



# Titanium Mobile

## Apps für iPhone und Android

- > So richten Sie Titanium Mobile optimal ein
- > So entwickeln Sie Ihre eigenen Apps plattformübergreifend
- > So bringen Sie Ihre App in die Stores

Norbert Usadel

**Titanium Mobile**  
**Apps für iPhone und Android**

Norbert Usadel



# **Titanium Mobile**

## **Apps für iPhone und Android**

**Mit 283 Abbildungen**

## **Bibliografische Information der Deutschen Bibliothek**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Alle Angaben in diesem Buch wurden vom Autor mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. Der Verlag und der Autor sehen sich deshalb gezwungen, darauf hinzuweisen, dass sie weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen können. Für die Mitteilung etwaiger Fehler sind Verlag und Autor jederzeit dankbar. Internetadressen oder Versionsnummern stellen den bei Redaktionsschluss verfügbaren Informationsstand dar. Verlag und Autor übernehmen keinerlei Verantwortung oder Haftung für Veränderungen, die sich aus nicht von ihnen zu vertretenden Umständen ergeben. Evtl. beigefügte oder zum Download angebotene Dateien und Informationen dienen ausschließlich der nicht gewerblichen Nutzung. Eine gewerbliche Nutzung ist nur mit Zustimmung des Lizenzinhabers möglich.

**© 2012 Franzis Verlag GmbH, 85540 Haar bei München**

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträgern oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt.

Die meisten Produktbezeichnungen von Hard- und Software sowie Firmennamen und Firmenlogos, die in diesem Werk genannt werden, sind in der Regel gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Der Verlag folgt bei den Produktbezeichnungen im Wesentlichen den Schreibweisen der Hersteller.

**Lektorat:** Anton Schmid

**Satz:** DTP-Satz A. Kugge, München

**art & design:** [www.ideehoch2.de](http://www.ideehoch2.de)

**Druck:** GGP Media GmbH, Pößneck

Printed in Germany

**ISBN 978-3-645-60160-3**

# Über dieses Buch


Dieses Buch vermittelt Ihnen den Einstieg in das Titanium-Framework. Sie erstellen in diesem Framework einen Code für die Android-Plattform und für das iPhone.

JavaScript-Kenntnisse werden vorausgesetzt, weil der Code eben in JavaScript geschrieben wird. Für Entwickler, die JavaScript nicht kennen, gibt es im Anhang dieses Buches einen Schnelleinstieg in diese Programmiersprache.

HTML und CSS werden zur Quelltext-Eingabe ebenfalls benutzt. Im Anhang findet sich auch dazu jeweils ein Abschnitt um die Basics zu erlernen. Ich hoffe, so können alle auf einen Stand gebracht werden, um einfache Apps mit Titanium Mobile zu programmieren. Im Anhang finden Sie zusätzlich eine Übersicht über kleine Helfer, die Ihnen das Programmieren von Apps leichter machen können. Während ich schrieb, benutzte ich ein MacBook Pro von Apple. Es kann zu kleinen Abweichungen in der Beschreibung der Entwicklungsumgebung kommen, wenn Sie Titanium auf einem Windows-Gerät installieren.

## Aufbau


In den ersten Kapiteln wird Ihnen der grundlegende Stoff über Titanium vermittelt: Wie Titanium installiert wird. Wo Sie was finden. Sie erlernen durch Code-Beispiele den Umgang mit Ihrer neuen Entwicklungsumgebung und was Sie mit ihr anstellen können. Danach wird der erlernte Stoff durch Workshops vertieft. Die Workshops, den Quelltext und eine nützliche Link-Liste zum Titanium-Framework können Sie sich kostenlos auf der Seite [www.appzitty.de/download](http://www.appzitty.de/download) herunterladen. Das ist die Seite des Autors, auf der Sie auch im Blog oder per e-mail Fragen stellen können. Es wird Ihnen dort mit Sicherheit weitergeholfen. Das Buch geht sozusagen dort weiter.



***Thanks for choosing  
Appcelerator!***

**Delivers the best experience**  
3,600 fully native mobile APIs and counting

**Across all major platforms**



**Using an open technology and ecosystem**  
Join 1.5 Million Web Developers

Die Appcelerator Titanium Development Platform stellt sich vor. Tauchen Sie in das Framework ein.

## Zielgruppe

Das Buch richtet sich an alle, die mit wenig Zeitaufwand Apps entwickeln wollen. In der Regel sind Sie Web-Entwickler, die JavaScript-Kenntnisse mitbringen, aber nicht Objective-C erlernen wollen oder nicht nur eine Plattform bedienen möchten.

## Über den Autor

Norbert Usadel lebt und schreibt im Ruhrgebiet. Studierte Architektur und Kunstgeschichte. Machte eine Ausbildung zum Programmierer und kam auf den Apple. Der Apple ließ ihn nicht mehr los.

Er arbeitet seit fast 20 Jahren selbstständig im Apple-Bereich. Der Autor schrieb in dieser Zeit Artikel über Podcasts, Datenverschlüsselung und Garageband für die Zeitschrift MacPraxis. Er ist Autor der Bücher »Inside iPod«, »Mit Xcode 4.2 und Objective-C fürs iPhone programmieren« und »App-Programmierung für Einsteiger«.

Der Autor schulte Menschen über die Themen SAP, Cobol, Filemaker, Shopsysteme, iPods, iPhones, iTunes, strukturierte Programmierung, Pagemaker, Photoshop und App-Programmierung.

Er entwickelt Apps und übernimmt Projekt-Leitungen für die App-Programmierung.

## Webseite zum Buch

Falls Sie Fragen und Anregungen haben, besuchen Sie meine Webseite [www.appzitty.de](http://www.appzitty.de). Dort dreht sich alles um die App-Programmierung. Im Bereich [www.appzitty.de/download](http://www.appzitty.de/download) liegen die einzelnen Titanium-Projekte aus diesem Buch für Sie bereit. Sie können mich unter der E-Mail-Adresse [buch@appzitty.de](mailto:buch@appzitty.de) erreichen. Es würde mich freuen, auf diesem Wege Kontakt mit Ihnen aufnehmen zu dürfen.

## Danksagung

Man ahnt es ja gar nicht, aber es sind viele Leute hinter den Kulissen nötig, um ein Buch möglich zu machen. Ich danke diesen Leuten, auch wenn sie hier nicht namentlich genannt sind. Sie können sich aber gerne angesprochen fühlen, wenn Sie diese Zeilen lesen. Stop. Zwei Leuten muss ich bei diesem Buch doch danken. Das ist zum einen mein Lektor Anton Schmid vom Franzis Verlag und zum anderen Dieter Reinelt, der mir bei den Programmierbeispielen zur Seite stand.



# Inhaltsverzeichnis

<b>1</b>	<b>Titanium Mobile und die anderen ...</b>	<b>13</b>
1.1	Einordnung .....	13
1.2	Was ist eine Web-App? .....	13
1.3	Was ist eine native App? .....	13
1.4	Der Vergleich .....	14
1.5	Fazit .....	15
1.6	Titanium und die anderen .....	15
1.6.1	PhoneGap .....	15
1.6.2	Rhodes .....	18
1.6.3	Sencha Touch.....	21
1.6.4	Titanium Mobile .....	22
1.7	Der Vergleich der Frameworks .....	24
<b>2</b>	<b>Titanium Mobile laden, installieren und benutzen .....</b>	<b>27</b>
2.1	Die Basis-Installation .....	27
2.1.1	Die Systemvoraussetzungen.....	29
2.1.2	Die Speichervoraussetzungen.....	30
2.2	Titanium Studio installieren .....	30
2.3	Installation von Titanium Mobile .....	35
2.3.1	Installation von Xcode.....	37
2.3.2	Die Installation des Android-SDK .....	40
2.3.3	Android für den Mac .....	40
2.3.4	Android für Windows.....	41
2.4	Der Android SDK Manager.....	44
2.5	BlackBerry .....	52
2.6	Testen, ob alles läuft – Der erste Test .....	59
<b>3</b>	<b>Arbeiten mit Titanium Mobile.....</b>	<b>67</b>
3.1	Wo finden Sie was? .....	67
3.2	Der Aufbau von Titanium Mobile.....	78
3.2.1	Die Menüleisten .....	80
3.2.2	Projekt anlegen und verwalten.....	95
3.2.3	Erscheinungsbild ändern.....	97
3.2.4	Der Debug-Bereich.....	98

3.2.5	Die Views des Debuggers .....	101
3.3	<b>Aptana, der Editor und Debugger.....</b>	<b>102</b>
3.4	<b>KitchenSink .....</b>	<b>105</b>
3.4.1	Anwendungen laufen lassen.....	121
3.4.2	Anwendungen im Android-Emulator schneller laufen lassen (Fastdev).....	124
4	<b>Die Reise durch das Titanium-Framework.....</b>	<b>127</b>
4.1	<b>Windows, Views und Controls .....</b>	<b>128</b>
4.1.1	Windows.....	128
4.1.2	Views.....	129
4.2	<b>Tab-Group und Tabs .....</b>	<b>130</b>
4.3	<b>Das Table-View.....</b>	<b>135</b>
4.4	<b>Ein Table-View-Beispiel .....</b>	<b>141</b>
4.5	<b>Picker .....</b>	<b>146</b>
4.5.1	Date and Time Picker .....	147
4.6	<b>Web-View .....</b>	<b>150</b>
4.7	<b>Netzwerkzugriffe .....</b>	<b>151</b>
4.7.1	JSON, XML.....	153
4.7.2	YQL-Datenbank .....	154
4.8	<b>Geolocation .....</b>	<b>156</b>
4.9	<b>Maps .....</b>	<b>157</b>
4.10	<b>Bilder und die Kamera, Videos und Sound.....</b>	<b>159</b>
4.10.1	Bilder.....	159
4.10.2	Sound.....	161
4.10.3	Videos .....	162
5	<b>Eine SQLite-Datenbank abbilden.....</b>	<b>165</b>
5.1	<b>Das Projekt anlegen .....</b>	<b>165</b>
5.2	<b>Anlegen der Datenbank .....</b>	<b>169</b>
5.3	<b>Import der Datenbank.....</b>	<b>175</b>
5.4	<b>Erzeugen der App .....</b>	<b>178</b>
6	<b>Twitter-App.....</b>	<b>187</b>
7	<b>Der Pizza-Palast.....</b>	<b>201</b>
8	<b>App in den Store .....</b>	<b>235</b>
8.1	<b>App in die Android-Markets.....</b>	<b>235</b>
8.1.1	Erstellung eines Schlüsselpaares.....	236
8.1.2	Überprüfen des Keystores.....	237

8.1.3	Die Android-App bauen .....	238
8.1.4	Die App einstellen.....	239
8.1.5	Richtlinien des Android Market .....	239
8.1.6	Andere Plattformen .....	240
<b>8.2</b>	<b>App in den Apple Store .....</b>	<b>244</b>
8.2.1	Anmelden als Entwickler .....	244
8.2.2	Zertifikate .....	246
8.2.3	iPhone zum Testen benutzen.....	251
8.2.4	App ID erstellen.....	252
8.2.5	iOS Development Provisioning Profile anlegen.....	253
8.2.6	iTunes Connect.....	254
8.2.7	Die App in Titanium bauen .....	256
<b>9</b>	<b>Anhang.....</b>	<b>257</b>
9.1	Wichtige Titanium-Links .....	257
9.2	HTML5 .....	268
9.3	CSS .....	270
9.4	JavaScript .....	272
9.5	Frameworktools .....	274
9.5.1	XUI Framework .....	274
9.5.2	Lawnchair .....	275
9.5.3	Urban Airship .....	276
9.5.4	Mobile Spec .....	276
9.5.5	Sony Ericsson WebSDK .....	276
	<b>Stichwortverzeichnis.....</b>	<b>279</b>



# 1 Titanium Mobile und die anderen ...

In diesem Kapitel erhalten Sie eine Einschätzung, was Sie mit Titanium machen können. Sie erhalten ebenso einen Überblick über die anderen Frameworks, die ähnliches für die App-Programmierung versprechen.

## 1.1 Einordnung

Anfänglich müssen ein paar Begriffe geklärt werden. Sie haben sich ein Buch über einen Cross-Compiler gekauft, um native Apps zu erstellen. Sie möchten einen Code erstellen und zwei Plattformen damit bedienen. Sie wollen dabei darauf verzichten, Objective-C zu erlernen, aber mittels HTML, CSS und JavaScript Apps für das iPhone und die Android-Plattform erstellen. Dann lesen Sie weiter.

Es wird grundsätzlich zwischen Web-Apps und nativen Apps unterschieden. Die Unterschiede zwischen beiden Typen von Apps verschwimmen immer mehr. Es gibt mittlerweile Web-Apps, die Sie nicht mehr von nativen Apps unterscheiden können. Es folgt eine Gegenüberstellung von Web-Apps und nativen Apps.

## 1.2 Was ist eine Web-App?

Eine Web-App ist im Grunde eine Webseite die von einem Browser aufgerufen wird. Der Inhalt der aufgerufenen App-Website kann alles mögliche enthalten.

Sei es ein Bestellformular für Ihre Pizza, den Wetterbericht oder den täglichen Leit-spruch des Tages. Web-Apps sind zum größten Teil kostenlos. Die Web-App wird nicht auf dem Smartphone (bzw. Ihrem System) installiert, sondern über den Browser aufgerufen.

## 1.3 Was ist eine native App?

Native Apps sind auf dem Smartphone direkt installiert und können dessen Ressourcen voll nutzen. Das können z.B. das GPS-Modul, Lautsprecher oder die Kamera sein. Der Quelltext wird vom Compiler in einen binären Code umgesetzt.

Für das iPhone werden diese Apps nur über den App-Store vertrieben. Apple behält sich vor, welche App in den Store kommt. Bei anderen Herstellern, die das Smartphone-Betriebssystem Android nutzen, ist es einfacher, Apps zu veröffentlichen.

Bei den nativen Apps stehen dem Entwickler mehr Möglichkeiten für die Entwicklung zur Verfügung, da durch die Programmierung auf alle Hardwarekomponenten des Smartphones zugegriffen werden kann.

## 1.4 Der Vergleich

### Web-Apps

Vorteile:

- Sie sind einfach zu erstellen. Webdesign reicht für viele Apps aus.
- Die Kosten der Entwicklung sind geringer.
- Die App läuft auf jedem Smartphone in einem Web-Browser.
- Fehler können sofort behoben werden.
- Die Entwicklungszyklen sind schneller.

Nachteile:

- Viele Hardwarefunktionen des Smartphones können nicht genutzt werden.

### Native Apps

Vorteile:

- Sie können alle Komponenten des Smartphones nutzen.
- Die Entwicklungsumgebungen der Smartphones bieten einen hohen Programmierkomfort.

Nachteile:

- Sie sind evtl. auf eine für Sie fremde Hochsprache in der Programmierung angewiesen (C++, Objective-C etc.).
- Sie haben evtl. keinen Einfluss auf Fehlerbehebungen.
- Der Entwicklungszyklus ist langsam, weil Sie keinen Einfluss auf den Testzyklus haben. Dies trifft auf Apps für das iPhone zu, da Apple sich vorbehält, die Apps zu testen, bevor diese in den App-Store gelangen.
- Die Entwicklungskosten sind höher, weil nicht auf gängige Programmier-techniken zurückgegriffen werden kann.

## 1.5 Fazit

Durch Crossplattforming können Sie aus einer Web-App eine native App machen, und haben Zeit und Geld gespart. Sie können jederzeit zweigleisig fahren. Entwickeln Sie erst die Web-App und schieben sie die native App nach.

Entscheiden Sie selbst, wie Sie Ihre App gestalten. Fakt ist, dass Smartphones heutzutage immer online sind, und eine Web-App teilweise als »native Web-App« angesehen werden kann. Die Grenzen verschwimmen. Das kann Ihnen aber auch egal sein, da Sie die Möglichkeit haben, durch einen Quelltext, eine App für die gängigen Smartphone-Plattformen zu entwickeln, ohne auf einen langen Entwicklungszyklus schauen zu müssen.

Benutzen Sie Titanium Mobile, haben Sie die Möglichkeit Apps einfacher, kostengünstiger und schneller auf den Markt zu bringen.

## 1.6 Titanium und die anderen

Titanium ist aber nicht der einzige Cross-Plattform-Compiler. Es existieren noch andere auf dem Markt. Die wichtigsten werden hier kurz vorgestellt.

### 1.6.1 PhoneGap

PhoneGap ist ein Open-Source-Tool, das Sie unter <http://www.phonegap.com/download> downloaden können. Es wurde von Nitobi Software entwickelt. Im Oktober 2011 wurde Nitobi (inkl. PhoneGap) von Adobe aufgekauft. Das Programm soll in die Apache Software Foundation übergeben werden und unter einem anderen Namen weitergeführt werden.



**Bild 1.1:** Die Homepage von PhoneGap

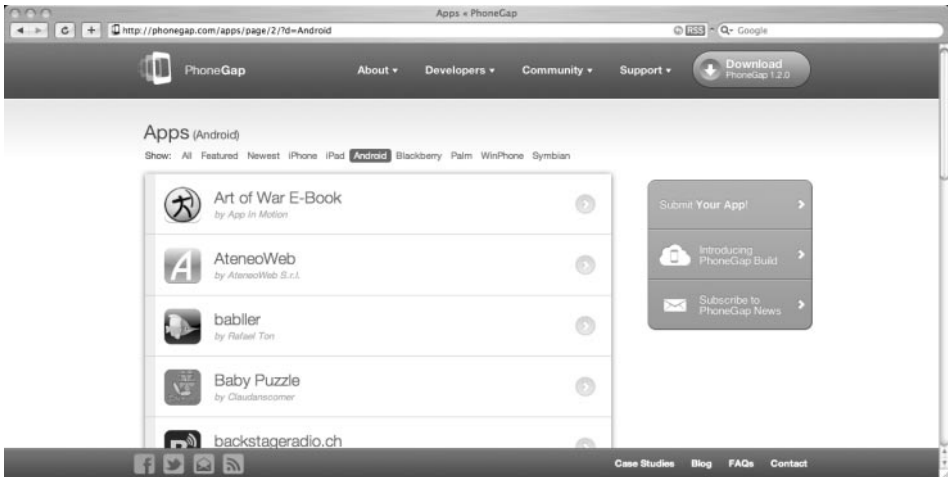
Es unterstützt iPhone, Android, Blackberry, Palm, Windows Mobile und Symbian. Hier haben Sie für Ihre Entwicklung die volle Bandbreite für alle auf dem Markt existierenden Smartphones. Haben Sie einmal eine Web-App in HTML codiert, können Sie mit PhoneGap für nahezu alle Smartphone-Plattformen Apps entwickeln. Die Anwendungslogik wird in JavaScript codiert.

PhoneGap ist ein Wrapper, der es erlaubt, auf Gerätefunktionen wie GPS-Ortung, Adressbuch, Beschleunigungssensoren etc. zuzugreifen.

	iOS iPhone/ iPhone 3G	iOS iPhone 3GS and newer	Android	OS 4.0-4.7	OS 5.x	OS 6.0+	WebOS	WP7	Symbian	Blackberry
ACCELEROMETER	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
CAMERA	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
COMPASS	✗	✓	✓	✗	✗	✗	✗	✓	✗	✓
CONTACTS	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓
FILE	✓	✓	✓	✗	✓	✓	✗	✓	✗	✗
GEOLOCATION	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MEDIA	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗
NETWORK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (ALERT)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
NOTIFICATION (SOUND)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

**Bild 1.2:** Übersicht über die Features von PhoneGap

Dass die Portierung und das Erstellen von Apps mit PhoneGap durchaus funktioniert, finden Sie unter diesem URL: <http://www.phonegap.com/apps>. Dort sehen Sie eine Zusammenstellung von Apps, die mit PhoneGap entwickelt wurden.



**Bild 1.3:** Übersicht der entwickelten Apps auf der PhoneGap-Plattform

Wenn Sie PhoneGap einsetzen, bedenken sie, dass die Anwendung, die Sie kreiert haben, immer noch eine Anwendung ist, die in einem Browserfenster läuft. PhoneGap ist einfach umzusetzen und bedient die meisten Handy-Plattformen. Die Installation ist nicht einfach. Wie es mit PhoneGap weiter geht, bleibt nach der Übernahme durch Adobe abzuwarten, da es nicht klar ist, unter welcher Flagge das Framework laufen wird.

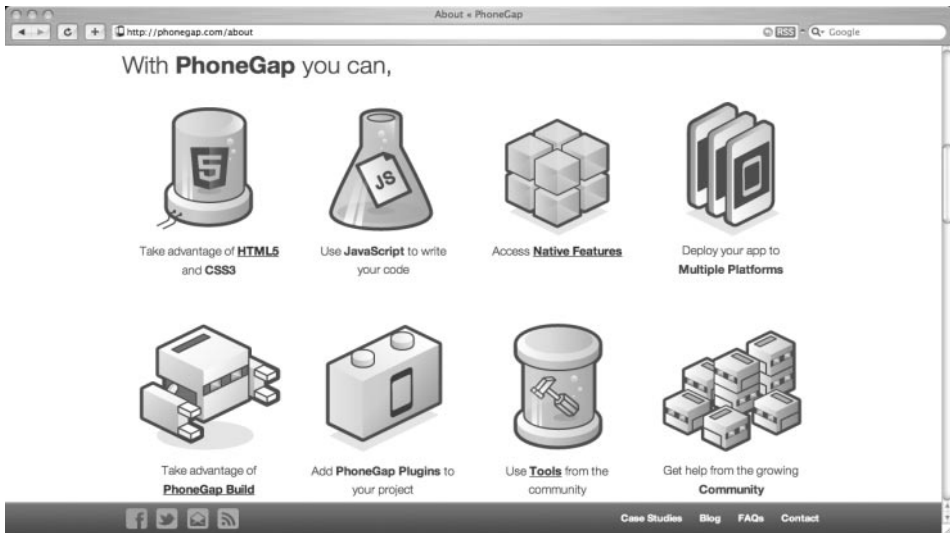


Bild 1.4: Das Leistungsspektrum von PhoneGap

## 1.6.2 Rhodes

Rhodes ist ein umfassendes Open Source Framework. Der Service, den Rhodes unter [www.rhobile.com](http://www.rhobile.com) anbietet, verspricht viel, er ist aber nicht einfach zu bedienen.

Sie entwickeln Ihre App in dem Framework einmal, und sie wird dann für alle gängigen mobilen Betriebssysteme ausgeworfen. Für die Entwicklung von Apps mit Rhodes sind HTML-, CSS- und Ruby-Kenntnisse erforderlich. Rhodes wurde in Ruby geschrieben, deswegen sind Ruby-Kenntnisse für das Entwicklungssystem zwingend erforderlich. Ruby ist eine leicht zu erlernende objektorientierte Programmiersprache. Sie wird im Anhang dieses Buches kurz vorgestellt.



Bild 1.5: Die Homepage von Rhomobile.com

Im Zusammenhang mit diesem Konzept bietet Rhomobile vier Produkte an:

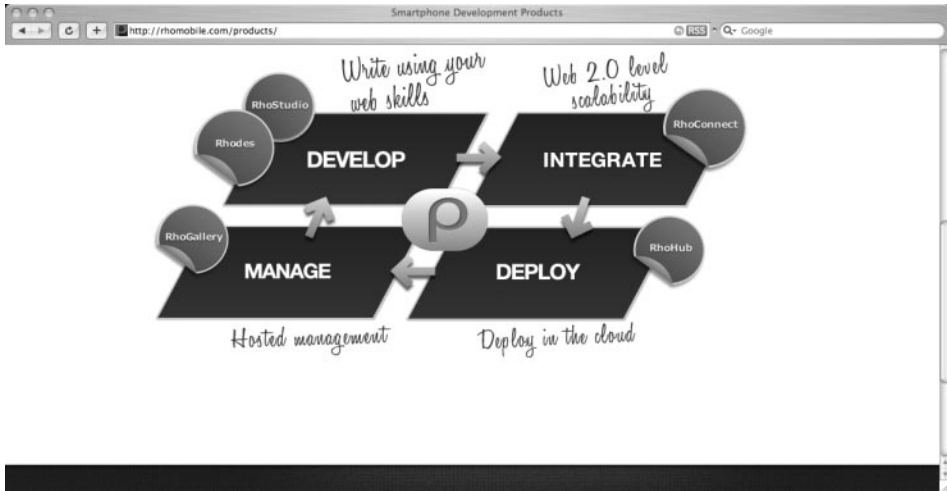
### **Rhodes Studio,**

die eigentliche Entwicklungsumgebung, in der native Apps mit Ruby oder JavaScript programmiert werden. Zu Beginn starten Sie mit der Installation von **Rhodes Studio** unter dem Link <http://rhomobile.com/products/rhostudio/>

### **RhoConnect,**

einen Synchronisationsassistenten, der Online-Daten immer aktuell für ein Smartphone zur Verfügung stellt. Er steht für die Integration und Übertragung der Daten. Dabei vereinfacht er das Parsen der Daten im JSON- und XML-Format.

Die Daten stehen im Offline-Zustand jederzeit zur Verfügung. Der Service ist mit Kosten verbunden. Die Preise starten bei 5000 Dollar für eine Anwendung bis 100 Benutzern.



**Bild 1.6:** Die vier Produkte von Rhomobile

### RhoHub,

eine Software-as-a-Service-Lösung, in der Anwendungen auf einer Netz-Plattform online entwickelt werden können. Für das Ändern der Daten wird in einem Online-Editor gearbeitet. Für iPhone-Daten steht ein spezieller Launcher zur Verfügung. Man kann mit ihm den offline arbeitenden iPhone-Simulator starten und die App so testen. Die in mehrere Reiter aufgeteilte Entwicklungsumgebung bietet durch den Reiter *Build* die Möglichkeit, die Apps online zu generieren und herunterzuladen. Dies funktioniert für Windows Mobile, Android und Blackberry. Für iPhones muss man den lokalen Weg über Xcode wählen. Die Lizenz-Bedingungen von Apple verbieten alles andere. Rhodes bietet in diesem Service eine kostenlose Basisversion an, diese erlaubt aber keine privaten Anwendungen. Will man eine App online generieren, lässt man den Code dort. So sind die Anwendungen unter diesem Account für jedermann zugänglich. Will man drei private Anwendungen generieren, so kostet das 20 Dollar. Im Falle einer kommerziellen Entwicklung auf dem RhoHub, fordert Rhodes 500 US-Dollar.

### RhoGallery,

verteilt Anwendungen an die Benutzer. Der Service bietet eine zentrale Benutzerverwaltung, sowie eine automatische Bereitstellung von Programmen für den Anwender. Ein weiteres Features ist das Sperren von Anwendungen, wenn das Endgerät gestohlen wurde, oder die Endverbraucher-Lizenz erloschen ist.

Mit Rhodes lassen sich eher native Apps erstellen als mit PhoneGap. Das kann sogar online passieren, ist aber kostenpflichtig. Der Support und die Dokumentation sind eher mager. Sie müssen zudem Ruby als Programmiersprache erlernen, um das Programm bedienen zu können. Ich fand es kryptisch, in einem Terminalfenster zu arbeiten, um die ganzen Prozesse der Gestaltung einer App mit Ruby-Befehlen anzustoßen.

```
Norbert-Usadels-MacBook:~ norbertusadel$ cd contacts
Norbert-Usadels-MacBook:contacts norbertusadel$ rhogen model Contact first_name,
last_name,email_address,buisness_number
Generating with model generator:
[ADDED] app/Contact/index.erb
[ADDED] app/Contact/edit.erb
[ADDED] app/Contact/new.erb
[ADDED] app/Contact/show.erb
[ADDED] app/Contact/index.bb.erb
[ADDED] app/Contact/edit.bb.erb
[ADDED] app/Contact/new.bb.erb
[ADDED] app/Contact/show.bb.erb
[ADDED] app/Contact/contact_controller.rb
[ADDED] app/Contact/contact.rb
[ADDED] app/test/contact_spec.rb
Norbert-Usadels-MacBook:contacts norbertusadel$
```

Bild 1.7: Anlegen eines Datenmodells im Terminalfenster unter Rhodes.

Es fallen bei Rhodes keine Kosten für die Entwicklung an. Rhodes bietet wenig Möglichkeiten zum Debuggen. Will man seine App über die Plattform vermarkten, fallen Lizenzgebühren an.

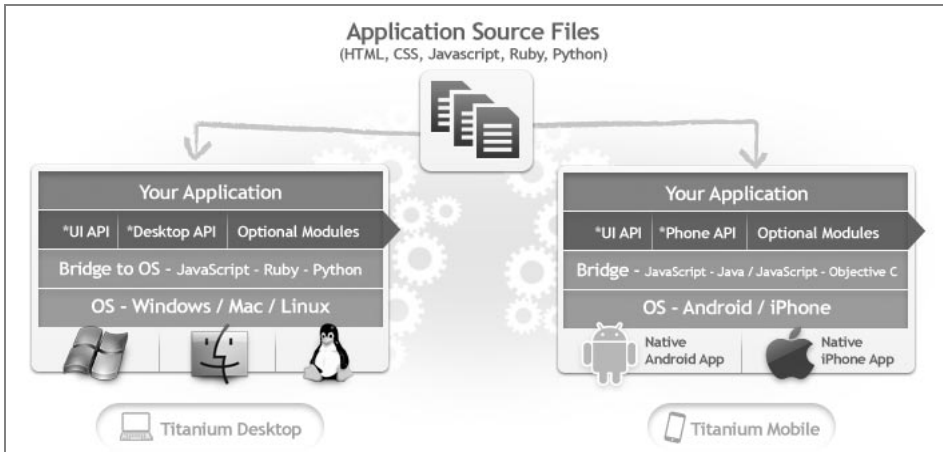
### 1.6.3 Sencha Touch

Sencha Touch <http://www.sencha.com/> ist ein HTML5-Entwicklerframework. Hiermit können Sie nach neuestem Entwicklerstandard Web-Apps für das mobile Web entwickeln. Dies ist durchaus als Kampfansage gegen die App-Stores zu sehen. Es ist durchaus einfacher, Web-Apps zu entwickeln als für jede Plattform eine native App.



Bild 1.8: Homepage von Sencha Touch





**Bild 1.10:** Die Grafik zeigt anschaulich, was Titanium zu leisten vermag.

Quelle: <http://www.appcelerator.com>

Die Software Titanium nimmt wenig Platz auf der Festplatte ein und ist schnell. Sie ist aber ein mächtiges, kleines Tool.

Die Entwicklung von iPhone-Apps bleibt auf die Mac-Umgebung beschränkt. Android-Apps kann man jedoch von allen Betriebssystemen aus erstellen. Man braucht die SDKs der einzelnen Plattformen. Die Einschränkung ist eben, dass das SDK fürs iPhone nur auf dem Mac läuft.

Titanium selbst ist ein Open-Source-Projekt und ist kostenlos. Es wird ständig weiterentwickelt. Titanium bietet Support gegen Geld an. Man kann sich dort Support-Videos kaufen oder sich direkt von Entwicklern beraten lassen. Derzeit nutzen ca. 1,5 Millionen Entwickler das Modul. Das ist eine große Community. Namhafte Hersteller nutzen ebenfalls Titanium.

Titanium produziert mehr nativen Code als PhoneGap. Viele Software-Anbieter nutzen Titanium auch, um Ihre Desktoplösungen auf andere Plattformen zu portieren. So wird aus Windows-Anwendungen auch ein Programm für Mac OS X oder Linux. Es ist einfach die Vielfalt, die Titanium zu einem mächtvollen Tool macht. Der Support ist außergewöhnlich, auch wenn er Geld kostet. Als Anfänger kommen Sie aber auch einfach mit der kostenlosen Dokumentation aus, die auf den Webseiten von Titanium geboten wird. Diese sind übersichtlich und gut aufgearbeitet.

Titanium hat einen Geschwindigkeits-Nachteil. Jeder View wird mit seinem eigenen Java-Code dargestellt. Der Code kann nicht auf andere Web-Views übertragen werden. Die Apps werden dadurch langsam.

Titanium ist Java-lastig. Es bietet aber eine schöne Anzahl von Modulen und APIs, die man in seine eigene Entwicklung einfließen lassen kann. Diese sind teilweise kostenpflichtig und lassen sich unter der Adresse <https://marketplace.appcelerator.com/landing> auswählen und laden.

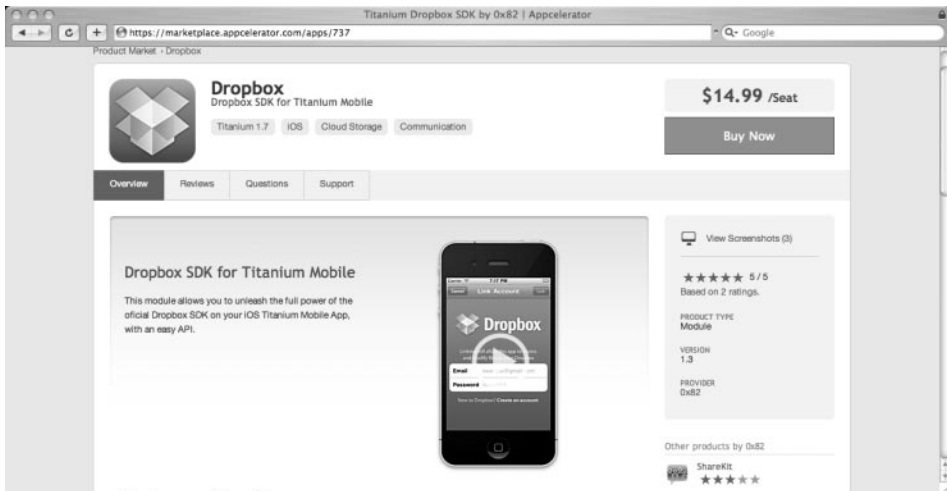


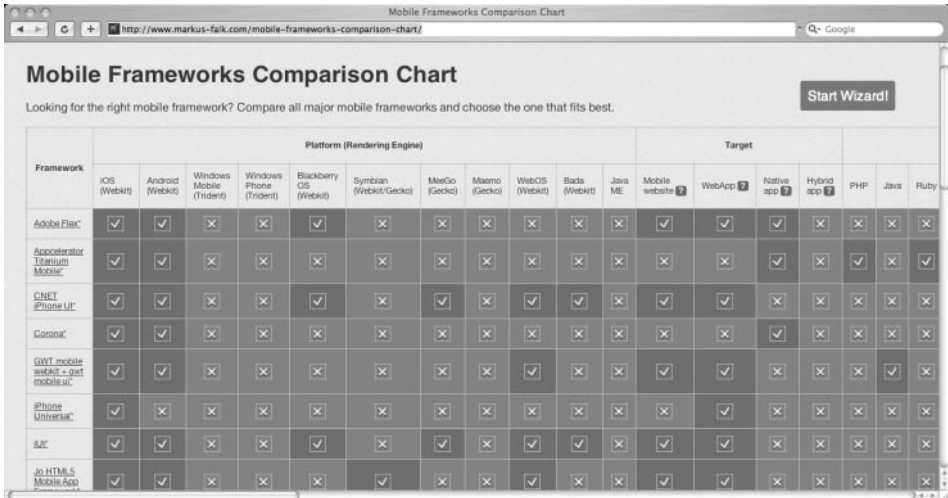
Bild 1.11: Dropbox-Modul für die App-Entwicklung aus dem Appcelerator-Marketplace.

## 1.7 Der Vergleich der Frameworks

Welches Programm Sie auch einsetzen, Sie haben die Qual der Wahl. Die hier vorgestellten Programme helfen dabei, native Anwendungen aus HTML, CSS und JavaScript zu bilden. Sie erhalten einen Überblick über alle Frameworks, die es für die mobile App-Entwicklung gibt, unter dieser Adresse:

<http://www.markus-falk.com/mobile-frameworks-comparison-chart/>

Die dort zu findende Mobile Frameworks Comparison Chart fasst alle Frameworks zusammen. Durch eine Suchabfrage gelangen Sie zu einem Framework, das für Ihr Projekt am besten passt.



**Mobile Frameworks Comparison Chart**

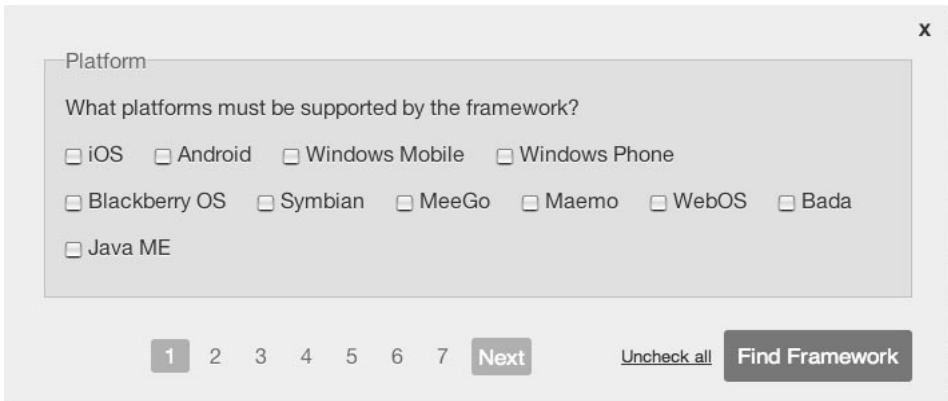
Looking for the right mobile framework? Compare all major mobile frameworks and choose the one that fits best.

[Start Wizard!](#)

Framework	Platform (Rendering Engine)										Target							
	iOS (WebKit)	Android (WebKit)	Windows Mobile (Trident)	Windows Phone (Trident)	BlackBerry OS (WebKit)	Symbian (WebKit/Gecko)	MeeGo (Gecko)	Maemo (Gecko)	WebOS (WebKit)	Bada (WebKit)	Java ME	Mobile website	WebApp	Native app	Hybrid app	PHP	Java	Ruby
Adobe Flash	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗
Accelerator Titanium Mobile	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✓
CHET Phone UI	✓	✓	✗	✗	✓	✗	✓	✗	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗
Corona	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
GVYT mobile webkit + gwt mobile ui	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✓	✗
Phone Universal	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
UI	✓	✓	✗	✗	✓	✗	✓	✗	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗
HTML5 Mobile App	✓	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗

Bild 1.12: Die Mobile Frameworks Comparison Chart zeigt Ihnen alle Frameworks

In der unteren Abbildung sehen Sie einen Teil der Suchabfrage, die zu dem richtigen Framework für Ihr Projekt führen soll.



**Platform**

What platforms must be supported by the framework?

☐ iOS
 ☐ Android
 ☐ Windows Mobile
 ☐ Windows Phone

☐ BlackBerry OS
 ☐ Symbian
 ☐ MeeGo
 ☐ Maemo
 ☐ WebOS
 ☐ Bada

☐ Java ME

1 2 3 4 5 6 7 **Next** [Uncheck all](#) **Find Framework**

Bild 1.13: Suchabfrage um das richtige Framework zu finden



## 3 Arbeiten mit Titanium Mobile

In diesem Kapitel erlernen Sie den Umgang mit der Entwicklungsumgebung Titanium. Sie erfahren wie man erste Projekte anlegt. Sie laden Ihr erstes Beispiel-Projekt und erhalten die ersten Code-Beispiele.

### 3.1 Wo finden Sie was?

Einen guten Einstieg für Ihre Projekte finden Sie im Developer-Center von Titanium. Unter der URL <http://developer.appcelerator.com/documentation> finden Sie die Dokumentationen für Ihre Arbeit.

Dort sind die APIs, die Programmier-Schnittstellen, für Sie dokumentiert.

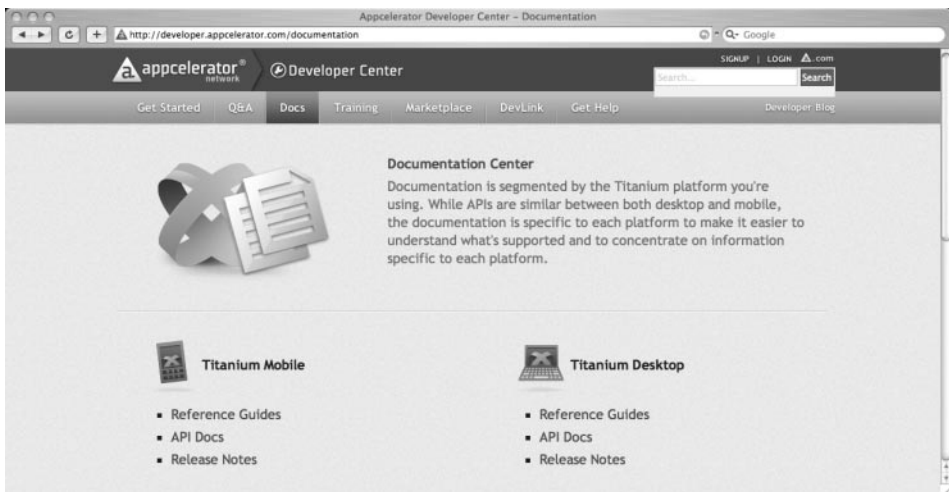


Bild 3.1: Das Developer-Center von Appcelerator

Es stehen umfangreiche, gut dokumentierte PDFs zur Verfügung. Die Hauptauswahl ist jeweils für Titanium Desktop und Titanium Mobile untergliedert in:

- Reference Guides
- API Docs
- Release Notes

In den Reference Guides stehen Ihnen Beispielprogramme, Lernvideos und Workshops zur Verfügung.

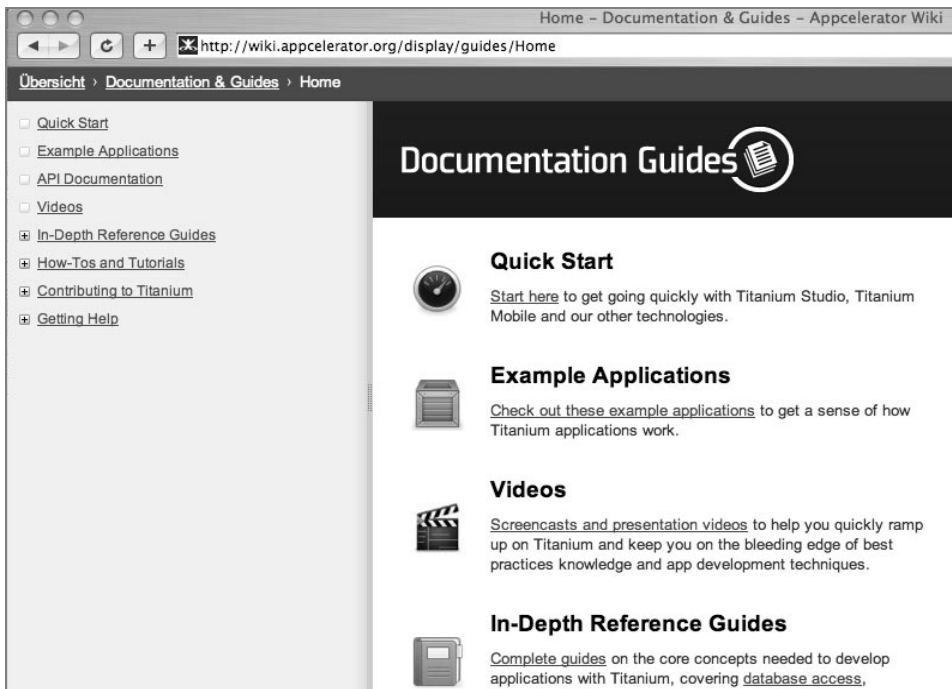


Bild 3.2: Documentation Guides

Unter dem Menüpunkt API Docs finden Sie eine Dokumentation zu den Programmierschnittstellen. Die untere Auflistung zeigt Ihnen die wichtigsten Features der APIs, die unterstützt und dokumentiert werden:

- Native apps built using web technologies
- Apps are compiled and run locally (offline available)
- Full support for HTML5 and CSS3
- Support for all native platform UI controls
- Third-party JavaScript support such as jQuery, Dojo, etc.

- Support for in-application SQL database
- Support for Geolocation (compass, geolocation, forward/reverse lookup)
- Support for Camera (taking Photos, playing and recording Video)
- Support for Photo Album (reading and writing)
- Support for Contacts Database / Address Book
- Support for Streaming Audio and Recording Audio, Audio Input Levels, Mic etc
- Support for Vibration
- Support for Social APIs such as Facebook Connect, Twitter etc.
- Support for Yahoo YQL
- Support for Web Services via REST, SOAP
- Support for native Maps
- Support for Push Notifications
- Support for in-application Email
- Support for in-application SMS, Telephone
- Support for Filesystem (reading, writing, etc)
- Support for Gestures (such as Shake)
- Support for Platform and Device capabilities
- Support for complex native views such as Coverflow, Image viewers, Table views, Grouped views, Composites, etc.
- Completely extensible via Module API for building your own controls or extending capabilities at compile-time

Unter der Seite <http://developer.appcelerator.com/apidoc/mobile/latest> erhalten Sie die Zusammenfassung aller wichtigen Titanium-APIs.

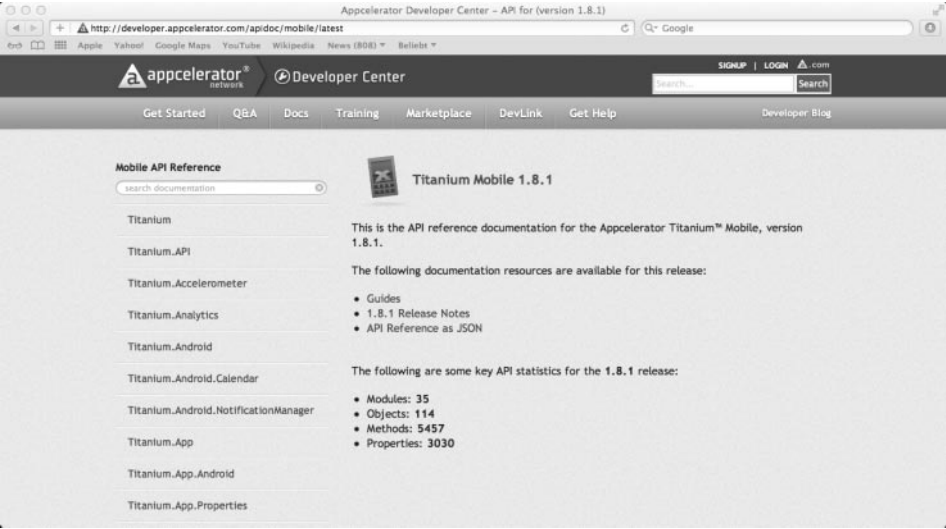


Bild 3.3: Überblick über die APIs von Titanium Mobile erhalten Sie im Developer Center.

Wenn Sie die Schnittstellen einer API benötigen, klicken Sie in der linken Seite auf eine API. In dem Beispiel unten sehen Sie die Beschreibung zu *Titanium.Android*. Sie wird auf der Seite <http://developer.appcelerator.com/apidoc/mobile/latest/Titanium.Android-module> dargestellt.

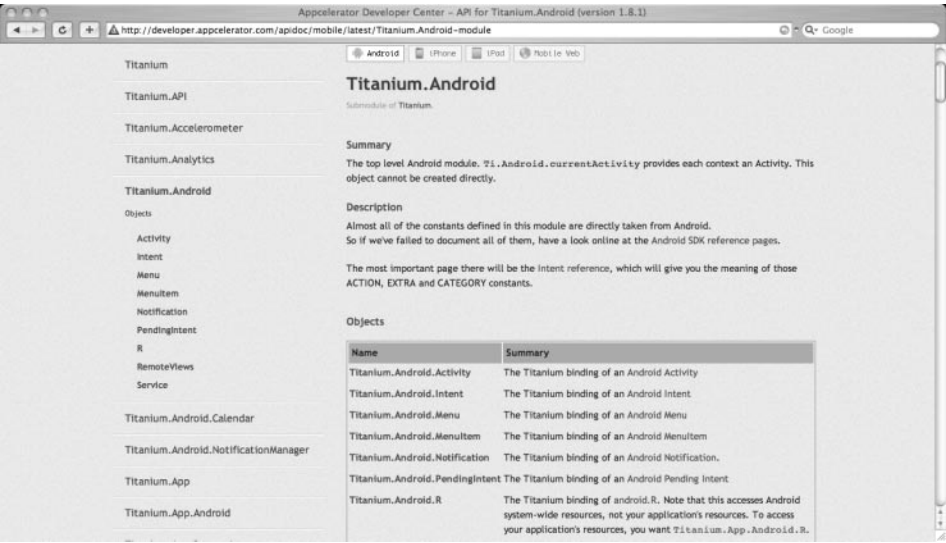
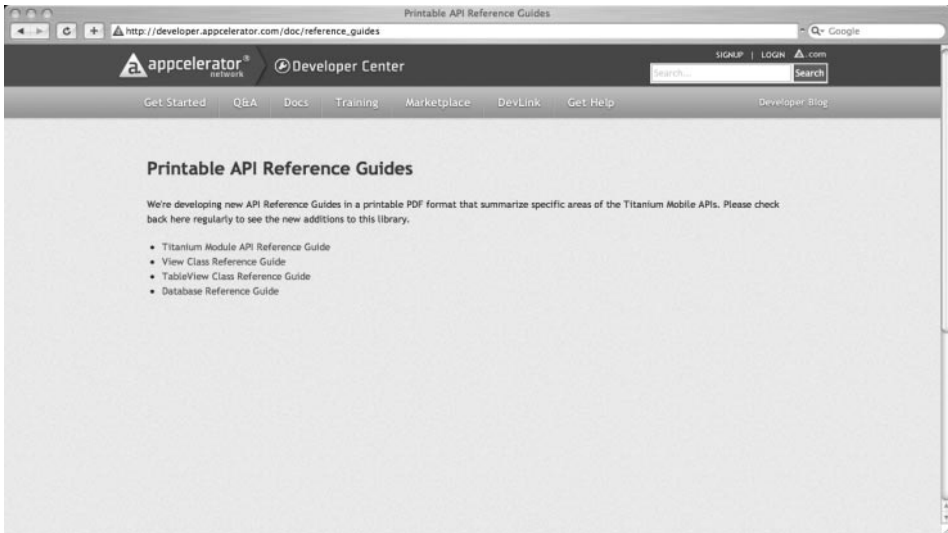


Bild 3.4: Beispiel für die Dokumentation der Titanium.Android-API.

Möchten Sie die kompletten Dokumentationen der APIs wählen Sie diese Seite:

[http://developer.appcelerator.com/doc/reference\\_guides](http://developer.appcelerator.com/doc/reference_guides)

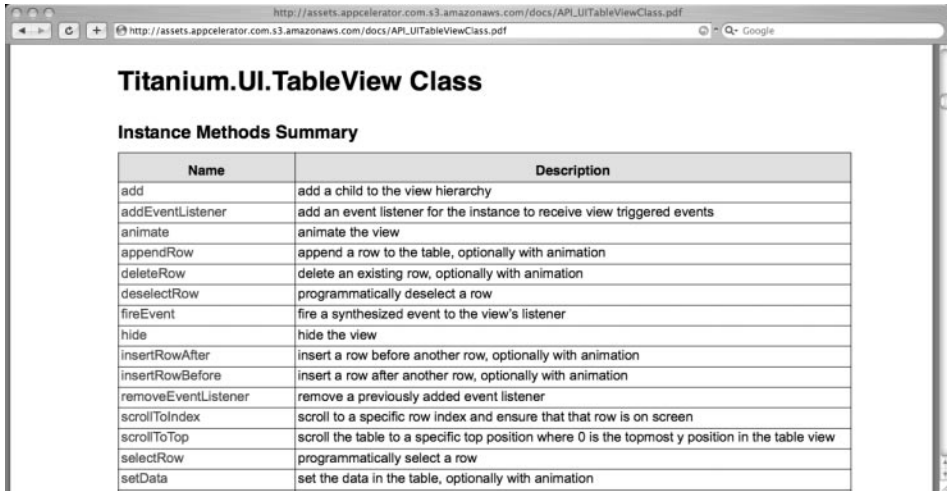


**Bild 3.5:** Die API Reference Guides

Die Dokumentationen liegen dort im PDF-Format aus. Diese sind in vier Gruppen gegliedert:

- Titanium Module API Reference Guide
- View Class Reference Guide
- TableView Class Reference Guide
- Database Reference Guide

Sie werden diese Dokumente ständig nutzen, wenn Sie Ihre Projekte in Titanium erstellen. In der unteren Abbildung sehen Sie die Zusammenfassung über die Titanium.UI.TableView Class.



Name	Description
add	add a child to the view hierarchy
addEventListener	add an event listener for the instance to receive view triggered events
animate	animate the view
appendRow	append a row to the table, optionally with animation
deleteRow	delete an existing row, optionally with animation
deselectRow	programmatically deselect a row
fireEvent	fire a synthesized event to the view's listener
hide	hide the view
insertRowAfter	insert a row before another row, optionally with animation
insertRowBefore	insert a row after another row, optionally with animation
removeEventListener	remove a previously added event listener
scrollToIndex	scroll to a specific row index and ensure that that row is on screen
scrollToTop	scroll the table to a specific top position where 0 is the topmost y position in the table view
selectRow	programmatically select a row
setData	set the data in the table, optionally with animation

Bild 3.6: Auszug aus der Titanium.UI.tableView Class

In der Dokumentation sind auch Programmierbeispiele zu den einzelnen Programmierschnittstellen enthalten. Im Folgenden sehen Sie ein kleines Beispiel für ein TableView. Die paar Zeilen Code finden Sie in der Dokumentation der Titanium.UI.tableViewClass-Reference. Das Codebeispiel zeigt Ihnen, wie einfach es sein kann, mit Titanium Mobile etwas für zwei Plattformen zu programmieren. Legen Sie einfach ein neues *Titanium Mobile Project* an. Öffnen Sie die Datei *app.js*. Löschen Sie deren Inhalt und kopieren den unteren Inhalt in die Datei. Danach können Sie das Projekt in den jeweiligen Simulatoren laufen lassen.

```
// app.js
// Create a window
var myWindow = Titanium.UI.createWindow({fullscreen:false});
// Create the table view -- it's a collection of various
// beverages
var myData = [{title:'Coffee'}, {title:'Tea'}, {title:'Juice'}, {title:'Ice
water'}];
var myTableView = Titanium.UI.createTableView({data:myData});
// Create one more row, and append it to the TableView
var myRow = Titanium.UI.createTableViewRow();
myRow.title = "Hot chocolate";
myTableView.appendRow(myRow);
// Add an event listener to respond to clicks in the TableView
myTableView.addEventListener('click', function(e) {
// Report which item was clicked
Titanium.UI.createAlertDialog({
title:'Alert',
message:('Click at index: '+e.index)
}).show();
});
```

```
});
// Add the table to the window
myWindow.add(myTableView);
myWindow.open();
```

Den Build-Prozess stoßen Sie mit dem grünen Button an. So haben Sie einen Code für zwei Plattformen.



Bild 3.7: Start der Simulatoren.

Die unteren Abbildungen zeigen Ihnen das Projekt auf dem iPhone-Simulator und dem Android-Emulator.



Bild 3.8: Table-View für das iPhone



Bild 3.9: Table-View für Android

Sie können einzelne Module für Ihre App-Entwicklung käuflich erwerben. Der Marketplace von Appcelerator bietet Ihnen dazu die Möglichkeit. Dort können Sie aber ebenso Module, die Sie programmiert haben, für andere anbieten. Laut Angaben von Appcelerator nutzen 1,5 Millionen Entwickler weltweit Titanium. Die Module bekommen Sie unter der URL: <https://marketplace.appcelerator.com/landing>

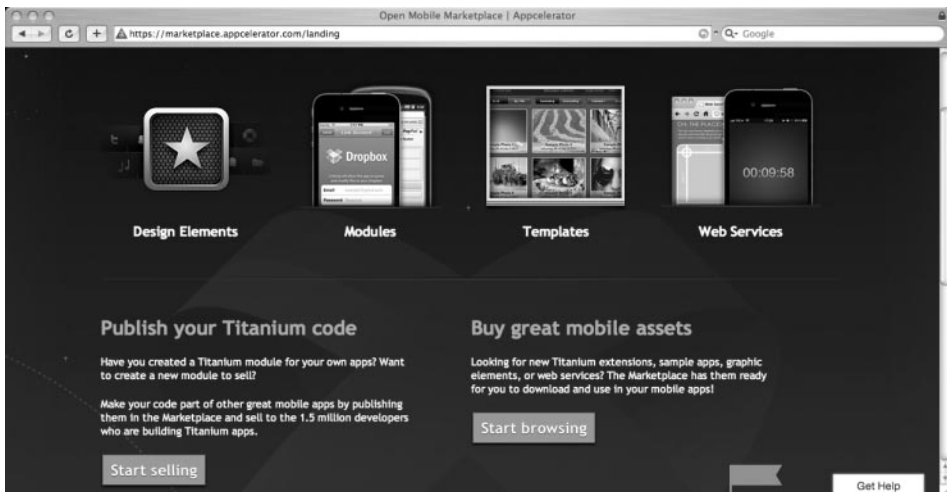
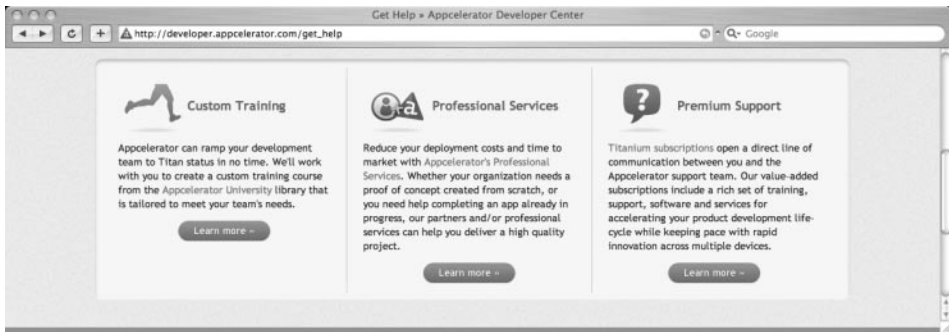


Bild 3.10: Der Marktplatz von Appcelerator

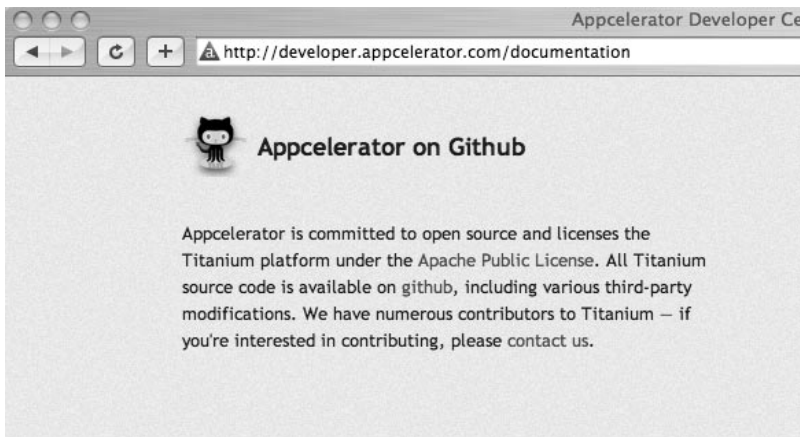
Wenn Sie mehr Hilfe als den kostenlosen Support brauchen, können Sie über die Seite [http://developer.appcelerator.com/get\\_help](http://developer.appcelerator.com/get_help) professionelle Hilfe buchen. Appcelerator bietet drei Formen der kostenpflichtigen Hilfe an.

- Custom Training
- Professional Services
- Premium Support



**Bild 3.11:** Professionelle, kostenpflichtige Hilfe von Titanium

Viele Programmbeispiele finden Sie verlinkt auf der Homepage von GitHub unter der Adresse <https://github.com/appcelerator>. GitHub ist ein Online-Dienst, der Code-Beispiele aktuell für seine Mitglieder zur Verfügung stellt.



**Bild 3.12:** Codebeispiele für Titanium finden Sie auf GitHub

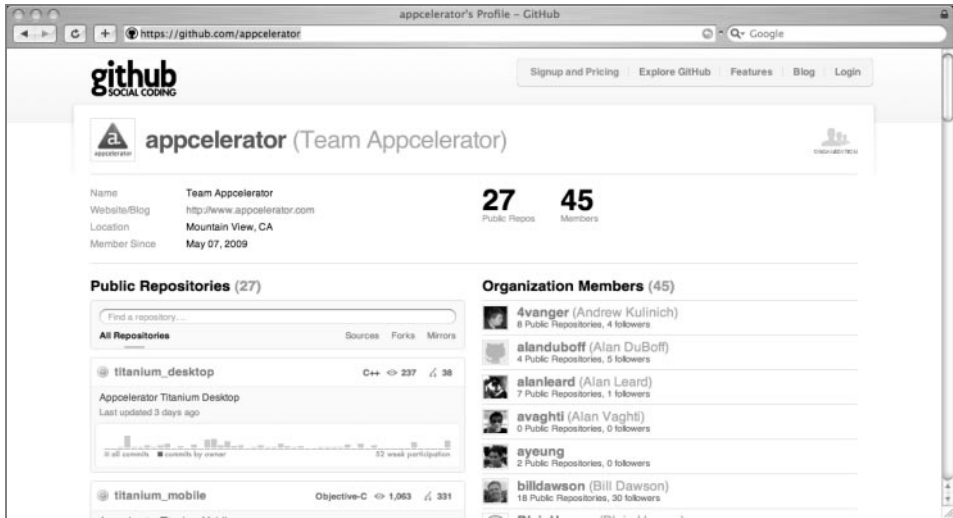


Bild 3.13: Codebeispiele sind hier gespeichert

Klicken Sie auf ein Code-Beispiel, so werden Ihnen im nächsten Fenster weitere Details über das Beispiel angezeigt, hier können Sie auch den Quelltext laden. Viele Beispiele finden Sie unter der angegebenen Adresse auf GitHub. Diese werden von der Entwicklungsgemeinschaft ständig aktualisiert.

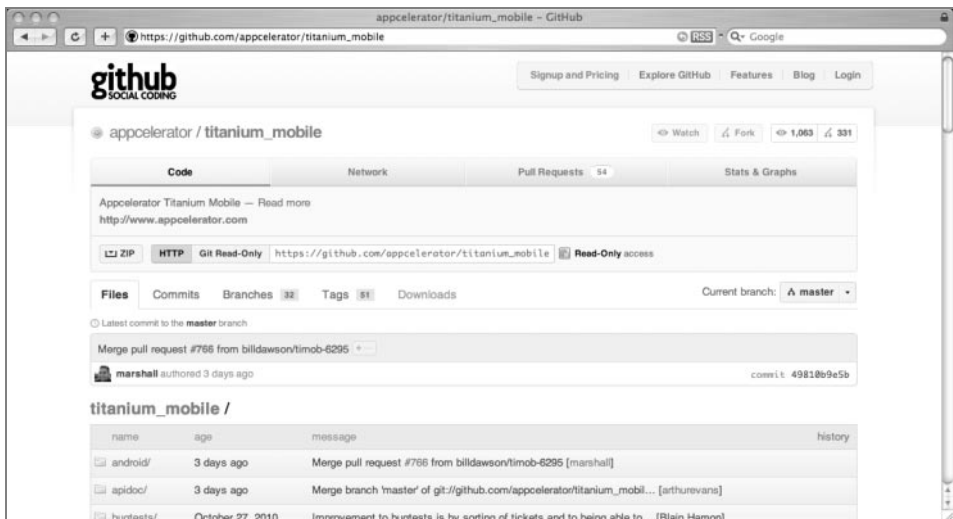


Bild 3.14: Code-Beispiele auf GitHub im Detail

Die Code-Beispiele unterliegen der Apache Public License. Eine Beschreibung dieser Lizenz finden Sie unter <http://de.wikipedia.org/wiki/Apache-Lizenz>. Sie wird dort so beschrieben:

»Man darf Software unter dieser Lizenz frei in jedem Umfeld verwenden, modifizieren und verteilen. Wenn man sie verteilt, muss eindeutig darauf hingewiesen werden, welche Software unter der Apache-Lizenz verwendet wurde und dass diese vom Lizenzgeber (name of copyright owner) stammt. Eine Kopie der Lizenz muss dem Paket beiliegen. Änderungen am Quellcode der unter der Apache-Lizenz stehenden Software brauchen nicht zum Lizenzgeber zurückgeschickt zu werden. Eigene Software, die unter Apache-Lizenz stehende Software verwendet, braucht nicht unter der Apache-Lizenz zu stehen. Die eigene Software darf nur dann Apache heißen, wenn eine schriftliche Genehmigung der Apache Foundation vorliegt. Eigene Werke, die auf einem unter der Apache-Lizenz stehendem Originalwerk basieren, müssen:

- eine Kopie der Apache-Lizenz beinhalten,
- bei modifizierten Dateien an auffälliger Stelle angeben, dass sie modifiziert sind,
- alle Original-Urheberrechtsvermerke beibehalten,
- falls das Originalwerk eine Textdatei namens »NOTICE« enthält, die dort enthaltenen Urhebervermerke der verwendeten Dateien auf eine in der Lizenz genauer vorgeschriebene Art und Weise enthalten.«

Sie unterwerfen sich also auch dieser Vereinbarung wenn Sie Ihren Titanium-Code auf GitHub zur Verfügung stellen.

Wenn Sie selber Code veröffentlichen wollen, müssen Sie sich dort anmelden. Weitere Hilfe bekommen Sie unter: <http://help.github.com>

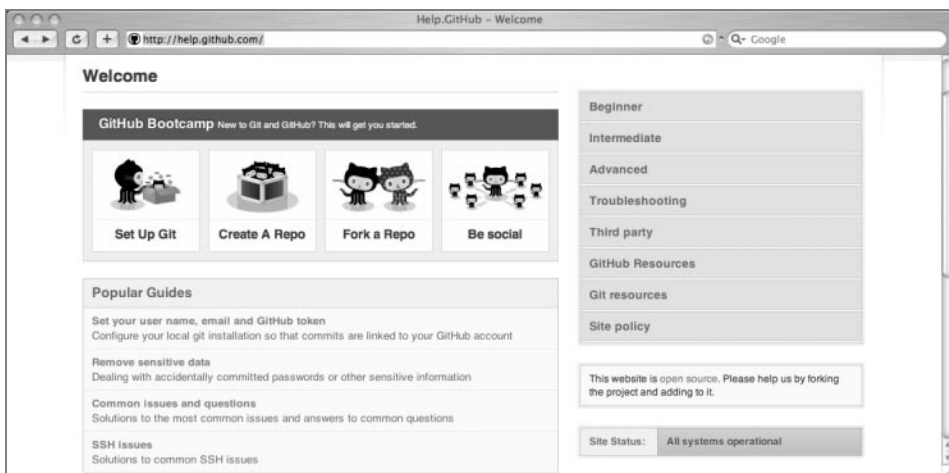


Bild 3.15: Hilfe für Ihre GitHub-Anmeldung

# Stichwortverzeichnis

## A

Adhoc-Vertrieb 257  
 AJAX-Monitor View 99  
 Android Debug Bridge 44  
 Android SDK Manager 43  
 Android Virtual Device Manager 48  
 Android-Emulator 63  
 Android-SDK 39  
 Apache Public License 77  
 API Docs 68  
 App ID 258  
 app.js 62  
 Aptana 102  
 Aptana Jaxer 103  
 asymmetrisches Verschlüsselungsverfahren 253

## B

BlackBerry 52  
 BlackBerry Code Signing Keys 52  
 BlackBerry Desktop Software 53  
 BlackBerry RIM SDK 54  
 BlackBerry-SDK 52  
 Breakpoints View 99  
 build-Verzeichnis 123  
 Bundle-Identifizier 259

## C

Certificate Signing Request 252  
 Certificates 255  
 Code-Vervollständigung 102  
 Console View 99

## D

Date and Time Picker 148  
 Datenaustausch 154  
 Debug View 99  
 Debug-Ansicht 101  
 Debug-Bereich 99  
 Developer-Center 67

## E

Eclipse-Plug-in 102  
 ein iOS Development Certificate 252  
 Entwicklungszyklus 14  
 Expressions View 99, 101

## F

Firefox-Add-On 171  
 FTP-Client 102

## G

Geolocation 156  
 GitHub 75, 106

## H

HTML5 22  
 HttpClient-Methode 152

## I

iAd-Netzwerk 261  
 ImageView 160  
 iOS Provisioning 254  
 iPhone-Simulator 63  
 iTunes Connect 260

**J**

Java-Bibliothek 22  
 JSON 189  
 JSON-Format 154

**K**

KitchenSink 105

**M**

Maps 158  
 Marketplace 74  
 Mobile Frameworks Comparison Chart 24

**N**

Native Apps 13

**O**

Open Source Framework 18  
 Ordnerstruktur 107  
 Outline View 99

**P**

Packages 46  
 PhoneGap 15, 17  
 Picker 147  
 Project Explorer 96  
 Provisioning Portal 259

**R**

Reference Guide 68  
 Resources folder 62  
 Resources-Verzeichnis 123  
 RhoConnect 19  
 Rhodes 18, 20  
 RhoGallery 20  
 RhoHub 20  
 Rhomobile 19  
 Ruby 20

**S**

SaaS 20

Scripts View 99

SDKs 27  
 Sencha Touch 21  
 Software-as-a-Service 20  
 Sound 162  
 SQL-Code 155  
 SQLite Manager 170

**T**

Tab 130  
 Tab-Group 130  
 tiapp.xml 62  
 tiapp.xml-Datei 123  
 Titanium Mobile 22, 35  
 Titanium Studio 27, 30  
 Titanium.Media.createVideoPlayer 163  
 Titanium.Media.Sound-Modul 162  
 Titanium.UI.createTableView 135, 136  
 Titanium.UI.createWebView 151  
 Titanium.UI.PickerColumn-Klasse 150  
 Titanium.UI.PickerRow-Klasse 150  
 Titanium.UI.TableViewRow 136, 140  
 Titanium.UI-Modul 128  
 Titanium-Mobile-Projekt 107  
 Twitter-App 189

**U**

UDID 258  
 Unique Device Identifier (UDID) 257

**V**

Variables View 99, 101  
 Versionskontrollsysteme 103  
 Videos 163  
 View 129

**W**

Web-App 13  
 Web-View 151  
 Wrapper 16

**X**

Xcode 37

XML 154

**Y**

YQL 155



# Titanium Mobile

## Apps für iPhone und Android

Entwickeln Sie Ihre Apps plattformübergreifend. Hier lernen Sie den eleganten Weg, wie Sie aus einfachem HTML5- und JavaScript-Code mit Titanium Mobile native Apps für Android und iOS-Geräte erstellen.

### ► Titanium Mobile optimal einrichten

Titanium Mobile setzt sich aus dem Titanium Studio und den Entwicklungswerkzeugen von Apple und Google zusammen. Welche Programme Sie installieren, wie Sie mit der Benutzeroberfläche von Titanium arbeiten und wie Sie dann Ihre ersten Apps auf den Emulatoren testen, wird hier beschrieben.

### ► Erste Beispiel-Apps erstellen

Um in die App-Entwicklung einzusteigen, ist es wichtig, durch praktische Beispiele Erfahrungen zu sammeln. Lernen Sie hier Schritt für Schritt, wie Sie ein Projekt anlegen, den Code mit den von Ihnen gewünschten Funktionalitäten ausstatten und die App dann auf den Emulatoren testen.

### ► Apps verkaufen

Wer eine App entwickelt hat, möchte sie auch in den App Store von Apple bringen oder bei Google Play vermarkten. Hier erfahren Sie, was Sie dazu brauchen und wie Ihre App schnell in den Stores landet.

### Aus dem Inhalt:

- Titanium Mobile im Vergleich mit anderen Frameworks
- Titanium Mobile laden, installieren und benutzen
- Der Aufbau von Titanium Mobile
- Projekt anlegen und verwalten
- Aptana, der Editor und Debugger
- Anwendungen laufen lassen
- Windows, Views und Controls
- Tab-Group und Tabs
- Das Table-View
- Picker
- Web-View
- Netzwerkzugriffe
- Geolocation, Maps
- Bilder und die Kamera, Video und Sound
- Eine SQLite-Datenbank abbilden
- Twitter-App
- Apps in die Stores bringen

### Über den Autor:

Norbert Usadel arbeitet seit 20 Jahren selbstständig. In dieser Zeit schulte er ca. 2000 Menschen über die Themen SAP, Cobol, Filemaker, Shopsysteme, iPods, iPhones, iTunes, Strukturierte Programmierung, Pagemaker, Photoshop und App-Programmierung. Außerdem entwickelt er Apps und übernimmt Projektleitungen für die App-Programmierung.

### Auf [www.buch.cd](http://www.buch.cd)

Der komplette Quellcode des Buches



30,– EUR [D]

ISBN 978-3-645-60160-3

Besuchen Sie unsere Website

[www.franzis.de](http://www.franzis.de)