

**PhoneGap/Cordova
(cross-platform)**

PhoneGap ou Cordova : PhoneGap était initialement un outil propriétaire mais a été cédé à la fonction Apache et renommé pour l'occasion Cordova. À terme, PhoneGap désignera un ensemble d'outils propriétaires qui utilisent Cordova comme brique de base, mais pour l'instant, les deux peuvent s'utiliser comme synonymes car ils s'installent et se manipulent exactement de la même manière. Seul le nom de la commande change : `phonegap` ou `cordova`, mais leurs syntaxes et options sont identiques. À l'IUT, elle est installée sous le nom de `phonegap`.

De quoi s'agit-il : D'un outil qui permet de développer son application comme une application web (HTML/JavaScript/CSS) pour la compiler automatiquement pour différentes plateformes cibles, comme iOS ou Android. La technique utilisée est d'embarquer un navigateur directement dans l'application (p.ex. WebView sous Android). D'ailleurs, c'est quelque chose que vous pourriez faire directement, sans utiliser PhoneGap/Cordova. Mais dans ce cas vous n'auriez pas accès aux différentes fonctionnalités du téléphone (capteurs, contacts, appareil photo, etc). Le principal atout de PhoneGap/Cordova est de mettre à votre disposition ces fonctionnalités depuis le code JavaScript de votre application.

1 Configuration

Pour compiler l'application vers la plateforme Android, il faut que nos commandes puissent trouver le SDK. Nous allons donc modifier les variables d'environnement suivantes.

```
export ANDROID_HOME=/net/opt/Android-sdk/android-sdk-linux
export PATH=$PATH:/net/opt/Android-sdk/android-sdk-linux/tools:
/net/opt/Android-sdk/android-sdk-linux/platform-tools
```

Si vous voulez que ces changements restent valables dans le future, ajoutez ces même lignes dans votre fichier `.bashrc`

2 Premiers pas

1. Lancez un émulateur Android (par exemple, via le studio)
2. Depuis le terminal, placez-vous dans le répertoire où vous voulez créer votre application
3. Tapez `phonegap create hello` (le nom de l'appli sera hello)
4. Rendez-vous dans le répertoire `hello` avec le navigateur de fichier pour observer ce qui a été généré. Comme vous pouvez constater, nous avons là une arborescence de projet web classique. Jeter un oeil au dossier `platforms` (vide pour le moment).
5. Vous pouvez tester le rendu depuis votre navigateur web, en lui donnant le chemin (`file://chemin_vers_votre_appli`). La page `index.html` consiste en une image de robot et un message "connecting to device" qui n'aboutit par car nous sommes dans le navigateur.

6. Nous allons maintenant générer une appli Android à partir de ces sources. Dans le répertoire de votre application, tapez `phonegap prepare android`. C'est fait. Votre projet Android est maintenant dans le dossier `platforms`. Explorez avec le navigateur de fichier, ça devrait vous sembler familier.
7. Pour lancer l'application dans l'émulateur, tapez `phonegap emulate android`, toujours depuis le répertoire `hello` (c'est très lent la première fois, mais ensuite ça ira...). Si votre téléphone est branché, vous pouvez aussi l'exécuter directement dessus (`phonegap run android`).
8. Tentez de modifier votre fichier `index.html`, par exemple en ajoutant une boîte alerte `alert("coucou")` ; puis testez le résultat.

3 Votre application

Vous l'aurez compris, une application PhoneGap/Cordova, c'est une application web qui utilise des accès aux fonctionnalités du téléphone. Nous allons donc nous concentrer sur ce dernier aspect.

1. Explorez certaines des APIs disponibles sur <http://docs.phonegap.com/plugin-apis/>
2. Jusqu'à la fin de la séance, créez une application de votre choix, avec pour seule contrainte celle d'utiliser plusieurs (voire beaucoup) de ces APIs, tout en proposant des fonctionnalités cohérentes à l'utilisateur.
3. Autant que possible, je vous conseille de tester votre application en local (navigateur web) pour gagner du temps, en utilisant l'émulateur (ou téléphone) seulement de temps en temps... sinon vous passerez votre temps à attendre.