Mit CD

On-Demand-Entwicklung + + + Firebird + + + Qt-Development

S&S

entwickler

entwickler magazin

Software, Systems & Development

5.08 | September / Oktober | www.entwickler-magazin.de

Remote Support

Fernwartungslösungen von einfach bis mächtig

Web Services mit Attachments

Der Standard MTOM im Fokus

OpenCms

Software as a Service

Die Zukunft der Businessanwendungen

Sonderdruck der Firma arlanis

Maps on Rails

Geocoding mit dem Ruby-Framework

Einfacher und schneller

XQuery Update in Aktion mit IBM DB2

Delphi Win32

Speicher managen mit FastMM

CD-Inhalt

Testversionen

- Advantage Database Server 9.0

Software

Software
Ruby 1.8.6 und Ruby Gems 1.2.0
OpenCMS 7.0.5
Mondrian 3.0
FastMM 4.84

Alle Infos auf Seite 5

Datenträger enthält nur Lehr- oder Infoprogramme

On-Demand-Anwendungsentwicklung mit Salesforce.com

Platform as a Service

„Software as a Service" und „Platform as a Service" sind vergleichsweise neue IT-Konzepte. Dennoch zeichnet sich immer deutlicher ab, dass diese neuen Technologien aus der IT-Welt bald nicht mehr wegzudenken sein werden. Tatsächlich ist die Möglichkeit, Software sowie Entwicklungsplattformen einfach nur zu mieten, und damit keinen Aufwand mehr zu haben mit Installation, Wartung und Backups, faszinierend. In diesem Artikel werden die Eigenschaften solcher Systeme beleuchtet und Beispiele für die Softwareentwicklung „on-demand" gegeben.

von Christian Metzger und Andreas Holubek

Die Idee hinter Software as a Service (SaaS) ist es, Anwendungen nicht mehr zu installieren, sondern – solange sie benötigt werden – zu mieten und via Internetbrowser zu nutzen. Eigentlich geht die Idee jedoch sehr viel weiter: tatsächlich wird nicht nur die Software gemietet, sondern auch ihre Wartung, die Datenbanken, Backups, die erforderlichen Sicherheitssysteme sowie natürlich die ständige Verfügbarkeit.

Der konsequente nächste Schritt ist Platform as a Service (PaaS). Hierbei werden nicht On-Demand-Anwendungen, sondern die diesen zugrundeliegende Entwicklungsplattform über das Internet genutzt, um darauf wiederum neue On-Demand-Anwendungen zu programmieren. Der Vorteil für Entwickler: sie benötigen keine eigene IT-Infrastruktur für das Erstellen ihrer Applikationen mehr – beispielsweise Entwicklungswerkzeuge, Application Server, Testumgebungen, Sicherheitslösungen sowie Backup und Desaster Recovery. Die technischen Voraussetzungen für das Umsetzen der eigenen Idee in Softwarecode reduzieren sich auf einen Internetanschluss und einen Rechner.

Zwei populäre Beispiele für solche On-Demand-Plattformen sind Force.com [1] von salesforce.com [2] und die Google AppEngine [3]. Sie erlauben es, Anwendungen über den Browser zu entwickeln, anzupassen und über das Internet zu vermarkten. Betrachtet man die Gesamtheit der bereits bestehenden „As-a-Service"-Konzepte, lässt sich schon heute eine IT-Welt vorstellen, in der tatsächlich fast alles als Dienstleistung angeboten und „in the cloud", also außerhalb des eigenen Unternehmens, genutzt wird. Die Gesamtheit dieser Dienstleistungen wird entsprechend als *Cloud Computing* bezeichnet. Dabei umfassen die verschiedenen Angebote wie Storage as a Service, User Interface as a Service, Logic as a Service, Integration as a Service, Database as a Service und eben auch Development as a Service entweder Einzelservices oder ganze Bündel von Dienstleistungen.

In den folgenden Abschnitten wird vorgestellt, wie Entwickler eine On-Demand Platform nutzen können. Am Beispiel von Force.com von salesforce.com wird gezeigt, wie bestehende On-Demand-Anwendungen mittels Customization an die eigenen Bedürfnisse angepasst sowie neue Applikationen auf der Plattform programmiert werden können.

Customization – Die Plattform an eigene Wünsche anpassen

Jede On-Demand-Plattform kommt mit einem Satz an vorgefertigten Komponenten, Workflows und einigem mehr. Je umfassender diese „Out-of-the-Box"-Funktionen sind, desto größer in der Regel der praktische Wert einer solchen Entwicklungsumgebung. Force.com stellt beispielsweise folgende Möglichkeiten zum Customizing zur Verfügung [4]:

- Erstellung von *Custom Objects*: Erstellung von Datenbanktabellen und die Verlinkung untereinander mit beispielsweise Master-Detail Relationship.
- Erstellung von Feldern
- Erstellung von Seitenlayouts: Es ist möglich, benutzerdefinierte Layouts für die Eingabe- und Anzeigeseiten zu erstellen. In einem nächsten Schritt können die Layouts Berechtigungs- und Zugangsprofilen zugeordnet werden. Das Ziel ist es, verschiedenen Benutzern individuelle Datensichten zur Verfügung zu stellen. Die komplette Gestaltungsfreiheit von Oberflächen erhält man durch den Einsatz von Visual Force, einem ergänzenden Tool zum völlig freien GUI-Design.
- Validierungsregeln für Feldeingaben: Validierungsregeln bestimmen, wie die Feldinhalte nach der Eingabe auf Richtigkeit geprüft werden. Entwickler können beispielsweise Felder, je nach dem Wert eines anderen Feldes, als bedingt erforderlich definieren. Oder sie können sicherstellen, dass Zahlen in einem angegebenen Bereich liegen, beispielsweise, dass der Rabatt weniger als 30 % beträgt.
- Workflows: Abläufe, Geschäftsregeln und Genehmigungsprozesse lassen sich frei gestalten und abhängig von Erstellung und Feldinhalten ausführen. Benachrichtigungsprozesse sind ebenso realisierbar.
- Berichte und Dashboards zu allen Feldern und Objekten: Berichte können sich auf alle Felder, die im System eingestellt sind, beziehen. Sie können in tabellarischer Form oder grafisch als Dashboards ausgegeben werden. Zudem lassen sie sich mit vorab definierten Kennzahlen abgleichen.
- Userverwaltung, Rollen, Rechte und Profile: Auch die Userverwaltung wird komplett als Funktionalität von der

© Software & Support Verlag GmbH

Platform as a Service zur Verfügung gestellt – inklusive Rechte, Rollen und Profile der Anwender.

- Security und Access-Control: Auch in Sachen Sicherheit und Zugangskontrolle stellt eine solche Plattform einen kompletten Baukasten zur Verfügung. Dieser erleichtert die Anpassung und bietet die Systemsicherheit betreffend verschiedene Optionen (Zugriffsbeschränkung, IP-Adressen bedingt, https, etc.).

Der nächste Schritt: Softwareentwicklung on-demand

Jede On-Demand-Plattform stellt ihre eigenen Möglichkeiten bereit, um Anwendungen zu entwickeln. Leider gibt es hierbei noch keine Standardisierung, sodass die Migration von Plattform A zu Plattform B immer noch Schwierigkeiten hervorruft. Hat man sich jedoch für eine Plattform entschieden, stehen einem sehr viele Wege offen.

Für die Softwareentwicklung auf Force.com steht der so genannte Apex Code, eine Java-ähnliche, auf die Plattform angepasste Programmiersprache, zur Verfügung. In dieser Sprache geschriebene Programme werden vollständig innerhalb der Force.com-Laufzeitumgebung, also auf dem On-Demand-Server, ausgeführt. Im Gegensatz zum Ansatz mit JavaScript bestehen keine Einschränkungen hinsichtlich der Performance, wie sie durch die Übertragung und clientseitige Ausführung von Skripten entstehen. Apex-Code-Programme werden online geschrieben, zur Plattform übertragen, übersetzt und lassen sich im Anschluss aus unterschiedlichen Umgebungen heraus nutzen (Abbildung 1).

Überträgt ein Entwickler Apex-Code-Skripte auf die Force.com-Plattform, werden diese in einem ersten Schritt überprüft und anschließend in einen abstrakten Satz von Instruktionen übersetzt. Diese Instruktionen werden im Metadata-System gespeichert und können vom Apex-Code-Laufzeitinterpreter ausgeführt werden. Wenn der Force.com-Anwender den Apex Code ausführen möchte, werden die übersetzten Instruktionen aus dem Metadata-System geholt und mittels des Laufzeitinterpreters abgearbeitet. Das Ergebnis dieses Vorgangs wird im Anschluss zum Anwender gesendet.

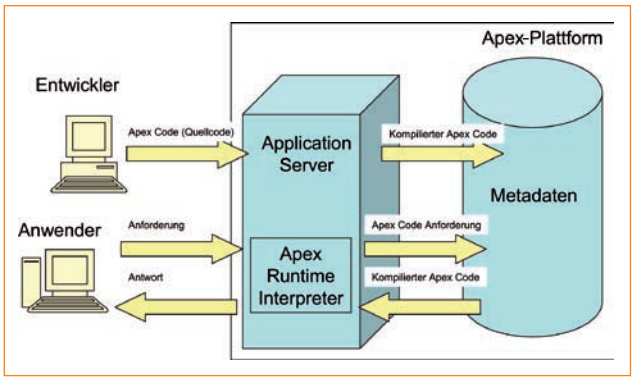


Abb. 1: Apex-Code in der Entwicklung und zur Laufzeit

Die Apex-Code-Sprache ist sehr stark an Java angelehnt, enthält aber integrierte Abfrage- und Manipulationsmöglichkeiten für die in der Force.com-Instanz gespeicherten Datensätze. Die in die Sprache integrierte Unterstützung umfasst dabei die meisten von der Apex-Plattform bekannten Idiome:

- Die Datenmanipulation mittels *INSERT, UPDATE, DELETE* inklusive der dazugehörigen Ausnahmebehandlung ist über entsprechende Schlüsselworte möglich.
- Eine integrierte Abfragesprache auf Basis der *Sforce Object Query Language* (SOQL) ermöglicht den Zugriff auf einzelne Objekte oder Objektmengen.
- Steuer- und Flussanweisungen (zum Beispiel *for*-Schleifen), die eine Batchverarbeitung ermöglichen, wurden integriert.
- Es besteht die Möglichkeit, Objekte zu sperren, um Konflikte zu vermindern.
- Methoden in Apex Code können als öffentliches API anderen Systemen (wie AJAX-basierten Clients) zur Verfügung gestellt werden.

Mittels Apex Code ist es möglich, Businessfunktionalität direkt in eine Force.com-Anwendung einzufügen oder in dieser zu nutzen. Darüber hinaus können Events, Buttons, Datensatz-Updates oder auch Controls mit mehr Logik versehen werden. Zahlreiche weitere Informationen zu Apex Code sind in der Language Reference (Apex Code Language Reference als PDF von salesforce.com) zu finden.

Damit die Entwicklung auch Spaß macht, bringt Force.com entsprechende Werkzeuge mit. So lassen sich die einzelnen Artefakte leicht ausprobieren. Zudem werden sie automatisch mit jenen in der Plattform synchronisiert.

Web Services als Grundlage der Programmierung

Der Zugriff auf die gespeicherten Daten ist mittels Web Services möglich. Nicht weiter verwunderlich, erzwingen SaaS und PaaS eine solche Herangehensweise doch geradezu. Mit dieser Technik erhält man gleichzeitig die größtmögliche Freiheit bei Design und Funktionalität von Anwendungen. Im Gegenzug muss natür-

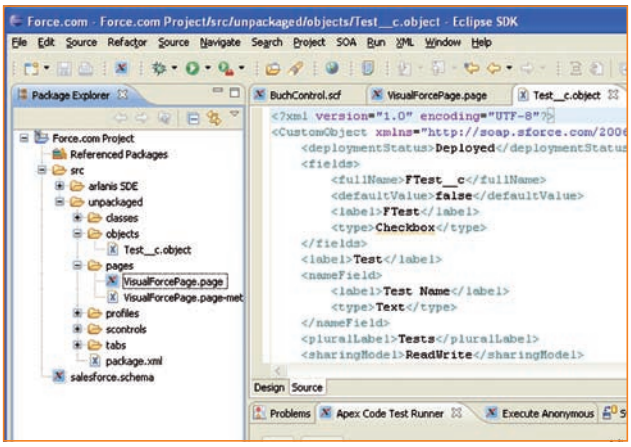


Abb. 2: Development as a Service (Entwicklung und Veröffentlichung eines Custom-Objekts auf der Plattform)

© Software & Support Verlag GmbH

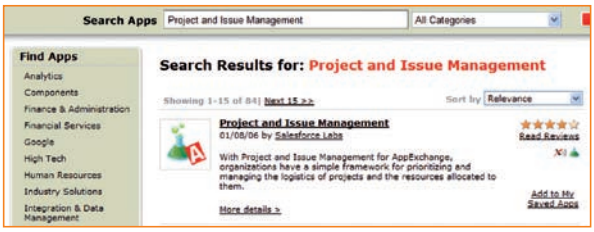


Abb. 3: Finden und Installieren von Anwendungen in den eigenen Anwendungsbereich

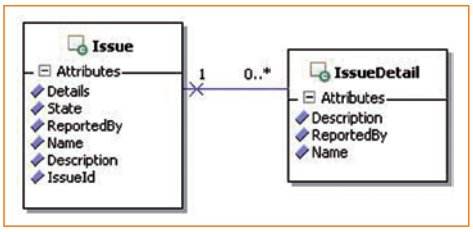


Abb. 4: Benutzerdefinierte Objekte und Attribute

lich speziell darauf geachtet werden, wie genau mit den Daten umgegangen wird. Man sollte hier mit einem Testaccount arbeiten oder zumindest eine Sicherung des Datenbestands vornehmen.

Web Services werden im Wesentlichen durch eine Serviceschnittstelle (WSDL) als auch durch Nachrichten (SOAP, XML) beschrieben. Diese Technologie ermöglicht es, mit beliebigen Sprachen, wie zum Beispiel Java, C# oder auch JavaScript, auf eine Force.com-Instanz zuzugreifen. Der folgende Quelltext zeigt das Anlegen eines neuen Kontakts in einem bestehenden Account. Dazu wird sowohl der Name des Kontakts als auch ein Account-Objekt übergeben. Die in die Programmiersprache der Plattform integrierte Transaktionssteuerung wird genutzt, um den neuen Kontakt dauerhaft zu speichern. Aus der Funktion wird die ID des Kontakts zurückgegeben.

```
webServiceId createContact(String a, Account acc) {
 Contact c = new Contact(LastName = a, AccountId = acc.Id);
 insert c;
 commit;
 return c.id;
}
```

Die Deklaration als Web Service ermöglicht den Zugriff von beliebigen Clients aus, aus allen möglichen Szenarien. So lassen sich Bibliotheken von nachnutzbaren Artefakten schaffen, welche wiederum in die verschiedensten Anwendungen integriert werden können.

Anwendungsarten auf Force.com

Salesforce.com hat selbst einige On-Demand-Anwendungen auf Force.com entwickelt. Dazu gehören beispielsweise Applikationen für das Customer Relationship Management (CRM) als auch für den Service- und Supportbereich.

Ihre wahre Stärke zeigt die Force.com-Plattform jedoch angesichts der mittlerweile mehr als 800 unterschiedlichen Businessapplikationen, die Kunden, Softwarehäuser oder freie Entwickler darauf entwickelt haben [5]. Über AppExchange, den Onlinemarktplatz für auf Force.com entwickelte Anwendungen, können etwa 800 solcher Softwareangebote bezogen werden. Grundsätzlich lassen sich Anwendungen, die auf Force.com entwickelt wurden, in eine der drei folgenden Kategorien einteilen:

- *Native Anwendungen*: Native Anwendungen werden ausschließlich mithilfe der Metadaten und des Apex Builders entwickelt. Sie benutzen weder direkt das interne API noch haben sie Referenzen oder Links auf andere Anwendungen.
- *Composite-Anwendungen*: Composite-Anwendungen bestehen aus einer Kombination von Metadaten und *low-level*-Funktionalität des Apex API. Solche Anwendungen ermöglichen zum Beispiel die Entwicklung kundenspezifischer Oberflächen oder die Integration neuer Komponenten für das Benutzerinterface. Interessant ist diese Möglichkeit auch, wenn existierende On-Demand-Anwendungen nach Apex portiert werden. Externe Services oder auch externe Komponenten lassen sich weiterverwenden.
- *Client-Anwendungen*: Diese Applikationen nutzen exklusiv das Apex Client API. Sie benutzen zum Beispiel das Web Service API und haben eine von salesforce.com getrennte Benutzeroberfläche.

Praxisbeispiel – Entwicklung einer On-Demand-Anwendung auf Force.com

Der folgende Abschnitt zeigt die Entwicklung einer ersten Anwendung mit dem Apex Builder (eine ausführlichere Version ist unter [4] zu finden, die vollständigen Quelltexte unter [6]). Dabei handelt es sich um einen Issue Manager, also ein Ticketsystem im weitesten Sinn, jedoch in sehr kleiner Ausbaustufe. Eine solche Anwendungsgrundform findet sich zum Beispiel als Grundstein jeder modernen Bug- und Feature-Tracking-Anwendung oder im Supportbereich von Firmen. Dank der On-Demand-Plattform muss sich der Entwickler lediglich um die Anwendung selbst Gedanken machen. Laufzeitumgebung, Entwicklungswerkzeuge und Deployment liefert Force.com. Eine Applikation benötigt

- benutzerdefinierte Objekte (für die späteren Daten)
- eine Seite zum Anzeigen und Editieren der Objekte
- eine Reportseite
- einen Platz zum Veröffentlichen der Anwendung

Anlegen von benutzerdefinierten Objekten

Anwendungen auf Force.com sind in der Regel datenzentriert, das bedeutet, das gesamte Design der Software wird um die einzelnen Datenobjekte herum angelegt. Die Entwicklung startet demzufolge am besten mit der Definition der zu erfassenden Daten und ihrer Gruppierung zu Datenobjekten. Das kann zum Beispiel ein Adressobjekt mit den typischen Feldern Stadt, Postleitzahl und Straße sein.

- **Datenobjekt *Issue***: In diesem benutzerdefinierten Objekt werden alle wichtigen Daten eines Falls (Issue) abgelegt. Das sind natürlich die Beschreibung, bei welchem Kunden oder Benutzer dieser Fall zu sehen ist sowie der aktuelle Status.
- **Datenobjekt *IssueDetail***: Manchmal ist es notwendig, weitere Informationen über einen gewissen Zeitraum zu sammeln und dem Fall zuzuweisen. Das sind zum Beispiel Rückfragen beim Kunden, Diskussionen zur Lösung der Aufgabe und erklärendes Material. Da es sich später um eine beliebige Menge an Informationen handeln kann, wird eine

© Software & Support Verlag GmbH

Master-Detail-Verbindung zum Issue verwendet.

Die Entwicklung der Datenobjekte erfolgt mit der integrierten Entwicklungsumgebung (zu finden auf der Seite der benutzerdefinierten Objekte, Abbildung 5). Nachdem das Objekt angelegt ist, werden ihm seine Felder zugeordnet (Abbildung 6). Sehr gut lässt sich die Wiederverwendung von vorhandenen Objekten und Anwendungen erkennen. So trägt der Entwickler zum Beispiel in das Feld *Reportet von* einen direkten Link zu einem Kontakt ein – in diesem Fall einem Objekt der bereits installierten Salesforce-CRM-Anwendung.

Die Anwendungsseite

Die Hauptseite der Anwendung nimmt das Objekt *Issue* sowie das davon abhängige Objekt *IssueDetail* auf. Als Erstes wird das *Issue*-Objekt mit der Seite verbunden. Genauer gesagt wird also das Layout für die Seite um das Objekt herum gestaltet. Das hat den Vorteil, dass man sich die Seite schon vorab ansehen kann. Wenn noch keine Anwendung existiert, kann die Seite auch manuell über den Link *Show all Tabs* mit anschließender Wahl der Seite in den Vordergrund gebracht werden. Dieser Link findet sich ganz rechts, am Ende der Leiste der verfügbaren Tabs in der Plattform (Abbildung 7). Nach der Seite wird nun das Layout für das Objekt selbst festgelegt. Dazu verwendet man am besten den Objekt-Layout-Editor (Abbildung 8).

Reports, S-Controls und eigener Quelltext

Bei der Nutzung der Anwendung werden im Laufe der Zeit mehr und mehr Einträge erfasst. Dabei ist es natürlich wünschenswert, wichtige Informationen im Überblick darzustellen. Solche Informationen sollte der Entwickler wiederum auf einer eigenen Seite zusammenfassen. Zur Reportseite gehören ein S-Control und die entsprechende Anwendungsseite. Das S-Control enthält dabei die Logik für das Summieren der einzelnen Issues und die Verbindungsdetails zum Salesforce.com-Account (Listing 1). Das S-Control selbst wird mit einer eigenen Seite verbunden und beim Anwählen derselben dargestellt. Wie man schön sieht, wird direkt die darunterliegende Plattform genutzt (Abbildung 9).

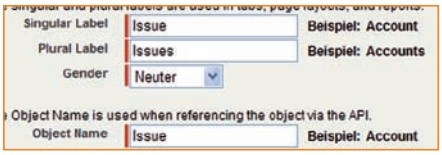
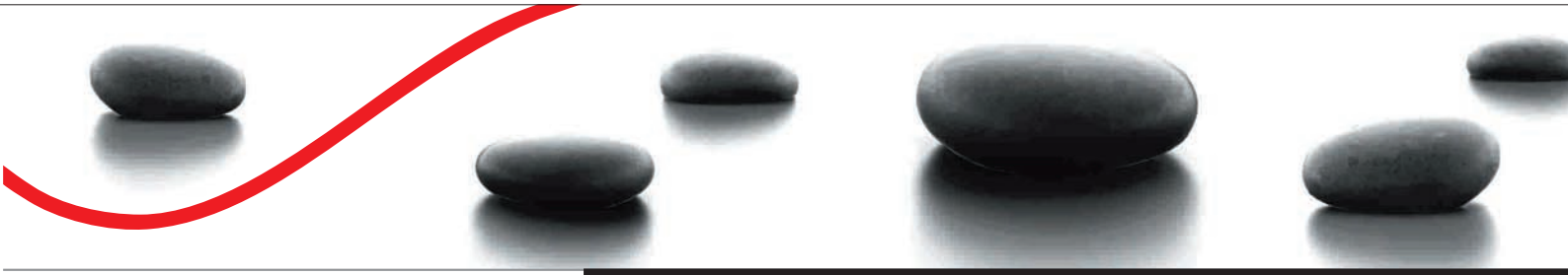


Abb. 5: Anlegen des Objekts *Issue*

Abb. 6: Felder zum Objekt hinzufügen

Der Lohn der Arbeit – Packen und Veröffentlichen

Nachdem eine Anwendung fertiggestellt ist, gilt es, sie zur Nutzung frei zu geben. Dabei unterscheidet man die private Nutzung, auch als Packaging bezeichnet, und die öffentliche Nutzung, das Publishing. Egal für welche Art der Verteilung und späteren Nutzung man sich entscheidet, der Packaging-Prozess ist der erste Schritt. Packaging bezeichnet dabei genau ge-



arlanis Universal Data Converter UDC

Der arlanis Universal Data Converter UDC bietet die optimale Lösung, um komplexe Daten verschiedener Systeme in komplexe Daten anderer Systeme umzusetzen.

Die Entwicklungsumgebung ermöglicht es sowohl einfache 1:1 Konvertierungen zu erzeugen als auch sehr komplexe Konvertierungen durchzuführen.

/ Datenintegration / Datenmigration / Datenkonvertierung

/ hochperformant / plattformunabhängig / 100% Java

/ easy-to-use / universell verwendbar / SOA-ready

/ integrierbar in jede Anwendung / Stand-Alone

/ große Datenmengen in Batch-Verarbeitung ausführbar

arlanis

Covering the last mile of integration.

/ arlanis Software AG / www.arlanis.com / info@arlanis.com

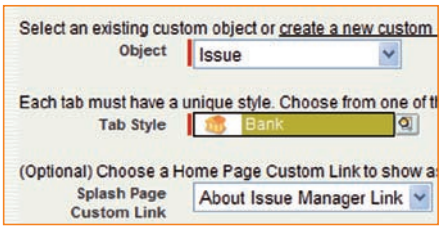


Abb. 8: Erzeugen einer Seite für unser Objekt

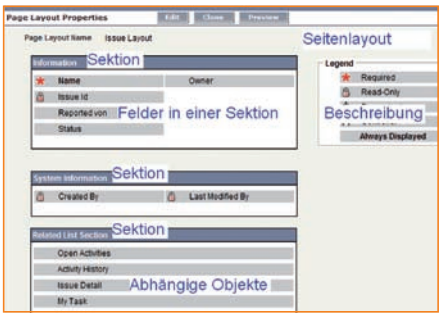


Abb. 9: Layout der Seite anlegen

nommen das Sammeln aller Artefakte der Anwendung und die Veröffentlichung als ein Paket in einem privaten Bereich.

Ist das Package fertig, wird es in den privaten Bereich der Plattform geladen, der nur für den Entwickler-Account sichtbar ist. Jedoch werden Anwendungen, die in diesem Bereich veröffentlicht sind, durch eine eindeutige URL beschrieben (und können auch schon „von außen" genutzt werden). Bisher wurde die Anwendung lediglich einem privaten Benutzerkreis zur Verfügung gestellt. Das hat zum einen den Vorteil, dass ein ausgiebiger Test möglich ist und zum anderen, dass die Anwendung sehr selektiv an bestimmte Benutzer gegeben werden kann. Dieser enge Kreis von Usern hat dabei normalerweise auch Verständnis für die eine oder andere Limitierung des Systems. Was aber noch wichtiger ist: Der private Kreis der Nutzer kennt normalerweise den Entwickler der Anwendung oder hat sie sogar bei ihm in Auftrag gegeben.

Der nächste Schritt ist die Veröffentlichung der Anwendung, das Publishing. Dafür sind folgende Schritte erforderlich:

- Publisher-Profil erzeugen
- Anwendung mit Publisher-Profil verbinden und Metainformationen zur Verfügung stellen
- Anwendung für das Review Board vorbereiten und absenden

Ist der Prozess des Publishings erfolgreich abgeschlossen, befindet sich die Anwendung auf AppExchange – dem Onlinemarktplatz von salesforce.com. Potenzielle Kunden können dort die neue Anwendung finden, auswählen, testen und dauerhaft nutzen. Die Entwickler müssen keinen eigenen Downloadbereich verwalten.

Fazit

Software as a Service und Platform as a Service sind den Kinderschuhen bereits entwachsen. Das Modell hat sich in allen Unternehmensgrößen durchgesetzt und wird in vielen Fachbereichen genutzt. Auch IT-Analysten wie Gartner prognostizieren für die Technologie ein exponentielles Wachstum. Entwickler genießen mit Platform as a Service den Vorteil, sich tatsächlich ausschließlich auf die Anwendung selbst konzentrieren zu können. Die Entwicklungsinfrastruktur übernimmt der Anbieter der Plattform, für die Vermarktung der Software wird ein Onlinemarktplatz zur Verfügung gestellt.

Abb. 10: Eine einfache Reportseite

Listing 1

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
                                    Transitional//EN">
<html>
<head>
<title>Issue Manager Report</title>
<script
src=https://www.salesforce.com/services/lib/ajax/
                    beta3.3/sforceclient.js?browser=true
 type="text/javascript"></script>

<script language="javascript">
<!--
function initPage() {
 sforceClient.registerInitCallback(pageSetup);
 sforceClient.init("{!API_Session_ID}",
  "{!API_Partner_Server_URL_70}", true);
}

function pageSetup() {
 sforceClient.setBatchSize(10);
 var qr = sforceClient.query(
  "Select Name From Issue__c Where State__c = 'Neu'");
document.getElementById("divMain").innerHTML +=
 "Neu: " + qr.size + "<br>";
qr = sforceClient.query(
 "Select Name From Issue__c Where State__c =
                                    'In Arbeit'");
document.getElementById("divMain").innerHTML +=
 "In Arbeit: " + qr.size + "<br>";
qr = sforceClient.query(
 "Select Name From Issue__c Where State__c = 'Fertig'");
document.getElementById("divMain").innerHTML +=
 "Fertig: " + qr.size + "<br>";}
//-->
</script>

</head>
<body onload="initPage();">
 <DIV id="divMain"
style="WIDTH: 368px; POSITION:relative;HEIGHT:424px"
                    ms_positioning="GridLayout"></DIV>
</body>
</html>
```

Andreas Holubek arbeitet als Vice President Engineering für die arlanis Software AG. Sein besonderes Interesse gilt SOA, EAI und den verschiedenen Persistenztechniken. Er ist als Autor und als Referent auf verschiedenen Konferenzen bekannt. Daneben beschäftigt er sich mit der Einführung und Umsetzung von modernen Technologien und Spezifikationen in der Praxis.

Christian Metzger, CEO der Arlanis Software AG, hat bereits zahlreiche Salesforce.com-Projekte erfolgreich eingeführt und umgesetzt.

Links & Literatur

[1] Force.com Platform: force.com

[2] Salesforce.com: salesforce.com/de

[3] Google App Engine: appengine.google.com

[4] Christian Metzger, Andreas Holubek: Das Salesforce.com Entwicklerhandbuch. entwickler.press 2007

[5] AppExchange: www.salesforce.com/appexchange

[6] Quelltexte zum Buch: entwickler-press.de (nach „Salesforce" suchen)

Salesforce.com CRM: salesforce.com/de/product

arlanis

arlanis Software AG

Mannheimer Straße 97
D-60327 Frankfurt am Main
Tel: +49-69-241829-26
http://www.arlanis.com

© Software & Support Verlag GmbH