# Training Mobiele Apps – Ministerie van Defensie - Leeswijzer

Oktober-november 2018 – Peter Kassenaar ([info@kassenaar.com](mailto:info@kassenaar.com))

## De training bestaat uit twee delen:

* Werken met Framework 7 voor de user interface en het maken van de app.
* Werken met PhoneGap/Cordova voor het bouwen van de app op device.

We doorlopen onderstaande stappen ook in de training. Ik heb hier de aanwijzingen beknopt gehouden. Als je op enig moment ‘vast’ komt te zitten, maak hiervan dan een aantekening. Dan gaan we in de training op dat punt verder.

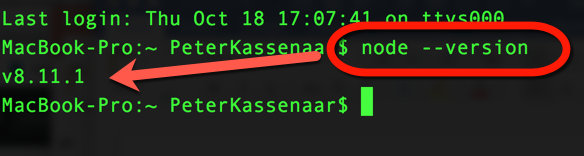
Waarschuwing: het installeren van alle afhankelijkheden voor app-ontwikkeling met PhoneGap is een lastig werkje. Er kan *van alles* mis gaan. En elk systeem is anders. Hieronder staan de algemene aanwijzingen. Ze staan ook beschreven in:

* Boek, *Hoofdstuk 4, PhoneGap lokaal installeren* (stap 3, Apache Ant nu overslaan)
* Online : <https://evothings.com/doc/build/cordova-install-windows.html> (Engelstalig, maar iets moderner).

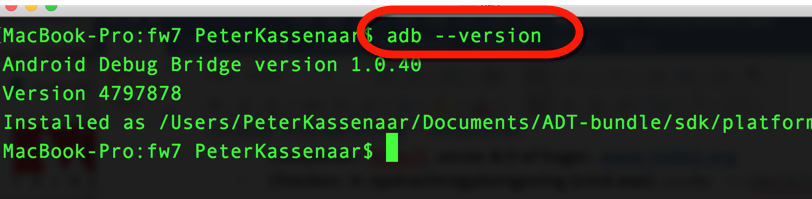
*Tip: Print bovenstaande webpagina uit, dan kun je stap voor stap afvinken wat je al hebt gedaan.*

## Globale Voorbereiding – Installaties

* Installeren: NodeJS, versie 8.0 of hoger. [www.nodejs.org](http://www.nodejs.org)
* Checken: in opdrachtregelomgeving (cmd.exe): node --version.



* Installeren: Git. Dit is een versiebeheersysteem dat door Cordova/PhoneGap op de achtergrond gebruikt wordt: <https://git-scm.com/> .
* Installeren Cordova/PhoneGap : npm install -g cordova. Zie <https://cordova.apache.org/#getstarted> voor achtergrondinformatie.
* Installeren Java: Installeer de Java JDK (let op: als er een map Java JRE aanwezig is op je computer, is dit *niet* voldoende).
  + Volg de installatie instructies in het boek (vanaf p.76), of in de online webpagina. Uiteraard zullen de versienummers al weer anders/nieuwer zijn.
* Installeren: Android Tools. Volg de installatiestappen in het boek, of uit de online pagina.
* Testen: adb (*Android Debug Bridge*) moet aanwezig zijn op je computer: adb –version.



Nu kun je in principe een nieuw Cordova/PhoneGap project beginnen, met de opdracht cordova create myDemoApp. Maar dit gaan we in het tweede deel van de training ook uitgebreid doen, besteed er voor nu nog niet al te veel tijd aan.

## Leeswijzer boek PhoneGap

* *Hoofdstuk 1* – Doorlezen. Ter kennisname.
* *Hoofdstuk 2* – Doorlezen, ter kennisname. In het boek wordt BootStrap gebruikt, in de training gaan wij Framework7 gebruiken. Je mag de stappen uitvoeren voor je begrip, maar het hoeft niet. We doen dit ook (later) in de training.
* *Hoofdstuk 3* – Voor nu, overslaan.
* *Hoofdstuk 4* – Installatie instructies voor installeren op je eigen pc. Belangrijk. Zie ook hiervoor.
* *Hoofdstuk 5* – Algemene achtergrondkennis. Doorlezen. Wordt in de training besproken.
* *Hoofdstuk 6* – Idem. Nog niet in detail doorlezen. Wordt in de training besproken.
* *Hoofdstuk 7* – Idem.
* *Hoofdstuk 8* – Publicatie in Google Play. Mag je doorlezen voor het algemene begrip (en om je eigen apps te publiceren), maar is voor MinDef waarschijnlijk minder van belang.
  + P.216-228 Het *builden* en *signeren* van de app is gewijzigd sinds het boek. In de training bespreken we de actuele situatie
* *Hoofdstuk 9* – Publicatie in Apple App Store – idem. Je mag het doorlezen voor je eigen informatie, maar voor MinDef mogelijk minder van belang?

# Voorbeeld barcode scanner plugin

Gebruikte plugin voor de barcodescanner:

cordova plugin add phonegap-barcode-scanner

documentatie: <https://github.com/phonegap/phonegap-plugin-barcodescanner>

# Video – afspelen en downloaden

1. Eerst: een lijst met af te spelen video’s hebben. Bijvoorbeeld hardcoded in de app, of anders downloaden via Ajax.
   1. Mijn voorbeeld: 'http://www.kassenaar.nl/video/video1\_20mb.mp4'
   2. Het voorbeeld staat in het project ./phonegap-www.
   3. Het voorbeeld staat in de pagina ./pages/download.html.
2. Checken of het file al bestaat. Een tekstvoorbeeld:

* <https://www.raymondcamden.com/2014/07/01/Cordova-Sample-Check-for-a-file-and-download-if-it-isnt-there>
* <https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-file/> - Zo ja, Cordova file plugin – voor toegang tot lokale(!) bestandssysteem.
* <https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-file-transfer/> - Zo nee, File Transfer, voor up- en downloaden van bestanden.

1. Als de video is gedownload of al bestaat – Video afspelen. In de voorbeeldcode wordt daarvoor de streaming media plugin gebruikt. <https://github.com/nchutchind/cordova-plugin-streaming-media> en de opdracht window.plugins.streamingMedia.playVideo(store + fileName);
2. Zo nee, downloaden volgens artikel Raymond Camden, en daarna de video player starten.

Documentatie over cordova-plugin-file: <https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-file/>

Documentatie cordova-plugin-file-transfer:

<https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-file-transfer/>

Documentatie cordova-plugin-streaming-media:

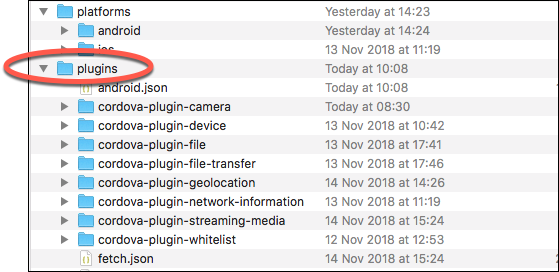
<https://github.com/nchutchind/cordova-plugin-streaming-media>

* Kijk ook naar de overige opties – deze hebben we in de training niet allemaal besproken.

File verwijderen uit sandbox binnen de app:

* Nog niet gelukt, maar het zou op deze wijze mogelijk moeten zijn: <https://www.webniraj.com/2017/08/22/cordova-deleting-a-file-using-the-file-plugin/>
* Of: <https://ourcodeworld.com/articles/read/28/how-to-delete-file-with-cordova> (dat is grotendeels dezelfde code, maar nogmaals, het is nog niet gelukt – lijkt niet goed te werken op deze wijze).

## Screenshot/overzicht van gebruikte plugins



# Algemene werkwijze

Complexe plug-ins bijna altijd volgens het principe:

* Plugin aanroepen
  + *Success* callback voor als het goed gaat,
  + *Error* callbackfunctie voor als het fout gaat
  + *Options* object om plug-in te configureren.
  + Error/Options zijn vaak optioneel (maar geven meer mogelijkheden tot finetuning).

# Vragen laatste dag:

* Verschil tussen *component* en *page*:
  + Page is een statisch geheel, alleen <div> tag, met verder andere html-tags.
  + Component is een dynamisch geheel, <template> plus <script> tag.
  + Hierin moet minimaal een leeg object worden teruggegeven. return {}
  + Het return-object kun je uitbreiden met pageInit, functions enzovoort.
* De index-pagina is hierop een uitzondering. Deze is (voor zover ik weet) altijd statisch.
  + Je *kunt* echter wel een script-tag op de pagina plaatsen, om dynamische functies uit te voeren. Zie bijvoorbeeld het voorbeeld met Google Maps op de homepage. Maar voor zover ik weet kun je geen dynamische componenten zoals Gauge, Accordeon enzovoort gebruiken.
* Barcode scanner – stappenplan:
  + Toevoegen van de juiste plugin cordova plugins add phonegap-plugin-barcodescanner
  + Documentatie lezen hoe je deze plugin kunt gebruiken. In dit geval op de site van de maker (NIET bij PhoneGap zelf): <https://github.com/phonegap/phonegap-plugin-barcodescanner>. Maar: check dit per plugin.
  + Probeer, na het toevoegen, een stukje copy/paste code te vinden dat je kunt kopieren. In ons voorbeeld staat dat in phonegap-www, pagina ./pages/barcode.html
  + De app moet worden voorbereid. In ons voorbeeld is dat:
    - Het toevoegen van de afbeelding van het gewenste voertuig.
    - Het maken van een nieuwe pagina met informatie over dit voertuig.
    - Het uitbreiden van de route-tabel in app.js met een verwijzing naar de voertuigpagina.
    - Het maken van een knop die de barcodeplugin aanroept. Hierop wordt het attribuut @click toegevoegd om de functie aan te roepen die de barcodescanner start. In ons voorbeeld is dat de functie scan.
  + De barcode plugin bestaat ook weer uit het aanroepen van een functie. Deze heeft weer twee (of drie) parameters: de success-callbackfunctie en de error-callbackfunctie
  + De derde parameter, opties, heb ik nu niet gebruikt.
  + De barcodescanner wordt geopend met de functie cordova.plugins.barcodeScanner.scan().
  + Gebruik de + en – tekens in Visual Studio Code om de code samen te vouwen en zo overzicht te houden over je (geneste) functies!
  + In de success-callbackfunctie wordt gecheckt of een goede code is gescand. Bij een resultaat (in ons voorbeeld: QR\_CODE of CODE\_128) wordt naar een bepaalde pagina in de app gesprongen (voertuigXXX.html).
  + Dit is ‘gewone’ JavaScript. Er wordt gebruik gemaakt van if() en else en meer.
  + De operator || betekend “OR” dus als de code een qr-code is, OF een barcode, dan wordt de code in dat betreffende blok uitgevoerd.
  + result.format === 'QR\_CODE' || result.format === 'CODE\_128'.
  + In het voorbeeld wordt nu gebruik gemaakt van een :
    - QR-code – spring naar pagina binnen de app
    - Barcode – spring naar google.
    - Anders – toon een alert()
  + Dit is gedaan met standaard if-statements.
* Koppelen van Visual Studio Code en Chrome voor debuggen
  + Gebruik de Chrome Debugger Extension in VS Code – eenmalige installatie via <https://marketplace.visualstudio.com/items?itemName=msjsdiag.debugger-for-chrome>.
  + De toegevoegde waarde voor Framework7-projecten is gering. Dit komt omdat je alleen breekpunten kunt plaatsen in JavaScript-bestanden. Niet in HTML-bestanden/componenten zoals in FW7 worden gebruikt.
  + Maar: als je je berekeningen uitsplitst naar verschillende JavaScript-bestanden – zoals Mark heeft gedaan – kan het weer heel nuttig zijn omdat je voor het debuggen niet hoeft te schakelen tussen twee verschillende omgevingen.
  + Stel in VS Code in het paneel Debug (Ctrl+Shift+D) een launch configuration in. Deze kan er bijvoorbeeld zo uitzien:

{

// Use IntelliSense to learn about possible attributes.

// Hover to view descriptions of existing attributes.

// For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387

"version": "0.2.0",

"configurations": [

{

"type": "chrome",

"request": "launch",

"name": "Launch Chrome against localhost",

"url": "http://localhost:5000",

"webRoot": "${workspaceFolder}/www"

}

]

}

Let op de aanpassing van de poort van je eigen (interne) webserver (hier: 5000) en de mapnaam. In PhoneGap-projecten is dit /www.