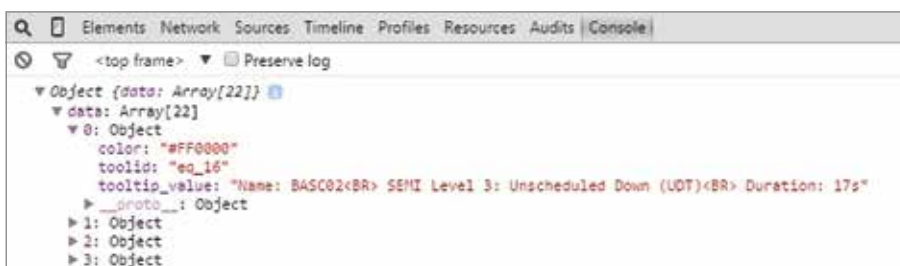


Abbildung 8: Das entstandene JSON-Objekt

```

/*Serveraufruf um aktuelle Daten zu erhalten*/
apex.server.process („MyProcess“, { /*Übergabeparameter*/ }, {
  /*Callback-Funktionen*/
  success: function(pData) {
    //pData enthält die Serverantwort
    /* Funktion wird im Falle einer erfolgreichen Datenabfrage
    ausgeführt --> Weiterverwendung pData nach Wunsch, z.B.
    zur Aktualisierung der Farben und Tooltips*/
  },
  error: function(pData) {console.log („error“, pData);}
});

/*Funktion zur Aktualisierung der Farbe*/
function setToolColor(tool_id, color){
  $( „#“ + tool_id ).attr („fill“, color);
}

```

Listing 8: Verwendung des „apex.server“-Namespace und Beispielfunktion zur Farb-Aktualisierung

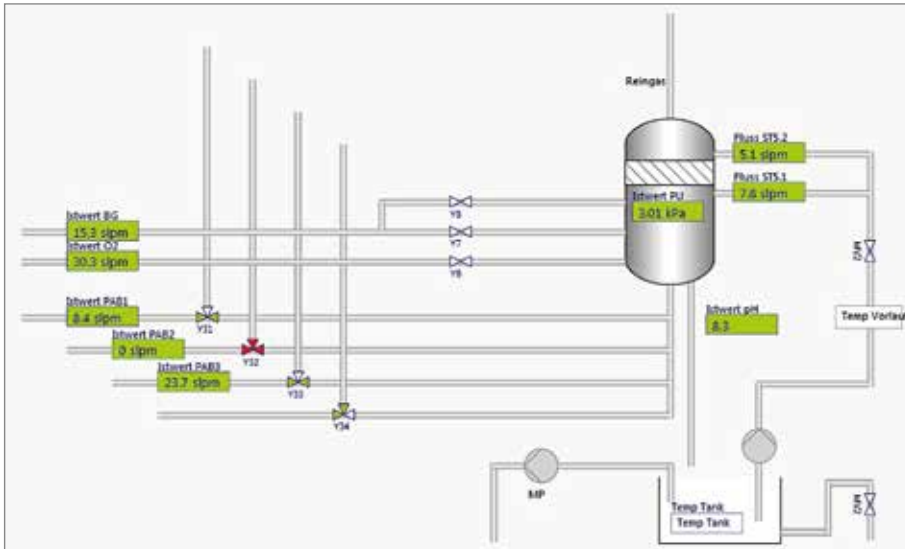


Abbildung 9: Prozessvisualisierung mit SVG

Grafiken waren keine Performance-Probleme bei der Darstellung in verschiedensten Browsern und Endgeräten feststellbar. Wer

Interesse hat, findet über das Projekt unter „<https://tfmportable.ais-automation.com/apex/?p=400:700>“ eine Live-Demo.



Frank Schubert
frank.schubert@ais-automation.com



Angela Wobar
angela.wobar@ais-automation.com

Heatmap und ADO in der Praxis

Florian Feicht, Trivadis GmbH

Aufgrund der rasant wachsenden Datenmengen gewinnt der Lebenszyklus der Daten in der Datenbank immer mehr an Bedeutung. Müssen die Archivdaten auf teurem SAN liegen? Können weniger genutzte Daten komprimiert werden? Gibt es einfachere Möglichkeiten? Die Oracle-Antwort in der Version 12c ist ein in die Datenbank integriertes Information-Lifecycle-Management-System (ILM). Dazu wurden die Features Heat Map und Automatic Data Optimization (ADO) eingeführt.

Seit der Version 9i bietet die Oracle-Datenbank die Möglichkeit, Daten zu komprimieren. Dadurch benötigen diese im Idealfall weniger Speicherplatz und der Lesezugriff wird beschleunigt. Leider ist es aufgrund des Performance-Overheads nicht möglich, Komprimierung für die gesamte Datenbank auf Knopfdruck zu aktivieren. Die Entscheidung darüber, wo Komprimierung sinnvoll ist, erfordert gute Kenntnisse der Applikation. Dazu musste bisher eine aufwändige Auswertung er-

stellt werden. Außerdem ist es oft nicht sinnvoll, ganze Tabellen zu komprimieren, sondern nur die selten benutzten Einträge. Eine Methode, verschiedene Storage-Klassen am Datenbankserver automatisiert zu verwalten, stand nicht zur Verfügung.

12c bringt die Übersicht

Mit Einführung der Version 12c wurde ein Information Lifecycle Management imple-

mentiert. Eine automatisierte Verwaltung der Daten durch ihren „Lebenszyklus“ ist dadurch sehr einfach möglich. Um den aktuellen Lifecycle-Stand zu überwachen, wird die Heat Map genutzt (siehe Abbildung 1). Mit diesem Feature werden die Zugriffe auf die Datenbankobjekte zentral protokolliert. Dadurch entsteht eine feingranulare Übersicht auf folgende Ebenen:

- Block
- Extent

- Segment
- Objekt
- Tablespace

Schreib- und Lesezugriffe werden separat protokolliert. Somit entsteht eine Karte der am häufigsten verwendeten Objekte. Im Zusammenhang mit der Heat Map wird von Daten gesprochen, die „hot“, „warm“ und „cold“ sind. In der View „V\$HEAT_MAP_SEGMENT“ (siehe Listing 1) ist dies online im Speicher protokolliert. Die Informationen werden periodisch in die Datenbank geschrieben. Der Zugriff erfolgt über die entsprechenden „DBA/USER/ALL“-Views. Hier sind unter anderem „DBA_HEATMAP_TOP_OBJECTS“ und „DBA_HEAT_MAP_TOP_TABLESPACES“ hilfreich. Gespeichert sind die Daten der Heat Map im „SYSAUX“-Tablespace.

Zusätzlich steht das Package „DBMS_HEAT_MAP“ zur Verfügung, mit dem verschiedene Auswertungen erstellt werden können (siehe Listing 2). Zu beachten ist, dass Objekte im „SYSTEM“- und „SYSAUX“-Tablespace von der Überwachung ausgenommen sind. Außerdem sind interne Systemjobs, DDLs, Redefinitions und die Sammlung der Statistiken nicht in der Heat Map protokolliert.

Die Heat Map aktivieren

Die Heat Map ist ein fest im Kernel integriertes Feature. Sie wird mit dem Befehl „ALTER SYSTEM SET heat_map=ON;“ aktiviert. Der Parameter kann auf System- oder auf Session-Ebene gesetzt sein, wodurch eine einfache Deaktivierung für Wartungsarbeiten ermöglicht wird. Durch Aktivierung der Heat Map entsteht ein Overhead abhängig von der Gesamtlast des Systems. Dies muss je nach Applikation genau untersucht werden.

Um die gesammelten Daten sinnvoll verwenden zu können, kommt das nächste Feature zum Einsatz, die Automatic Data Optimization (ADO). Vereinfacht gesprochen, werden hier ADO-Regeln (Policies) auf die Heat Map gesetzt, die mit entsprechenden Aktionen verbunden werden können. Grundsätzlich gibt es zwei Möglichkeiten:

- Storage Tiering
- Compression Tiering

Storage Tiering ermöglicht die Verwaltung mehrerer Storage-Klassen in der Datenbank. Diese werden über verschiedene Tablespaces, die auf unterschiedlichen physikalischen Storage liegen, abgebildet. Mit „ALTER TABLE config_mgr ILM ADD POLICY TIER TO tbs_nas_storage;“ lässt sich die Tiering Policy auf Tabellen- beziehungsweise auf Partitions-Ebene aktivieren.

Das Keyword „TIER TO“ definiert die Storage-Policy. So wird die Tabelle „config_mgr“ beziehungsweise das Segment „config_mgr“ nach einer definierbaren Belegung des momentanen Tablespace verschoben. Es wird nur die Tabelle verschoben, die Indizes bleiben unberührt (siehe MOS Doc ID 2025280.1).

Im Hintergrund wird das betroffene Segment mit „MOVE“ bewegt. Dadurch entsteht ein exklusiver Lock. Falls dieser Lock wegen blockierender Sessions nicht erreicht werden kann, wird die Policy nicht ausgeführt. Die Ausführung wird im nächsten Wartungsfenster beziehungsweise bei der nächsten manuellen Ausführung wiederholt.

Am häufigsten ist die Anwendung von Storage Tiering bei partitionierten Tabellen. Dabei erfolgt ein „MOVE“ auf Partitionsebene. In der Version 12c wurde für Partitionen die Möglichkeit „MOVE ONLINE“ eingeführt, die hier nicht genutzt wird. Leider ist kein Tiering nach Zugriffszeit der Segmente auf Basis der Heat-Map-Daten vorgesehen. Dies kann allerdings über sogenannte „Custom“ Policies erfolgen. Die Syntax ist dazu um „ON FUNKTIONSDNAME“ zu erweitern. Jede Funktion, die als Rückgabewerte einen „BOOLEAN“-Datentyp hat, kann genutzt werden. Sobald die Policy einmal erfolgreich etabliert ist, wird sie automatisch deaktiviert. Für eine weitere Ausführung

ist eine manuelle Reaktivierung mithilfe von „ENABLE POLICY“ erforderlich.

Die zweite Möglichkeit der Policy-Definition ist das Compression Tiering. Dabei lassen sich Policies auf Zeilen-, Segment- oder Group-Ebene definieren. Diese führen eine Komprimierung auf Basis der Zeitstempel der letzten Zugriffe durch. Hier kann sowohl der Kompressions-Algorithmus als auch die Auswahl der zu bewertenden Heat-Map-Daten beeinflusst werden.

Dazu ein Syntax-Beispiel für Advanced Row/Low Securefile Compression auf Block-Ebene: „ALTER TABLE config_mgr ILM ADD POLICY ROW STORE COMPRESS ADVANCED SEGMENT AFTER 30 DAYS OF NO MODIFICATION;“. Die Syntax sieht hier zwar „ROW“ vor, im Hintergrund wird allerdings jeweils der ganze Block geprüft. Das bedeutet, dass der gesamte Block dem ausgewählten Kriterium entsprechen muss.

Stärkere Kompressions-Algorithmen setzen eine Oracle-Hardware voraus. „COMPRESS FOR QUERY“ und „COMPRESS FOR ARCHIVE“ aktivieren die Hybrid Columnar Compression sowie „MEDIUM“- beziehungsweise „HIGH“-SecureFile-Kompression. Jedes Kompressionsverfahren komprimiert die Indizes zusätzlich mit der Standard-Komprimierung. Für die Betrachtung der Zugriffszeit der Heat-Map-Daten stehen „NO ACCESS“, „NO MODIFICATION“, „CREATION“ und „LOW ACCESS“ zur Verfügung. Bei Compression Tiering auf Segment-Basis kann analog zum Storage Tiering die „ON“-Klausel mit eigener Funktion verwendet werden.

Damit die Ausführung der Policy erfolgt, sind gesammelte Statistiken für die involvierten Objekte notwendig. Der Umfang einer Group ist derzeit nicht dokumentiert. Die Policy auf Segment-Ebene führt im Wartungsfenster „MOVE COMPRESS FOR OLTP“ durch. Dafür muss genug freier Platz



Abbildung 1: Heat Map im Enterprise Manager

im Tablespace vorhanden sein. Außerdem erfolgt wieder ein exklusiver Lock.

Bei der zeilenbasierten Variante werden aus der Heat Map die zu komprimierenden ROWIDs generiert. Anschließend wird mit „SELECT ROWNUM FROM config_mgr FOR COMPRESS 2“ über die Session-Statistik „HSC OLTP Space Saving“ das Einspar-Potenzial ermittelt. Leider ist nicht erkennbar, auf welche Art die einzelnen Rows komprimiert werden.

Diese Policies bieten den großen Vorteil, dass die jeweilige Applikation nicht angepasst werden muss. Storage und Compression Tiering kann transparent für die Applikation und gezielt auf dafür geeigneten Objekten erfolgen. Somit lassen sich die Auswirkungen auf die Performance deutlich verringern und die Komprimierung wirkungsvoller einsetzen. Row-basierte Policies erfolgen alle fünfzehn Minuten, Segment-basierte im Wartungsfenster der Datenbank. Alternativ steht mit „DBMS_ILM_ADMIN“ ein Package zur Administrator-gesteuerten Ausführung der Policies zur Verfügung. Damit lassen sich auch weitere Parameter konfigurieren. Aufeinander aufbauende Policies sind problemlos möglich. Die Ausführung der Policies kann über die entsprechenden „DBA/USER/ALL“- Views (siehe Listing 3) überwacht werden.

Einschränkungen von Heat Map und ADO

Für die in diesem Artikel beschriebenen Features ist die Enterprise Edition und zusätzlich die „Advanced Compression“-Option erforderlich. Leider ist die Heat Map und damit auch ADO nur in Non-CDB-Datenbanken nutzbar. Darüber hinaus gibt es Einschränkungen für bestimmte Datentypen. Die ADO-Policies funktionieren nur in eine Richtung. Demzufolge gibt es keine Dekomprimierung oder ein Zurückschieben auf den ursprünglichen Tablespace.

Der zweite Fall lässt sich über den Umweg einer Custom-Funktion mit entsprechender Heat-Map-Auswertung in der Storage-Tiering-Policy realisieren. Bei deren Einsatz ist vorab zu prüfen, ob im Ziel-Tablespace ausreichend Platz vorhanden ist – eine automatische Prüfung erfolgt nicht. Zudem ist zu beachten, dass die automatische Ausführung der ADO-Policies im Wartungsfenster der Datenbank nicht funktioniert, wenn

„Supplemental Logging“ aktiviert ist. Als Alternative bleibt die manuelle Ausführung. Nach zweimaliger fehlerhafter Ausführung werden die Policies automatisch deaktiviert.

Anwendungsfälle aus der Praxis

Im ersten Beispiel (Dokumentenverwaltung) legt die Applikation Dokumente und Metadaten in der Datenbank ab. Auf Kundenwunsch sollen alle Blöcke, die seit drei Monaten nicht geändert worden sind, automatisch komprimiert werden. Dafür kommen die Advanced Row Compression für die Heat-Tabellen und Low SecureFile Compression zum Einsatz. Um die Anforderung umzusetzen, wird nach Aktivierung der Heat Map auf allen Tabellen mit „ALTER TABLE document_mgr ILM ADD POLICY ROW STORE COMPRESS ADVANCED ROW AFTER

3 MONTH OF NO MODIFICATION;“ eine Automatic-Data-Optimization-Policy erstellt. Diese Policy komprimiert nur die Blöcke, die dem geforderten Zeitfenster entsprechen, der Rest bleibt unberührt. Dadurch minimiert sich das Risiko von „Chained Rows“ bei Updates auf „hot“-Daten deutlich.

Verschiedene Performance-Tests haben ergeben, dass der Overhead durch Aktivieren der Heat Map bei maximal zehn Prozent liegt. Aufgrund der durch Komprimierung deutlich kleineren Segmente konnte der Gesamt-Overhead aber sehr gering gehalten werden, um deutlich Speicherplatz zu sparen. Eine Monitoring-Lösung für die Überwachung der Policy-Ausführung ist unbedingt erforderlich (siehe Listing 4).

Im zweiten Beispiel stehen im Unternehmen verschiedene Storage-Klassen zur Verfügung. Durch die interne Verrechnung der Storage-Belegung soll nach Anforderung der Datenbank-Abteilung, wo immer mög-

```
SELECT object_name
       ,track_time
       ,segment_write AS seg_w
       ,segment_read AS seg_r
       ,full_scan
       ,lookup_scan
FROM gv$heat_map_segment;
```

OBJECT_NAME	TRACK_TIME	SEG_W	SEG_R	FUL	LOO
CONF_OBJ_C1	09.05.2015 20:15:38	YES	NO	YES	NO
CONF_SERVIC	09.05.2015 20:15:38	YES	NO	NO	YES

Listing 1

```
SELECT owner
       ,segment_name
       ,block_id
       ,writetime
FROM TABLE (
  dbms_ilm.heat_map.block_heat_map (
    owner=>'DOCMGR'
    ,segment_name=>'CONFIG_MGR')
);
```

Listing 2

```
SELECT policy_name
       ,policy_type
FROM dba_ilmpolicies;
```

POLICY_NAME	POLICY_TYPE
P443	DATA MOVEMENT
P445	DATA MOVEMENT

Listing 3

lich und sinnvoll, günstiger – wenn auch langsamer – Storage verwendet werden. Dabei sind verschiedene Varianten denkbar: zum einem eine Standard-Storage-Tiering-Policy, die Segmente nach Belegung der Tablespace verschiebt, und zum anderen eine „ON FUNCTION“- Storage-Tiering-Policy, die anhand der Daten aus der Heat Map über die Storage-Klasse entscheidet.

Aufgrund der detaillierten Auswertungsmöglichkeit wird die zweite Variante aus-

gewählt. Das erstellte Konzept sieht vor, dass alle Daten, auf die seit vier Wochen nicht zugegriffen wurde, im günstigeren Storage gespeichert werden können. Dazu definiert man zunächst High-End- und Low-End-Tablespaces. Zur Auswertung der Heat-Map-Daten wird eine Funktion erstellt (siehe Listing 5). Sie wird mit „ALTER TABLE ILM ADD POLICY TIER TO low_end_tbs ON not_accessed;“ in die Applikations-Tabellen per Policy eingebunden.

Um tatsächliche Platzeinsparungen zu ermöglichen, laufen regelmäßig Jobs, die „RESIZE DATAFILE“ auf dem High-End-Storage durchführen. Zusätzlich ist auf der Low-End-Seite zu beachten, dass genug freier Speicherplatz in den Tablespaces zur Verfügung steht. Ein weiterer wichtiger Punkt des Konzepts ist das Monitoring der Heat-Map-Views in Bezug auf alle bereits verschobenen Objekte. Da die Storage-Tiering-Policy kein erneutes Verschieben der Daten ermöglicht, muss manuell geprüft werden, ob die Segmente wieder „hot“ sind. Diese Überwachung erfolgt mithilfe der kumulierten Ergebnisse aus den Heat-Map- und ILM-Views (siehe Listing 6).

```
SELECT task_id
       ,job_name
       ,job_state
FROM dba_ilmresults;
```

TASK_ID	JOB_NAME	JOB_STATE
854	ILMJOB2760	COMPLETED SUCCESSFULLY
853	ILMJOB2752	COMPLETED SUCCESSFULLY

Listing 4

```
CREATE FUNCTION not_accessed(objn IN NUMBER)
RETURN BOOLEAN
IS
  days NUMBER:=0;
BEGIN
  SELECT sysdate - segment_read_time
  INTO days
  FROM dba_heat_map_segment
  WHERE object_name = (SELECT object_name FROM user_objects WHERE
object_id = objn);

  IF (days > 28)
  THEN
    RETURN true;
  ELSE
    RETURN false;
  END IF;
END;
```

Listing 5

```
SELECT e.object_owner
       ,e.object_name
       ,r.job_state
       ,h.full_scan
FROM dba_ilmresults r
     ,dba_ilmevaluationdetails e
     ,dba_heat_map_segment h
     ,dba_segments s
WHERE r.task_id = e.task_id
     AND e.object_owner = h.owner
     AND e.object_name = h.object_name
     AND e.object_owner = s.owner
     AND e.object_name = s.segment_name
     AND r.job_state = 'COMPLETED SUCCESSFULLY'
     AND s.tablespace_name = 'LOW_END_TBS';
```

Listing 6

Fazit

Heat Map und Automatic Data Optimization erleichtern den Einsatz der Advanced-Compression-Features in der Oracle-Datenbank deutlich. Die Aktivierung der Heat Map gibt ein klares Bild über die Nutzungshäufigkeit der einzelnen Objekte in verschiedenen Detailstufen. Entsprechende Automatic-Data-Optimization-Policies nutzen die gesammelten Daten automatisiert zu konkreten Aktionen. Der Datenbank-Administrator hat damit die Möglichkeit, die Auswirkungen auf die Zugriffszeiten gering zu halten und gleichzeitig, wo es sinnvoll ist, Storage-Einsparungen zu erreichen.

Die Verwendung von Storage Tiering kann verschiedene Storage-Klassen am Datenbankservers einfach verwalten. Für den Einsatz der Automatic Data Optimization ist anforderungsspezifisch ein Designkonzept erforderlich. Dabei ist das Monitoring unbedingt zu berücksichtigen. Zu bedenken ist allerdings, dass die neuen Features Heat Map und ADO für die Multitenant-Architektur noch nicht freigegeben sind.



Florian Feicht
florian.feicht@trivadis.com

Oracle Database Standard Edition 2

Johannes Ahrends, CarajanDB GmbH

Seit dem 1. September 2015 ist es endlich möglich, das Upgrade der Oracle Standard Edition oder Standard-Edition-1-Datenbank auf das aktuelle Release 12.1.0.2 durchführen. Aber bevor die Euphorie zu groß wird: Es gibt einige gravierende Änderungen, die in diesem Artikel näher beschrieben werden.

Zunächst ein wenig Historie: Bis einschließlich Version 12.1.0.1 gab es einen Download für alle Editionen – also Enterprise Edition, Standard Edition, Standard Edition 1 und Personal Edition. Einzig die kostenlose Express Edition war nicht enthalten. Mit der Freigabe von 12.1.0.2 im Juli 2014 plötzlich die Überraschung: Die Installationen von Standard Edition, Standard Edition 1 (SE1) und Personal Edition sind ausgegraut, die Version 12.1.0.2 ist also den Enterprise-Edition-Anwendern vorbehalten. Diese Hängepartie bleibt über ein Jahr bestehen und erst am 1. September 2015 verkündet Oracle endlich die Verfügbarkeit von 12.1.0.2 für die Standard Edition 2 (SE2). Statt einer Freigabe für die bestehenden Editionen gibt es also jetzt eine neue Edition.

Lizenzierung der SE2

Zunächst einmal bedeutet diese Freigabe, dass Unternehmen, die die Standard Edition beziehungsweise SE1 lizenziert haben, das aktuelle Oracle Release 12.1.0.2 immer noch nicht einsetzen können. Die Release-Verfügbarkeit für diese beiden Editionen endet mit der Version 12.1.0.1. Wer auf die Version 12.1.0.2 oder spätere Releases wechseln möchte oder besser gesagt muss, der benötigt entweder eine Enterprise Edition oder die neue SE2. Oracle bietet hier folgende Migrationsoptionen an:

- Unternehmen, die die Standard Edition einsetzen, können kostenfrei auf die SE2 wechseln.
- Unternehmen, die die SE1 einsetzen, können für eine moderate Erhöhung des Supportbetrages um zwanzig Prozent auf die SE2 wechseln.

Bevor die SE2 statt der bisherigen Editionen eingesetzt werden darf, muss eine Migration der bestehenden Lizenzen durch Oracle erfolgt sein.

Einschränkungen der SE2

Die Frage, die sich zunächst stellt, ist natürlich: Was hat sich ändert? Die Lizenzen sind genauso teuer wie die der Standard Edition und die Funktionalität ist ebenfalls identisch. Die Antwort ist: die Nutzung der Ressourcen! Während die Standard Edition auf Servern mit maximal vier Sockets und beliebig vielen Cores und Threads eingesetzt werden durfte, gibt es bei der SE2 folgende Einschränkungen:

- Der Server, auf dem die SE2 betrieben wird, darf maximal zwei Sockets haben. Dabei ist es zunächst unerheblich, ob tatsächlich ein Prozessor gesteckt ist oder nicht.
- Jede Datenbank kann maximal sechzehn Threads nutzen (und hier zählt Hyperthreading mit). Nicht gezählt werden allerdings die Hintergrundprozesse.

Unternehmen, die die Standard Edition auf Servern mit mehr als zwei Sockets installiert haben, was beispielsweise in virtuellen Umgebungen oft der Fall ist, müssen entweder neue Server mit maximal zwei Sockets kaufen oder sie müssen auf die Enterprise Edition wechseln.

Die Beschränkung auf maximal sechzehn Threads ist in den Augen des Autors nicht ganz so tragisch. An dieser Stelle hat er Verständnis dafür, dass Oracle aufgrund der Leistungsfähigkeit heutiger Prozessoren eine Bremse eingebaut hat. Betreibt

man mehrere Datenbanken auf einem Server, kann jede davon unterschiedliche Threads verwenden. Insofern eignet sich die Standard Edition immer noch hervorragend für den Betrieb in virtuellen Umgebungen.

SE2 Real Application Clusters

Oracle Real Application Clusters (RAC) ist auch in der SE2 enthalten. Allerdings gelten hier folgende Einschränkungen:

- Jeder Server darf maximal einen bestückten Socket und insgesamt maximal zwei Sockets haben.
- Jede Datenbank-Instanz ist auf acht Threads beschränkt.

Alle Unternehmen, die die Standard Edition mit Oracle RAC betreiben, werden hier sicherlich kreativ sein müssen. Server mit einem Socket findet man so gut wie gar nicht mehr auf dem Markt und selbst bei Zwei-Socket-Servern muss man den zweiten Prozessor herausziehen, um Lizenzkonform zu bleiben. Unternehmen, die RAC-Datenbanken mit mehr als zwei Servern betreiben, bleibt nur der Weg, entweder die Anzahl der Server zu reduzieren oder auf die Enterprise Edition zu wechseln. Alternativ bietet sich natürlich immer auch eine Virtualisierungslösung mit Hard-Partitioning an, bei der die Datenbank auf einen Socket festgelegt wird.

SE2-Funktionalität

Generell finden sich in der SE2 die gleichen Funktionen wie in der Standard Edition wieder. Unternehmen, die bislang die

SE1 eingesetzt haben, kommen also über den oben genannten „Uplift“ des Supportbeitrags in den Genuss, die RAC-Option einsetzen zu dürfen. Die Limitierung der Threads erfolgt über das schon aus der Enterprise Edition bekannte „CPU Caging“; über den Parameter „cpu_count“ und die Verwendung des Resource Manager ist also die Anzahl gleichzeitiger Threads limitiert. Allerdings ist der Resource Manager auf diese Nutzung eingeschränkt, eine generelle Freigabe in der SE2 gibt es nicht.

Eine interessante Funktion möchte der Autor an dieser Stelle exemplarisch

herausgreifen: Flashback Data Archive. Diese ursprünglich kostenpflichtige Option mit dem Namen „Total Recall“, steht seit Version 11.2.0.4 in jeder Edition kostenfrei zur Verfügung. Damit lassen sich Änderungen an Datensätzen sehr einfach und anwendungsneutral protokollieren. Die Änderungen werden in Partitionen gespeichert (auch in SE, SE1 und SE2) und nach einer bestimmten Zeit (etwa zwei Jahre) automatisch gelöscht. Nach Ansicht des Autors wird dieses hervorragende Feature viel zu selten eingesetzt.



Johannes Ahrends
johannes.ahrends@carajandb.com

Termine

Mai

09.05.2016
Regionaltreffen NRW (Vorabendevent der DOAG 2016 Datenbank) | Düsseldorf
Martin Schmittler, Andreas Stephan
regio-nrw@doag.org

10.05.2016 – 11.05.2016
DOAG 2016 Datenbank | Düsseldorf
datenbank.doag.org

11.05.2016
Regionaltreffen München/Südbayern
Franz Hüll, Andreas Ströbel
regio-muenchen@doag.org

12.05.2016
Regionaltreffen Stuttgart
Jens-Uwe Petersen, Anja Zahn
regio-stuttgart@doag.org

12.05.2016
DOAG 2016 Spatial & Geodata Day | Düsseldorf
Kushtrim Krasniqi
sig-spatial@doag.org

13.05.2016
DOAG Webinar: Oracle GoldenGate 12c Tipps and Tricks
Johannes Ahrends, Christian Trieb
sig-database@doag.org

19.05.2016
Regionaltreffen Nürnberg/Franken
André Sept, Martin Klier
regio-franken@doag.org

Juni

30.05.2016 – 31.05.2016
Berliner Expertenseminar mit Christian Pfundtner zum Thema "Oracle 12c Flex RAC und ASM" | Berlin
Cornel Albert
expertenseminare@doag.org

31.05.2016
Regionaltreffen Hamburg/Nord
Jan-Peter Timmermann
regio-nord@doag.org

02.06.2016
DOAG 2016 Logistik & IT | Hannover
logistik.doag.org

07.06.2016
Regionaltreffen NRW (APEX Community) | Düsseldorf
Martin Schmittler, Andreas Stephan
regio-nrw@doag.org

08.06.2016 – 09.06.2016
DOAG 2016 BI | Bonn
bi.doag.org

09.06.2016
Regionaltreffen Karlsruhe
Reiner Bünger
regio-karlsruhe@doag.org

10.06.2016
DOAG Webinar: Compression
Johannes Ahrends, Christian Trieb
sig-database@doag.org

13.06.2016
Regionaltreffen Osnabrück/Bielefeld/Münster
Andreas Kother, Klaus Günther
regio-osnabrueck@doag.org

14.06.2016
Regionaltreffen Rhein-Neckar | Mannheim
Frank Stöcker
regio-rhein-neckar@doag.org

16.06.2016
Regionaltreffen Bremen
Ralf Kölling
regio-bremen@doag.org

16.06.2016 – 17.06.2016
DOAG Vorstandssitzung | Berlin
DOAG Geschäftsstelle
office@doag.org

16.06.2016
Regionaltreffen Nürnberg/Franken
André Sept, Martin Klier
regio-franken@doag.org

20.06.2016
Regionaltreffen Jena/Thüringen
Jörg Hildebrandt
regio-thueringen@doag.org

21.06.2016
AOUG Anwenderkonferenz 2016
office@aoug.at

21.06.2016
Regionaltreffen Dresden/Sachsen
Helmut Marten
regio-sachsen@doag.org

23.06.2016
SOUG-DAY
sekretariat@soug.ch

Wir begrüßen unsere neuen Mitglieder

Persönliche Mitglieder

- Christian Kunzmann
- Randolf Geist
- Fabian Greis
- Jürgen Gizzi
- Michael Specker
- Mathias Mildnerberger
- Roland Hörmann
- Günther Stürner
- Markus Krisch
- Maik Wagner
- Kai Hof
- Heiko Dervedde

Firmenmitglieder DOAG

- BG-Phoenix GmbH, Klaus Amann
- @-yet GmbH, Wolfgang Straßer
- Berlin Hyp AG, Till Kasperbauer
- Georg Thieme Verlag KG, Erik Ebel
- Yellowstar Solutions GmbH, Osman Akdemir
- DERMALOG Identification Systems GmbH, Peter Elmers

Neumitglieder SOUG

- VP Bank AG, Thomas Moll
- Acceleris AG, Rainer Schaub
- SIX Group SA, César Ferreira
- Helsana-Gruppe, Marco Hürlimann
- Axedo Solutions AG, Angelika Braun
- Bison Schweiz AG, Andreas Roser
- LC Systems Engineering AG, Thilo Ketzscher
- LC Systems Engineering AG, Dietmar Schubnell
- Swiss Aviation Software Ltd., Matthias Link

Impressum

Red Stack Magazin wird gemeinsam herausgegeben von den Oracle-Anwendergruppen DOAG Deutsche ORACLE-Anwendergruppe e.V. (Deutschland, Tempelhofer Weg 64, 12347 Berlin, www.doag.org), AOUG Austrian Oracle User Group (Österreich, Lassallestraße 7a, 1020 Wien, www.aoug.at) und SOUG Swiss Oracle User Group (Schweiz, Dornacherstraße 192, 4053 Basel, www.soug.ch).

Red Stack Magazin ist das User-Magazin rund um die Produkte der Oracle Corp., USA, im Raum Deutschland, Österreich und Schweiz. Es ist unabhängig von Oracle und vertritt weder direkt noch indirekt deren wirtschaftliche Interessen. Vielmehr vertritt es die Interessen der Anwender an den Themen rund um die Oracle-Produkte, fördert den Wissensaustausch zwischen den Lesern und informiert über neue Produkte und Technologien.

Red Stack Magazin wird verlegt von der DOAG Dienstleistungen GmbH, Tempelhofer Weg 64, 12347 Berlin, Deutschland, gesetzlich vertreten durch den Geschäftsführer Fried Saacke, deren Unternehmensgegenstand Vereinsmanagement, Veranstaltungsorganisation und Publishing ist.

Die DOAG Deutsche ORACLE-Anwendergruppe e.V. hält 100 Prozent der Stammeinlage der DOAG Dienstleistungen GmbH. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. wird gesetzlich durch den Vorstand vertreten; Vorsitzender: Dr. Dietmar Neugebauer. Die DOAG Deutsche ORACLE-Anwendergruppe e.V. informiert kompetent über alle Oracle-Themen, setzt sich für die Interessen der Mitglieder ein und führen einen konstruktiv-kritischen Dialog mit Oracle.

Redaktion:

Sitz: DOAG Dienstleistungen GmbH
(Anschrift s.o.)
Chefredakteur (ViSdP): Wolfgang Taschner
Kontakt: redaktion@doag.org
Weitere Redakteure (in alphabetischer Reihenfolge): Julia Bartzik, Gaetano Bisaz, Mylène Diacquenod, Marius Fiedler, Marina Fischer, Klaus-Michael Hatzinger, Dr. Dietmar Neugebauer, Jan Peterskovsky, Fried Saacke

Titel, Gestaltung und Satz:

Alexander Kermas, DOAG Dienstleistungen GmbH (Anschrift s.o.)

Fotonachweis:

Titel: lightwise/123rf.com
Foto S. 16: © iqoncept/123rf.com
Foto S. 37: © studiostoks/123rf.com
Foto S. 43: © [jesadaphorn chaiinkeaw/123rf.com](http://jesadaphorn.chaiinkeaw/123rf.com)
Foto S. 47: © fernando.gregory/123rf.com
Foto S. 50: © gennady.kireev/123rf.com
Foto S. 79: © [grufnar/fotolia.com](http://grufnar.fotolia.com)

Anzeigen:

Simone Fischer, DOAG Dienstleistungen GmbH (verantwortlich, Anschrift s.o.)
Kontakt: anzeigen@doag.org
Mediadaten und Preise unter: www.doag.org/go/mediadaten

Druck:

adame Advertising and Media GmbH,
www.adame.de

Alle Rechte vorbehalten. Jegliche Vervielfältigung oder Weiterverbreitung in jedem Medium als Ganzes oder in Teilen bedarf der schriftlichen Zustimmung des Verlags. Die Informationen und Angaben in dieser Publikation wurden nach bestem Wissen und Gewissen recherchiert. Die Nutzung dieser Informationen und Angaben geschieht allein auf eigene Verantwortung. Eine Haftung für die Richtigkeit der Informationen und Angaben, insbesondere für die Anwendbarkeit im Einzelfall, wird nicht übernommen. Meinungen stellen die Ansichten der jeweiligen Autoren dar und geben nicht notwendigerweise die Ansicht der Herausgeber wieder.

Inserentenverzeichnis

CeMAT www.cemat.de	U 3	E-3 Magazin www.e-3.de	31	Softbase A/S www.softbase.com	S. 9
DBConcepts www.dbconcepts.at	S. 41	MuniQsoft GmbH www.muniqsoft.de	S. 3	Trivadis GmbH www.trivadis.com	U 4
dbi services ag www.dbi-services.com	S. 23	ORACLE www.oracle.com	U 2		
DOAG e.V. www.doag.org	S. 54, 58, 63	PROMATIS software GmbH www.promatis.de	29		

Wir haben die smarten Lösungen

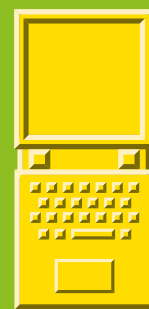
der Zukunft auf Lager

Erleben Sie die Weltneuheiten und Vordenker
der Intralogistikbranche zuerst auf der

CeMAT 2016

31. Mai – 3. Juni 2016 • Hannover • Germany
cemat.de

Leitthema:
smart
supply chain
solutions



Deutsche Messe

World leading trade fair for intralogistics &
supply chain management

CeMAT

APEX mit Trivadis

Web-Datenbank-Anwendungen
im Express-Tempo.



APEX GUIDELINES 2.1 VON TRIVADIS

Jetzt kostenlos als Download: www.trivadis.com/apex-guidelines

■ Setzen Sie APEX auf das richtige Gleis: mit Trivadis. Die Oracle- und APEX-Experten von Trivadis unterstützen Sie dabei, Ihre gewünschten Anwendungen effizient zu entwickeln, bestehende Applikationen zu optimieren, den Betrieb zu modernisieren und Ihre Kosten zu senken. Wir zeigen Ihnen, wie Sie mit Oracle APEX schneller und flexibler Ihre datenbankbasierten Web-Anwendungen erstellen – bis hin zu leistungsfähigen Frameworks für Mobile Enterprise oder Responsive Design. Sprechen Sie mit uns über Ihre APEX-Vorhaben. www.trivadis.com | info@trivadis.com

BASEL ■ BERN ■ BRUGG ■ DÜSSELDORF ■ FRANKFURT A.M. ■ FREIBURG I.B.R. ■ GENF
HAMBURG ■ KOPENHAGEN ■ LAUSANNE ■ MÜNCHEN ■ STUTTGART ■ WIEN ■ ZÜRICH

trivadis
makes IT easier. ■ ■ ■