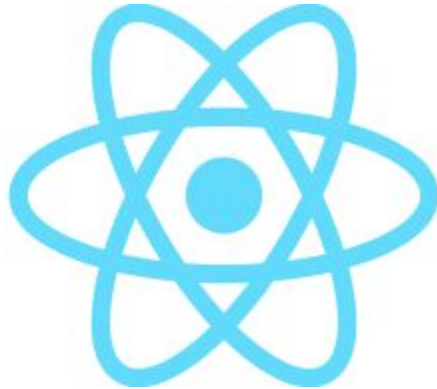


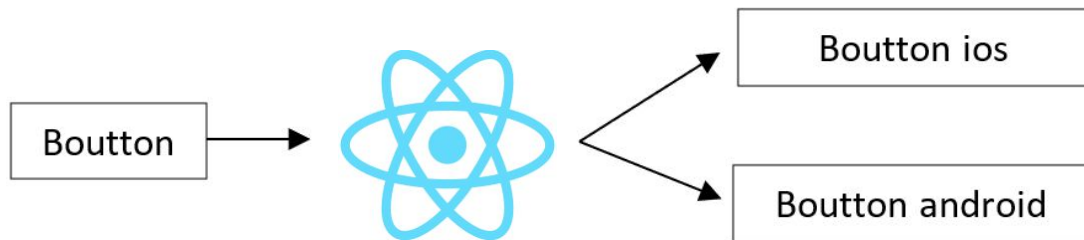
Dièse-Dev 2022

React Native



React Native c'est quoi ?

- Un framework, pas un langage
- Permet de développer des applications mobiles sur android et ios
- Utilise les éléments natifs des systèmes d'exploitation



=> Bonnes performances

- Un seul langage à apprendre : Javascript

JavaScript (JS)

- Langage très simple (syntaxe similaire à python ou C)
- Pas de cours de JS ici (regarder la syntaxe sur internet pour les boucles while et for, etc...)
- Vous apprendrez sur le tas
- React Native ne requiert pas un haut niveau de compréhension de JS

Description du projet

- App permettant de rechercher et d'afficher une liste de film depuis un URL
- Gérer les données de l'application en mettant des films en favoris



Un choix à faire avant de développer

- React Native permet de développer une app avec deux formats:
 - Un format rapide à initialiser et utiliser avec Expo, uniquement pour les app n'utilisant pas les composants du téléphone (caméra, gyroscope, etc...)
 - Un format natif plus long à initialiser pour les app utilisant les composants du téléphone
- On peut passer Expo à natif en "éjectant l'application"
- On utiliseras Expo (plus simple, permet aux utilisateurs iphone de ne pas avoir un mac)

Installation (1)

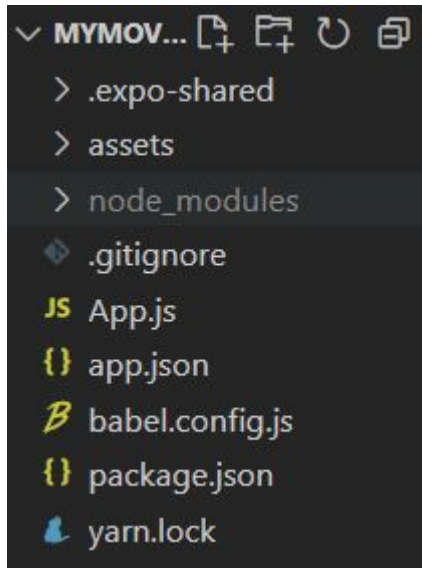
- Téléchargez node.js [ici](#) (windows mac) et installez le. (`sudo apt install nodejs` pour linux)
- Ouvrez un terminal et tapez
`npm install -g expo-cli` (windows)
ou
`sudo npm install -g expo-cli` (mac)
- Téléchargez Expo sur votre téléphone (possibilité de faire tourner un téléphone virtuel sur votre pc mais on ne verras pas ça)
- Mettez vous dans le dossier de votre choix avec `cd` et en copiant l'adresse via l'explorateur de fichier. Pour les windows, utilisez le cmd (cmd dans la barre de recherche)

Installation (2)

- Tapez `expo init nomApp` et sélectionnez “blank” pour créer votre projet
- `cd nomApp` pour se placer dans le dossier créé

Hiérarchie d'un projet React Native

- Ouvrez le dossier créé par React Native avec votre IDE favori (si vous n'en n'avez pas, téléchargez Visual Studio Code)



On s'en fiche d'un peu tout à part :

- App.js : le contenu de l'application, c'est le fichier sur lequel on va coder pour ajouter des choses sur l'app
- app.json : fichier qui contient toutes les informations sur l'application, notamment les modules qu'elle utilise (on parlera plus tard des modules)

Démarrer l'application

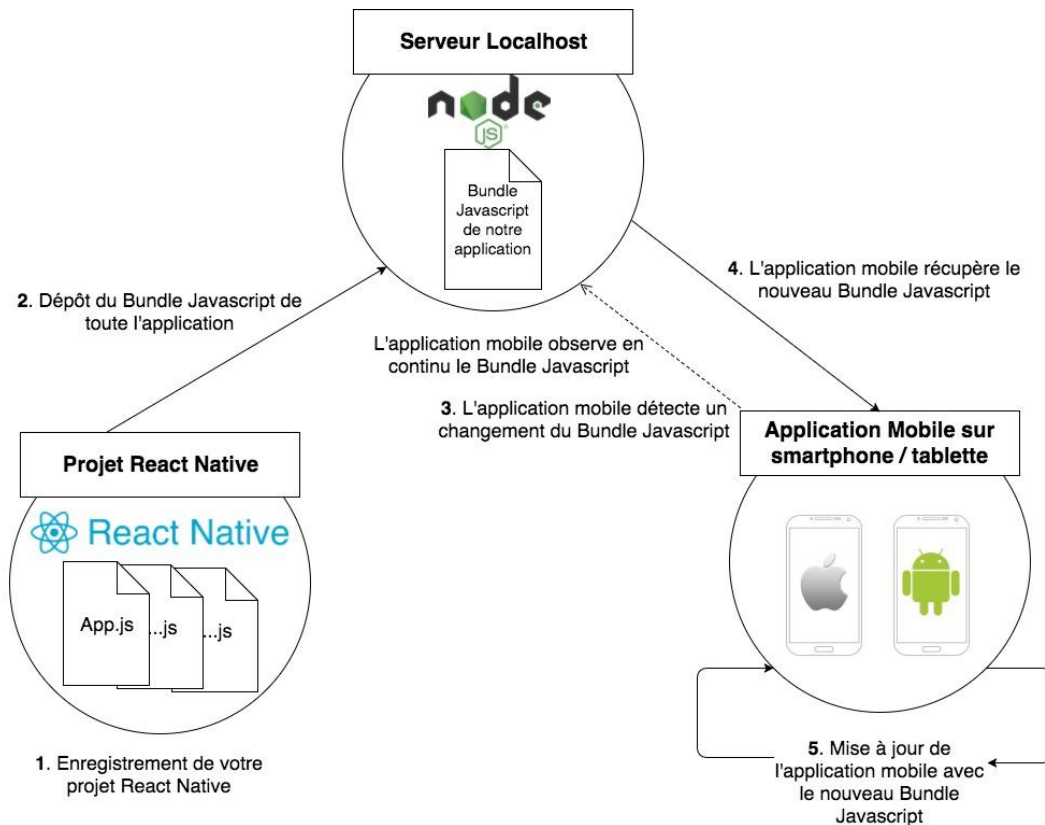
- Avant de démarrer l'application, il faut s'assurer que le téléphone et l'ordi soient connectés au même wifi (ça marche avec un partage de connexion)
- Tapez `npm start`, après un moment un QR code devrait apparaître
- Scannez le qr code depuis l'app Expo (android) où directement avec la caméra (ios)
- Après un chargement vous arrivez sur cette page:



Recharger l'application

- L'application se rechargera toute seule après un enregistrement (Live Reloading)
- Modifier le texte `Open up App.js to start working on your app !` et enregistrer les modifications (ctrl+S)
- Pas de compilation pendant le codage en React Native (grâce à JS)

Recharger l'application (Schéma)



(Bundle = fichier)

Les composants

- Concept clé de React Native
- Un component est un élément graphique affiché par l'interface utilisateur.
Toutes les vues React Native sont composées entièrement de composants.
- Les éléments graphiques, vues, boutons, textes, images, listes, etc., sont des composants. Si vous créez une vue “Home” avec une image et un texte, votre vue Home est un component et elle possède un component image et un component texte.
- Les composants sont réutilisables et très flexibles

Deux manières de coder un component

- Avec React Native on peut coder un component avec une syntaxe de classe ou de fonction selon si on préfère les langages orientés objets ou fonctionnel (vous saurez ce que c'est au S2)
- React Native initialise votre premier composant avec une syntaxe de fonction, mais ce cours sera fait avec la syntaxe de classe. Il faut donc réécrire un peu App.js

Deux manières de coder un component (2)

```
1 //App.js
2
3 import React from 'react';
4 import { StyleSheet, Text, View } from 'react-native';
5
6 export default class App extends React.Component{
7   render(){
8     return (
9       <View style={styles.container}>
10         <Text>Open up App.js to start working on your app!</Text>
11       </View>
12     )
13   }
14 }
15
16 const styles = StyleSheet.create({
17   container: {
18     flex: 1,
19     backgroundColor: '#fff',
20     alignItems: 'center',
21     justifyContent: 'center',
22   },
23 });
```

- Modifiez App.js comme ceci
- On voit bien que App.js est un component avec la ligne 6 (extends ~ reprend les propriétés de)

Deux types de component

- Les components React Native : éléments graphiques simples que l'on retrouve sur les applications mobiles natives : Text, Button, Image, ScrollView, View, etc...
Ces components existent déjà et sont mis à disposition par React Native.
- Component custom : components créés, reposant sur les components de bases

Exemple de component custom

- Si je crée le component custom “profil” suivant :
Il est en fait composé de plusieurs components de bases

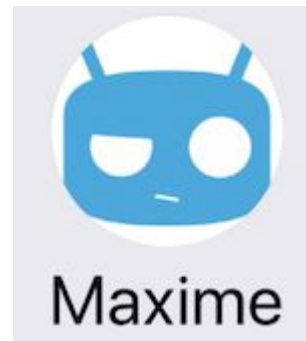
`component profil:`

`-component de base “vue”`

`-component de base “image” (tête bleue)`

`-component de base “texte” (Maxime)`

- On pourra alors réutiliser ce component autant de fois que l’on veut (Ex: dans la page d'accueil et dans les paramètres, sans avoir de doublons de code)



Prochain cours

- Comment coder nos components
- Comment spécifier l'apparence des components (text rouge, couleur de fond grise, etc...)
- Comment passer des informations entre les components