



Ionic Framework

SUPPORT DE COURS PAR NICOLAS FAZIO

7. Capacitor



7.1.C'est quoi Capacitor??

Historiquement, Ionic framework utilisait un Outil nommé « Cordova » développé par PhoneGap. Cet outil permettait de pouvoir accéder via des plugins, aux ressources natives des devices (appareil photo, GPS, Contacts, Gyroscope, Accéléromètre, Files System, etc...), il en existe énormément, mais ne sont pas tous maintenus.

L'équipe Ionic a alors décidé de sélectionner quelques plugins qu'ils ont « wrapper » dans des plugins « Ionic-Native ».

Avec le temps, l'équipe de Ionic a changé la manière de faire pour permettre un contrôle total des ressources système des devices. Ils ont alors planché sur un nouveau projet qui sera capable de résoudre les nombreux problèmes que cause Cordova.

Le projet voit le jour début 2018 et se nomme « Capacitor ». Il a pour but de fournir un accès système avec de nouveaux plugins et surtout, le processus de « build » de l'application change complètement.

Désormais, fini l'interpréteur embarqué avec l'application... Le code est directement délivré dans le langage natif du device choisi. Une révolution! On est dans du vrai « Cross-Platform » 🎉🎉🎉

Doc: <https://capacitor.ionicframework.com>

7.2.Installer Capacitor dans un projet Ionic

- \$ npm install --save @capacitor/core @capacitor/cli
- \$ npx cap init

7.3.Introduction aux plugins

Les plugins dans Capacitor permettent à JavaScript de s'interfacer directement avec les API natives.

Avec les plugins, une application web peut accéder à toute la puissance des API natives, en faisant tout ce qu'une application native traditionnelle peut faire. Les plugins sont particulièrement adaptés pour encapsuler des opérations natives courantes qui peuvent utiliser des API très différentes sur plusieurs plates-formes, tout en exposant une API multiplateforme cohérente à JavaScript.

De plus, la fonctionnalité Plugin de Capacitor permet aux équipes mixtes de développeurs natifs traditionnels et de développeurs Web de travailler ensemble sur différentes parties de l'application.

Capacitor auto génère des Hooks JavaScript sur le client, donc la plupart des plugins ont seulement besoin de construire un plugin natif Swift / Obj-C pour iOS, et / ou un Java pour Android.

Bien sûr, ajouter du JavaScript personnalisé pour un plugin est possible, et c'est comme fournir un paquet npm JavaScript.

Doc: <https://capacitor.ionicframework.com/docs/plugins>

7.4.Introduction au API Natives

Capacitor inclut un certain nombre d'API natives disponibles pour toutes les applications Capacitor. Ceux-ci peuvent être considérés comme des «core plugins» de Capacitor, et ils facilitent l'accès aux fonctionnalités les plus courantes sur chaque plateforme.

Pour ceux qui viennent de Cordova, les principales API de Capacitor couvrent une grande partie des plugins Cordova de base, et en incluent aussi de nouvelles.

Doc: <https://capacitor.ionicframework.com/docs/apis/>

7.5.Utilisation de l'API Camera

L'API Camera permet à un utilisateur de choisir une photo à partir de son album photo ou de prendre une photo. Sur iOS, cela utilise UIImagePickerController, et sur Android, cette API envoie une intention qui sera gérée par l'application Camera principale par défaut.

Doc:

<https://capacitor.ionicframework.com/docs/apis/camera>

```
import { Plugins, CameraResultType } from '@capacitor/core';

const { Camera } = Plugins;

async takePicture() {
  const image = await Camera.getPhoto({
    quality: 90,
    allowEditing: true,
    resultType: CameraResultType.Uri
  });
  var imageUrl = image.webPath;
  // can be set to the src of an image now
  imageElement.src = imageUrl;
}
```

!!!! Warning !!!

Ce plugin est actuellement non fonctionnel (06.06.18)