



Oracle Forms goes Mobile

Jan-Peter Timmermann, Jan-Peter Timmermann Consulting

Seit Jahren hört man immer wieder von aktuellen Technologien in der Entwicklung. Da sind Schlagworte wie „IT 4.0“, „Mobile“, „Cloud“ und so weiter im Gespräch. Es gibt mit Sicherheit gute Gründe, sich über diese Entwicklungen Gedanken zu machen und auch die Entwicklung der Unternehmensanwendungen in die eine oder andere Richtung zu überdenken.

Was geschieht jedoch, wenn eine unternehmenskritische Anwendung auf einer alten Technologie basiert – und das schon mehr als zwanzig Jahre? Das gesamte Wissen steckt in dieser Anwendung. Die Anwender arbeiten damit bereits seit einigen Jahren und das nahezu blind. Die Lösung deckt genau das ab, was das Unternehmen braucht.

Nun kommen aber immer mehr Anforderungen auf die Entwickler zu, auch Anwendungen für mobile Endgeräte anzubieten. Vieles, was für das mobile Endgerät gefordert wird, ist bereits in der Forms-Anwendung hinterlegt. Muss man nun alles neu entwickeln, wie schaut es aus mit Security, was ist, wenn kein Internet-Empfang besteht? So kommen komplett neue Fragen auf die Entwickler und die Betreiber zu.



Abbildung 1: Die Vorstellung von „Forms Mobile“

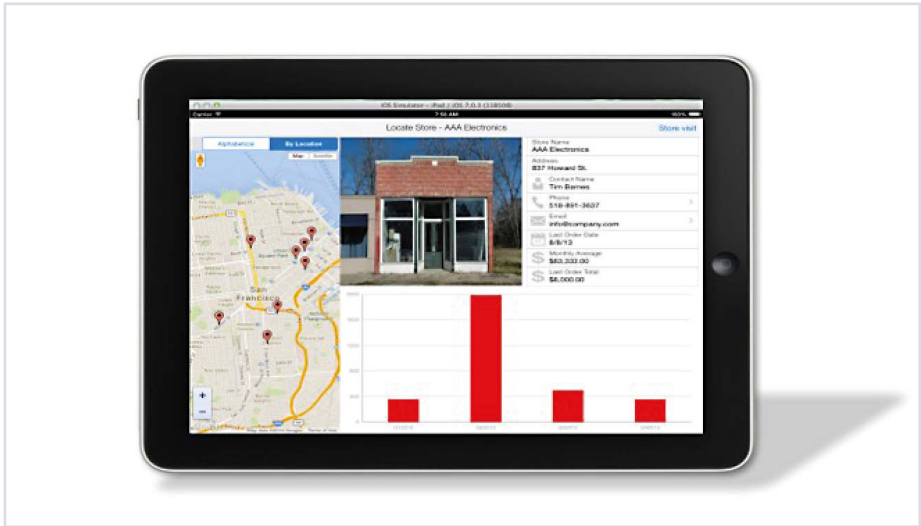


Abbildung 2: Mobile Erwartung

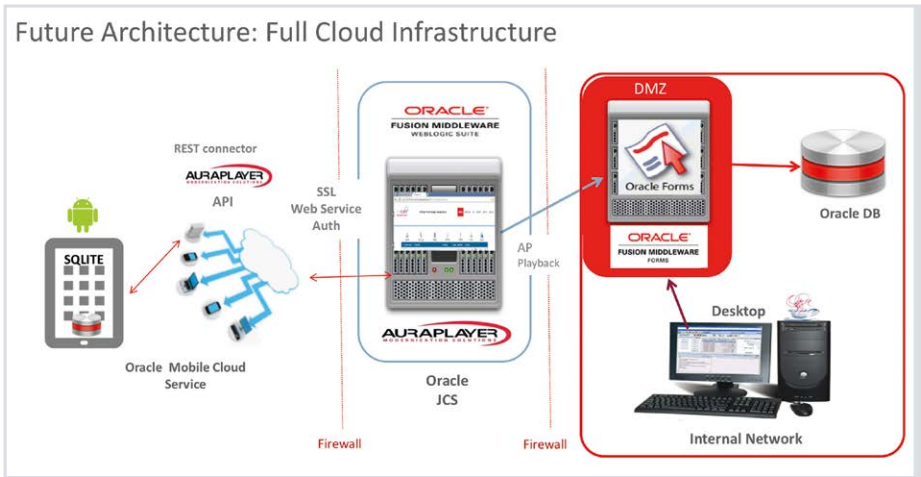


Abbildung 3: Eine mögliche Architektur

3 Steps to Mobile Success with Oracle Forms

- 1. Business process and User eXperience**
 - Identify the the Oracle Forms business process
 - Define the end-user experience
 - Create a Storyboard
- 2. Mobile APIs & Services**
 - Set up platform to manage services
 - Record your Oracle Forms process and expose REST services
 - Create your mobile APIs
- 3. Mobile App**
 - Develop mobile app against your mobile APIs, in the tools of your choice
 - Be agile, release early, release often
 - Measure success with Mobile Analytics

Abbildung 4: Die notwendigen Schritte

Jetzt hat Oracle im letzten Jahr endlich das lange versprochene Release Oracle Forms/Reports in der Version 12c auf den Markt gebracht. Oracle Forms 12c arbeitet mit den aktuellen JDKs zusammen, sodass hier von Seiten der Applika-

tions-Server ein sehr aktueller Stand vorgehalten ist.

Nun gibt es verschiedene Ansätze, mit diesen Themen umzugehen. Viele Entwickler werden auf eine neue Entwicklung hinarbeiten. Es gibt aber genauso den Trend,

bestehende Forms-Anwendungen in Richtung einer neuen Technologie zu migrieren. Leider birgt dieses Vorgehen diverse Risiken, auf die hier nicht weiter eingegangen wird. Trotzdem braucht es auch seine Zeit, eine neue Anwendung zu erstellen oder eine bestehende zu migrieren. Was geschieht in der Zwischenzeit?

Der Artikel beschreibt die Möglichkeit, eine bestehende Forms-Anwendung über einen Webservice für mobile Anwendungen bereitzustellen. Zuvor ist es wichtig zu verstehen, dass man nicht die Forms-Anwendung mobil bereitstellen möchte, sondern immer den Geschäftsprozess. Wenn man sich mit Forms-Anwendern oder -Entwicklern unterhält und sie dann fragt, was sie sich unter einer mobilen Forms-Anwendung vorstellen, stellt man sehr schnell fest, dass meistens ein bestimmtes Bild herauskommt (siehe Abbildung 1).

Man will natürlich die Möglichkeiten ausnutzen, die einem Smartphones und Tablets anbieten. Wer eine Adresse hat, erwartet auch, dass ihm diese in einer Karte angezeigt wird. Ebenso möchte man, dass der Kunde angerufen wird, wenn man auf seine Telefonnummer klickt. So sieht die Erwartungshaltung eher wie in Abbildung 2 aus.

Alle diese Informationen liegen innerhalb der Forms-Anwendung vor. Man muss sie nur als Webservice bereitstellen. Die Schritte sind bei allen Tools ähnlich. Es muss in jedem Falle ein Storyboard erstellt werden, das die Geschäftsprozesse abbildet. Dazu gibt es unterschiedliche Ansätze. Die einen packen in jeden Sourcecode weiteren Code, um den Prozess aufzuzeichnen, andere bevorzugen das Aufzeichnen der Forms-Schritte, etwa mit der Oracle Application Testing Suite.

Wir werden an dieser Stelle das Tool „AuraPlayer“ intensiver betrachten. Es ist als Tool insofern interessant, weil man nicht in den bestehenden Code eingreifen muss, sondern das Ausführen der notwendigen Schritte in einem Storyboard erstellt – ohne eine einzige Zeile Code schreiben zu müssen. Diese Technik funktioniert mit allen notwendigen Operationen, die im Forms-Umfeld bereitstehen. Das ist zum einen natürlich die Abfrage, zum anderen kann man auch Daten einfügen, ändern oder aber auch über den erstellten Webservice löschen.

Was das Tool nicht kann, ist das Erstellen einer mobilen Anwendung. Es ist ein-

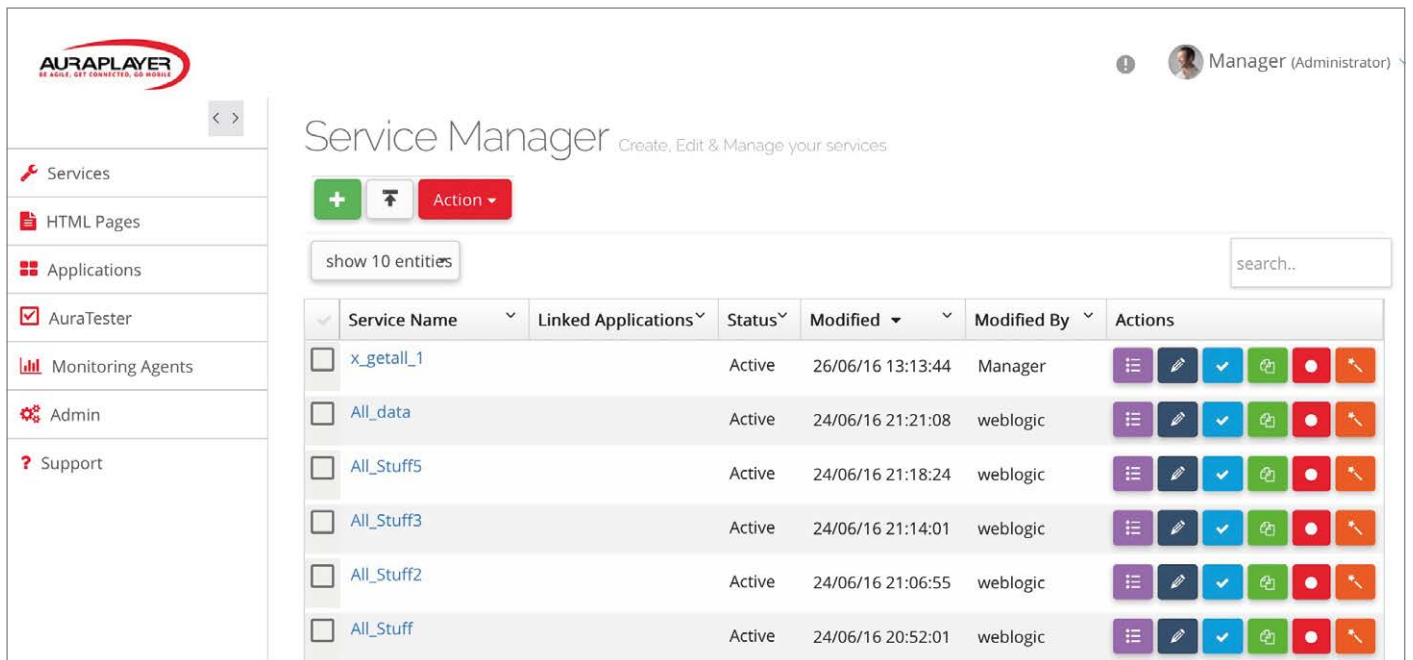


Abbildung 5: Der Service Manager

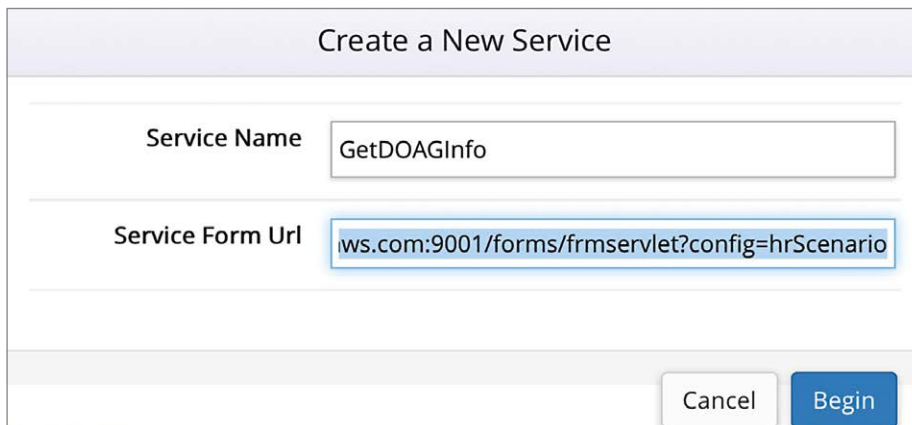


Abbildung 6: Das Storyboard starten

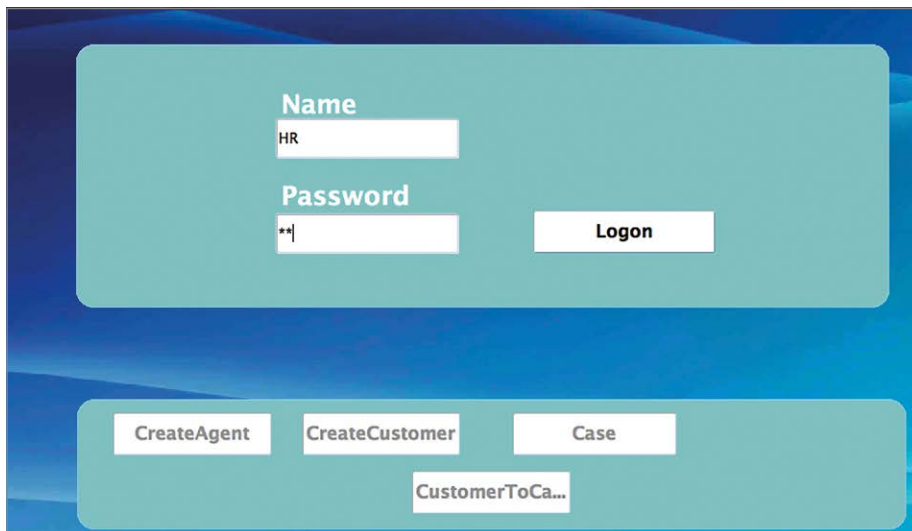


Abbildung 7: Die Anwendung starten

zig und allein dazu ausgelegt, den Webservice bereitzustellen. Dieser greift bei Bedarf immer auf die aktuelle Forms-Maske zu und führt dies zur Laufzeit im Hintergrund aus. Das heißt allerdings auch, dass alle Änderungen, die in dem Forms-Modul erfolgen, sich automatisch auf den Webservice durchziehen. Ebenso werden alle Validierungen, die innerhalb eines Forms-Moduls ausgeführt werden, auch durch den Webservice ausgeführt. *Abbildung 3* zeigt die Architektur der Forms-Umgebung mit dem AuraPlayer.

Es wurde hier der komplette Cloud-basierten Ansatz gewählt. Es ist zwar ebenso möglich, dies „in-house“ zu machen, der Oracle Mobile Cloud Service bietet jedoch verschiedene Vorteile, die man nutzen sollte. Bevor wir uns nun den einzelnen Schritten zuwenden, hier noch einmal ein Überblick darüber, was zu tun ist (*siehe Abbildung 4*). Der System-Manager ist das Steuerelement, um die Webservices zu erstellen und zu verwalten (*siehe Abbildung 5*).

Im ersten Schritt wird immer über den Service Manager ein Storyboard erstellt. Dazu startet man einfach eine Forms-Session, in der man sein Geschäftsmodell abbilden möchte. Aufrufparameter ist die URL der Forms-Anwendung (*siehe Abbildung 6*). Dieser Parameter kann sich sowohl auf eine Test- als auch auf eine Produktiv-Umgebung beziehen, da hier keine Eingriffe in der Anwendung erfolgen. Mit

diesem Aufruf werden die Forms-Anwendung gestartet und alle anschließend getätigten Schritte aufgezeichnet (siehe Abbildung 7). Das wird über die Steuerleiste bestimmt (siehe Abbildung 8).

Mit „Capture Parameter“ werden alle Items einer Maske in den Webservice übernommen, ebenso kann man durch Anklicken der Items entscheiden, welche Informationen bereitgestellt werden sollen. Die Entscheidung fällt hier für die Übernahme aller Items. Daraufhin wird der Webservice genau mit den Informationen erstellt, die man haben möchte (siehe Abbildung 9).

Es sind nun alle Items der Abfrage hinterlegt. Zudem ist die Abfrage aufgezeichnet worden. Man kann jetzt daraus über den Create Service einen Webservice erstellen und sich über die URL „`http://<HOSTNAME>:7001/ServiceManager/Macro/ExecMacro/GetDOAGInfo?wsdl`“ anzeigen lassen (siehe Listing 1).

Mit diesen Informationen lässt sich nun eine Web-Anwendung erstellen. Das

kann eine MCS-Anwendung sein, genauso könnte man die Information auch in eine Apex-Lösung einbauen. Automatisches Testen mit diesen Informationen ist ebenso möglich, denn man verfügt über alle Informationen, die zum Testen einer Forms-Anwendung nötig sind. Dazu muss man nur noch diesen Webservice zum Test Case machen (siehe Abbildung 10).

Dieser Test Case versetzt einen in die Lage, über Parameter/Items Validierungen daraufhin zu testen, ob die Anwendung Fehler erzeugt oder genau das Ergebnis liefert, das erwartet wird. Es besteht die Möglichkeit, über Textdateien verschiedene Eingaben zu steuern und die unterschiedlichen Eingaben gegen Parameter zu validieren. Zudem kann man feststellen, wie schnell die Forms-Anwendung antwortet.

Über die Validierungen legt man die Bedingungen dafür fest, ab wann ein Test erfolgreich durchlaufen ist. Natürlich gibt es hier die Möglichkeit, mehrere Test Cases zu einer Teststrecke zusammenzufas-

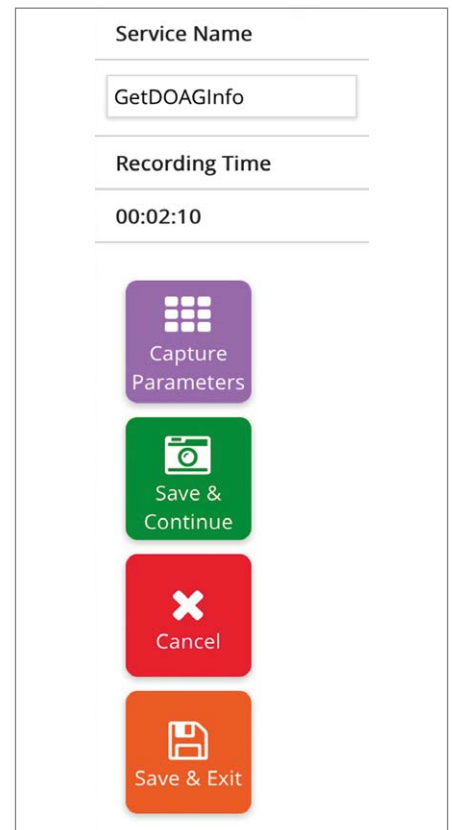


Abbildung 8: Die Steuerung

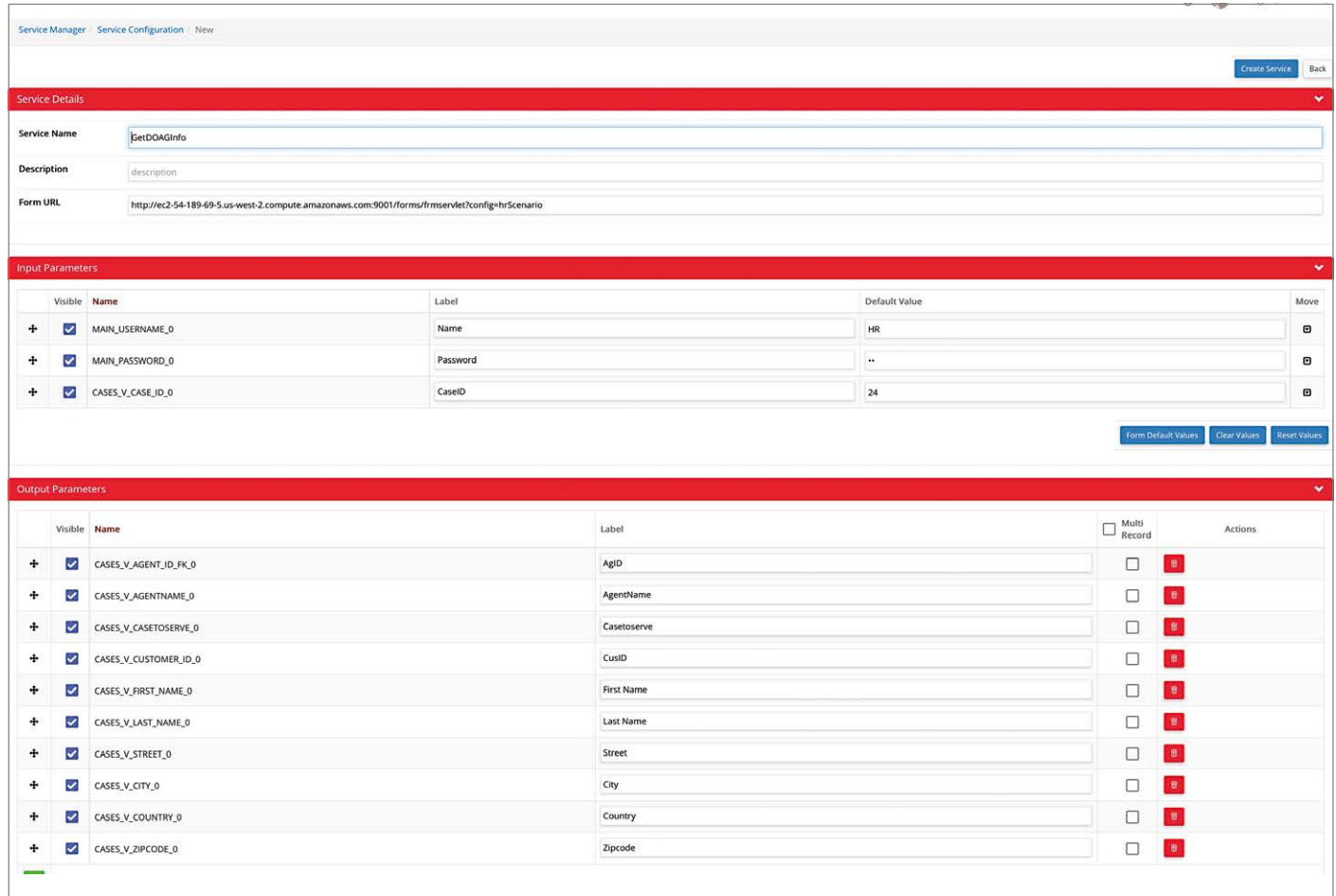


Abbildung 9: Den Webservice erstellen

```

<wsdl:definitions targetNamespace="http://services.auraplayer.com">
  - <wsdl:types>
    - <schema elementFormDefault="qualified" targetNamespace="http://services.auraplayer.com">
      - <element name="GetDOAGInfo">
        - <complexType>
          - <sequence>
            <element name="MAIN_USERNAME_0" type="xsd:string"/>
            <element name="MAIN_PASSWORD_0" type="xsd:string"/>
            <element name="CASES_V_CASE_ID_0" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
      - <element name="GetDOAGInfoResponse">
        - <complexType>
          - <sequence>
            <elementname="GetDOAGInfoReturn" type="impl:GetDOAGInfoResponse"/>
          </sequence>
        </complexType>
      </element>
      - <complexType name="GetDOAGInfoResponse">
        - <sequence>
          <element name="GetDOAGInfoMessage" type="impl:GetDOAGInfoMessage"/>
          <element name="GetDOAGInfoElements" type="impl:GetDOAGInfoElements"/>
        </sequence>
      </complexType>
      - <complexType name="GetDOAGInfoMessage">
        - <sequence>
          <element name="Error" type="string"/>
          <element name="PopupMessages" type="string"/>
          <element name="StatusBarMessages" type="string"/>
        </sequence>
      </complexType>
      - <complexType name="GetDOAGInfoElements">
        - <sequence>
          <element name="CASES_V_AGENT_ID_FK_0" nillable="true" type="xsd:string"/>
          <element name="CASES_V_AGENTNAME_0" nillable="true" type="xsd:string"/>
          <element name="CASES_V_CASETOSERVE_0" nillable="true" type="xsd:string"/>
          <element name="CASES_V_CUSTOMER_ID_0" nillable="true" type="xsd:string"/>
          <element name="CASES_V_FIRST_NAME_0" nillable="true" type="xsd:string"/>
          <element name="CASES_V_LAST_NAME_0" nillable="true" type="xsd:string"/>
          <element name="CASES_V_STREET_0" nillable="true" type="xsd:string"/>
          <element name="CASES_V_CITY_0" nillable="true" type="xsd:string"/>
          <element name="CASES_V_COUNTRY_0" nillable="true" type="xsd:string"/>
          <element name="CASES_V_ZIPCODE_0" nillable="true" type="xsd:string"/>
          <element name="CASES.V_CASE_ID_0)" nillable="true" type="xsd:string"/>
        </sequence>
      </complexType>
    </schema>
  </wsdl:types>

```

Listing 1

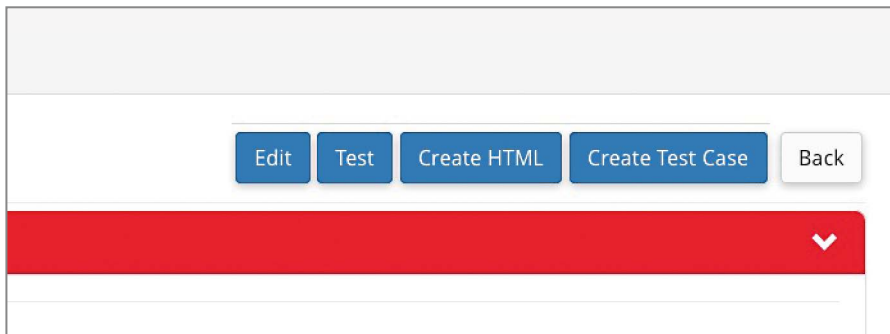


Abbildung 10: Den Test Case erstellen

sen und diese als Gesamtes zu betrachten (siehe Abbildung 11).

Fazit

Der Autor ist seit Jahren ein Freund von Oracle Forms, sieht sich aber immer häufiger dem Thema gegenüber, modern arbeiten zu müssen. Mit dem AuraPlayer hat er eine Möglichkeit gefunden, mit überschaubarem Aufwand seine bestehenden

Output Parameters		
Name	Label	Validations
CASES_V_AGENT_ID_FK_0	AgID	
CASES_V_AGENTNAME_0	AgentName	
CASES_V_CASETOSERVE_0	Casetoserve	
CASES_V_CASE_ID_0	CaseID	
CASES_V_CUSTOMER_ID_0	CusID	
CASES_V_FIRST_NAME_0	First Name	
CASES_V_LAST_NAME_0	Last Name	
CASES_V_STREET_0	Street	
CASES_V_CITY_0	City	
CASES_V_COUNTRY_0	Country	
CASES_V_ZIPCODE_0	Zipcode	
PopupMessages	popup messages	
StatusBarMessages	status bar messages	
Error	error	Fail if Response is NOT empty

Response Time Validation	
Timeout	10 seconds

Abbildung 11: Validierungen

Forms-Information zu nutzen und für mobile Anwendung bereitzustellen.

Das Tool dient nicht dazu, Forms abzulösen, sondern besteht vielmehr aus dem Zusammenspiel von Forms und Webservice. Es bietet damit die Möglichkeit, sehr schnell Anwendungen für das Web bereitzustellen. Der Webservice lässt sich als Plug-in für die MCS nutzen, man spricht ihn direkt an oder implementiert ihn in Apex – die Möglichkeiten sind vielfältig.



Jan-Peter Timmermann
jan.timmermann@jptu.de

