

Développement build et développement natif en React-Native

Développement natif :

Pour faire un développement natif avec React-Native, on opte pour l'option d'utiliser React-Native sans Framework :

Step 1: `npx @react-native-community/cli@latest init AwesomeProject`

Cette commande permet de démarrer un nouveau projet, ici le nom du projet est **AwesomeProject**

[illegible]

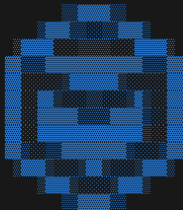
Step 2: Start Metro avec la commande npm start

```

❖ nbu-yvette@Desktop-06-ThinkPad-Centre-M900:~/AwesomeProject$ npm start

> AwesomeProject@0.0.1 start
> react-native start

Welcome to React Native v0.81
Starting dev server on http://localhost:8081



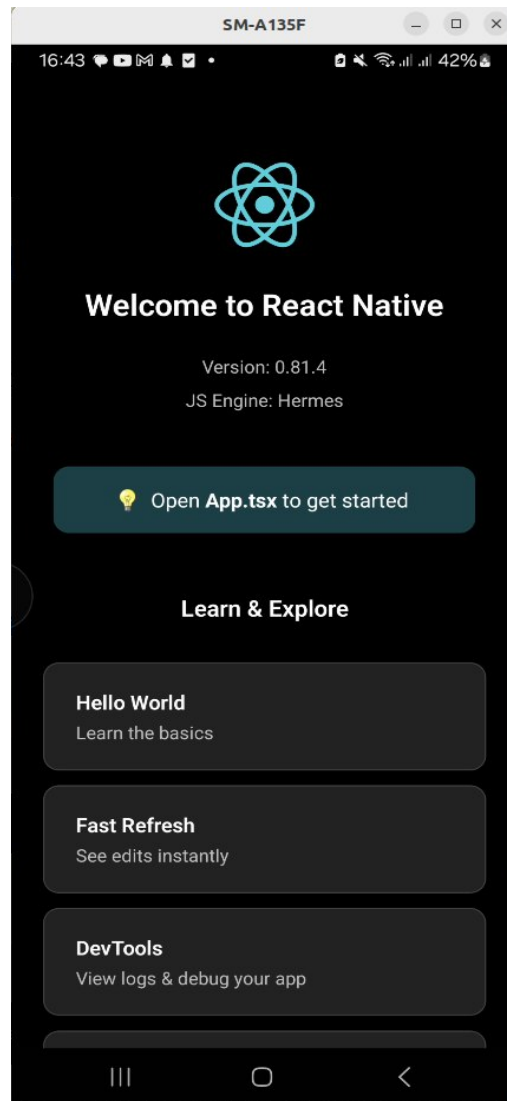
Welcome to Metro v0.83.2
Fast - Scalable - Integrated

INFO Dev server ready. Press Ctrl+C to exit.
INFO Key commands available:

  r - reload app(s)
  d - open Dev Menu
  j - open DevTools

BUNDLE ./index.js
INFO Connection established to app='com.awesomeproject' on device='SM-A135F - 14 - API 34'.
INFO JavaScript logs have moved! They can now be viewed in React Native DevTools. Tip: Type j in the terminal to open (requires Google Chrome or Microsoft Edge).
```

Step 3 : Démarrer l'application avec la commande `npm run android`



Développement build :

Le projet Perxonomiz tournait déjà en développement build, les différentes étapes sont les suivantes :

- `npx create-expo-app nom-projet`
- `npx expo prebuild`
- `npm run android`

Les différences clés notées :

- Expo fournit un environnement prêt-à-l'emploi.
- Pas besoin de toucher au code natif (Android/iOS)
- Builds personnalisés avec Expo EAS

- Avec le natif, on travaille directement avec React Native CLI
- Si besoin, possibilité d'implémenter directement du code natif (Kotlin ou Swift) pour des usages avancés
- on a `/android` dès le départ
- `index.js` lance l'application