



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

Inhaltsübersicht

Über dieses Buch

Aufbau

Zielgruppe

Über den Autor

Webseite zum Buch

Danksagung

1 Titanium Mobile und die anderen ...

1.1 Einordnung

1.2 Was ist eine Web-App?

1.3 Was ist eine native App?

1.4 Der Vergleich

1.5 Fazit

1.6 Titanium und die anderen

1.7 Der Vergleich der Frameworks

2 Titanium Mobile laden, installieren und benutzen

2.1 Die Basis-Installation

2.2 Titanium Studio installieren

2.3 Installation von Titanium Mobile

2.4 Der Android SDK Manager

2.5 BlackBerry

2.6 Testen, ob alles läuft - Der erste Test

3 Arbeiten mit Titanium Mobile

3.1 Wo finden Sie was?

3.2 Der Aufbau von Titanium Mobile

3.3 Aptana, der Editor und Debugger

3.4 KitchenSink

4 Die Reise durch das Titanium-Framework

4.1 Windows, Views und Controls

4.2 Tab-Group und Tabs

4.3 Das Table-View

4.4 Ein Table-View-Beispiel

4.5 Picker

4.6 Web-View

4.7 Netzwerkzugriffe

4.8 Geolocation

4.9 Maps

4.10 Bilder und die Kamera, Videos und Sound

5 Eine SQLite-Datenbank abbilden

5.1 Das Projekt anlegen

5.2 Anlegen der Datenbank

5.3 Import der Datenbank

5.4 Erzeugen der App

6 Twitter-App

7 Der Pizza-Palast

8 App in den Store ...

8.1 App in die Android-Markets

8.2 App in den Apple Store

A Anhang

A.1 Wichtige Titanium-Links

A.2 HTML5

A.3 CSS

A.4 JavaScript

A.5 Frameworktools

Stichwortverzeichnis

Über dieses Buch

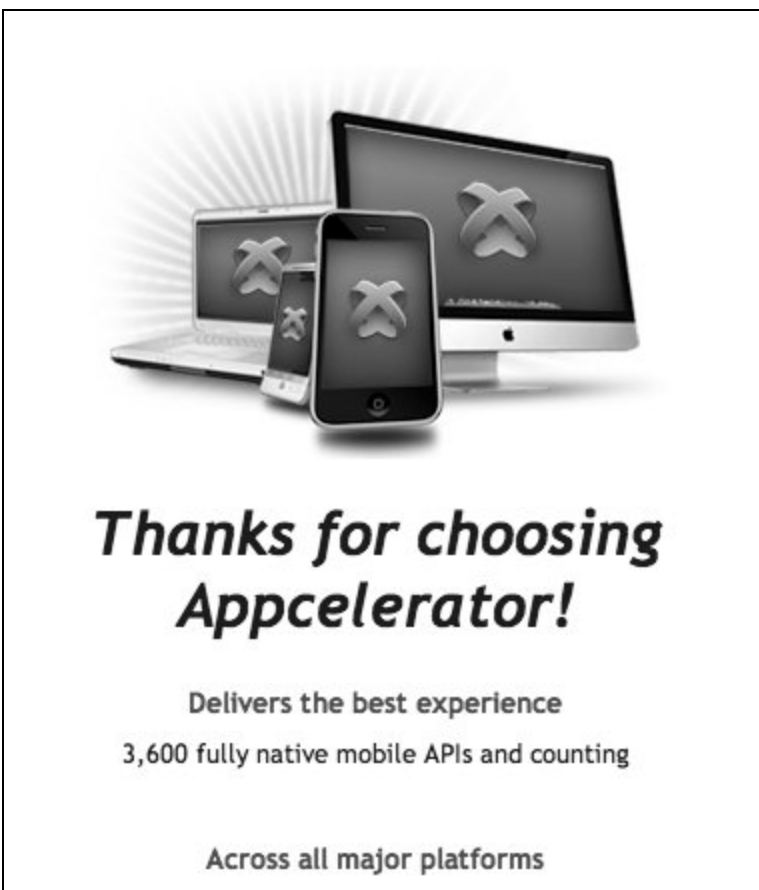
Dieses Buch vermittelt Ihnen den Einstieg in das Titanium-Framework. Sie erstellen in diesem Framework einen Code für die Android-Plattform und für das iPhone.

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 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.





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. Dort dreht sich alles um die App-Programmierung. Im Bereich www.appzitty/download liegen die einzelnen Titanium-Projekte aus diesem Buch für Sie bereit. Sie können mich unter der E-Mail-Adresse 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.

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 Leitspruch 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.

PhoneGap

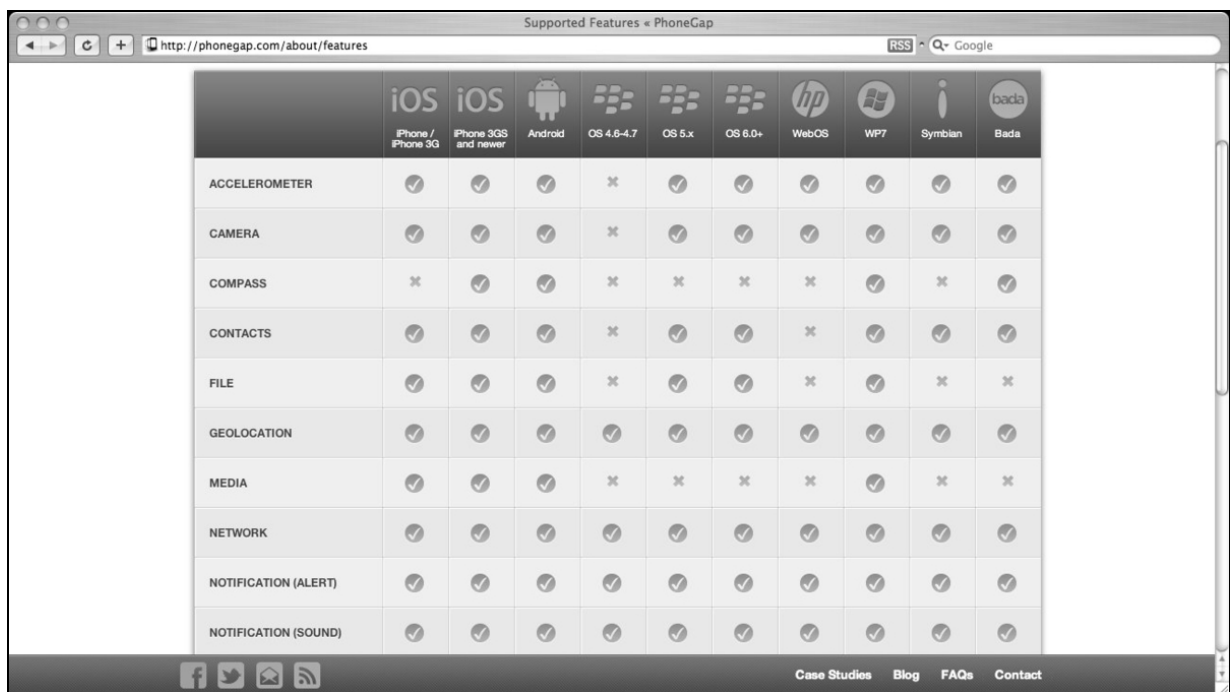
PhoneGap ist ein Open-Source-Tool, das Sie unter <http://www.phonegap.com/download> downloaden können. Es wurde von Nitobi Software entwickelt. Im Oktober 2011 wurde Nitobi (inkl. PhoneGap) von Adobe aufgekauft. Das Programm soll in die Apache Software Foundation übergeben werden und unter einem anderen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



The screenshot shows a web browser window displaying the 'Supported Features' page of PhoneGap. The page features a table with 11 columns representing different mobile operating systems and 10 rows representing various device features. Each cell in the table contains a checkmark (✓) indicating support or an 'x' indicating lack of support.

	iOS iPhone / iPhone 3G	iOS iPhone 3GS and newer	Android	OS 4.6-4.7	OS 5.x	OS 6.0+	WebOS	WP7	Symbian	Bada
ACCELEROMETER	✓	✓	✓	x	✓	✓	✓	✓	✓	✓
CAMERA	✓	✓	✓	x	✓	✓	✓	✓	✓	✓
COMPASS	x	✓	✓	x	x	x	x	✓	x	✓
CONTACTS	✓	✓	✓	x	✓	✓	x	✓	✓	✓
FILE	✓	✓	✓	x	✓	✓	x	✓	x	x
GEOLOCATION	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MEDIA	✓	✓	✓	x	x	x	x	✓	x	x
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.

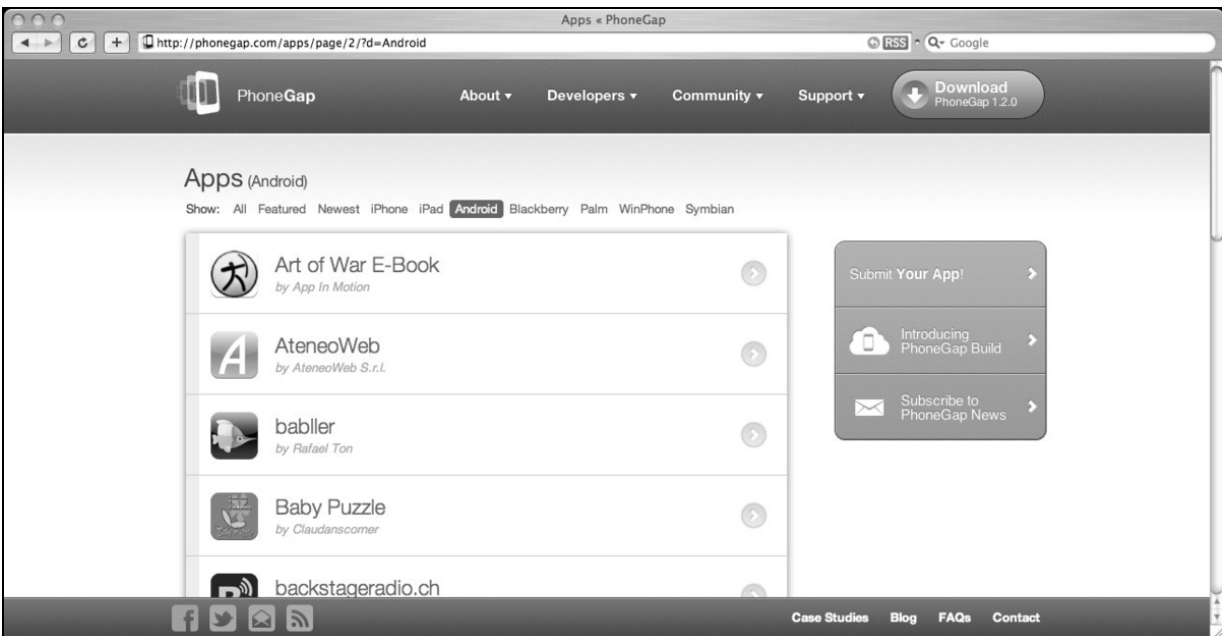


Bild 1.3 Übersicht der entwickelten Apps auf der PhoneGap-Plattform

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.

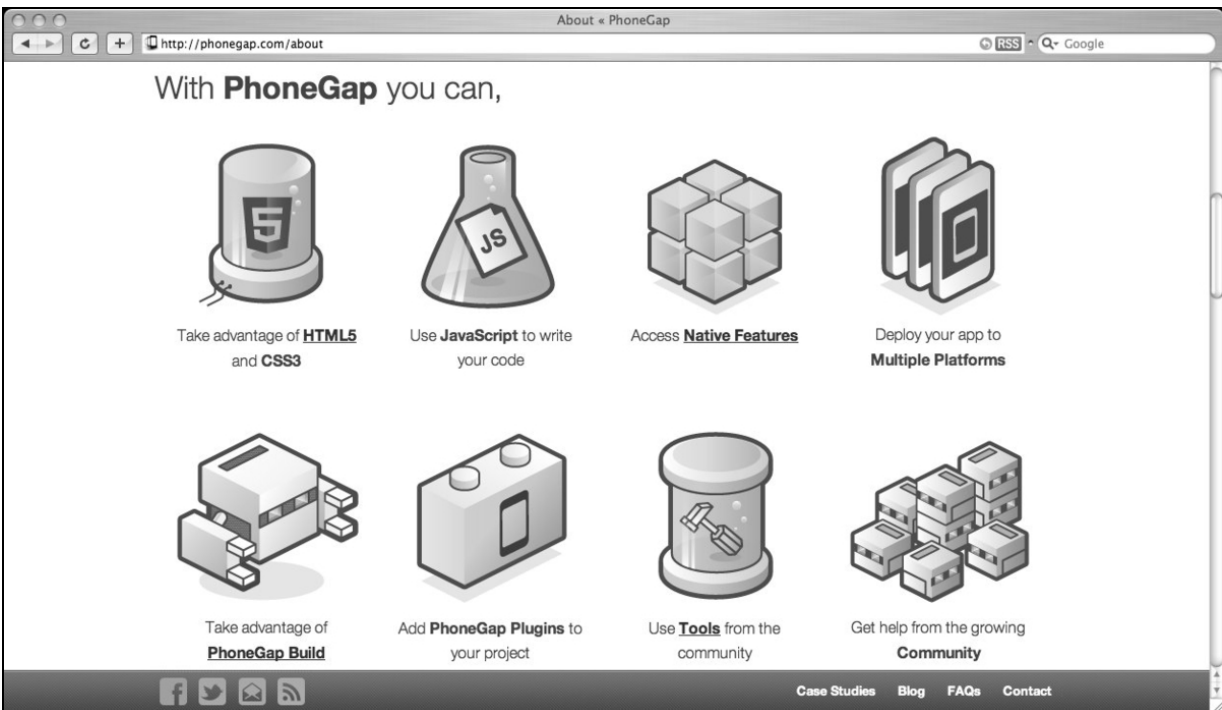


Bild 1.4 Das Leistungsspektrum von PhoneGap

Rhodes

Rhodes ist ein umfassendes Open Source Framework. Der Service, den Rhodes unter 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.

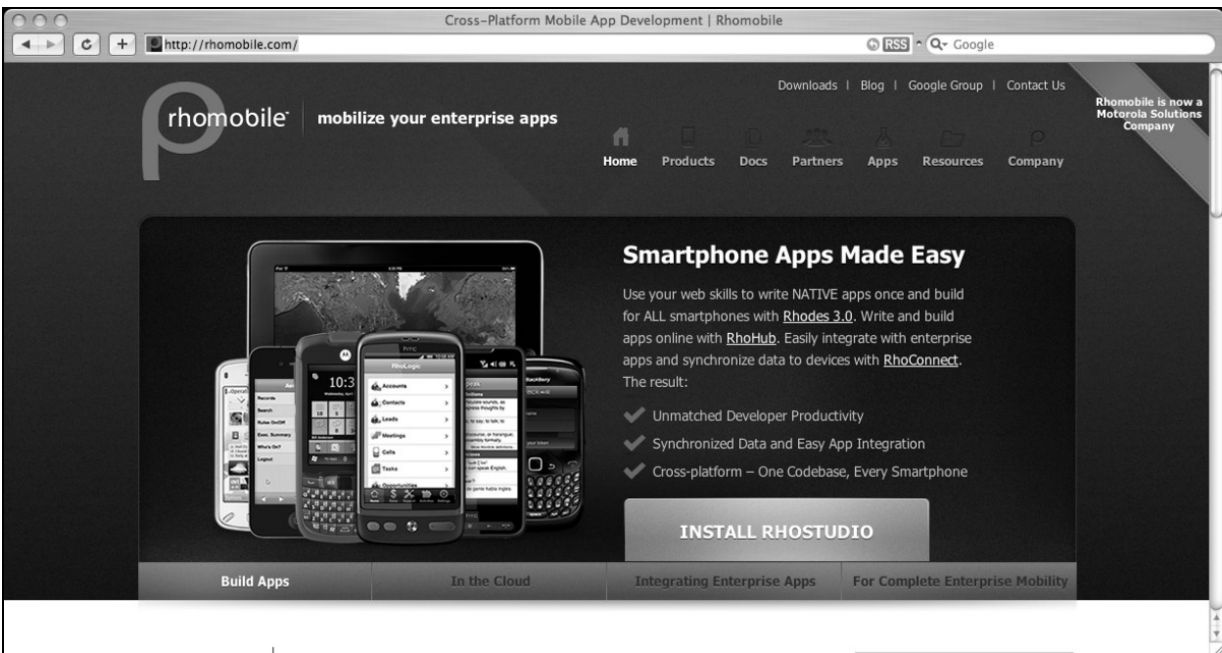


Bild 1.5 Die Homepage von Rhomobile.com

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.

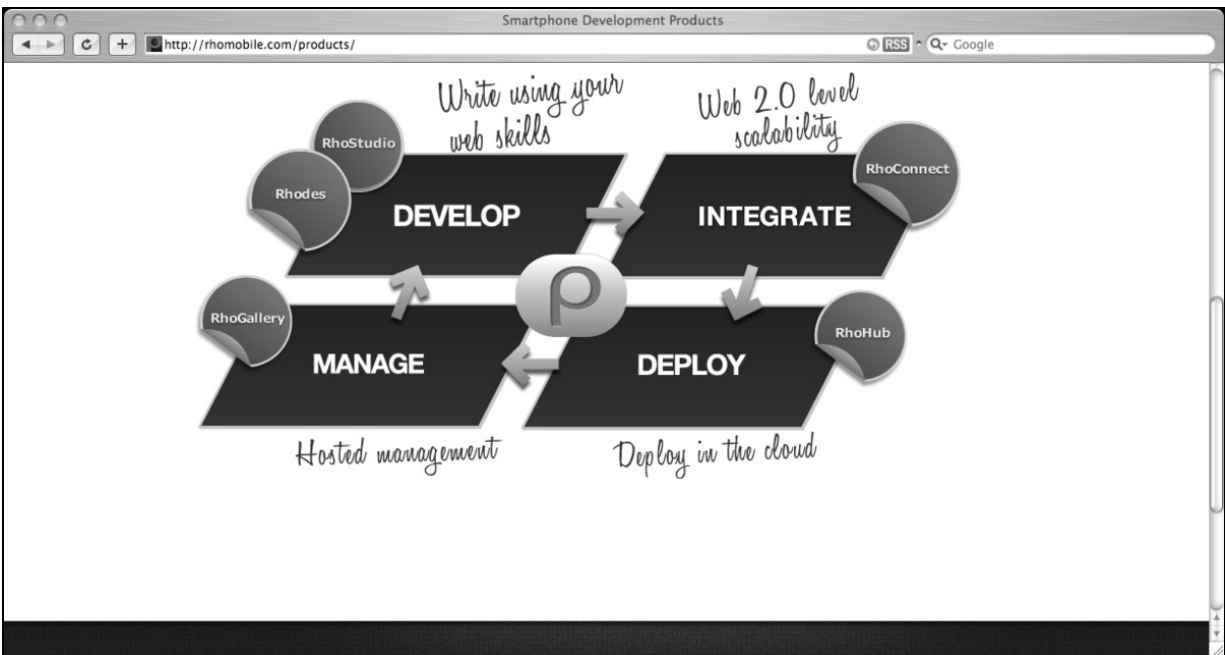


Bild 1.6 Die vier Produkte von Rhomobile

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 Lizenzbedingungen 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_adress,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.

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.

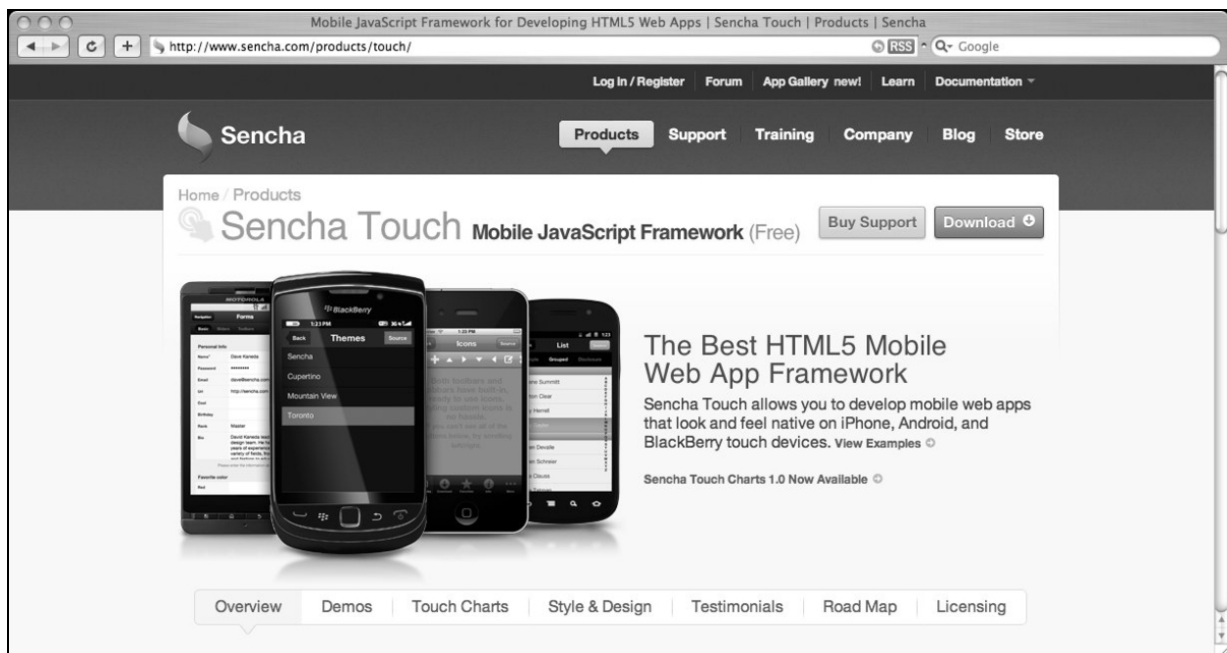


Bild 1.8 Homepage von Sencha Touch

Web-Apps werden in HTML5 geschrieben und können unabhängig von der Geräteplattform genutzt werden. Benutzer können so immer auch auf die neueste Version der App zugreifen, weil Sie ja im Web aktualisiert wurde. Sie

könnten erst mit Sencha Touch entwickeln und dann aus Ihrem Projekt native Apps erstellen lassen.

Titanium Mobile

Mit Titanium Mobile können Sie native Apps für das iPhone, Android und eingeschränkt für Blackberry-Smartphones entwickeln. Sie bedienen sich dabei der Syntax von JavaScript, HTML5 und CSS. Die einzelnen SDKs (Software-Developer-Kits) korrespondieren mit Titanium Studio, dem eigentlichen Programm, in dem Sie Ihre Apps entwickeln.

Für Webdesigner dürfte es kein Problem sein, schnell Erfolge zu erzielen. Objective-C braucht man für diesen Weg der App-Erstellung nicht zu erlernen.

Ist die Idee einer App umgesetzt, kann diese auf mehreren Plattformen, wie Android oder iOS für das iPhone, laufen. Nur einmal programmieren für mehrere Plattformen spart Zeit und Kosten.

Dasselbe Dokument lässt sich aber auch noch zu Desktopanwendungen für Windows, Mac und Linux konvertieren. Durch Titanium Desktop ist dies möglich. Wer HTML, CSS und JavaScript nicht nutzen möchte, kann seine Quelldokumente auch in Ruby oder Python schreiben. Für die Gestaltung der Oberfläche von Programmen und Apps stellt Titanium eine umfangreiche Java-Bibliothek zur Verfügung.

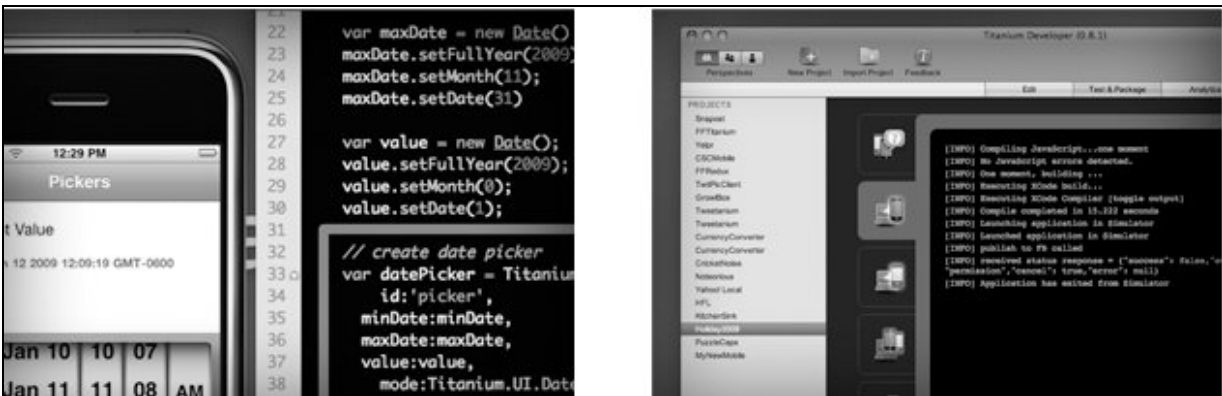


Bild 1.9 Einblicke in Titanium Mobile

Sie bekommen das Programm unter dieser Adresse:

<http://www.appcelerator.com>

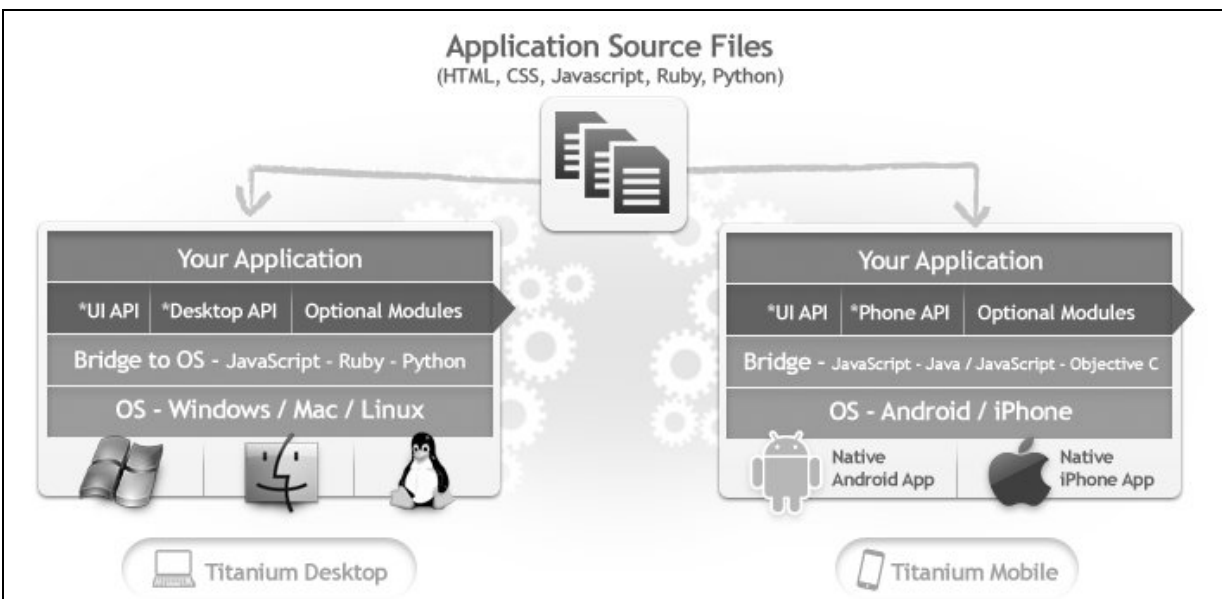


Bild 1.10 Die Grafik zeigt anschaulich, was Titanium zu leisten vermag. Quelle:

<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.

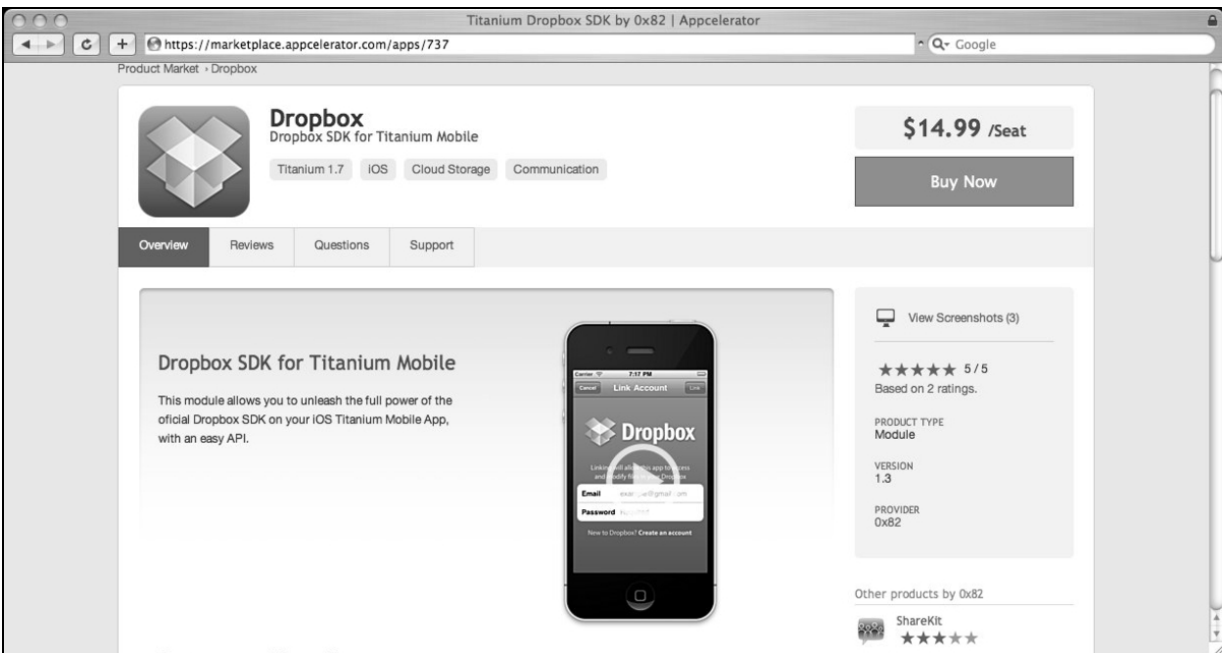


Bild 1.11 Dropbox-Modul für die App-Entwicklung aus dem Appcelerator-Marketplace.

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 Flex*	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗
Appcelerator Titanium Mobile*	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✓
CNET iPhone UI*	✓	✓	✗	✗	✓	✗	✓	✗	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗
Corona*	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗
GTK mobile webkit + gwt mobile ui*	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓	✗	✗	✗	✓	✗
iPhone Universal*	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
UI*	✓	✓	✗	✗	✓	✗	✓	✗	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗
Jo HTML5 Mobile App Framework	✓	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗

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

2 Titanium Mobile laden, installieren und benutzen

Dieses Kapitel zeigt Ihnen, wie Titanium Studio und Titanium Mobile installiert werden. Sie installieren zunächst Titanium Studio, mit dem Sie Desktopanwendungen erstellen können. Im zweiten Schritt binden Sie die SDKs (Software-Developer-Kits) ein, um Apps nativ erstellen zu können. Am Ende des Kapitels, lassen Sie Ihre erste App schon in den beiden Simulatoren für Android und das iPhone laufen.

2.1 Die Basis-Installation

Die ersten Schritte.

Das Programm Titanium bekommen Sie nach Anmeldung unter der URL

<https://my.appcelerator.com/auth/signup/offer/community>.

Nachdem Sie das Anmelde-Prozedere überwunden haben, ist Ihr Account in 60 Sekunden erstellt. Hierzu muss der Account durch einen Link in einer Bestätigungs-e-mail aktiviert werden. Im Folgenden installieren Sie zuerst das Programm Titanium Studio. Im zweiten Schritt ergänzen Sie das Programm mit den SDKs der einzelnen Hersteller, um mobile Programme zu erstellen und laufen lassen zu können. Sie bekommen Hilfe unter diesem Link:

<http://wiki.appcelerator.org/display/guides/home>. Das Wiki hält alles für den Start für Sie bereit: Quick Start, Beispiel-Programme, Videos, Reference-Guides, und spezielle Workshops für knifflige Themen.

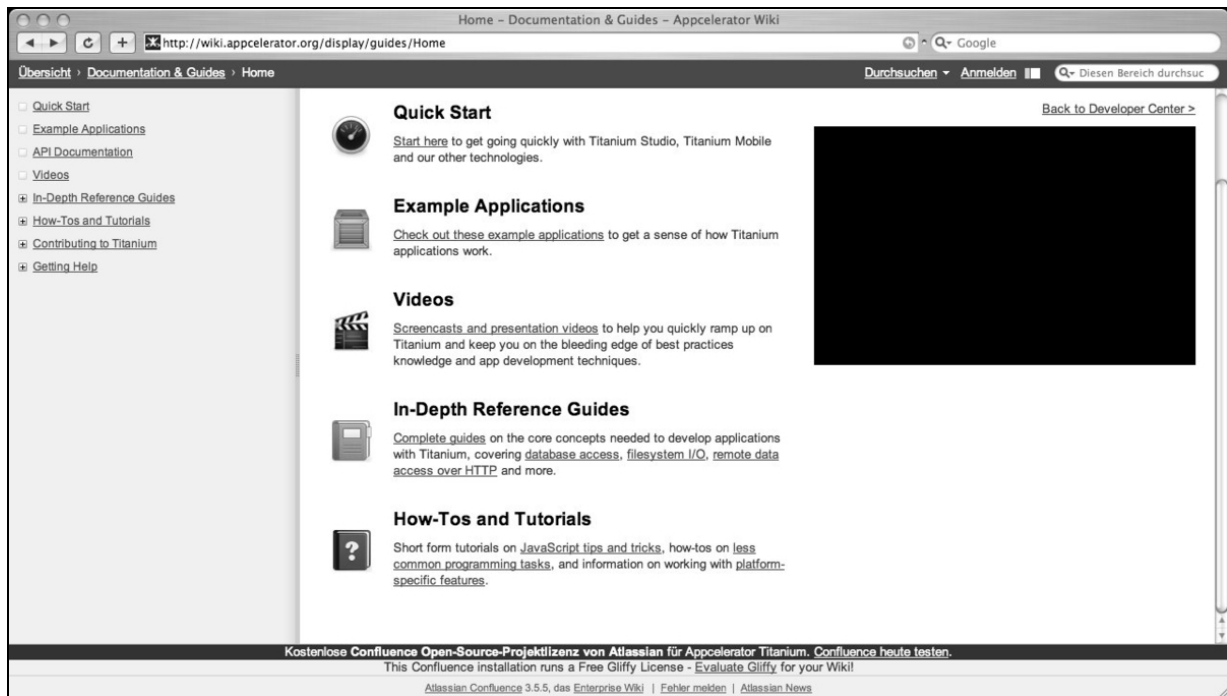


Bild 2.1 Das Hilfe-Center von Appcelerator

Für weitergehenden Support müssen Sie zahlen. Unter http://developer.appcelerator.com/get_help haben Sie die Übersicht über den kommerziellen Support von Appcelerator. Dort können Sie zwischen Custom Training, Professional Services und Premium Support auswählen.

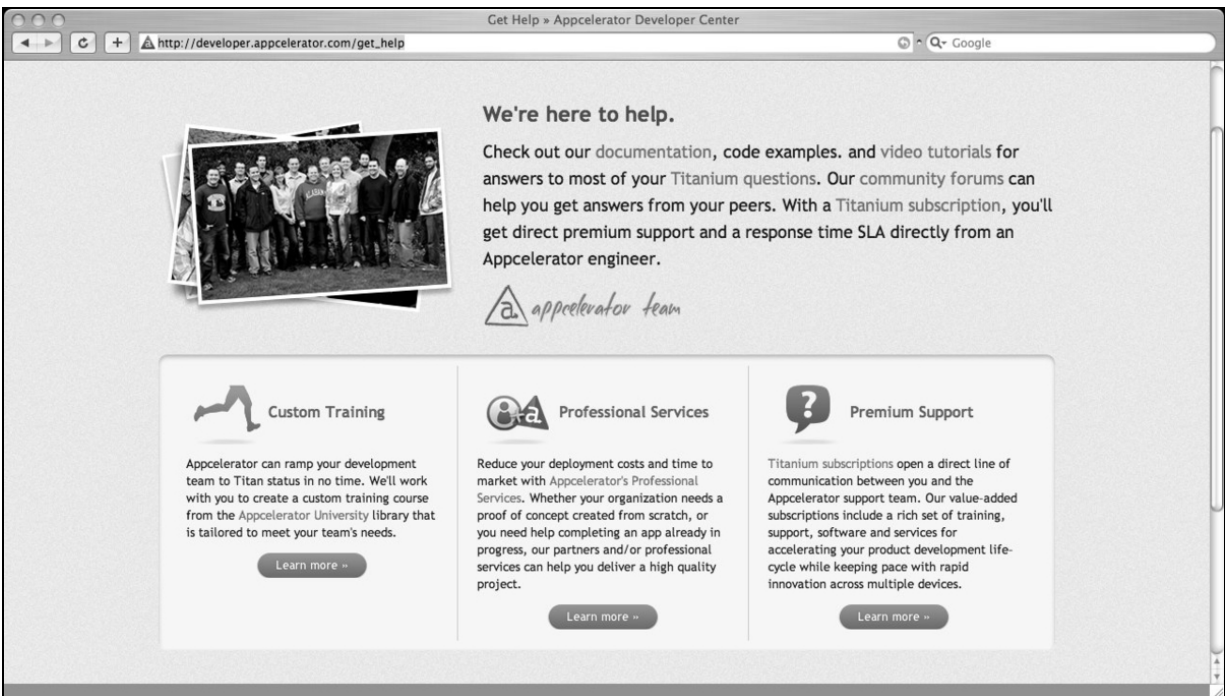


Bild 2.2 Professionelles Training für die Titanium-Entwickler

Titanium Studio läuft auf den Plattformen Mac OS X, Windows und Linux. Es werden Windows XP und Windows 7, nicht aber Windows Vista, NT, 98 und 95, unterstützt.

Aus einem Quellcode heraus können dann Desktopanwendungen für Windows, Mac OS und Linux erstellt werden. Viele bekannte Software-Hersteller wie zum Beispiel Ebay nutzen Titanium, um Ihre Programme für alle Betriebssysteme zur Verfügung stellen zu können.

Für den Start melden Sie sich unter der URL:

<https://my.appcelerator.com/auth/signup/offer/community>
an. Danach starten Sie den Download von Titanium Studio.

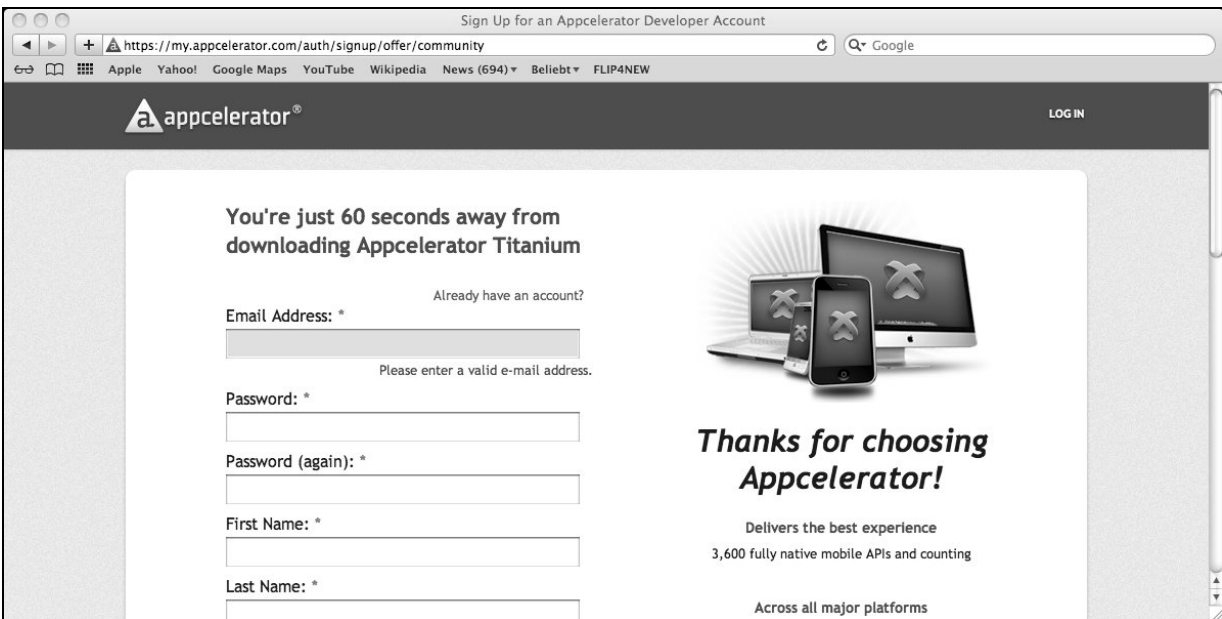


Bild 2.3 Das Anmelde-Prozedere bei Appcelerator

Die Systemvoraussetzungen

Titanium Studio läuft ab Mac OS X 10.6, Windows XP und Ubuntu Linux 10.04.

Operating System	Min Version	Max Version
Apple Mac OSX	10.6.X (Snow Leopard)	10.7 (Lion)
Windows XP (all editions, except <u>Starter Edition</u>)	Service Pack 3	Latest
Windows 7	All	Latest
Ubuntu Linux	10.04 LTS (Lucid Lynx)	10.04 LTS (Lucid Lynx)

⚠ Windows Vista, NT, 98 and 95 are not supported.

Bild 2.4 Die Systemvoraussetzungen für Titanium