

# Oracle Application Express 5.0

Marc Sewtz, Oracle America, Inc.

Mit Oracle Application Express 5.0 hat Oracle im April dieses Jahres das bei Weitem umfangreichste Release in der über zehnjährigen Geschichte von Application Express (Apex) vorgestellt.

Oracle Application Express 5.0 ermöglicht es, moderne, Datenbank-basierte Web-Anwendungen zu entwickeln und auszuliefern. Durch die Verwendung von „Responsive Design“ sind diese Anwendungen sowohl auf Desktops und Laptops als auch auf mobilen Endgeräten wie Smartphones und Tablets einsetzbar. Oracle Application Express ist als Feature ohne zusätzliche Lizenzkosten in jeder Edition der Oracle-Datenbank enthalten und ist zudem in die Oracle Database Cloud integriert. Dieser Artikel zeigt einen Überblick über wichtige Meilensteine in der Entwicklung von Oracle Application Express sowie über die Apex Community. Zudem werden einige der wichtigen Neuerungen in Apex 5.0 behandelt und aufgezeigt, wie Apex zusammen mit der Oracle-Datenbank die Grundlage dafür bildet, dass Anwendungen, die mit Apex entwickelt wurden, auch in Zukunft auf dem neuesten Stand der Technik bleiben können.

Oracle Application Express ist seit dem Jahr 2004 ein fester Bestandteil der Oracle-Datenbank. Zunächst unter dem

Namen „Oracle HTML DB“ veröffentlicht, wurde das Produkt im Jahr 2006 in „Oracle Application Express“ umbenannt und mit der Version 2.1 zum Front-End für die damals neu vorgestellte Oracle Database Express Edition (Oracle XE) erweitert. Mit der Version 3 kamen Interactive Reports, PDF Printing, Flash Charts sowie Migration von MS Access und Forms-Anwendungen hinzu. Version 4 brachte deklaratives JavaScript mit Dynamic Actions, Erweiterbarkeit durch Plug-ins, mobile Anwendungsentwicklung sowie eine umfassende Kollektion von mitgelieferten Geschäftsanwendungen.

Mit der in diesem Jahr veröffentlichten Version 5.0 wurde Apex in einem gut zweieinhalbjährigen Entwicklungszyklus grundlegend modernisiert. Sowohl die Apex-Entwicklungsoberfläche als auch das neue Universal Theme für Endanwendungen wurden unter Verwendung modernster Web-Technologien wie HTML5, CSS3 und Responsive Design völlig neu konzipiert. Mit dem neu eingeführten Page Designer steht zudem ein außergewöhnlich produktives Entwicklungswerk-

zeug bereit, das den Komfort traditioneller, Desktop-basierter Entwicklungsumgebungen mit der Flexibilität moderner, Web-basierter Anwendungen vereint.

## Anwendungsbereiche

Apex erlaubt die Entwicklung verschiedenster Web-Anwendungen, die sowohl auf Desktops als auch auf mobilen Endgeräten eingesetzt werden können. Das Spektrum der Anwendungsmöglichkeiten erstreckt sich dabei von einfachen, Web-basierten Spreadsheet-Alternativen bis hin zu komplexen Enterprise-Lösungen. So wurde etwa der Oracle Online Store (*siehe „<http://shop.oracle.com>“*) mit Apex entwickelt sowie verschiedene Komponenten der Oracle Cloud.

Apex richtet sich sowohl an professionelle Softwareentwickler als auch an Business User, die sogenannten „Citizen Developer“. Gartner definiert diese als Benutzer, die Anwendungen in ihrem Geschäftsbereich für sich und für andere Benutzer entwickeln (*siehe „<http://www.gartner.com/it-glossary/citizen-developer>“*). Während in der Vergan-



Abbildung 1: Anwendungsbereiche für Oracle Application Express



Abbildung 2: Drei-Tier-Architektur

genheit dazu häufig Desktop-Lösungen wie Microsoft Excel und Access zum Einsatz kamen, werden heute zunehmend Browserbasierte Plattformen eingesetzt, die von der EDV-Abteilung entweder On-Premise oder über Cloud-basierte Services bereitgestellt sind (siehe Abbildung 1).

Die Entwicklung von Apex-Anwendungen erfolgt deklarativ, der Entwickler wählt also einfach per Wizard Tabellen und Spalten aus, auf denen beispielsweise ein Bericht, ein Formular oder ein Chart erstellt werden soll. Apex generiert dann alle erforderlichen Komponenten für die

gewünschte Anwendung. Viele Anwendungsbereiche erfordern somit keinerlei fortgeschrittene Entwicklungskennnisse.

Für komplexere Business-Logik oder Datenbank-Abfragen kommen SQL- und PL/SQL-Kennnisse zum Einsatz, also Sprachen, mit denen Oracle-Entwickler zu-

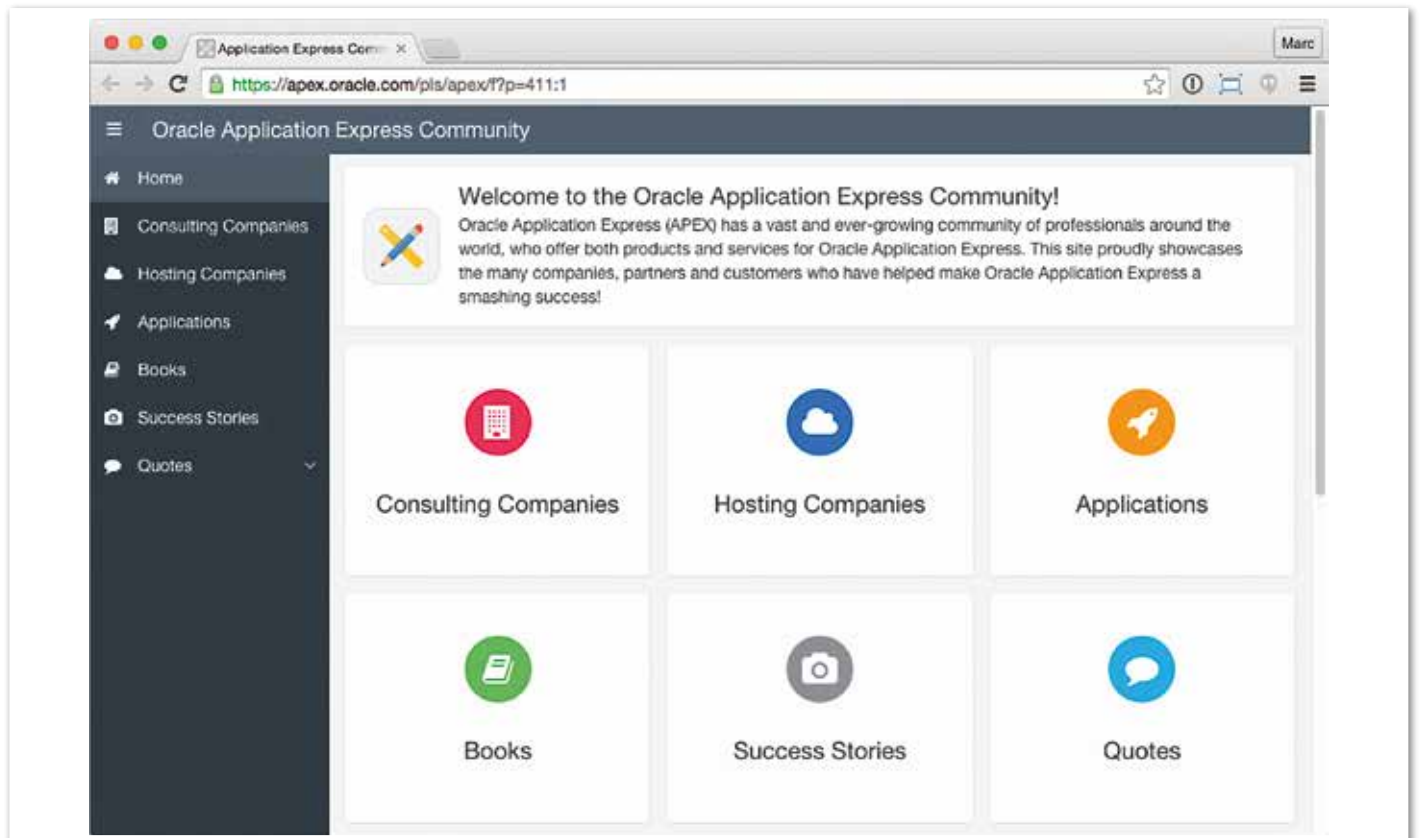


Abbildung 3: Apex Community Page

meist vertraut sind. Für die Anwendungsoberfläche stellt Apex ein leistungsstarkes HTML5-Framework bereit, das zahlreiche populäre HTML-, CSS- und JavaScript-Bibliotheken integriert.

So kommen unter anderem jQuery, jQuery Mobile, Less, Sass, Font Awesome und weitere Frameworks zum Einsatz. Apex ist zudem offen für die Integration anderer Third-Party-Bibliotheken sowie die Einbindung von eigenem JavaScript- und HTML-Code.

### Die Architektur

Apex bietet signifikante Vorteile gegenüber anderen Entwicklungs-Frameworks – insbesondere dann, wenn es zur Analyse und Verarbeitung von Daten in der Oracle-Datenbank eingesetzt wird. Da Apex direkt in die Datenbank integriert ist, finden alle Daten-intensiven Prozesse in unmittelbarer Nähe zu diesen Daten statt, ohne dass kostspielige und wiederholte Datenzugriffe zwischen Middle-Tier und Datenbank erforderlich sind. Ein Seitenzugriff mit Apex erfordert nur einen einzigen HTTP-Request vom Browser, der über den Middle-Tier-Listener an die Datenbank weitergereicht wird, sowie eine einzige HTTP-Response, die via Listener an den Browser zurückgeliefert wird (siehe Abbildung 2).

Apex bietet verschiedene Konfigurationsoptionen. Neben dem ursprünglich eingesetzten „mod\_plsql“ und Apache-Setup oder dem in die Datenbank eingebauten „Embedded PL/SQL Gateway“ kommt heute vor allem der von Oracle empfohlene Oracle REST Data Service (ORDS) zum Einsatz. Dieser kann sowohl mit WebLogic und GlassFish als auch mit Tomcat als Webserver betrieben werden. ORDS bietet gegenüber älteren Konfigurationen zusätzlich die Einbindung von REST-based Webservices sowie eine einfachere PDF-Printing-Konfiguration.

Die Definition einer Apex-Anwendung wird in Metadaten-Tabellen im Apex-Datenbank-Schema abgespeichert. Diese Tabellen sind vergleichbar mit dem Data Dictionary, mit dem die Oracle-Datenbank die Definition von Datenbank-Objekten speichert. Während das Data Dictionary Informationen über Tabellen, Spalten, Trigger, PL/SQL Packages etc. enthält, finden sich in den Apex-Metadaten-Tabellen Informationen über Anwendungen,

Seiten, Regionen, Formular-Felder, Templates, etc. Diese Metadaten werden zur Laufzeit von der Apex Engine ausgelesen, die mithilfe dieser Daten dynamisch den HTML-Code für die abgerufene Seite generiert.

Diese Laufzeit-Generierung von HTML-Seiten, basierend auf Metadaten, macht den gravierenden Unterschied von Apex zu anderen Web-Entwicklungs-Frameworks aus. Während diese Frameworks zwar häufig auch deklarative Werkzeuge zur Anwendungserstellung bieten, ist das Endprodukt jedoch zumeist eine Kollektion von generiertem Programmcode, der kompiliert und installiert werden muss.

Mit Apex wird die Definition der Anwendung unabhängig von der Laufzeit-Seitengenerierung gespeichert; somit ist es jederzeit möglich, Änderungen in der Anwendungsdefinition vorzunehmen, ohne die Anwendung neu generieren zu müssen. Zudem erlaubt dieser Ansatz, alle verwendeten HTML-Templates komplett mit neuen Templates auszutauschen. Das Benutzeroberflächen-Thema der Anwendung kann also ausgewechselt werden, ohne dass die Anwendungsdefinition davon betroffen ist.

Der Metadaten-basierte Ansatz gewährleistet zudem, dass Apex-Anwendungen für zukünftige Technologie-Trends gewappnet sind und neue Technologien sehr einfach aufgegriffen werden können, ohne dass die Anwendung neu entwickelt werden muss. Als Apex beziehungsweise HTML DB im Jahr 2004 auf den Markt kam, hatte noch niemand von HTML5 oder AJAX gehört, JavaScript war eine eher belächelte Scripting-Sprache, die bestenfalls zum Anzeigen und Verstecken von Komponenten einer Seite eingesetzt wurde, Flash war noch „cool“ und Klapp-Handys mit winzigen Bildschirmen waren das, was man gemeinhin unter „Mobile“ verstand. Während viele dieser Technologien mittlerweile überholt sind, kann eine Anwendung, die im Jahre 2004 mit HTML DB entwickelt wurde, auch heute noch problemlos auf Application Express 5.0 installiert werden und dort entweder unverändert weiterverwendet oder in wenigen Schritten auf eine moderne, dynamische und responsive HTML5-Oberfläche modernisiert werden.

### Die Apex Community

Apex erfreut sich einer stetig wachsenden und sehr aktiven Entwickler-Community.



Business Intelligence  
Managed Services  
Custom Development  
E-Business Suite

Wir sind dabei

2015  
**DOAG**  
Konferenz + Ausstellung

**Apps Associates GmbH**

Flughafenring 11  
D-44319 Dortmund

Tel.: +49 231 22 22 79-0

[www.appsassociates.com](http://www.appsassociates.com)

**ORACLE** Platinum  
Partner

Oracle schätzt die Zahl der Apex-Entwickler weltweit auf mehr als 400.000. Viele Entwickler engagieren sich im Apex-Diskussionsforum auf OTN und auf Social Media Sites wie Twitter und LinkedIn. Sie tauschen ihr Wissen zudem mit anderen Entwicklern über Blogs aus sowie als Autoren von mehr als dreißig Büchern zum Thema „Apex“. Vor gut einem Jahr in New York ins Leben gerufen, hat sich zudem eine mehr als 1.000 Entwickler umfassende Apex Community zusammengefunden, die regelmäßig rund um die Welt unabhängige Meetups zum Gedanken- und Ideenaustausch veranstaltet (siehe Abbildung 3).

Als zentrale Anlaufstelle für alle Apex-interessierten Entwickler hat Oracle kürzlich die Apex Community Website gestartet (siehe „<https://apex.oracle.com/community/>“), auf der Informationen zu mehr als 120 Consulting und Hosting Companies, Links zu öffentlichen Apex-Anwendungen sowie zahlreiche Success-Stories abgelegt sind. Zudem finden Entwickler dort Links zu zahlreichen Büchern über Apex sowie Erfahrungsberichte von anderen Kunden.

### Moderne HTML5-Benutzeroberfläche

Apex stellt Entwicklern eine Reihe weitgehend eigenständiger Komponenten zur Verfügung. Der Application Builder ist die eigentliche Entwicklungsumgebung, die es ermöglicht, schnell und einfach Anwendungen auf Objekten in der Datenbank zu definieren. Der SQL-Workshop bietet ein Browser-basiertes Interface zur Oracle-Datenbank, über die SQL-Abfragen gestartet sowie Datenbank-Objekte inspiziert und angelegt werden können. Es können PL/SQL-Code editiert und kompiliert werden sowie SQL-Skripte ausgeführt werden.

Die „Packaged Apps“, also die mitgelieferten Geschäftsanwendungen, lassen sich eigenständig verwalten und bereitstellen, ohne dass Zugriff auf andere Apex-Komponenten freigeschaltet werden müsste. Mit Team Development steht Entwicklern und Anwendern ein integriertes System zum Projekt-Management von Apex-Entwicklungsprojekten zur Verfügung. Insgesamt besteht Apex aus gut 2.000 individuellen HTML-Seiten, davon entfallen allein 1.000 auf den Application Builder.

Als Teil der Modernisierung der Apex-Benutzeroberfläche wurde jede einzelne Seite individuell inspiziert und aktualisiert. Die eingesetzten HTML-Templates sind komplett überarbeitet. Modernes, Grid-basiertes HTML-Markup hat älteren Code ersetzt, in dem noch HTML-Tabellen zum Einsatz kamen. HTML5-Charts ersetzen die Flash-Charts. Sämtliche Wizards wurden optimiert und in modale Dialoge umgewandelt. Alle verwendeten Grafiken und Icons hat man neu entworfen und als Vektorgraphiken realisiert, diese Grafiken sind jetzt auch auf modernen High-DPI/Retina-Bildschirmen gestochen scharf angezeigt.

### Der Page Designer

Oracle-Forms-Anwender sowie Entwickler, die mit anderen Desktop-basierten Entwicklungsumgebungen vertraut sind, sind es gewohnt, Komponenten wie Formularfelder per Drag & Drop auf einer Seite zu platzieren. Auch Apex hatte in früheren Versionen einen rudimentären „Drag & Drop“-Formular-Designer angeboten, mit dem Entwickler Formularfelder in einer Region platzieren konnten. Dieser Formular-Designer basierte jedoch auf HTML-Tabellen und mit dem Wechsel zu einem Grid-basierten Responsive Layout war es notwendig, diesen Formular-Designer zu aktualisieren. Endresultat ist der neue Page Designer.

Der Page Designer geht jedoch weit über das hinaus, was mit dem Formular-Designer möglich war. So kann statt einer einzelnen Region jetzt die gesamte Seite per Drag & Drop gestaltet werden. Der Page Designer ist nicht mehr nur auf Formular-Elemente beschränkt, sondern erlaubt das Hinzufügen und Editieren aller Komponenten einer Seite, einschließlich vollständiger Regionen, Dynamic Actions, Seiten-Prozesse und Berechnungen. Über das eigentliche Layout hinaus ist es möglich, alle Attribute einer Komponente in dieser integrierten Oberfläche zu bearbeiten. Oracle hat mit dem Page Designer in Apex 5.0 eine Entwicklungsumgebung geschaffen, die traditionellen Desktop-IDEs in Sachen Komfort um nichts nachsteht, die jedoch komplett für den Betrieb im Web-Browser konzipiert wurde und somit die volle Flexibilität moderner, Web-basierter Anwendungen bietet.

Die Anforderungen an den Page Designer haben es nötig gemacht, verschiedene Hilfskomponenten, Widgets und Bibliotheken zu modernisieren. So wurde beispielsweise ein vollständig neues Menu-Widget eingeführt, das neben dem Page Designer auch für die neuen, List-basierten Anwendungsmenus zum Einsatz kommt. Das sogenannte „Splitter Widget“, das im Page Designer den Komponenten-Tree vom Layout- und Property-Editor trennt, wurde auch dazu eingesetzt, um verschiedene Bereiche des SQL-Workshops zu modernisieren. Das Icon „List Widget“, das für die Komponenten-Gallery im Page Designer verwendet wird, kommt zusätzlich in den modernisierten Region und Item Wizards zum Einsatz.

### Modale Dialoge

Pop-up-Windows wurden in der Vergangenheit gern dazu eingesetzt, Dialoge zu simulieren, wie sie aus Desktop-Anwendungen bekannt sind. Durch den Missbrauch von Pop-up-Windows für unerwünschte Werbung und die daraus resultierenden zahlreichen Sicherheitsrisiken sowie die insgesamt schlechte Bedienbarkeit werden heutzutage eher modale Dialoge eingesetzt, also Dialoge, die die aktuelle Seite überlagern und dabei die aufrufende Seite dunkler darstellen und deaktivieren.

Während es mit älteren Versionen von Apex bereits möglich war, einzelne Regionen als modalen Dialog darzustellen, wurden in Apex 5.0 echte modale Dialoge eingeführt, eine komplette Apex-Seite kann also jetzt als modaler Dialog angezeigt werden, wobei alle Funktionen einer Apex-Seite zur Verfügung stehen. Es sind unabhängig von der aufrufenden Seite Berechnungen möglich sowie Prozesse, etwa zum Aktualisieren von Daten.

### Universal Theme

Das Konzept eines Benutzer-Interface-Theme wurde bereits in HTML DB 1.6 eingeführt. Ein Theme ist eine Kollektion aller Templates, die für die Entwicklung einer Geschäftsanwendung gebraucht werden könnten; es stehen zahlreiche Seiten-Templates, Regionen-Templates, Formfeld-Templates, Schaltflächen-Templates etc. zur Verfügung. Je nach Anwendungszweck wählt man passende Templates aus. Soll nun die Oberfläche einer

Anwendung geändert und ein neues Look & Feel eingeführt werden, bietet Apex die Möglichkeit, das Theme auszuwechseln; das aktualisiert automatisch alle referenzierten Templates.

Um zu gewährleisten, dass die Anwendung hinterher noch vernünftig aussieht, werden Template-Typen und -Klassen dazu verwendet, die jeweiligen Templates im alten Theme passenden Templates im neuen Theme zuzuordnen. Dieses Konzept hat gut funktioniert, solange unterschiedliches Anwendungsdesign primär über HTML realisiert wurde. Alle Themes und Templates unterschieden sich also signifikant voneinander.

Heutzutage vermeidet man es jedoch, Layout und Design direkt im HTML-Code zu definieren. Stattdessen wird das HTML-Markup darauf beschränkt, die Struktur einer Seite anzulegen, Container für Regionen und Formular-Felder bereitzustellen und die semantische Bedeutung der verschiedenen Komponenten zu definieren. Das eigentliche Layout und Design, die konkrete Positionierung von Regionen und anderen Elementen sowie das responsive Verhalten werden mit CSS definiert. Das hat zur Folge, dass der HTML-Code stark reduziert ist, die Seiten somit schneller laden und weniger Bandbreite erfordern, zumal referenzierte JavaScript-, CSS- und Image-Dateien, wenn sie einmal geladen sind, über den Cache abgerufen werden. Für Apex bedeutet dies konkret, dass sich der HTML-Code von einem Theme zum anderen nur noch minimal unterscheidet und auch die Zahl der erforderlichen Templates stark reduziert werden kann.

Natürlich gibt es nach wie vor den Bedarf, Komponenten unterschiedlich darzustellen. So sollen zum Beispiel Felder, die ausgefüllt werden müssen, anders angezeigt werden als optionale Felder. Eine Region, die zur Anzeige von Fehlermeldungen oder Warnungen verwendet wird, sollte anders aussehen als etwa eine Region, die als Container für einen Bericht verwendet wird, und ein Button zum Speichern anders als ein Button zum Abbrechen. Diese Unterschiede werden über CSS-Klassen definiert. Allerdings wäre es sehr ineffektiv, dutzende Templates anzulegen, die weitgehend das gleiche HTML Markup verwenden und sich nur in den CSS-Klassen

unterscheiden. Daher wurden in Apex 5.0 die sogenannten „Template-Optionen“ eingeführt.

Template-Optionen erlauben es, einzelne Templates für eine Vielzahl verschiedener Zwecke einzusetzen, indem für die jeweilige Komponente einfach unterschiedliche Template-Optionen deklarativ ausgewählt werden, die dann intern in die jeweiligen CSS-Modifier-Klassen übersetzt werden. Template-Optionen abstrahieren somit den kryptischen CSS-Klassen-Namen und zeigen dem Benutzer eine verständliche Beschreibung der Klasse. Template-Optionen werden auf der Komponenten-Ebene zusammen mit dem zugehörigen Template ausgewählt, was es möglich macht, ein einziges Template in vielen verschiedenen Iterationen einzusetzen.

Neben den unterschiedlichen Template-Typen haben ältere Themes zudem das Gesamt-Layout einer Anwendung definiert, also Farbschema, Schriftarten, Darstellung von Navigationselementen, gegebenenfalls Drop-Shadows sowie Gradients für Button und Regionen. Diese Aufgabe fällt in modernen Themes nun primär dem CSS-Stylesheet zu. Das Austauschen eines kompletten Theme mit allen zugehörigen Templates erfüllt somit nicht mehr den eigentlichen Zweck. Stattdessen wird lediglich ein anderes oder ergänzendes CSS-Stylesheet eingesetzt. Um dies deklarativ möglich zu machen, wurden mit Apex 5.0 sogenannte „Theme Styles“ eingeführt, mit denen verschiedene CSS-Stylesheets eingebunden werden können.

Apex 5.0 hat drei vordefinierte Theme Styles, den blauen Default-Style, eine graue Variante sowie das Vista-Theme, das an Oracle Standard Alta Look angelehnt ist. Weitere Theme Styles lassen sich sehr einfach mit dem Theme-Roller erstellen, mit dem zur Laufzeit neue Farbschemata ausprobiert werden können, etwa zum Nachbilden einer gewünschten Corporate UI. Der Theme-Roller erlaubt, die Darstellung von Komponenten weiter zu verändern, etwa mit Gradients, Drop-Shadows, Rounded-Corners etc. Wenn das gewünschte Design erreicht ist, lässt sich dieses als Theme Style abspeichern und auch für andere Anwendungen in der Organisation bereitstellen.

## Fazit

Seit mehr als zehn Jahren bietet Oracle Application Express den Kunden eine solide und verlässliche Plattform für die Entwicklung Web-basierter Anwendungen auf Daten in der Oracle-Datenbank. Das an Oracle Forms angelegte deklarative Konzept und die Verwendung bewährter Sprachen wie SQL und PL/SQL erleichtern den Einstieg insbesondere für bestehende Oracle-Anwender. Das Metadaten-basierte Konzept macht Apex-Anwendungen zukunftsicher und erlaubt es, kontinuierlich neue Technologien einzubinden.

Mit Apex 5.0 stellt Oracle eine moderne HTML5- und CSS3-Oberfläche bereit und hat zahlreiche aktuelle JavaScript-Bibliotheken integriert. Durch Responsive Design und Integration von jQuery Mobile können Apex-Anwendungen mit den verschiedensten Endgeräten verwendet werden – egal ob es sich dabei um Desktop, Laptop, Tablet oder Smartphone handelt.

Mit dem Page Designer und dem Universal Theme ist Oracle Application Express anderen Web-basierten Entwicklungsframeworks weit voraus und durch die enge Einbindung in die Oracle Database Cloud haben Kunden die freie Auswahl, ob sie ihre Anwendung On-Premise oder in der Cloud betreiben wollen.



Marc Sewtz  
marc.sewtz@oracle.com