

ionic 5

Créer des applications pour téléphones et
tablettes

Les applications pour téléphones et tablettes

Il existe 3 moyens pour qu'une application soit accessible sur un téléphone ou une tablette :

Une application web responsive, ou « designée » pour mobile

- L'utilisateur pourra accéder à l'application via internet
- Pas de possibilité d'accéder à toutes les fonctionnalités offertes par le smartphone : « téléphone », « messages », « appareil photo », « contact »,...
- En transformant cette application en PWA (progressive web app), le téléphone proposera de créer un raccourci vers le site internet (icône) comme pour une véritable application. De plus, la barre de navigation disparaîtra pour ressembler à une application native.

Une application native, créée en utilisant les languages et les plateformes de développement associées à chaque système d'exploitation utilisé par les téléphones Android, iOS, Windows phone,...

- Il s'agit d'une véritable application disponible sur les stores respectifs et directement accessible (icône).
- L'application peut avoir accès à toutes les fonctionnalités du téléphone.
- Malheureusement, il faut coder différemment pour chaque système d'exploitation : java pour Android, swift pour iOS, C# pour Windows Phone

Une application hybride

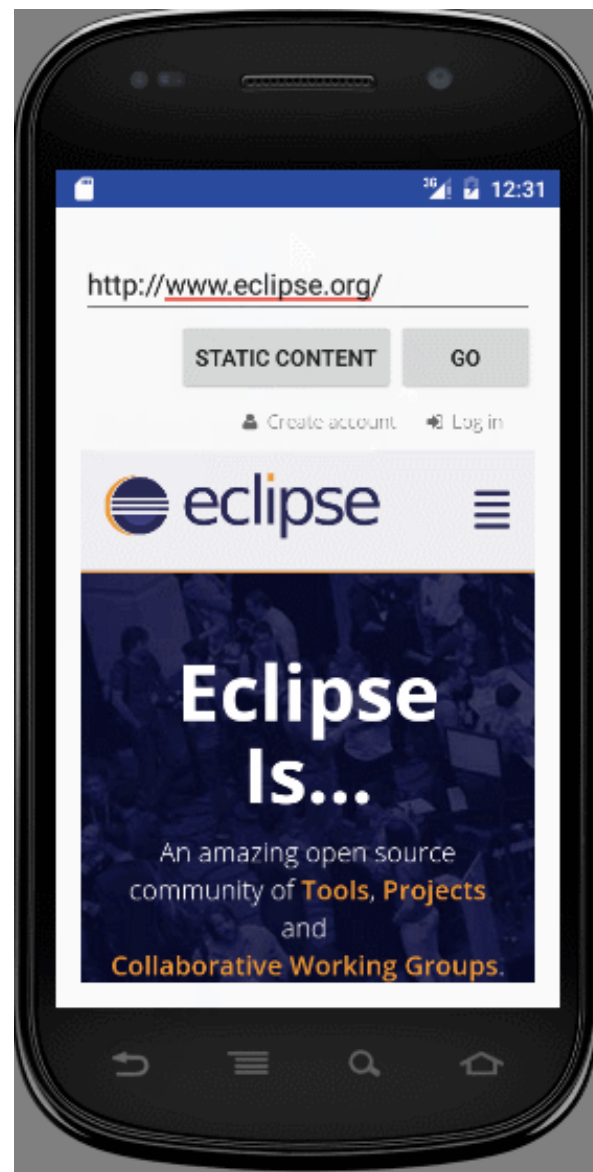
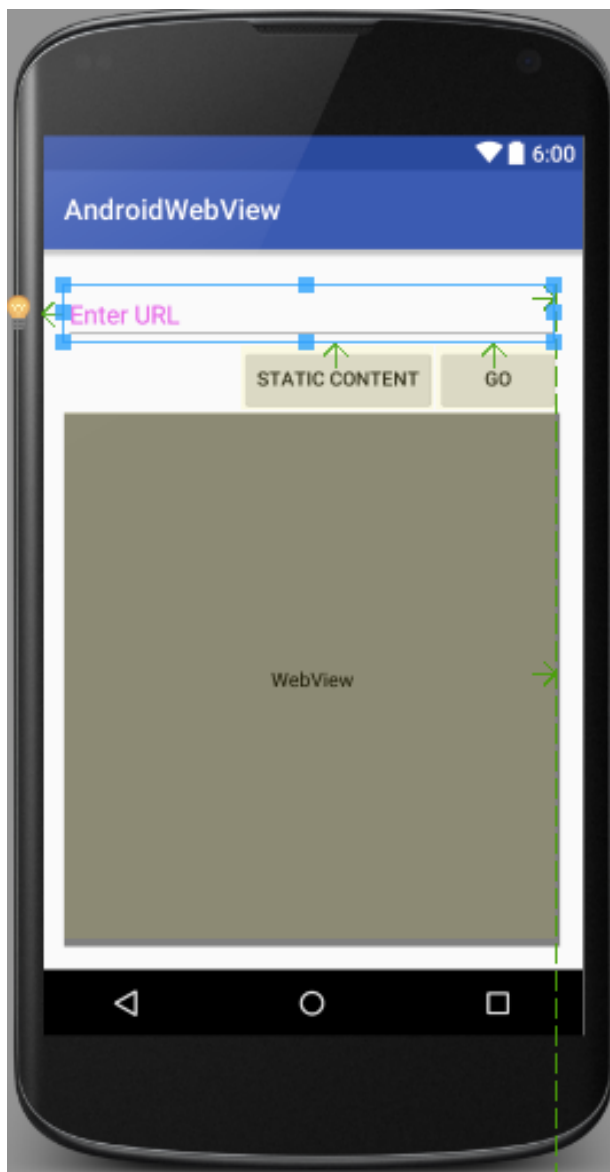
- L'application hybride va utiliser les langages de développement d'un site internet (html, css, javascript). Donc un seul langage de programmation.
- Elle va pouvoir utiliser les fonctionnalités des téléphones (accéléromètre, contacts,...) grâce à des **plugins**
- Elle sera construite pour devenir une véritable application native et être disponible via les plateformes de téléchargement Appstore, le Playstore, ...

Comment créer une application hybride ?

Cordova

Les smartphones et les tablettes possèdent des navigateurs web grâce auxquels les utilisateurs peuvent aller sur des sites internet.

Dans une application native, il est possible d'intégrer un composant qui permet l'affichage d'une WebView, c'est à dire capable d'interpréter les langages des navigateurs.



La création d'une application se ramène donc à la création d'une seule WebView plein écran qui interprètera le code html, js, css.

Ce travail de création de WebView est différent pour chaque plateforme mais unique pour toutes les applications iOS par exemple.

C'est cela que fait **cordova**

Il y a également des événements propres aux applications mobiles, cordova permet également d'avoir accès à ceux ci.

Supported Platforms/ Events	android	ios	Windows
deviceready	✓	✓	✓
pause	✓	✓	✓
resume	✓	✓	✓
backbutton	✓	✗	✓
menubutton	✓	✗	✗
searchbutton	✓	✗	✗
startcallbutton	✗	✗	✗
endcallbutton	✗	✗	✗
volumedownbutton	✓	✗	✗
volumeupbutton	✓	✗	✗
activated	✗	✗	✓

Dans de nombreuses applications, il est nécessaire d'utiliser les fonctionnalités natives du smartphone, inaccessible à travers un navigateur web (le téléphone, les contacts,...)
On réalise cela avec un **plugin**. Il en existe de nombreux disponibles permettant de couvrir l'ensemble des besoins.
Un plugin comporte des fichiers écrits en langage utilisé pour le développement des applications natives (swift ou java,...)
Cordova réalise un pont « bridge » entre le javascript et ces langages pour accéder aux fonctionnalités.

Le développement pour android et iOS requiert l'installation d'outils de développements spécifiques. (Voir : <https://cordova.apache.org/docs/en/9.x/guide/platforms/android/index.html>)

Il n'est malheureusement pas possible de construire la version finale d'une application pour iOS sans utiliser Xcode disponible uniquement sur mac.

Quelques commandes cordova :

Créer une nouvelle application cordova :

cordova create my-app

Ajouter une plateforme :

cordova platform add android

Ajouter un plugin :

cordova plugin add cordova-plugin-camera@^2.0.0

Supprimer un plugin :

cordova plugin rm StatusBar

Voir la liste des plugins :

cordova plugin ls

Voir l'état des exigences pour le développement sur une plateforme :

cordova requirements

Lancer l'application dans un émulateur :

cordova emulate android

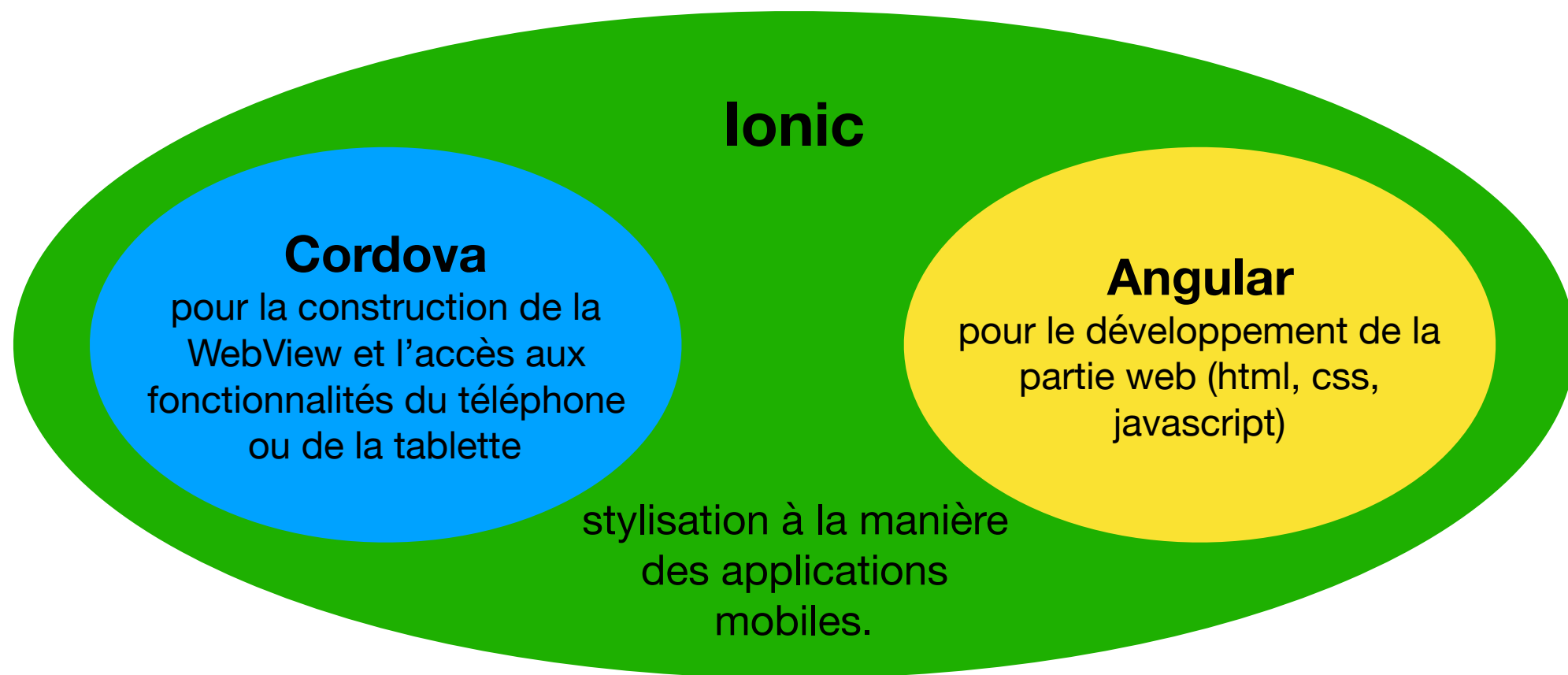
Lancer l'application sur le mobile connecté :

cordova run android

Construire l'application pour l'OS du mobile (pour android crée un .apk)

cordova build android --release --prod

Ionic



Quelques commandes ionic :

Créer une nouvelle application ionic :

ionic start my-app

Ajouter une page :

ionic generate page contact

Ajouter un composant :

ionic generate component contact/form

Ajouter un service :

ionic generate service api/user

Ajouter une plateforme :

ionic cordova platform add android

Ajouter un plugin :

ionic cordova plugin add cordova-plugin-camera
npm install @ionic-native-camera

Supprimer un plugin :

ionic cordova plugin rm cordova-plugin-camera

Lancer l'application dans un émulateur :

ionic cordova emulate android

Lancer l'application sur le mobile connecté :

ionic cordova run android

Construire l'application pour l'OS du mobile (pour android crée un .apk)

ionic cordova build android --release --prod

Le cycle de vie d'une application Ionic

