

Erstellen einer Mobilen Applikation mit PhoneGap und ADFmobile

**Wolfgang Nast
MT AG
Ratingen**

Schlüsselworte

Java Mobile PhoneGap ADFmobile RESTful JAX RS XML

Einleitung

Die Erstellung von Apps für die Plattformen Android und iOS wird zurzeit immer wichtiger, um seine Webanwendungen auch für Smartphones und Tablets ansprechend umzusetzen. Hier ist die Entwicklung von "nativen" Apps recht aufwendig, da jede Plattform eine eigene Sprache hat, die sich schnell weiterentwickelt. Hier gibt es auch den plattformübergreifenden Ansatz von Java mit PhoneGap. Hier werden die beiden Implementierungen PhoneGap und ADFmobile vorgestellt.

Überblick

Es wird davon ausgegangen, dass eine bestehende Webseite als App umgesetzt werden soll. Damit ergibt sich folgender Ablauf:

1. Erweitern der Seite um RESTful Services.
2. PhoneGap App für eine Plattform umsetzen.
 - 2.a Die einzelnen Seiten umsetzen.
 - 2.b Die Seiten mit den RESTful Services verbinden.
3. PhoneGap App für die nächste Plattform umsetzen.
 - 3.a Die einzelnen Seiten anpassen.
 - 3.b Die Anbindung der Seiten überprüfen.
4. Tests der Apps auf den Geräten.

Der Anfang

Zuerst muss Überlegt werden, welche Fähigkeiten einer Webseite auch auf der App zur Verfügung gestellt werden sollen. Normalerweise wird der Umfang reduziert. Umfangreiche Seiten sollten in mehrere Seiten zerlegt werden und gegebenenfalls einen festen Ablauf in die aufgeteilten Seiten bringen.

Die Beispiel Webseite

Da keine reale Webseite existiert, wird hier eine einfache Webseite erstellt. Die Aufgabe der Website ist sehr einfach gehalten. Es wird eine Kalender Applikation, mit der Termine verwaltet werden. Wichtig ist hier einfach und verständlich, vor schön und praktisch. Deshalb werden die Termine nur einfach chronologisch verwaltet. Auch gibt es keine Benutzerverwaltung, es ist ein Kalender, der allen gehört!

Bei der weiteren Auswahl der Vorgaben. Es wird die RESTful Service verwendet, um die Termine abrufen, anlegen, ändern und löschen zu können. Dazu wird eine Rest-Seite angelegt, wo alle geforderten Methoden abgebildet werden. Auf der selben Seite, können für eine einfache Anzeige, auch noch die HTML-Darstellung für die Webseite untergebracht werden. Die Daten sollten in einer Datenbank gehalten werden. Hier kann entweder JDBC direkt verwendet werden oder ein OR-Mapper wie EclipseLink verwendet werden.

Bei der Auswahl der Implementierung sind folgende Konstellationen wahrscheinlich GlassFish und PostgerSQL oder MySQL mit PhoneGap oder bei ADFmobile OracleDB und WebLogic. Es können natürlich auch ganz

andere Datenbanken und Webserver verwendet werden. Die kleinste Lösung, die hier gewählt wird ist nicht für die Produktion geeignet. Als DB eine H2 und als RESTful-Server Jersey2.2 mit JDK.

PhoneGap für Android

Mit Eclipse wird jetzt das Projekt angelegt. Nach dem alle Einstellungen für das Projekt vorgenommen wurden, kann jetzt mit den Seiten der App begonnen werden. Hier sind Seiten für die Übersicht mit allen Terminen, Termin anzeigen, Termin anlegen oder ändern und Termin löschen zu implementieren.

Der PageFlow ist recht leicht zu beschreiben:

1. Als Startseite bietet sich die Übersicht mit allen Terminen an.
2. Von hier kann durch auswählen eines Termins auf die bearbeiten Seite wechseln.
3. Von der bearbeiten Seite kann das löschen oder speichern anstoßen und kehrt zur Übersicht zurück.
4. Von der Übersicht kann man auch einen neuen Eintrag anlegen, der dann in der bearbeiten Seite angezeigt wird.
- 4.a Von der bearbeiten Seite kann das Speichern oder verwerfen des neuen Termins anstoßen und kehrt zur Übersicht zurück.

Nach dem die Seiten und der PageFlow(Verlinkung) getestet ist kann der Aufruf an den Server eingebaut werden.

PhoneGap für iOS

Mit Eclipse wird jetzt das Projekt angelegt. Nach dem alle Einstellungen für das Projekt vorgenommen wurden, kann jetzt mit den Seiten der App begonnen werden. Hier sind Seiten für die Übersicht mit allen Terminen, Termin anzeigen, Termin anlegen oder ändern und Termin löschen zu implementieren. Hier sind die Seiten an die neue Plattform anzupassen.

Der PageFlow ist recht leicht zu beschreiben:

1. Als Startseite bietet sich die Übersicht mit allen Terminen an.
2. Von hier kann durch auswählen eines Termins auf die bearbeiten Seite wechseln.
3. Von der bearbeiten Seite kann das löschen oder speichern anstoßen und kehrt zur Übersicht zurück.
4. Von der Übersicht kann man auch einen neuen Eintrag anlegen, der dann in der bearbeiten Seite angezeigt wird.
- 4.a Von der bearbeiten Seite kann das Speichern oder verwerfen des neuen Termins anstoßen und kehrt zur Übersicht zurück.

Der Aufruf an den Server kann direkt von der anderen Plattform übernommen werden.

Tests auf realen Geräten

Hier kommt eine der größeren Herausforderungen. Da die Android Geräte eigentlich nur über den PlayStore mit Software geladen werden, muss man hier die Sonderlösung für Entwickler wählen. Die Anwendung kommt dann über einen lokalen Store in das Gerät. Hier gibt es viele kleine Überraschungen, bis die App endlich auf dem Gerät angekommen ist.

Auch bei iOS gibt es nur den AppStore. Auch hier kann man einen lokalen Store erstellen. Und wieder kann man viele neue kleine Überraschungen kennen lernen, bis man es endlich geschafft hat die App auf seinem Gerät zu haben.

ADFmobile für iOS

Mit dem JDeveloper wird jetzt das Projekt angelegt. Nach dem alle Einstellungen für das Projekt vorgenommen wurden, kann jetzt mit den Seiten der App begonnen werden. Hier sind Seiten für die Übersicht mit allen

Terminen, Termin anzeigen, Termin anlegen oder ändern und Termin löschen zu implementieren.

Der PageFlow ist recht leicht zu beschreiben:

1. Als Startseite bietet sich die Übersicht mit allen Terminen an.
2. Von hier kann durch auswählen eines Termins auf die bearbeiten Seite wechseln.
3. Von der bearbeiten Seite kann das löschen oder speichern anstoßen und kehrt zur Übersicht zurück.
4. Von der Übersicht kann man auch einen neuen Eintrag anlegen, der dann in der bearbeiten Seite angezeigt wird.
- 4.a Von der bearbeiten Seite kann das Speichern oder verwerfen des neuen Termins anstoßen und kehrt zur Übersicht zurück.

Wie zu erwarten haben sich nicht die Aufgaben geändert, sondern nur die Tools, mit denen man zum gleichen Ziel kommt.

Auch hier kann man den Aufruf zum Server aus den vorherigen Umsetzungen übernehmen.

ADFmobile für Android

Mit dem JDeveloper wird jetzt das Projekt angelegt. Nach dem alle Einstellungen für das Projekt vorgenommen wurden, kann jetzt mit den Seiten der App begonnen werden. Hier sind Seiten für die Übersicht mit allen Terminen, Termin anzeigen, Termin anlegen oder ändern und Termin löschen zu implementieren. Hier sind die Seiten an die neue Plattform anzupassen.

Der PageFlow ist recht leicht zu beschreiben:

1. Als Startseite bietet sich die Übersicht mit allen Terminen an.
2. Von hier kann durch auswählen eines Termins auf die bearbeiten Seite wechseln.
3. Von der bearbeiten Seite kann das löschen oder speichern anstoßen und kehrt zur Übersicht zurück.
4. Von der Übersicht kann man auch einen neuen Eintrag anlegen, der dann in der bearbeiten Seite angezeigt wird.
- 4.a Von der bearbeiten Seite kann das Speichern oder verwerfen des neuen Termins anstoßen und kehrt zur Übersicht zurück.

Der Aufruf an den Server kann direkt von der anderen Plattform übernommen werden.

Tests auf realen Geräten Teil 2

Hier sollte einem die Erfahrung aus der Implementierung mit PhoneGap die meisten Probleme umschiffen lassen.

Je mehr unterschiedliche Geräte man Testet umso mehr kleine Anpassungen kommen dazu.

Zum Schluss

Aus der kleinen App lässt sich mit einiger Arbeit und persönlichen Vorlieben, eine umfangreichere App erstellen. Hier sind besonders die Möglichkeiten eines mobilen Gerätes zu betrachten.

Kontaktadresse:

Wolfgang Nast
MT AG
Balcke-Dürr-Allee 9
D-40882 Ratingen

Telefon: +49 (0) 2102 30961-0
Fax: +49 (0) 2102 30961- 101
E-Mail: Wolfgang.Nast@mt-ag.com
Internet: www.mt-ag.com