

Inhoud

1	Kennismaken met PhoneGap	1
	Kennismaken met PhoneGap	2
	Native apps versus hybride apps	3
	Waarom PhoneGap gebruiken?	5
	PhoneGap versus Cordova	7
	Geschiedenis – Nitobi, Adobe en Apache	7
	PhoneGap op internet	9
	Versies van PhoneGap	10
	Geen kosten verbonden aan PhoneGap-ontwikkeling	11
	Manieren om PhoneGap te gebruiken	11
	PhoneGap Build	12
	Lokale installatie	13
	PhoneGap Developer App	14
	Benodigde voorkennis	15
	Tips voor meer leesvoer	17
	Apparaten	17
	De ontwikkelomgeving inrichten	20
	Editor en browser	20
	Andere tools	21
	Debugging voor Android	22
	Debugging voor iOS	23
	Oefenbestanden downloaden	25
	Voorbeeldapp in de stores	25
	Conclusie	26
	Praktijkoefeningen	27

2	Uw eerste PhoneGap-app	31
	 Beginnen met een HTML-app	32
	De startpagina ontwerpen	32
	De vervolgpagina maken	34
	Eisen voor de structuur van de app	36
	De app inpakken	37
	 Aanmelden bij PhoneGap Build	38
	 De app uploaden	40
	 De app installeren	42
	QR-code Reader	42
	De app terugvinden	45
	Troubleshooting	46
	 De app bijwerken	47
	Build cycle	48
	 Conclusie	49
	 Praktijkoefeningen	50
3	PhoneGap Build configureren	53
	 Configuratieopties bij PhoneGap Build	54
	 Eenvoudige configuratieopties instellen via Settings	55
	Eigen icon uploaden	56
	Nieuwe build testen	58
	 Uitgebreide configuratieopties instellen met config.xml	59
	Documentatie voor config.xml	59
	Een config.xml maken	60
	Config.xml uitbreiden	62
	Icon en splash screen	62
	Retina- en andere hogeresolutieschermen	63
	Doelplatform instellen	64
	Oriëntatie instellen	64
	Overige instellingen	64
	 Config.xml testen	66
	Settings bekijken	66
	 Hydration gebruiken	67
	Wat doet Hydration?	67
	Hydration inschakelen	68
	De app updaten	69
	Hydration uitschakelen	70
	 Conclusie	71
	 Praktijkoefeningen	72

4	PhoneGap lokaal installeren	75
	Randvoorwaarden bij lokale installatie	76
	SDK's per platform	76
	Emulator per platform	78
	Installatie voor Android	78
	Installatievolgorde	79
	Lokale installatie voor Android-development	79
	Stap 1. Controleer Java	80
	Stap 1b. Installeer Java	81
	Stap 2. Java aan PATH toevoegen	83
	Stap 3 Apache Ant toevoegen	85
	Stap 3a. Apache Ant aan PATH toevoegen	86
	Stap 4. Omgevingsvariabelen instellen	88
	Stap 5. Android Developer Tools installeren	88
	Stap 6. De Android SDK Manager starten	91
	Stap 6a. Android Developer Tools aan PATH toevoegen	92
	Stap 7. NodeJS controleren en installeren	93
	Stap 8. Cordova installeren	94
	Een lokaal Cordova-project maken	95
	Een platform toevoegen	96
	Android Emulator installeren	98
	De app starten in de emulator	99
	De structuur van Cordova-projecten	100
	De basisapp aanpassen	102
	De app draaien op eigen device	104
	Opdrachten van de Cordova CLI	104
	Cordova lokaal installeren op een Mac	106
	Conclusie	108
	Praktijkoefeningen	109
5	Basisplug-ins gebruiken	111
	Wat zijn PhoneGap plug-ins?	112
	PhoneGap 3.0 en hoger	113
	Oorspronkelijke plug-ins	114
	Documentatie over plug-ins	115
	Plug-ins gebruiken bij PhoneGap Build	116
	Plug-ins lokaal installeren	117
	Informatie bij de online voorbeeldcode	119
	Apparaatinformatie opvragen via de plug-in Device	120
	De plug-in toevoegen	120
	De HTML aanpassen	121
	De scriptpagina schrijven	122

De event deviceready	124
Multi page versus SPA	124
De plug-in Network information	126
Network Information toevoegen	127
De HTML aanpassen	127
JavaScript schrijven voor de plug-in Network Information	128
Events van de plug-in Network information	130
De plug-in InAppBrowser	132
InAppBrowser toevoegen	132
De HTML uitbreiden	133
JavaScript schrijven voor de plug-in	134
Conclusie	136
Praktijkoefeningen	137
6 Complexe plug-ins gebruiken	141
Algemene werking van asynchrone plug-ins	142
De plug-in Camera	143
De plug-in Camera toevoegen	144
De HTML aanpassen	144
Camera openen via deviceready	145
Success- en error-callback	147
Foto ophalen uit camera roll	148
destinationType en sourceType	148
Tips voor foto's	149
Geolocation gebruiken	151
Plug-in Geolocation toevoegen	152
HTML aanpassen	152
Stijl toevoegen voor kaart	153
JavaScript schrijven voor geolocation	154
Success- en error-callback schrijven	155
De code debuggen	156
Een markering (speld) plaatsen op de kaart	157
Een serie markeringen plaatsen	158
Meer opties voor geolocation	160
De functies watchPosition() en clearWatch()	161
Meer tips voor plug-ins	163
Conclusie	165
Praktijkoefeningen	166

7	Werken met plug-ins van derden	169
	Plug-ins van derden vinden en gebruiken	170
	Waar vind ik plug-ins van derden?	170
	Plug-ins per platform	170
	Plug-ins installeren	173
	Git installeren	173
	Een BarcodeScanner maken	175
	De plug-in toevoegen	176
	HTML schrijven voor de BarcodeScanner	177
	JavaScript voor de BarcodeScanner	177
	BarcodeScanner uitbreiden	179
	Delen via sociale media	182
	X-services en Eddy Verbruggen	182
	Social Sharing plug-in toevoegen	182
	HTML schrijven voor Social Sharing	183
	JavaScript schrijven voor Social Sharing	184
	Delen naar specifieke platformen via shareVia	186
	Meer opties voor Social Sharing	188
	Advertenties tonen in PhoneGap-apps	189
	Waarom advertenties?	189
	Algemeen stappenplan	191
	Stap 1. Aanmelden bij AdMob	192
	Stap 2. Nieuwe app aanmelden (Monitize)	193
	Stap 3. AdMob-plug-in installeren	196
	Stap 4. JavaScript code voor de AdMob-plug-in	197
	Stap 5. Test de app	198
	Overige opties van de AdMob-plug-in	199
	Meer plug-ins die samenwerken met externe diensten	199
	Conclusie	201
	Praktijkoefeningen	202
8	Apps publiceren in Google Play	205
	Publicatie voorbereiden	206
	Waarschuwing vooraf	206
	Bronmateriaal bij publicatie	207
	Naam en package-id	208
	Screenshots maken	209
	Promotievideo's	210
	Icon maken	212
	Beschrijving en trefwoorden	212
	Een developeraccount afsluiten	213
	Startpunten voor publicatie	214

Android-apps signeren	216
Debugversie ongeschikt voor publicatie	216
Opties voor signeren	217
Stap 1 – Maak een keystore	219
Stap 2 – Maak bestand ant.properties	221
Stap 3 – Build in release mode	222
Stap 3a – Build failed?	222
Stap 4 – Ondertekenen met jarsigner	223
Stap 4a – Verificatie	225
Stap 5 – zipalign	225
Apps signeren via PhoneGap Build	226
Vergrendeling opheffen	227
Build uitvoeren	228
Android-apps publiceren in Google Play	229
APK uploaden	229
Winkelvermelding instellen	230
Prijzen en distributie	232
De app publiceren	233
De app updaten	234
Conclusie	235
Praktijkoefeningen	236
9 Publiceren in de Apple App Store	239
Publicatie voorbereiden	240
Profielen, certificaten en meer	241
Provisioning profiles	242
Workflow bij signeren	243
Stap 1 – Certificate Signing Request	244
Stap 2 – Maak certificaat op basis van CSR	246
Stap 3 – Installeer Distribution Certificate in Keychain Access	250
Stap 4 – Een App ID maken	251
Stap 5 – Distribution Provisioning Profile	254
Stap 5a – Profielen beheren via Xcode	256
Stap 6 – App bouwen met Distribution Provisioning Profile	256
Workflow bij distribueren	260
Stap 1 – Nieuwe app aanmelden bij iTunes Connect	260
Stap 2 – Versie-informatie en andere eigenschappen opgeven	261
Stap 3 – Prijsinformatie en beschikbaarheid opgeven	264
Stap 4 – Build uploaden	265
Stap 5 – Submit for review	267
Stap 6 – Waiting for review	268
App Rejected?	270

iOS apps bij PhoneGap Build	271
.p12-bestand exporteren	272
Download Provisioning Profile via portal	273
Key toevoegen aan PhoneGap Build	273
Bronnen voor meer informatie	274
PhoneGap Developer Directory	275
Conclusie	276
Praktijkoefeningen	277
Index	279

Kennismaken met PhoneGap

U wilt als webdeveloper graag eigen apps publiceren, maar hebt geen of weinig ervaring met Java of Objective C? Dan is PhoneGap geknipt voor u. PhoneGap is een JavaScript-bibliotheek die het mogelijk maakt webapps om te zetten naar standalone applicaties. Apps die u op deze wijze maakt, kunt u publiceren in de Apple App Store, de Google Play Store of de Windows Phone Store. In dit inleidende hoofdstuk maakt u kennis met PhoneGap en Cordova (en wat het verschil tussen die twee nu eigenlijk is) en leest u over welke voorkennis u moet beschikken om succesvol apps te kunnen maken op basis van uw mobiele websites.

In dit hoofdstuk:

Wat is PhoneGap en wat is het niet?

PhoneGap versus Cordova – wat is het verschil?

Manieren om PhoneGap te gebruiken.

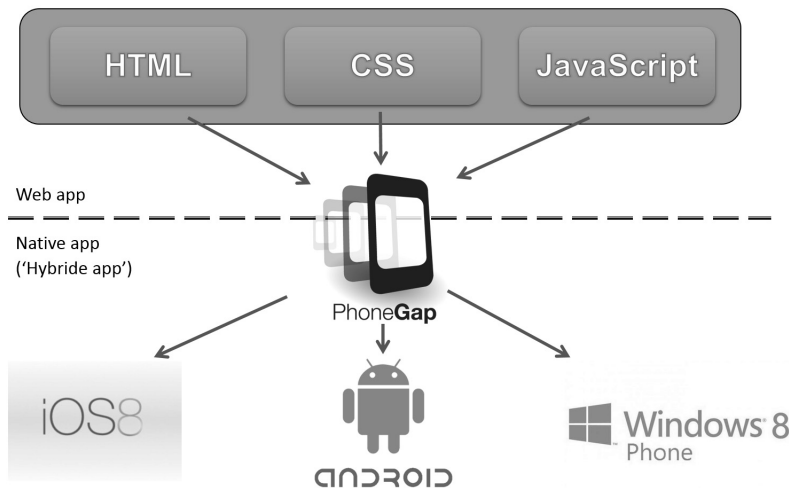
Indeling van dit boek.

De ontwikkelomgeving inrichten.

Kennismaken met PhoneGap

Er zijn allerlei manieren om apps te ontwikkelen voor smartphones en tablets. Misschien programmeert u al jarenlang Java-toepassingen. In dat geval is het een kleine stap om Android-apps te leren maken. Maar hoe zit het dan met het eveneens populaire iOS-platform van Apple? Op deze devices zijn Objective-C en Swift de toonaangevende programmeertalen. En misschien wil uw opdrachtgever ook apps uitbrengen op Windows Phone-gebied of een Universele Windows 10-app publiceren. Dan moet u dezelfde app nogmaals schrijven, maar nu in C# (de programmeertaal op het Microsoft .NET-platform). Welkom PhoneGap!

“Met PhoneGap schrijft u eenmalig uw applicatie in de talen die op het web bekend zijn: HTML, CSS en JavaScript. Daarna kunt u dezelfde broncode publiceren naar alle toonaangevende mobiele platformen: Android, iOS en Windows Phone.”



Afbeelding 1.1 PhoneGap verpakt uw website als standalone app voor de populairste mobiele platforms.

PhoneGap is als het ware de lijm tussen een webapp (die op een webserver draait) en een standalone app, die te installeren is op een mobiele telefoon of tablet. We gaan hier in de rest van het boek nog uitgebreid op in. In afbeelding 1.1 ziet u alvast welke centrale rol PhoneGap speelt bij het maken van mobiele apps op basis van webtechnieken.



Drie platforms

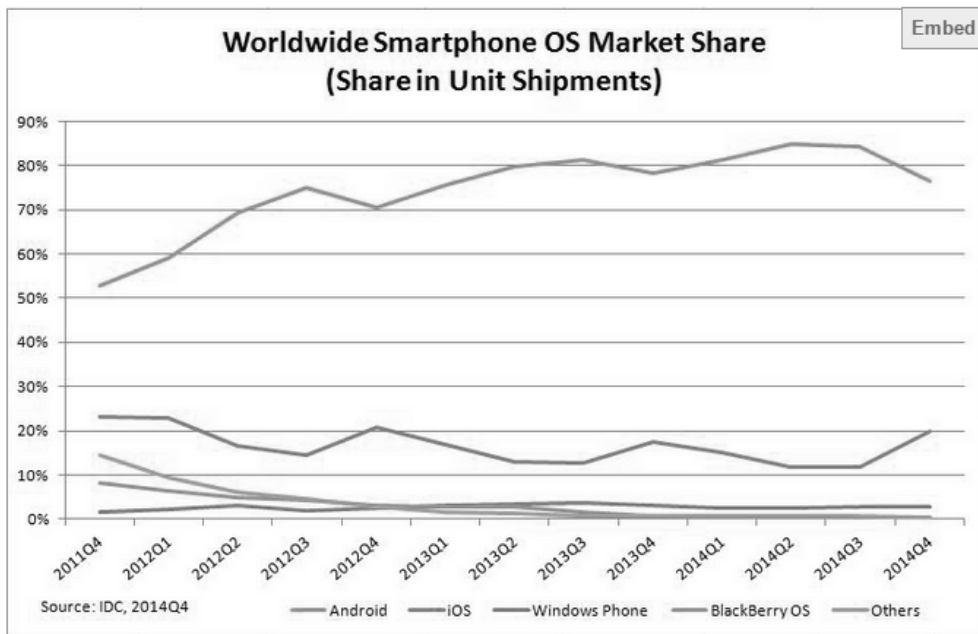
De belangrijkste mobiele platforms zijn Android, iOS en Windows Phone. Andere platforms (BlackBerry, Tizen, Amazon FireOS, Symbian) blijven buiten beschouwing. Met de huidige versies van PhoneGap kunt u er ook geen applicaties meer voor maken. Ook in dit boek worden ze niet besproken. Moet u toch nog (bijvoorbeeld) een BlackBerry-app opleveren, gebruik dan een oudere versie van PhoneGap, bijvoorbeeld PhoneGap 2.9.

Native apps versus hybride apps

Zeker, het schrijven van een applicatie in de programmeertaal die speciaal bij het betreffende platform hoort, biedt de meeste flexibiliteit en de beste performance. We noemen dit ook wel *native apps*. Maar het probleem werd hiervoor al geschetst. Zodra u de eenmaal geschreven applicatie wilt uitbrengen op een ander platform, merkt u dat u vrijwel opnieuw kunt beginnen. Immers: Java-code wordt niet begrepen door iPhones en iPads, terwijl Objective C-code niet wordt ondersteund op Android-apparaten en Windows-smartphones. Een native app uitbrengen op de drie populaire platformen kost daarmee drie keer tijd en drie keer budget.

Dit zijn de huidige populaire platforms en de native programmeertalen die daar bij horen:

- **Android** Het populairste mobiele platform. Het marktaandeel van Android nadert in Europa de 80% (op het moment van schrijven van dit boek). Java is de programmeertaal die hiervoor wordt gebruikt.
- **iOS** Het besturingssysteem van Apple. Het marktaandeel ligt tussen de 12% en 20%, afhankelijk van het land. Apps voor iOS worden geschreven in Objective C of Swift, of een combinatie hiervan.
- **Windows Phone** Het mobiele platform van Microsoft. Het marktaandeel is nog niet hoog met ongeveer 3% tot 5% van



Period	Android	iOS	Windows Phone	BlackBerry OS	Others
Q4 2014	76.6%	19.7%	2.8%	0.4%	0.5%

Afbeelding 1.2 Met enige regelmaat verschijnen updates over de stand van zaken in de mobiele markt. Het algemene beeld is dat Android verreweg het grootste is (bron: idc.com).

het aantal verkochte mobiele apparaten, maar mogelijk gaat Windows 10 hier verandering in brengen. De programmeertaal die native wordt gebruikt is C# met XAML.



Let op uw doelgroep

Als algemene tip kan worden gegeven dat wanneer u apps minimaal voor Android uitbrengt, u waarschijnlijk het grootste deel van uw doelgroep al bereikt. Maar let wel op specifieke groepen. Bijvoorbeeld in de medische- en zakelijk/economische wereld is het aandeel van Apple/iOS veel groter dan onder thuisgebruikers. In dat geval moet publicatie in de Apple App Store uw eerste doel zijn. Doe altijd een goed doelgroepenonderzoek voor de app die u gaat maken.

Waarom PhoneGap gebruiken?

Alle mobiele platformen zijn zonder meer geschikt om mobiele webapps te tonen. Ga bijvoorbeeld maar eens naar **m.facebook.com**, **m.nos.nl** of **www.nu.nl** op een smartphone. Dit zijn in feite gewone websites die met een framework als Twitter Bootstrap of jQuery Mobile (of met gewoon handgecodeerde HTML en CSS) zijn geoptimaliseerd voor weergave op een mobiel apparaat.

Waarom zou u dan toch de moeite nemen een aparte mobiele app te maken? Er zijn verschillende redenen voor te bedenken.

- **Standalone app** Door een standalone app te maken van een mobiele website bespaart u de bezoeker de moeite om een browser (Safari, Chrome, Internet Explorer Mobile) te hoeven starten en handmatig het webadres in te hoeven voeren. Het

is gebruikersvriendelijk. PhoneGap is in deze functie vooral een *wrapper* rondom uw webapp.

- **Extra functies** Via PhoneGap krijgt u toegang tot veelgebruikte apparaatsensors zoals de camera, accelerometer, contactenlijst, het kompas en meer. Dit kan niet in een gewone website. Hoewel veel functionaliteit inmiddels ook beschikbaar is via HTML5 API's (denk aan `Geolocation` en `localStorage`), biedt HTML5 geen mogelijkheden om te gaan met native device sensors of aanvullende diensten zoals delen op sociale media of Push Notification. PhoneGap vervult in dat geval een *bridge*-functie.
- **Marketing en promotie** Door een app te publiceren zet u zichzelf en uw bedrijf of organisatie op de kaart. U bent rechtstreeks aanwezig in de broekzak van duizenden mensen. Dat heeft niks met techniek te maken, maar alles met psychologie, marketing en promotie. PhoneGap is in dat geval een promotionele tool.



In dit boek: wrapper en bridge

In dit boek kijken voornamelijk naar de eerste twee aspecten. U leert als eerste hoe u de wrapperfunctie van PhoneGap optimaal inzet met PhoneGap Build en via het lokale buildproces. Daarna gaat u aan de slag met de bridgefunctie van PhoneGap. U leert hoe u bijvoorbeeld foto's maakt met de camera of vanuit uw app QR-codes kunt scannen. Marketing en promotie van apps laten we vervolgens over aan de afdeling communicatie van uw bedrijf. Die zijn daar veel beter in.

PhoneGap versus Cordova

Zodra u met PhoneGap aan de slag gaat komt u ongetwijfeld ook het begrip *Cordova* tegen, of misschien hebt u dit al eens gehoord. Om de verschillen en overeenkomsten tussen deze twee technieken goed te kunnen begrijpen, is een klein geschiedenislesje op zijn plaats.

Geschiedenis – Nitobi, Adobe en Apache

Nitobi is het bedrijfje dat PhoneGap rond 2009 oorspronkelijk ontwikkeld heeft. Internetgigant Adobe (bekend van Photoshop, Illustrator en Dreamweaver) zag de potentie van PhoneGap en heeft Nitobi in 2011 gekocht en daarmee het product PhoneGap. Omdat er inmiddels echter al een grote opensourcecommunity rondom PhoneGap geformeerd was en Adobe deze niet van zich wilde vervreemden, is de broncode van PhoneGap teruggegeven aan de opensourcegemeenschap.



Afbeelding 1.3 Van Nitobi naar Adobe PhoneGap naar Apache Cordova. Voor ons als gebruikers maakt het niet zoveel uit.

Onder beheer van de Apache Foundation (www.apache.org) werd opensourceonderhoud van de tool gegarandeerd. Maar omdat Adobe de rechten op de merknaam PhoneGap behield, moest Apache er een andere naam voor verzinnen. Dat is Cordova geworden.



Een merknaam

Het belangrijkste verschil is eigenlijk van juridische aard: zodra u een uiting ziet met de naam PhoneGap er in (zoals PhoneGap Build), dan weet u dat dit een merknaam is, en dus eigendom is van Adobe. Als u het begrip Cordova ziet, dat weet u dat u met de opensourcevariant te maken hebt.

Adobe verwoord het verschil op de volgende manier:

“You can think of Apache Cordova as the engine that powers PhoneGap, similar to how WebKit is the engine that powers Chrome or Safari.

(...)

Over time, the PhoneGap distribution may contain additional tools that tie into other Adobe services, which would not be appropriate for an Apache project. PhoneGap will always remain free, open source software and will always be a free distribution of Apache Cordova.”

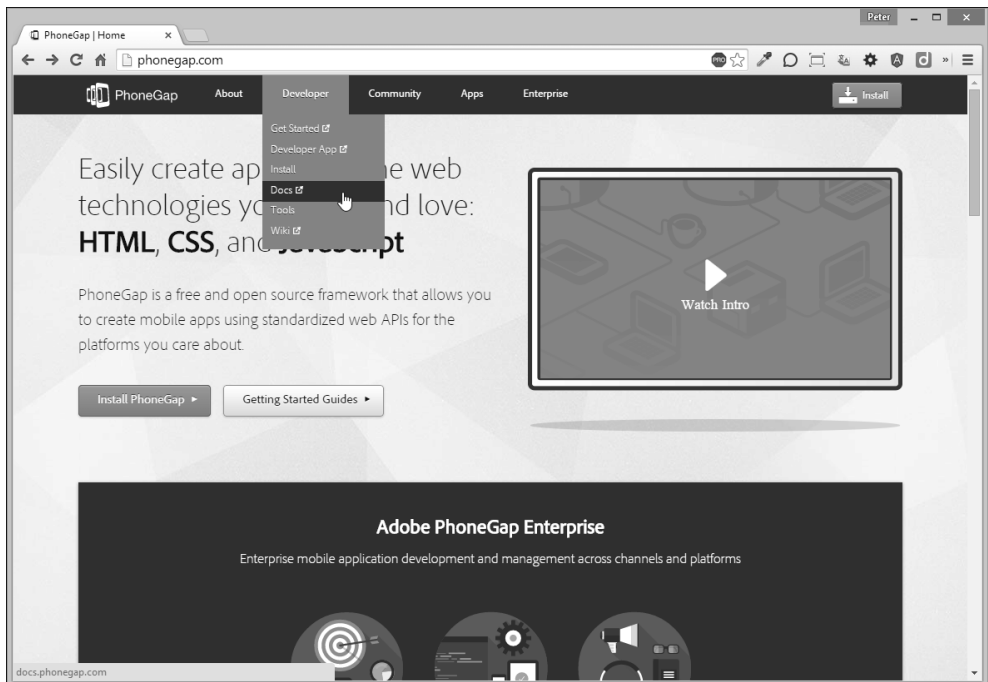
— Brian Leroux, Adobe

Technisch gezien zijn PhoneGap en Cordova dus twee verschillende dingen. In de praktijk is er echter amper onderscheid te merken. Zelfs de opdrachten in de *command line interface (CLI)* zijn in versie 4.0 weer gelijkgetrokken. In dit boek worden – net als op de rest van internet – de termen PhoneGap en Cordova

door elkaar gebruikt. We zullen PhoneGap vooral gebruiken als het gaat om de online omgeving PhoneGap Build en Cordova vooral als we hebben over de lokale installatie ervan, op uw eigen Mac of pc.

PhoneGap op internet

De homepage van PhoneGap is te vinden op **phonegap.com**. Via de site kunt u het framework downloaden, online tutorials volgen, documentatie lezen, video's bekijken, het PhoneGap-blog volgen en meer. Ook is dit het startpunt voor de officiële documentatie. Kies hiervoor de optie **Developer, Docs** uit het hoofdmenu, of voeg **docs.phonegap.com** direct toe aan uw favorieten.



Afbeelding 1.4 De homepage van PhoneGap op internet. Bezoek vooral de sectie met documentatie om meer te leren over de PhoneGap-achtergronden.

Versies van PhoneGap

Zoals we verderop zullen zien zijn er verschillende manieren om PhoneGap te gebruiken. U kunt een online omgeving gebruiken om PhoneGap-apps te bouwen (deze heet PhoneGap Build) of u kunt PhoneGap lokaal installeren op uw computer. Hoewel PhoneGap dus nog maar amper vijf jaar oud is, kent het al een rijke versiehistorie. Dit zijn de belangrijkste mijlpalen

- **PhoneGap 1.x** De basisversie uit de jaren 2010-2012. Er was ondersteuning voor zeven mobiele platforms en de basis-mogelijkheden van de apparaten (API's voor de camera, contactenlijst, accelerometer) werden ondersteund.
- **PhoneGap 2.x** Het aantal API's werd uitgebreid en toegevoegd aan de PhoneGap-core. Elk platform had zijn eigen installatietool: een wizard voor Windows, een .dmg-schijfimage voor MacOS en een commandlinetool voor Linux. Er ontstonden talloze plug-ins die de mogelijkheden van PhoneGap verder uitbreidden.
- **PhoneGap 3.x** De PhoneGap *core* en de API's werden van elkaar gescheiden. Elke API is sindsdien een aparte plug-in die geïnstalleerd moet worden, net als plug-ins van derden. Dit betekent dat de PhoneGap core los van de plug-ins kan worden geüpdatet en omgekeerd. Sinds PhoneGap 3.0 wordt NodeJS gebruikt als basisplatform. Er zijn geen aparte installatieprogramma's meer voor Windows, Macintosh en Linux.
- **PhoneGap 4.x** Beschikbaar sinds januari 2015. Er zijn talloze bugfixes, performanceverbeteringen en een aantal wijzigingen in de opensourcelicentie aangebracht. Platforms die wereldwijd minder dan 5% marktaandeel hebben worden niet meer vanuit het team ondersteund. Functioneel gezien is de werking van 4.x zo goed als gelijk aan 3.x.



In dit boek: Cordova 4.2.0

In dit boek gebruiken we Cordova versie 4.2.0 uit het voorjaar van 2015. Dit was op het moment van schrijven de nieuwste versie. De nieuwste versie die via PhoneGap Build werd aangeboden was versie 3.6.3. Functioneel gezien zijn de verschillen niet groot. Alle voorbeelden en oefeningen in dit boek kunt u volgen met alle PhoneGap-versies sinds PhoneGap 3.2.

Geen kosten verbonden aan PhoneGap-ontwikkeling

Een van de belangrijkste dingen om te onthouden is dat het met PhoneGap mogelijk is geworden om zonder kosten apps te ontwikkelen. PhoneGap zelf is open source en daarmee gratis te downloaden en te gebruiken. De belangrijkste ontwikkeltools voor alle platforms zijn gratis verkrijgbaar (in de afzonderlijke hoofdstukken gaan we hier nog op in), van PhoneGap Build is een gratis versie beschikbaar en er zijn tal van opensourcewebeditors beschikbaar. In principe kunt u alles gratis uitvoeren, tot aan het testen op uw (Android-)device aan toe.

Pas op het moment dat u een app in een van de Apps Stores wilt publiceren, moet u de creditcard trekken. Daar gaan we vanaf hoofdstuk 8 dieper op in.

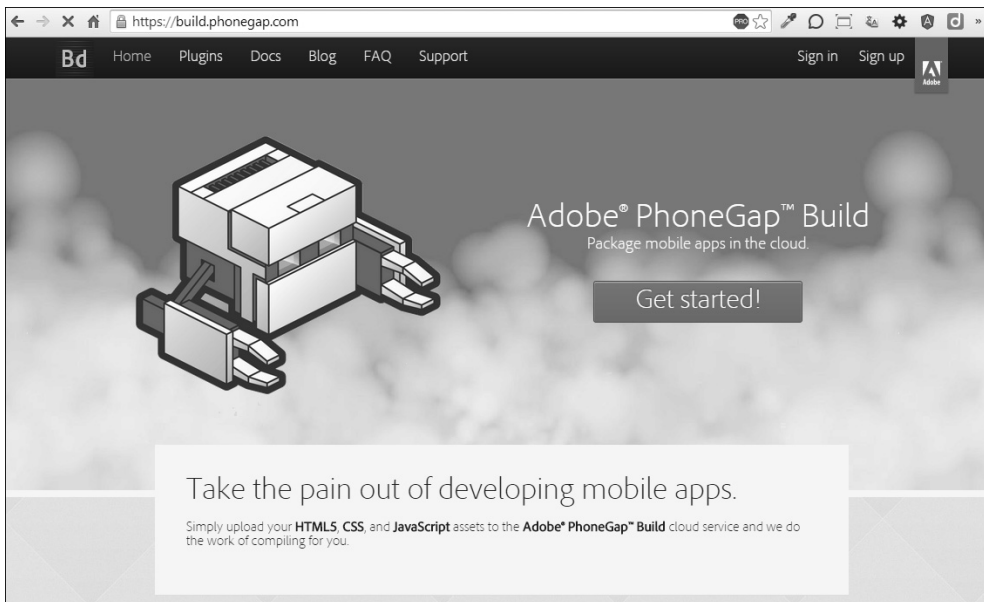
Manieren om PhoneGap te gebruiken

Zoals u misschien al hebt begrepen zijn er verschillende manieren om te profiteren van alle PhoneGap-mogelijkheden. In de volgende paragrafen worden ze kort genoemd, waarna we in de rest van het boek uitgebreid ingaan op de twee meest gebruikte methoden.

PhoneGap Build

De eenvoudigste manier om te beginnen is waarschijnlijk door gebruik te maken van PhoneGap Build. Dit is een online tool die Adobe in 2012 heeft gelanceerd. U hoeft er niks voor te downloaden of te installeren. Het stappenplan kan waarschijnlijk niet eenvoudiger:

- 1 Maak een app op uw eigen computer en test hem in de browser (u maakt dus eigenlijk een mobiele website).
- 2 Maak een Adobe-account aan bij PhoneGap Build of gebruik uw bestaande Adobe ID als u al een Creative Cloud-abonnement hebt.
- 3 Verpak de site in een zipbestand en upload de zip naar PhoneGap Build.
- 4 Wacht twee of drie minuten terwijl Build het werk voor u doet. Download daarna de app als .apk (Android), .ipa (iOS) of



Afbeelding 1.5 Zonder moeite apps maken met de online dienst PhoneGap Build. Geen installatie benodigd.

.xap (Windows Phone). Of, nog eenvoudiger, scan de QR-code die PhoneGap Build u toont en installeer de app rechtstreeks op uw telefoon.

Als u maar zo nu en dan een appje maakt, snel resultaat wilt of geen zin hebt in ingewikkelde installatieprocedures is PhoneGap Build een buitengewoon geschikt startpunt. Vanaf hoofdstuk 2 maakt u kennis met deze tool.

Lokale installatie

Wanneer u meer grip wilt op het complete proces en niet afhankelijk wilt zijn van online diensten – die zoals bekend wel eens kunnen uitvallen of op zich laten wachten – dan is een lokale installatie van PhoneGap het overwegen waard. Er zijn dan wel wat hordes te nemen (installatie van NodeJS, Apache Ant, de Java

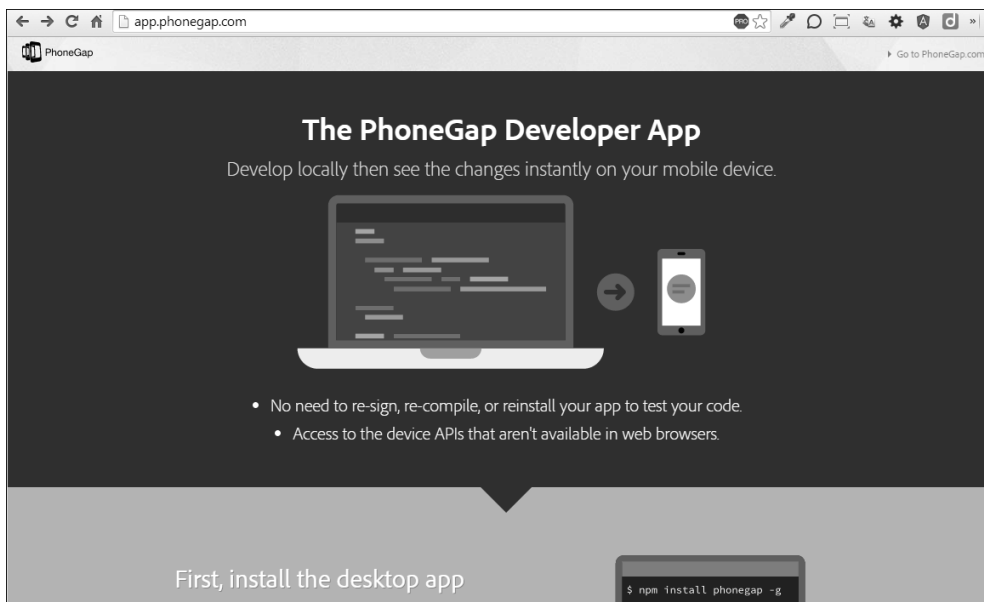


Afbeelding 1.6 Een lokale installatie van PhoneGap/Cordova biedt de meeste flexibiliteit.

SDK en meer), maar het resultaat is de moeite waard. U kunt dan volledig onafhankelijk ontwikkelen en kunt elke kleine aanpassing aan de app snel beoordelen. Hiervoor is het wel nodig dat u over het platform beschikt waarvoor u wilt ontwikkelen. Met andere woorden: het is niet mogelijk om iOS-apps te maken op een Windows-machine. Het omgekeerde kan ook niet. U kunt geen Windows Phone-apps maken op een Mac. Een lokale installatie van Cordova wordt vanaf hoofdstuk 4 behandeld.

PhoneGap Developer App

In het najaar van 2014 is nog een derde mogelijkheid aan het rijtje PhoneGap-mogelijkheden toegevoegd: de PhoneGap Developer App. In de App Stores van Apple, Google Play en Microsoft kunt u de PhoneGap Developer App downloaden. U kunt daarna de app lokaal ontwikkelen (op uw eigen computer) en deze rechtstreeks beoordelen op uw telefoon, zonder dat u hem apart hoeft te installeren op het apparaat. Hiervoor serveert de PhoneGap com-



Afbeelding 1.7 Met de PhoneGap Developer App kunt u een app rechtstreeks vanuit de browser testen op uw telefoon. De techniek is echter nog niet volmaakt.

mand line interface de applicatie op een bepaald IP-adres en een bepaalde poort in uw netwerk (bijvoorbeeld `http://192.168.1.100:8000`) en de PhoneGap Developer App ‘kijkt’ naar dit adres. Hij laadt de app die u hebt gemaakt dan binnen zijn eigen lege kapstokomgeving en u ziet de app zoals een eindgebruiker hem ook zal ervaren.

Wijzigingen die u vervolgens lokaal aanbrengt kunnen snel worden gesynchroniseerd met de versie in de Developer App, zonder dat een nieuwe compilatie- en installatieslag nodig is. Wilt u hier meer over weten, kijk dan op app.phonegap.com.



Geen Developer App

In dit boek maken we geen gebruik van de PhoneGap Developer App. Hoewel de techniek veelbelovend is (en ook andere frameworks op deze manier experimenteren) was de techniek op het moment van schrijven nog niet goed genoeg om serieus mee te werken. Regelmatig crashte de Developer App, of moesten we hem handmatig afsluiten en daarna opnieuw opstarten om aangebrachte wijzigingen binnen de app te zien. Het vermeende tijdsvoordeel werd daarmee direct teniet gedaan. Reden voor ons om deze app op dit moment nog terzijde te leggen en ons te concentreren op PhoneGap Build en op een lokale installatie van PhoneGap.

Benodigde voorkennis

Dit boek maakt deel uit van de *Web Development Library* (www.webdevelopmentlibrary.nl). In elk deel wordt een op zichzelf staande techniek besproken die te maken heeft met webdevelopment. Andere, gerelateerde technieken worden bekend ver-

ondersteld. Zo betaalt u alleen voor datgene wat u echt nodig heeft.

- In dit boek gaan we in op PhoneGap/Cordova 4.2. Aan het werken met PhoneGap liggen de webtechnieken HTML, CSS en JavaScript ten grondslag. We gaan er dan ook van uit dat u hier voldoende ervaring mee hebt.
- We gebruiken bij de voorbeelden soms andere bibliotheken zoals jQuery (jquery.com) en Twitter Bootstrap (getbootstrap.com). Meestal leggen we bij de codevoorbeelden kort uit wat er gebeurt, echter zonder al te diep in te gaan op deze randzaken. Ze worden bekend verondersteld.
- Wanneer we aan de slag gaan met plug-ins om bijvoorbeeld de camera op een apparaat te gebruiken, wordt deze aangesproken via JavaScript. U moet bekend zijn met JavaScript-functies en -variabelen en het principe van asynchrone callbackfuncties in JavaScript.
- Veel documentatie of aanvullende instructievideo's zijn alleen beschikbaar in het Engels. We verwijzen vaak naar online (Engelstalige) documentatie.



Wat hoeft u niet te weten?

Diepgaande kennis van andere JavaScript-bibliotheken (zoals jQuery) of -frameworks is niet nodig. Ook hoeft u niks te weten van server-sided talen zoals PHP, Python, Ruby, Java of C#. U mag specifieke tools als Eclipse (Java) of Xcode (Objective C) gebruiken, maar noodzakelijk is dit niet.
