

Zugriff auf Daten und Gerätefunktionen von mobilen Endgeräten mit ADF Mobile

Jürgen Menge
Oracle Deutschland B.V. & Co. KG
München

Schlüsselworte

Oracle ADF, Oracle ADF Mobile, Android, iPhone, iPad

Einleitung

Das Framework Oracle ADF ermöglicht bereits seit einigen Jahren die Entwicklung mobiler Anwendungen für den Einsatz in Unternehmen. In der Variante **ADF Mobile Browser** laufen diese Anwendungen auf einem zentralen Applikations-Server und das mobile Endgerät stellt lediglich die Inhalte im Browser des Mobilgerätes angepasst dar. Diese Arbeitsweise setzt eine ständige Netzverbindung voraus und ist in ihren Möglichkeiten der Nutzung von spezifischen Gerätefunktionen eingeschränkt.

Mit **ADF Mobile** können nun auch Anwendungen entwickelt werden, die im Offline-Modus auf dem Mobilgerät ablauffähig sind und lokale Datenbanken sowie spezifische Gerätedienste, wie z.B. Mail, SMS, Kamera, Adressbuch, GPS u.a. nutzen können.

Da diese Anwendungen auch web-basierte Inhalte integrieren können, werden sie als **Hybride Anwendungen** bzw. **Hybride Clients** bezeichnet.

Die hybride ADF-Anwendung läuft in einem separaten Applikations-Container auf dem mobilen Endgerät. Die native Applikationsschicht unterstützt drei wesentliche Funktionen:

1. gerätespezifische Darstellung, die an das Look & Feel des Gerätes angepasst ist
2. Anzeigen von Webinhalten (Content) innerhalb der Applikation (Web View)
3. Integration mit systemspezifischen Gerätediensten, wie z.B. Kamera und Adressbuch.

Bei der Darstellung der Inhalte kommen HTML5 und populäre JavaScript-Frameworks (PhoneGap, jQuery) zum Einsatz.

Mit ADF Mobile ist es möglich, eine Applikation aus sogenannten Features zusammenzusetzen (*adfmf-feature.xml*), die unterschiedliche Technologien verwenden. Wie in der Abb. 1 zu erkennen, können innerhalb der Web View folgende Technologien miteinander kombiniert werden:

- Server HTML (Remote URL)
Die Anwendungslogik befindet sich auf einem zentralen Server, der das HTML erzeugt.
- ADF Mobile AMX Pages
AMX-Seiten, die lokal ausgeführt werden und zur Laufzeit HTML5 und JavaScript erzeugen.
- Lokales HTML
HTML-Seiten, die lokal auf dem Gerät ausgeführt werden.
- Native View
Komponenten, die speziell für eine bestimmte Plattform (z.B. in Objective-C) entwickelt werden und damit nicht plattform-übergreifend zur Verfügung stehen.

Erst zum Zeitpunkt des Deployments wird aus der Applikation eine ausführbare Anwendung erzeugt, indem die plattform-spezifischen Artefakte (Deployment-Deskriptoren, Metadaten, Projekt Files, ...) für die jeweilige Plattform (iOS, Android) generiert und mit den notwendigen Bibliotheken zu einem Bundle zusammengefasst werden. Dabei werden Werkzeuge aus dem SDK der jeweiligen Zielplattform genutzt.

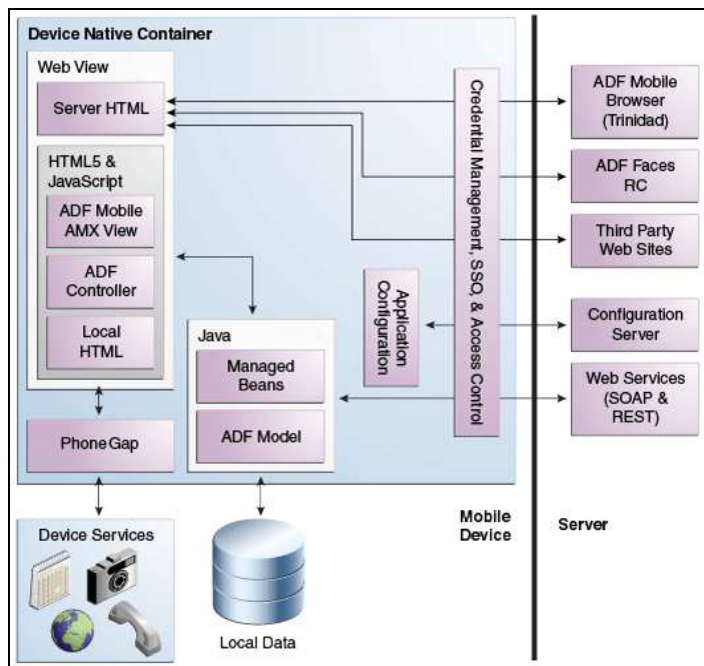


Abb. 1: Hybride Architektur von mobilen ADF-Anwendungen (Developer Guide Oracle ADF Mobile)

Zugriff auf Unternehmensdaten

Für Unternehmens-Anwendungen spielt der Zugriff auf die Unternehmensdaten eine wichtige Rolle. Mit ADF Mobile stehen hier zwei Möglichkeiten zur Verfügung.

Zunächst ist aus der Applikation ein Aufruf von **Web Services** (SOAP oder REST) möglich. Die Nutzung von Web Services im Backend setzt allerdings voraus, dass eine Verbindung zum Server hergestellt werden kann. Kann dies nicht garantiert werden, besteht die Alternative darin, die Anwendungsdaten lokal auf dem Gerät zu speichern und zu bestimmten Zeitpunkten mit zentralen Datenbanken zu synchronisieren.

Für die **lokale Datenhaltung** können in ADF Mobile wahlweise SQLite oder die Berkeley DB zum Einsatz kommen. Da die Berkeley DB in der aktuellen Version das SQLite API 3.7.1 implementiert, können für beiden Datenbanken die gleichen Funktionsaufrufe (SQL) verwendet werden.

Die folgende Übersicht stellt typische Anwendungsfälle und deren empfohlene Implementierung mit ADF Mobile dar:

Einfache Online-Anwendungen	User Interface ↔ Web Service Data Control ↔ Web Service (REST/XML oder SOAP)
Mittlere bis komplexe Online-Anwendungen	User Interface ↔ Bean Data Control ↔ POJO ↔ Web Service Data Control ↔ Web Service (REST/XML oder SOAP)
	User Interface ↔ Bean Data Control ↔ POJO ↔ Web Service (JSON)
Offline-Anwendungen	User Interface ↔ Bean Data Control ↔ POJO ↔ Lokale DB
	Lokale DB ↔ POJO ↔ Web Service Data Control ↔ Web Service (REST/XML oder SOAP)
	Lokale DB ↔ POJO ↔ Web Service (JSON)

Datenzugriff über Web Services

In AMX-Seiten können SOAP- oder REST-basierte Web Services über die entsprechenden Data Controls aufgerufen werden.

Für SOAP-basierte Web Services (1.1, 1.2) können im JDeveloper die Web Service Data Controls auf Basis der WSDL erzeugt werden. Eine Absicherung der Web Services durch Security Policies ist möglich.

REST-basierte Web Services verwenden die Standard-HTTP-Methoden GET, POST, PUT und DELETE. Die entsprechenden URL Service Data Controls können ebenfalls im JDeveloper generiert werden, wobei nur die Basic Authentication über HTTP/HTTPS unterstützt wird. ADF Mobile fügt der Anwendung automatisch die entsprechenden Web Service Policies für HTTP bzw. HTTPS hinzu. REST-basierte Web Services können alternativ über einen REST Web Service-Adapter, d.h. ohne Data Control, aufgerufen werden. Aufgrund der eingeschränkten Java-Funktionalität auf mobilen Endgeräten sind Web Service Proxies/Clients nicht einsetzbar.

Lokale Datenhaltung

SQLite ist eine leichtgewichtige relationale Datenbank (SQL92) mit ACID-Eigenschaften (Atomizität, Konsistenz, Isolation, Dauerhaftigkeit), die bei Verwendung in der Applikation eingebettet wird. Da SQLite keine Authentifizierung über Benutzer oder Rollen unterstützt, kann ADF Mobile den notwendigen Schutz der Daten über eine Verschlüsselung (Encryption) gewährleisten.

Die Verbindung zur lokalen Datenbank wird programmatisch über die Methode *getConnection()* von *java.sql.Connection* hergestellt. Existiert die angegebene Datenbank noch nicht, wird sie beim Connect automatisch angelegt. Danach können beliebige SQL-Skripte aufgerufen werden. Alternativ kann die Datenbank in das Installationspaket der Applikation gepackt werden.

Nutzung nativer Gerätefunktionen

ADF Mobile stellt in der Entwicklungsumgebung des JDeveloper ein generisches Data Control mit Gerätefunktionen zur Verfügung (S. Abb. 2). Über das *Device Data Control* kann das Java API des Frameworks *PhoneGap* auf komfortable Weise durch den Entwickler genutzt werden.

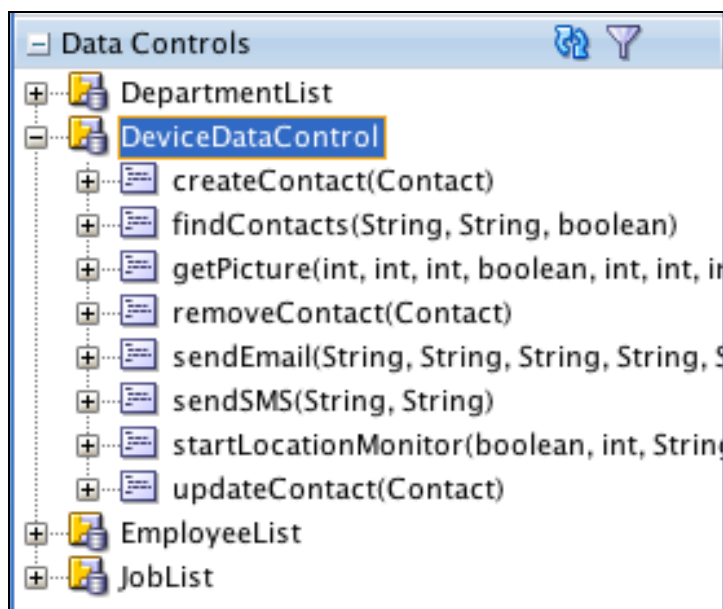


Abb. 2: Data Control für native Gerätefunktionen

Das Data Control für native Gerätefunktionen bietet folgende Möglichkeiten:

- Anlegen, Ändern und Suchen von Kontakten im Adressbuch.
- Senden und Empfangen von Mails
- Senden und Empfangen von SMS
- Bestimmen des Standorts (Geolocation)
- Auswahl von Bildern im Filesystem
- Aufnehmen von Bildern mit der Kamera

Kontaktadresse:

Dr. Jürgen Menge
Oracle B.V. & Co. KG
Riesstr. 25
D-80992 München

Telefon: +49 (0) 89-1430 2239
Fax: +49 (0) 89-1430 2150
E-Mail juergen.menge@oracle.com
Internet: www.oracle.com