Tutoriel Apache Cordova

Ce tutoriel va couvrir l'installation de Cordova, ainsi que des différentes plateformes et plugins. L'objectif est de faire un tour des fonctionnalités proposées par Apache Cordova. Pour ce faire, nous allons installer et utiliser 4 plugins. La marche à suivre est indiquée ci-dessous :

- **Installation et création du projet :**

npm install -g cordova

cordova create tuto com.exemple.tuto TutoCordova
ou cloner la branche base_tuto : https://github.com/Quarren/tuto_cordova

- **Ajout des plateformes :**

cordova platform add browser
cordova platform add electron // si vous voulez tester, mais certains plugins ne fonctionnent pas sur Electron. Pour tester sur Android, il faut installer un émulateur (c'est long et compliqué, mais c'est expliqué en annexes).

- **Pour lancer l'application :**

cordova run <platform>
On peut spécifier le navigateur avec l'option --target=<navigateur>
Par exemple : cordova run browser --target=Chrome
Certaines fonctionnalités ne fonctionnent que sur certains navigateurs (la batterie fonctionne sur Chrome mais pas sur Firefox par exemple).

- **Plugin geolocation :**

cordova plugin add cordova-plugin-geolocation

Remplacer le placeholder par la clef API Weather dans la fonction getWeather() :
Clef API : `862214dafbb83dc2e46db490e7bc9efb`

Dans index.js, remplir la fonction getWeatherLocation() qui fait appel aux méthodes du plugin :
navigator.gelocation.getCurrentPosition()

Cette fonction renvoie un objet Position avec les propriétés suivantes : coords.latitude et coords.longitude.
Appeler la méthode getWeather(latitude, longitude) qui prend comme paramètres les latitude et longitude que vous venez de récupérer.

>>> Documentation <<<

- **Plugin battery status :**

cordova plugin add cordova-plugin-battery-status

Dans la fonction onDeviceReady, ajouter 3 eventListeners sur l'objet windows pour les événements suivants :
- batterystatus : se produit à chaque fois que la charge de la batterie change d'au moins 1 %, ou que l'appareil est branché/débranché.

- batterylow & batterycricital : se produisent lorsque la charge passe en dessous du palier ‘batterie faible/critique’ (peut différer selon les appareils).

Chacun de ces événements renvoie un objet Status avec les propriétés suivantes :
- level : le pourcentage restant de la batterie ;
- isPlugged : booléen vrai si la batterie est branchée, faux sinon.

Chacun des eventListeners doit appeler une fonction, définie ailleurs, qui alerte ou affiche dans la console le niveau de charge restant et si l’appareil est branché ou non.

>>> Documentation <<<

- **Plugin camera :**

cordova plugin add cordova-plugin-camera

Ajouter un bouton « ajouter camera » avec un id, passer cet id dans le querySelector sur lequel est ajouté l’eventListener dans la partie caméra d’index.js.
Vous pouvez ensuite tester la caméra dans la partie HTML.

>>> Documentation <<<

- **Plugin InAppBrowser :**

cordova plugin add cordova-plugin-inappbrowser

Ajouter un bouton « test IAB » dans la partie HTML, passer son id en paramètre du querySelector dans la partie IAB d’index.js.
Passer l’url de votre choix en paramètre de la méthode open() de l’objet InAppBrowser et tester le résultat.

>>> Documentation <<<

Pour aller plus loin :

- Vous pouvez essayer d’implémenter des plugins supplémentaires
- Vous pouvez essayer de créer votre propre plugin pour Apache Cordova