

PhoneGap

Mobile Cross-Plattform-Entwicklung mit Apache Cordova und Co. --> Mit einem Geleitwort von Michael Brooks,
PhoneGap Core Contributor

von
Marcus Ross

1. Auflage

PhoneGap – Ross

schnell und portofrei erhältlich bei beck-shop.de DIE FACHBUCHHANDLUNG

Thematische Gliederung:

PDA & Handheld Programmierung

dpunkt.verlag 2013

Verlag C.H. Beck im Internet:

www.beck.de

ISBN 978 3 89864 824 0

Inhaltsverzeichnis

1	Entwickeln mit PhoneGap	1
1.1	Das Cross-Plattform-Problem	1
1.2	Wozu braucht man PhoneGap?	1
1.3	Die Architektur von PhoneGap	2
1.4	PhoneGap oder Apache Cordova?	4
1.5	Zusammenfassung	7
2	Von der Idee zur App	9
2.1	Von der App-Idee zur Skizze	9
2.2	Ein Storyboard erstellen	10
2.3	Einen ersten Prototyp entwickeln	11
2.3.1	Das UI-Framework festlegen	12
2.3.2	Die richtigen Tools für die Features auswählen	14
2.4	Die App implementieren	15
2.4.1	Die Entwicklungsumgebung einrichten	15
2.4.2	Den App-Lifecycle berücksichtigen	15
2.4.3	Eine Versionsverwaltung für den Code nutzen	16
2.4.4	Die Anwendung testen lassen	16
2.4.5	Die Fehler finden	16
2.5	Die App über die Marktplätze ausliefern	17
2.5.1	Apples App Store	17
2.5.2	Google Play Store	17

2.5.3	Microsoft Store	18
2.5.4	BlackBerry's App World	18
2.5.5	Samsung Hub (Bada)	18
2.5.6	HP App Catalog (webOS)	19
2.6	Zusammenfassung	19
3	Die Zielplattformen	21
3.1	»myDevice« – Ihre erste Cross-Plattform-App	21
3.2	iOS	23
3.2.1	Die Entwicklungsumgebung installieren	24
3.2.2	Die erste App für iOS	27
3.2.3	Der Netzwerkzugriff in iOS	31
3.2.4	Bilder und Icons anpassen	31
3.3	Android	32
3.3.1	Die Entwicklungsumgebung installieren	33
3.3.2	Die erste Android-App	37
3.4	Windows Phone 7	48
3.4.1	Die Entwicklungsumgebung installieren	49
3.4.2	Die erste App für Windows Phone 7	50
3.4.3	Wenn Bilder nicht angezeigt werden	53
3.5	Windows Phone 8	54
3.5.1	Die Entwicklungsumgebung installieren	54
3.5.2	Die erste App für Windows Phone 8	56
3.6	BlackBerry	58
3.6.1	Die Entwicklungsumgebung installieren	58
3.6.2	Die erste App für BlackBerry	64
3.6.3	Auf einem physischen Gerät installieren	66
3.6.4	Verschiedene Geräte-Simulatoren nutzen	66
3.7	webOS	68
3.7.1	Die Entwicklungsumgebung installieren	68
3.7.2	Die erste App für webOS	71

3.7.3	Applikationsmetadaten	74
3.7.4	Mit anderen webOS-Versionen testen	75
3.8	Bada	76
3.8.1	Die Entwicklungsumgebung installieren	76
3.8.2	Die erste Bada-App	77
3.8.3	Einblick in die Bada-Projektstruktur	81
3.9	Zusammenfassung	82
4	Hardwarezugriff I – häufig genutzte Features	85
4.1	Das Gerät (Device)	85
4.1.1	Ein komplettes Beispiel	86
4.2	Der Lifecycle (Events)	87
4.2.1	deviceready – das Startevent	88
4.2.2	pause – die Applikation legt sich schlafen	89
4.2.3	resume – die App wacht auf	90
4.2.4	online – eine Verbindung mit dem Internet	92
4.2.5	offline – keine Verbindung mit dem Internet	93
4.3	Die Kamera	94
4.3.1	Ein kurzes Beispiel	95
4.3.2	Das Options-Objekt	96
4.3.3	camera.getPicture – ein Foto machen	98
4.4	Geolocation	105
4.4.1	geolocation.getCurrentPosition – Positionsdaten auslesen	108
4.4.2	geolocation.watchPosition – Daten automatisch aktualisieren	111
4.4.3	geolocation.clearWatch – Datenaktualisierung aufheben	114
4.5	Der Kompass	117
4.5.1	Ein kurzes Beispiel	117
4.5.2	compass.getCurrentHeading – die aktuelle Richtung bestimmen	120
4.5.3	compass.watchHeading – Richtung automatisch aktualisieren	123
4.5.4	compass.clearWatch – Richtungsaktualisierung aufheben	125
4.5.5	Praxisbeispiel	127
4.6	Zusammenfassung	128

5	Hardwarezugriff II – erweiterter Zugriff	129
5.1	Das Adressbuch (Contacts)	129
5.1.1	contacts.create – neue Kontakte erstellen	139
5.1.2	contacts.find – Kontakte finden	142
5.2	Der Beschleunigungssensor (Accelerometer)	145
5.2.1	Ein kurzes Beispiel	146
5.2.2	accelerometer.getCurrentAcceleration – Beschleunigungswerte erfassen	147
5.2.3	accelerometer.watchAcceleration – Beschleunigungswerte aktualisieren	150
5.2.4	accelerometer.clearWatch – Datenaktualisierung beenden	154
5.2.5	Praxisbeispiel	157
5.3	Die Verbindungsqualität (Network)	158
5.3.1	Ein komplettes Beispiel	158
5.4	Die Benachrichtigung (Notification)	160
5.4.1	Ein kurzes Beispiel	160
5.4.2	notification.alert – native Meldung ausgeben	161
5.4.3	notification.confirm – nativen Auswahldialog anzeigen	164
5.4.4	notification.beep – das Smartphone piepen lassen	167
5.4.5	notification.vibrate – das Smartphone vibrieren lassen	169
5.5	Geräte-Events II – (Tasten- & System-Events)	170
5.5.1	batterystatus – Akku-/Gerätestatus auslesen	170
5.5.2	batterylow – Warnung des Ladestandes	172
5.5.3	batterycritical – Warnung vor leerem Akku	172
5.5.4	backbutton – die Zurücktaste registrieren	173
5.5.5	menubutton – die Menütaste registrieren	174
5.5.6	searchbutton – die Suchtaste registrieren	175
5.5.7	startcallbutton – die Anruftaste registrieren	175
5.5.8	endcallbutton – Anruf beenden registrieren	176
5.5.9	volumedownbutton – die Lautstärke-Taste	176

5.5.10	volumeupbutton	178
5.5.11	Praxisbeispiel	178
5.6	Zusammenfassung	179
6	Testen & Debuggen	181
6.1	Der Smartphone-Emulator Ripple	181
6.1.1	Ripple installieren	182
6.1.2	Ripple mit einer PhoneGap-Anwendung nutzen	184
6.2	Der WebInspector	187
6.2.1	WebInspector unter Windows	187
6.2.2	WebInspector unter Mac OS	189
6.2.3	Eine Webanwendung debuggen	189
6.3	WeinRe für PhoneGap	192
6.3.1	Installation per npm-Paket	192
6.3.2	WeinRe verwenden (Beispiel unter iOS)	195
6.4	Zusammenfassung	199
7	Plugins einbinden	201
7.1	Beispiel »Die Taschenlampe«	201
7.1.1	Die Zutaten	202
7.1.2	Das Rezept	202
7.2	Beispiel »Barcode-Reader«	206
7.2.1	Die Zutaten	207
7.2.2	Das Rezept	207
7.2.3	Die Barcodescanner-API	214
7.3	Beispiel »Augmented Reality«	216
7.3.1	Die Zutaten	217
7.3.2	Das Rezept	218
7.4	Fazit	220

8	PhoneGap Build – der Cloudservice	221
8.1	Erste Schritte mit dem Cloudservice	221
8.2	Ein lokales Projekt in die Cloud bringen	227
8.3	Anmerkungen zur Installation	230
8.4	Fazit	230
	Stichwortverzeichnis	231