

# Mobile Möglichkeiten

Wer heute eine mobile Webseite erstellen möchte, wird vor eine Vielzahl von Entscheidungen gestellt: Kostenlos oder kostenpflichtig? Entwicklung auf Code-Ebene oder per WYSIWYG? Serverseitige oder clientseitige Lösungen? Mittlerweile gibt es eine Menge unterschiedlicher Lösungsansätze, von denen jeder seine eigenen Vor- und Nachteile mit sich bringt.

Die optimale Lösung hängt in der Praxis davon ab, welche Anforderungen Sie an Ihre mobile Webseite stellen - und wie viel Zeit und Budget für das Projekt zur Verfügung stehen. Wie so oft gilt auch hier: je mehr Aufwand, desto größer (in der Regel) der Nutzen. So existieren Lösungen, um ohne große Vorkenntnisse und teilweise sogar kostenlos innerhalb von fünf Minuten mit einer simplen mobilen Website online gehen zu können. Andere Lösungen erfordern grundlegende Erfahrungen im Bereich der Web-Programmierung und können Wochen oder sogar Monate an Entwicklungszeit dauern, um sich angemessen zu präsentieren.

Vier grundsätzliche Möglichkeiten stellt Ihnen dieser Artikels vor: die Nutzung von WYSIWYG-ähnlichen Editoren; clientseitige Lösungen, die auf JavaScript basieren; serverseitige Lösungen, die eine Gerätedatenbank als Basis nutzen; sowie der Mix aus den beiden letztgenannten. Teilweise können Sie dabei Inhalte aus bereits bestehenden Seiten übernehmen, in einigen Fällen müssen Sie die mobile Webseite von Grund auf neu aufbauen.

## Wie von Zauberhand

Wenn Sie weder viel Zeit noch Geld in eine mobile Seite investieren möchten, eignen sich die zahlreichen „Wizard“-Anbieter, mit deren Service Sie ganz im Stile von WYSIWYG eine simple mobile Webseite „zusammenklicken“ können. In der Regel läuft es dabei so,

dass Sie einen Link zu einem RSS-Feed, einer Facebook-Fanpage, einem Twitter-Profil oder ähnliche Dienste in einer Eingabemaske eintragen. Der Wizard holt sich die entsprechenden Informationen von der angegebenen Datenquelle, bietet anschließend die Möglichkeit, z. B. den Seitentitel, Beschriftung von Menüpunkten oder die Farben zu ändern, und speichert anschließend die gewählten Einstellungen in der Datenbank des Anbieters. Einige Wizards erlauben Ihnen auch, eigene Seiten und Inhalte anzulegen, meist ist diese Möglichkeit allerdings auf wenige Seiten und Inhalte beschränkt. Wer eine komplexe Lösung erstellen möchte, sucht hier also vergeblich.

**Geringer Aufwand, geringer Nutzen:** Die Einfachheit solcher Dienste ist auch gleichzeitig ihre Schwäche: sollen erweiterte Funktionen wie ein Kontaktformular eingebunden werden, stoßen viele dieser Dienste bereits an ihre Grenzen. Auch was die Kompatibilität mit verschiedenen Endgeräten angeht, sind die Möglichkeiten eher beschränkt. Oft wird nur eine iPhone- oder eine Webkit-Version aus den Daten erzeugt. Einfachere Geräte, die etwa keine Unterstützung für JavaScript bieten, bleiben dabei schnell außen vor. Auf der Kostenseite bieten alle Anbieter eine kostenlose Basisversion ihres Dienstes an. Wenn Sie etwa die Zahl der Unterseiten erweitern oder die möglichen Zugriffe pro Monat steigern möchten, sind Sie mit einer günstigen Monatspauschale ab ca. 5 Euro dabei.

**Wirenode:** „The mobile website creator“ heißt es, loggt sich ein Benutzer in seinen Wirenode-Account ein. Wirenode bietet ein simples Interface, mit dem Sie sich eine mobile Seite zurechtclicken können. Sie wählen aus verschiedenen Seitentemplates aus und erstellen anschließend über einen Richtext-Editor Seiteninhalte. Auch verschiedene Widgets wie z. B. die Einbindung der eigenen Twitter-Timeline werden angeboten.

In der Basisversion ist Wirenode kostenlos und erlaubt es Ihnen, eine Site zu erstellen. Diese ist unter <http://benutzername.wirenode.mobi> erreichbar. Bei den vier kostenpflichtigen Versionen (zwischen



4,90 Euro und 190 Euro) können Sie mehrere Sites pro Account angelegen und eine eigene Domain verwenden.

**Google Sites:** Auch Google bietet mit Google Sites seit kurzem die Möglichkeit, mobil optimierte Websites zu erstellen. „Mobilize your business“ heißt es auf der entsprechenden Produktseite. Entwickler können dabei aus einer Reihe von mobilen Templates für verschiedene Anlässe wählen (etwa „Restaurant“ oder „Local Businesses“) und über ein intuitives WYSIWYG-Interface Seiten erstellen, Inhalte anlegen und über einen Richttext-Editor formatieren. Über sogenannte „Gadgets“ bietet auch Google Sites die Möglichkeit, einen Twitter-Stream, das aktuelle Wetter oder eine Google Map in die Seite zu integrieren. Dabei ist jedoch Vorsicht geboten, denn auf Grund fehlender Geräteadaption bekommen auch alte Geräte ohne JavaScript-Unterstützung die Gadgets ausgeliefert, wodurch es zu Fehlern in der Darstellung kommen kann.

**Onbile:** Ein sehr aufgeräumtes Interface, im positiven wie im negativen Sinne, bietet onbile seinen Nutzern. Positiv, da wirklich jeder, der einen Browser bedienen kann, ohne Probleme innerhalb von nur fünf Minuten mit einer mobilen Website an den Start gehen kann. Negativ, da die Möglichkeiten für Benutzer hier am restriktivsten sind. Die Anzahl der Unterseiten (vier inkl. Startseite) ist vorgegeben und kann derzeit weder erweitert werden, noch können Seiten entfernt werden. Als Benutzer dürfen Sie eines der rund 15 vorgegebenen Templates wählen, anschließend noch farbliche Anpassungen vornehmen und ein Logo hochladen. Alles in allem eine nette Spielerei, aber selbst für sehr kleine Unternehmen kaum eine ernsthafte Alternative, auch wenn die optische Gestaltung der Bedienoberfläche bei Onbile wohl für den Testsieg reichen würde. Auch die Tatsache, dass der Dienst vollständig kostenlos ist, ändert nichts mehr daran, dass Onbile höchstens dazu dient, um das Auge mit einem hübschen Interface zu erfreuen, als ernsthaft mobile Websites zu erstellen.

### Clientseitige JavaScript Frameworks

In der Entwickler-Community stoßen sie auf ein großes Interesse und genießen einen hohen Stellenwert: mobile JavaScript Frameworks wie jQuery Mobile, SenchaTouch oder zepto.js. Spätestens seit dem Release des iPhones, das erstmals verdeutlichte, was auf mobilen Geräten im Browser möglich sein kann, gibt es eine ganze Reihe an Entwicklern, die sich auf die neue Touch-Geräte Generation spezialisiert haben und entsprechende Frameworks bereitstellen. Diesen Frameworks kommt zu Gute, dass sie auf



www.wirenode.com



www.google.com/sites/help/intl/en/mobile-landing-pages/mlpb.html



www.onbile.com

HTML5 aufbauen und damit für die Zukunft gerüstet sind. Diese Zukunftsfähigkeit hat wiederum zur Folge, dass deren „Vergangenheitstauglichkeit“ leidet, also dass hier Geräte, denen es an Unterstützung der recht neuen Standards mangelt, nicht ausreichend berücksichtigt werden. Vor allem ältere Geräte ohne JavaScript-Unterstützung, die gerade in so wichtigen Entwicklungsländern wie Indien, Brasilien oder der Türkei noch die überwiegende Mehrheit darstellen, können von den JavaScript-lastigen Frameworks nicht hinreichend bedient werden. Zwar ist es mit jQuery Mobile oder zepto.js noch möglich, durch „progressive Enhancement“, intelligentes Markup und klugen Einsatz von Stylesheets (bspw. durch Media Queries) zumindest eine mobile Seite mit Grundfunktionalitäten abzubilden, das ansonsten sehr ausgereifte SenchaTouch versagt hier jedoch auf ganzer Linie.

**Gut, für gewisse Zielgruppen:** So schön, einfach und cross-browser-kompatibel derartige Frameworks auch angepriesen werden, so sind sie im Hinblick auf mögliche Zielgruppen durchaus kritisch zu betrachten und sollten nicht zu vorschnell eingesetzt werden. In Indien beträgt der Marktanteil des iPhones bspw. gerade einmal ca. 5% und auch Android-Geräte sind dort mit einem Marktanteil von unter 15% nicht wirklich breit vertreten. Wer also für einen Kunden aus Indien, dem nahen Osten oder Südamerika arbeitet, der sollte sich gut überlegen, ob er mit SenchaTouch nicht vielleicht an der Zielgruppe des Kunden vorbei entwickelt.

Doch genug zu den Nachteilen, selbstverständlich bieten diese Frameworks Entwicklern von mobilen Websites eine ganze Reihe echter Vorteile, die den Einsatz wirklich reizvoll und lohnenswert machen. Ein eben schon erwähnter Vorteil ist die Zukunftstauglichkeit: HTML5 eignet sich auf Grund seiner Eigenschaften als generische Markup-Sprache hervorragend für die Entwicklung mobiler Websites. Hinzu kommt, dass Entwicklern ein ganzes Arsenal an Tools zur Verfügung steht, die bereits auf den allermeisten Geräten der aktuellen und kommenden Generationen funktionieren. Ob Buttons, die in ihrem Verhalten an die Buttons des Geräte-Betriebssystems angelehnt sind, oder stilvolle Slide-Effekte beim Aufruf einer neuen Unterseite. Um all diese visuellen Schmankerl muss sich der Entwickler keine Gedanken mehr machen. Zusammenfassend kann festgehalten werden, dass JavaScript-Frameworks eine einfache und gute Möglichkeit sind, um schnell Ergebnisse zu erzielen. Ohne auf eine serverseitige Sprache zurückzugreifen, ist es dabei jedoch nicht ohne weiteres möglich, die Inhalte einer bereits bestehenden Seite im Nachhinein zu übernehmen.

**jQuery Mobile:** jQuery Mobile basiert vollständig auf dem verbreiteten JavaScript-Framework jQuery, nutzt also dessen bewährten Funktionsumfang und gibt Entwicklern somit ein Tool an die Hand, um innerhalb kurzer Zeit ansehnliche, bedienbare mobile Websites zu erstellen. Ausgangspunkt für jQuery Mobile ist sauberes HTML5 Markup, das über vordefinierte data-role-Attribute semantisch erweitert wird. So können Sie einen Seitenbereich über `<div data-`



www.jquerymobile.com



www.sencha.com

role="header" data-position="fixed"></div>  
einfach zu einem feststehenden Seitenheader deklarieren. jQuery Mobile bietet eine Reihe solcher Zusatzfunktionen, die über unaufdringliches JavaScript realisiert werden. Wie der große Bruder jQuery steht jQuery Mobile unter GPL 2.0 und MIT zur freien Verfügung.

**Sencha Touch:** Der Name impliziert es bereits: die Ausrichtung von Sencha Touch liegt ganz klar auf Geräten mit Touch-Bedienung. Da dies in der Hauptsache neuere Geräte sind, ist es auch nur wenig verwunderlich, dass das komplette Framework auf JavaScript aufbaut und für die Erstellung einer mobilen Website mit Sencha Touch nicht eine einzige Zeile HTML5-Markup benötigt wird. Obgleich Sencha von sich behauptet, das erste „HTML5 Mobile Web App Framework“ zu sein. Die gesamte Entwicklung einer mobilen Website bzw. Web App basiert auf JavaScript, für die visuelle Gestaltung kommt CSS zum Einsatz. Kompatibel ist Sencha zudem nur mit Webkit-Browsern (Safari, Chrome, ...), Firefox und Opera Nutzer bleiben außen vor. Trotz des kommerziellen Backgrounds, Sencha Touch wird von der Sencha Inc. entwickelt, steht die Library aktuell noch unter der GPL 3.0 zur freien Verfügung. Dies gilt sogar für kommerzielle Projekte.

**zepto.js:** „The aerogel-weight mobile javascript framework“. Das ausgesprochene Ziel von zepto.js ist es, eine JavaScript Library anzubieten, die nur 2-5 Kilobyte an Größe mitbringt und dem Entwickler dennoch alle Aufgaben abnimmt, für die sonst mühsame Fleißarbeit

fällig würde. Die Syntax von Zepto.js ist stark an jQuery angelehnt. Entwickler, die bereits Erfahrungen mit jQuery oder anderen Frameworks gesammelt haben, werden sich schnell damit zurecht finden. Das Ziel, die Größe von 5 KB nicht zu überschreiten, erreicht zepto.js dadurch, dass mobile Randgruppenbrowser wie der Internet Explorer aber auch Opera Mobile derzeit außen vor gelassen werden. So funktioniert zepto.js, was sich zum aktuellen Zeitpunkt, ähnlich wie auch jQuery Mobile, noch in der Beta befindet, am besten auf Webkit-Browsern (iPhone, iPad, Android, ...), aber auch im Firefox Mobile.

### Serverseitige Verarbeitung von Anfragen

Eine weitere Möglichkeit, um Websites für mobile Geräte zu optimieren, ist die serverseitige Erzeugung von mobil optimierten Seiten. Dies passiert in der Regel durch die Ermittlung des aktuellen Client-Geräts durch den vom Browser gesendeten User Agent Header. Um auf ein Endgerät angemessen reagieren und eine tatsächlich optimierte Seite ausgeben zu können, sind bei dieser Methode Kenntnisse über die verwendete Kombination aus Gerät, Betriebssystem und Browser notwendig. Zu diesem Zweck gibt es Datenbanken mit ausführlichen Informationen über diverse am Markt erhältliche Geräte, die Angaben über die Bildschirmgröße, unterstützte HTML-Formate oder den Gerätetyp (z. B. Desktop, Mobile, Tablet, ...) enthalten. Diese Methode hat jedoch zwei Nachteile: zum einen prüft sie nur auf theoretische, nicht auf tatsächliche Geräte-Features. Sie kann also ermitteln, dass ein Gerät JavaScript-Unterstützung



www.zeptojs.com



www.fitml.com

bietet, nicht jedoch, ob der Benutzer JavaScript-Unterstützung vielleicht clientseitig deaktiviert hat. Zum anderen ist das Verfahren insofern unzuverlässig, als dass der User Agent in den Browser-Einstellungen verschleiert oder geändert werden kann. Außerdem kann es zu Problemen kommen, wenn der User Agent noch zu neu ist, um von der Datenbank erkannt zu werden. Zumindest das erste Problem können Sie durch cleveres Markup umgehen, indem Sie z. B. gleichzeitig eine JavaScript- und eine Non-JavaScript-Version der Website ausliefern. Beim zweiten Problem können Sie lediglich auf schnelle Updates der Anbieter der entsprechenden Datenbank hoffen. Der große Vorteil bei der serverseitigen Erzeugung einer Seite ist, dass Sie ein Dokument schon vor der Auslieferung auf das Endgerät auf die möglicherweise beste Darstellung hin optimieren können. Große JavaScripts, die viel Ladezeit benötigen, müssen so bspw. gar nicht erst an den Browser ausgeliefert werden. Bilder können Sie bereits auf dem Server auf die maximale Höhe und Breite des Displays skalieren. Das spart Akkuleistung, Übertragungskosten und lädt zudem die Seite schneller.

**WURFL:** Pionier auf dem Gebiet der Gerätedatenbanken ist das Wireless Universal Resource File, kurz WURFL. Es enthält viele nützliche Geräteinformationen, die von der Open Source Community bis heute intensiv gepflegt werden. Im Kern besteht WURFL aus einem unheimlich großen XML-File mit Angaben über die Geräte. Mittlerweile gibt es aber Implementierungen in sämtlichen relevanten Sprachen wie PHP, Java, .NET, Python oder Perl. Durch die freiwillige Pflege der Open Source Community ist WURFL nicht immer auf dem neusten Stand und enthält zudem zu einigen Geräten fehlerhafte Angaben, die erst spät behoben werden. Ein kleines Manko, dafür erhalten Sie mit WURFL eine ansonsten solide Datenbasis, die für kommerzielle Projekte einmalig mit 1500 Dollar zu Buche schlägt. <http://wurfl.sf.net/>


**DeviceAtlas:** Genau wie auch WURFL besteht der DeviceAtlas aus einer Datenbank mit unzähligen Informationen über diverse Endgeräte. Anders als WURFL wird der DeviceAtlas jedoch kommerziell vertrieben. Eine Lizenz mit 1 Mio Datenbankabfragen ist für 399 Dollar im Jahr erhältlich. Für diesen stolzen Preis erfährt der DeviceAtlas jedoch Unterstützung von großen Unternehmen aus dem mobilen Geschäft: Nokia, Sony Ericsson oder auch Vodafone werden als Partner aufgeführt. Entwickler, die in ihren Projekten gern kommerzielle Unterstützung erhalten möchten, dürften sich beim DeviceAtlas gut aufgehoben fühlen. <http://www.deviceatlas.com/>

### Das Beste aus beiden Welten

Für eine gute User Experience im Bereich der mobilen Webseitenstellung hat sich in vielen Fällen ein Mittelweg aus den beiden letztgenannten Technologien bewährt. Eine serverseitige Erkennung des Zielgeräts und anschließend eine clientseitige progressive Erweiterung der ausgelieferten Seite, je nachdem, welche Features ein Gerät tatsächlich unterstützt. Ältere Geräte erhalten dabei eine uneingeschränkt bedienbare mobile Seite, mit allen Inhalten, serverseitig skalierten Bildern und ohne JavaScript. Neue Geräte mit JavaScript-Unterstützung und hoch aufgelöstem Touch-Display erhalten eine aufgehübschte Nutzerführung mit Bildern in hoher Auflösung, Effekten wie CSS3 Transformationen und jQuery-Mobile-Erweiterungen.

**fitml.com:** Diesen Weg verfolgt Sevenval mit seinem Cloud-Service fitml.com. Mit einer eigenen, FITML genannten Markup-Sprache können Entwickler XML-Dokumente erstellen, die je nach Gerät zum Teil völlig unterschiedlich dargestellt werden. So bekommen technisch ausgereifte Geräte ansprechende und hochwertige HTML5-Seiten ausgeliefert, die mit JavaScript und CSS3 angereichert werden, ohne dabei jedoch alte Geräte ohne HTML5 Support auszuschließen. Der Vorteil dieser Lösung ist die Markup-Abstraktion, so dass Sie nur FITML verwenden müssen, die Cloud-Plattform kümmert sich anschließend um die optimale Ausgabe auf allen Geräten. Dabei werden sowohl serverseitige Parameter als Ausgangsbasis, als auch zusätzliche clientseitige Abfragen zur weiteren Fein-Optimierung verwendet. Kleinere Seiten können dort kostenlos entwickelt werden. Für größere Auftritte bietet die Plattform abhängig von der Anzahl der monatlichen Page Impressions unterschiedliche Preismodelle. (Hinweis: der Autor dieses Textes arbeitet für Sevenval, das Unternehmen hinter fitml.com).

Autor » Links » Bild- und Quellennachweis



**Manuel Bieh**

Als Software-Entwickler kümmert sich Manuel Bieh bei der Sevenval GmbH um die Weiterentwicklung der Plattform fitml.com. Dafür steht er als Developer Relations Manager im direkten Kontakt mit der Community. Gleichzeitig unterstützt er auch Entwickler bei ihren ersten Gehversuchen mit FITML.

Link-Code      @@  
 Twitter-Hashlag      #wsm1111-@@  
 Twitter-Account      manuelbieh