

TP : Cordova + Accès matériels + BDD

But

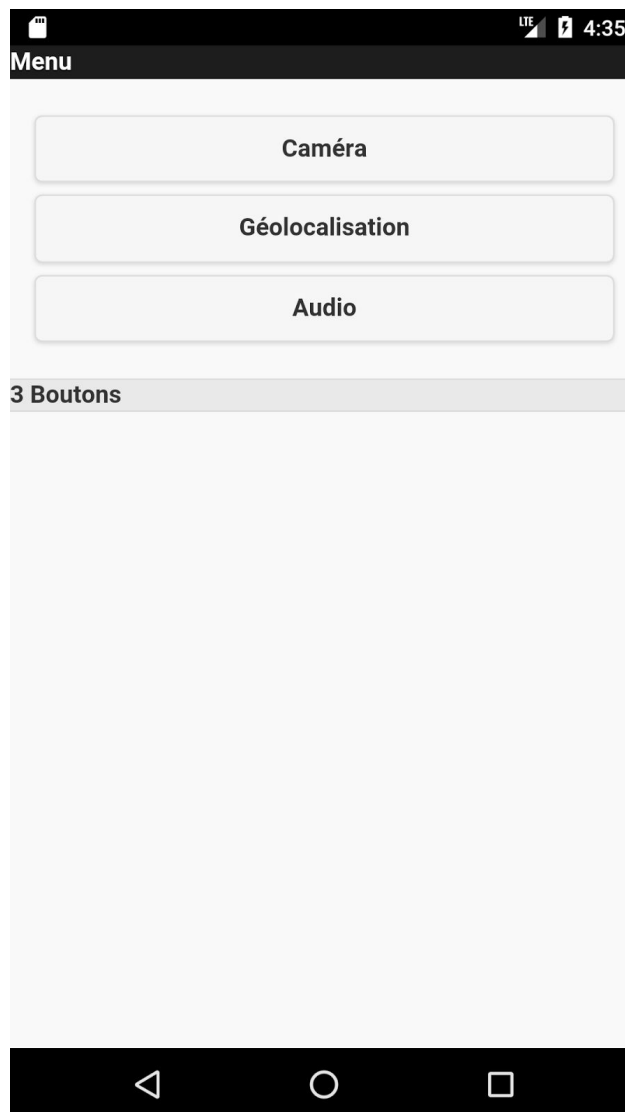
Le but de ce tp est de vous faire manipuler les différents plugins incontournable du framework cordova : Caméra, Media (audio), Géolocalisation, et SQLite (base de donnée).

3 parties qui correspondent aux 3 boutons ci-dessous vont constituer ce tp. Plus une dernière partie visant à manipuler une BDD local à l'appareil.

Il est conseillé de faire le tp uniquement via le téléphone mobile et de ne pas utiliser de navigateur pour débbugger car certaines fonctionnalités ne fonctionne que sur un mobile.

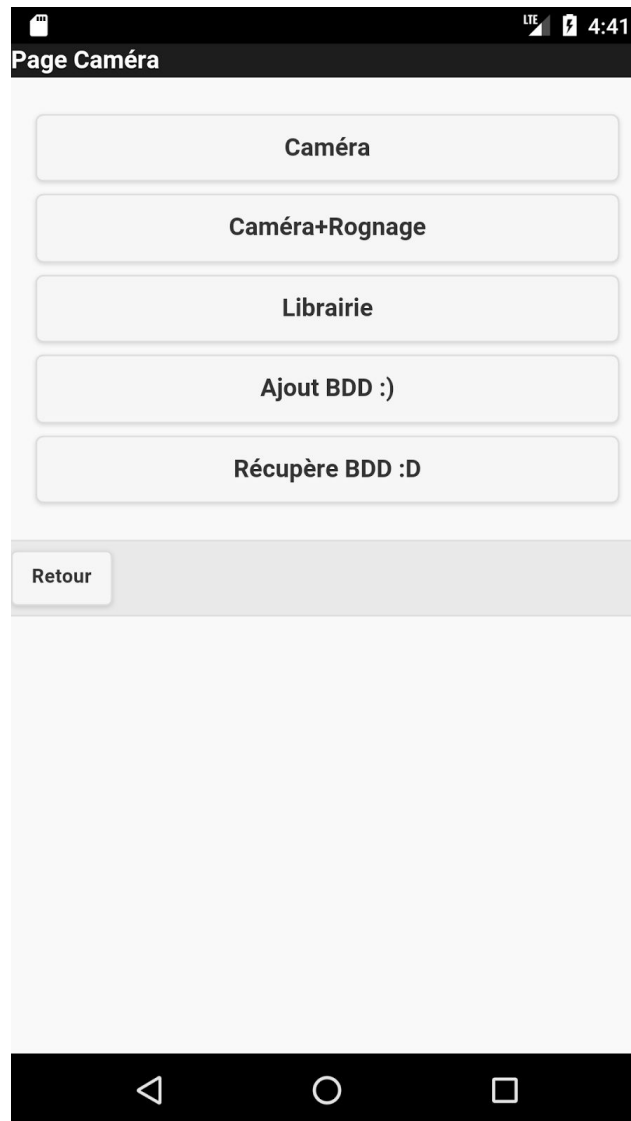
Aidez-vous de la documentation officiel de cordova :

<http://cordova.apache.org/docs/en/latest/>



Part 1 Caméra

Voici la vue qu'il faudra reproduire (ne pas tenir compte des 2 derniers boutons) :



Elle permettra de prendre une photo avec sa caméra, de la rogner, et de récupérer une image à partir de sa librairie.

1. Introduction :

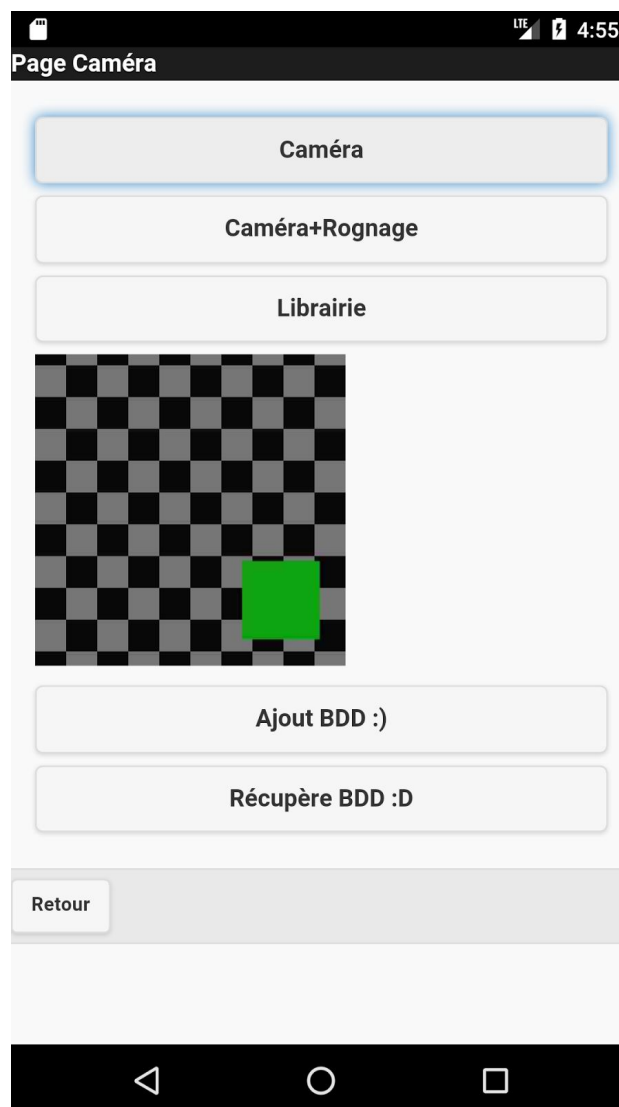
Pour commencer, on se concentre sur les 3 premiers boutons.

Ajouter le plugin nécessaire à l'utilisation de la caméra et assurez-vous qu'il fonctionne en faisant une alerte de la variable "navigator.camera".

2. Bouton "Caméra" :

1. Créer le bouton
2. Ajouter un écouteur qui active une fonction "capturePhoto()"
3. Remplir cette fonction pour qu'elle lance la caméra du téléphone
4. Ajouter une balise de type "Image" sous le bouton "Caméra" avec l'attribut src vide
5. Faire en sorte que lorsqu'une photo est prise avec la caméra cela ajoute l'url de l'image dans l'attribut src de la balise de type "Image"

Le résultat devrait-être similaire à celui-ci.



3. Bouton “Caméra+Rogner”

1. Reprendre le code de la fonction caméra pour l’injecteur dans une autre fonction appelée “captureEditPhoto()”
2. Modifier le code pour que le téléphone demande à rogner l’image une fois capturé
3. Modifier la qualité de la capture pour qu’elle prenne moins de place
4. Modifier la taille de l’image pour qu’elle prenne moins de place également

4. Bouton “Librairie”

1. Reprendre le code de la fonction caméra pour l’injecteur dans une autre fonction appelée “getPhoto()”
2. Modifier le code pour que le téléphone aille chercher une photo dans la librairie du téléphone

5. Affichage

Si cela n’est pas déjà fait. Faite en sorte que lorsqu’on clique sur chacun des boutons, cela affiche la photo récupérée dans notre balise de type “Image”.

Part 2 Géolocalisation

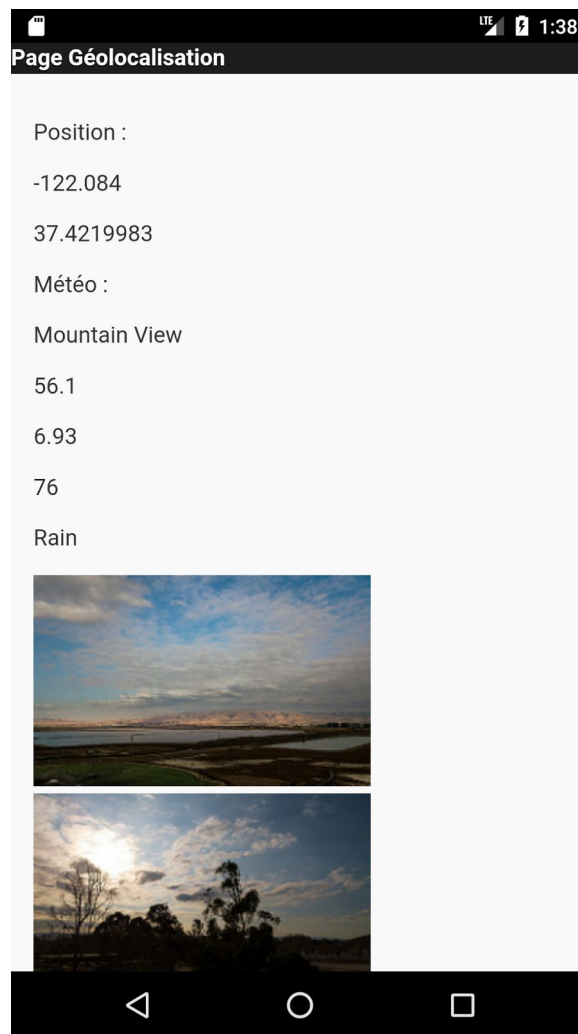
Deuxième partie de ce tp, la géolocalisation. L'objectif de cet exercice est d'apercevoir et d'utiliser certaines possibilités liés aux coordonnées GPS.

Rappel : Avant toute chose, bien s'assurer que le plugin de géolocalisation a été ajouté au projet. Pour cela, vous pouvez vérifier qu'il y a un bien une dépendance renseignant la géolocalisation dans le fichier config.xml.

Comme ceci : `<plugin name ="cordova-plugin-geolocalisation"/>`

Auquel cas, il faudra utiliser cette commande à la racine de votre projet :

- `cordova plugin add cordova-plugin-geolocation`



1 . Afficher dans les balises longitude et latitude, les coordonnées adéquates, à l'aide la fonction *navigator.geolocation.getCurrentPosition(onSuccess, onError)*. A vous donc de créer les fonctions onSuccess et OnError.

2. Maintenant que nous avons vu comment récupérer des coordonnées et les utiliser, nous allons pouvoir jouer avec la fonction *navigator.geolocation.getCurrentPosition*. Remplacer les fonctions précédemment écrites afin qu'elles renseignent cette fois ci les balises météo en plus des coordonnées GPS. Pour cela, vous parserez un JSON reçu depuis l'API OpenWeather (clé API et URL disponibles en entête de fichier index.js). Libre à vous d'utiliser une autre API si vous le souhaitez mais n'oubliez pas qu'il faudra récupérer la clé API.

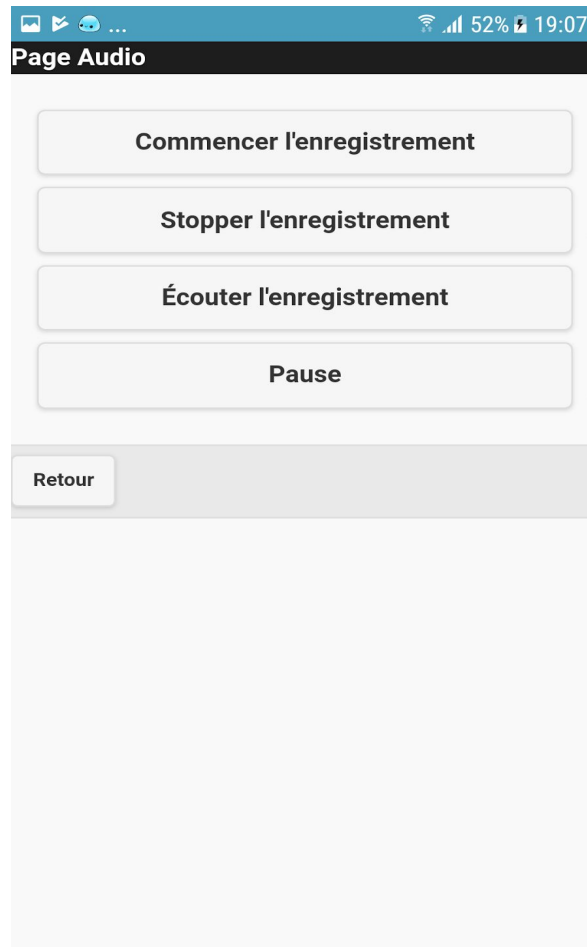
3. Dernière tâche, on va récupérer et afficher une liste d'images renseignés selon nos coordonnées. Pour cela on utilisera l'API Flickr (clé API et URL disponibles en entête de fichier index.js). Ecrire les fonctions permettant de parser le fichier contenant la liste d'images et les afficher à la suite de la balise "pictures".

Part 3 Audio

Le but de cette partie est de manipuler les différentes fonctionnalités que nous offre Cordova pour l'audio.

Avant de commencer, il faut penser à ajouter le plugin "cordova-plugin-media" à la racine de votre projet. Assurez-vous qu'il fonctionne en faisant une alerte de la classe "Media".

Reproduisez la vue suivante.



1. Pour le bouton "Commencer l'enregistrement", initialisez le média après que l'appareil soit prêt. Ecrivez une fonction qui vérifie qu'un média n'est pas déjà instancié et qui commence à enregistrer la voix.
2. Pour le bouton "stopper l'enregistrement", écrivez une fonction qui arrête l'enregistrement.
3. Pour le bouton "Ecouter l'enregistrement", écrivez une fonction qui lit l'enregistrement.
4. Pour le bouton "Pause", écrivez une fonction qui met le fichier audio en cours de lecture sur pause.

Part 4 Base de données

Lien utile : <https://github.com/litehelpers/Cordova-sqlite-storage#readme>

1. Introduction

Ajouter le plugin nécessaire à l'utilisation d'une base de données SQLite et assurez-vous qu'il fonctionne en utilisant la fonction adéquat.

Reprenons le code de la partie 1 en y ajoutant les deux derniers boutons qui permettent d'ajouter et de récupérer une image dans une BDD.

2. Bouton "Ajout BDD"

1. Ajouter un écouteur qui active une fonction "addImgBDD()" quand on clique sur le bouton d'ajout
2. Remplir la fonction pour qu'elle insert l'emplacement de l'image dans la bdd
3. Faite en sorte que lorsque l'ajout en BDD se passe bien une alert affiche "SUCCESS" et quand ça se passe mal affiche le message d'erreur.

3. Bouton "Récupère BDD"

1. Ajouter un écouteur qui active une fonction "retrieveImgBDD()" quand on clique sur le bouton de récupération
2. Remplir la fonction pour qu'elle récupère l'emplacement de l'image précédemment enregistré en BDD
3. Modifier l'affichage de la balise de type "Image" une fois l'emplacement récupéré
4. Faite en sorte que lorsque la récupération en BDD se passe bien une alert affiche "SUCCESS" et quand ça se passe mal affiche le message d'erreur.

Plus

Pour ceux qui ont fini le TP en avance vous pouvez :

- Changer l'icône de votre application.
- Essayer de lancer l'application sur un émulateur iOS (/\ Pour ceux possédant un SDK iOS /\).
- Regarder les différents plugins disponible sur le site de cordova. Voir essayer d'en implanter.
- Publier votre application sur drive pour pouvoir la télécharger.