==

홈페이지(WEB)를 쉽게 앱(Android,Ios)만들기(구름IDE, React-Native,NodeJs,Expo, feat조코딩)

==

웹개발자도 쉽게 어플리케이션을 만드는 방법을 찾던 중 조코딩님의 ‘구름IDE를 이용한 3분만에 안드로이드와 아이폰 어플만들기’를 보았고, 이 방법이면 앱 개발자가 아니어도 충분히 훌륭한 앱을 만들 수 있겠다라는 생각이 들었습니다. 그리고 조코딩님의 유튜브 찍은날짜가 1년이 넘어, 영상과 다른부분이 있어 혼자 시행착오하고 성공한 결과를 정리및 공유드립니다.

진행할 내용은 구름IDE를 활용한 리액트 네이티브로 웹뷰를 활용해 앱을 만드는 것 입니다. 리엑트네이티브가 웹개발자가 앱을 만듬에 있어 강력한 이유는 우리가 ios를 만들기 위해 swift 를 공부 및 알 필요도 없으며 안드로이드언어 또한 마찬가지입니다. 그러므로 최종 완성은 콘텐츠 영역은 html이며, 앱 배포에 필요한 패키징 처리만 아이폰 안드로이드로 하는 하이브리드 앱을 완성하는 것 입니다.

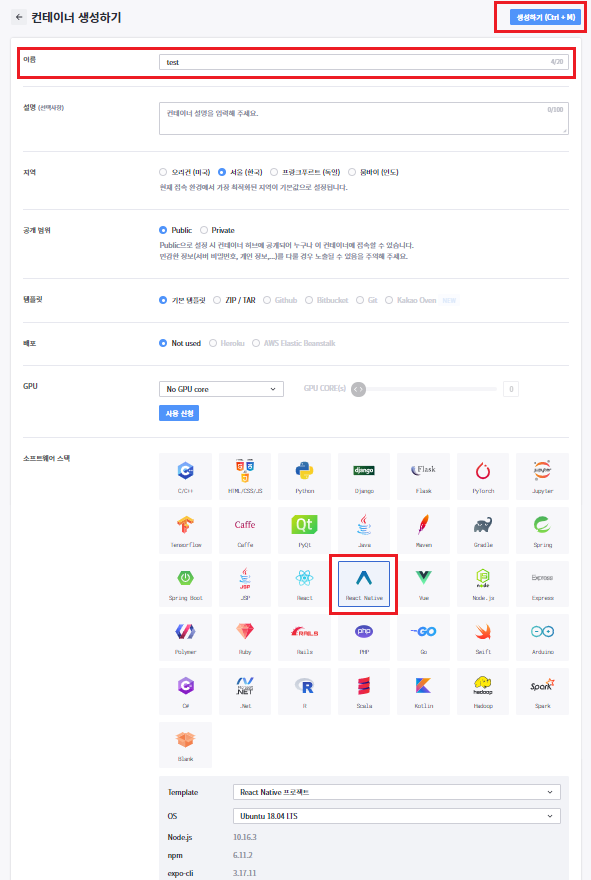
==

구름IDE는 도커관련서비스로 IDE(Integrated Development Environment)에서 알 수 있듯이 코딩을 편하게 할 수 있도록 통합개발환경을 제공해줍니다. 그리고 도커관련 서비스란 우리가 설치하려 했던 개발 환경을 미리 다 설치된 개발환경을 제공해줍니다.

(우리가 필요한 개발환경 : NodeJs, ReactNative, Expo)

<https://ide.goorm.io/>

에서 회원가입 & 로그인 후 대시보드에 들어가 “새컨테이너0/5”를 눌러줍니다.



빨간색부분만 작성 및 체크 후 다른건 건들일 필요 없이 생성하기 버튼을 클릭 후 컨테이너가 완성되면 실행합니다.

개발

===========

expo에서 제공하는 웹페이지를 앱 상에 띄우는 기능 install

expo install react-native-webview (codeview로작성)



App.js를 클릭하여 내용을 전체 삭제 후 아래코드를 복사 후 완성된 웹 url로 설정합니다.

import \* as React from 'react';

import { WebView } from 'react-native-webview';

export default class App extends React.Component {

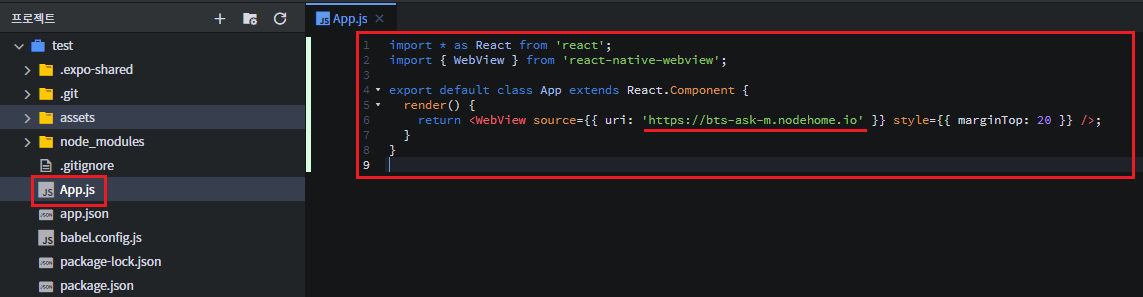
render() {

return <WebView source={{ uri: '원하는url' }} style={{ marginTop: 20 }} />;

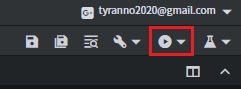
}

}

(codeview로작성)



여기까지 진행한 후 우측 상단의 play버튼을 클릭하면



우리가 완성되면 갖게될 어플리케이션을 미리 확인하고 테스트를 할 수 있습니다.

우리의 스마트폰으로 테스트를 위해 준비할 준비물 :

아이폰 : App Store - Expo Go

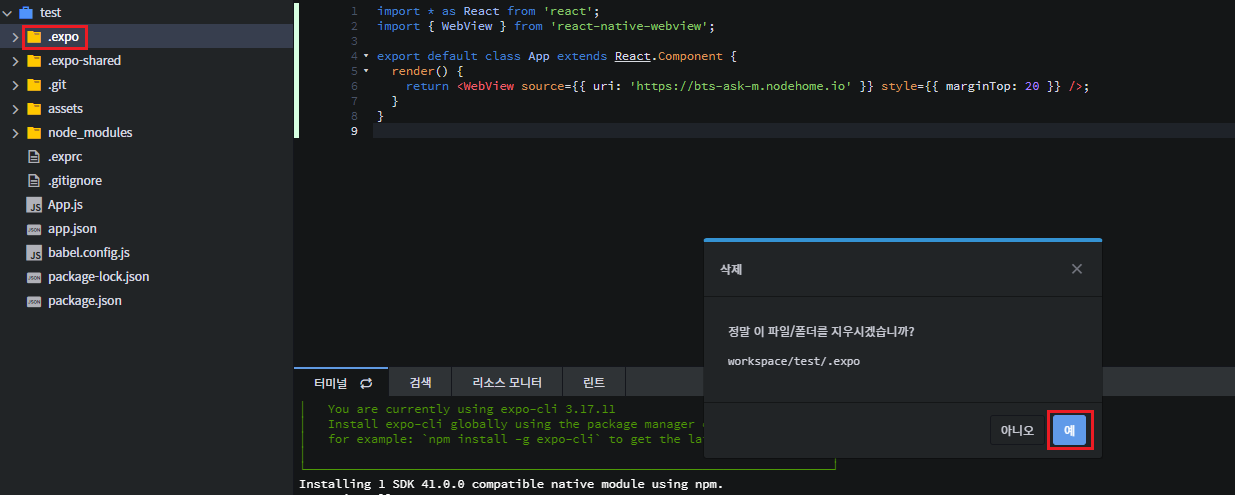
안드로이드 : Play 스토어 - Expo



안드로이드의경우 어플에 들어가 Scan Qr Code를 통해 해당 큐알을 찍으면 아래 게이지바가 진행되며, 아이폰은 카메라로 QR을 찍어 어플에 들어가면 게이지바가 진행됩니다.

게이지가 완료되고 다시한번 QR을찍으면 정상적으로 어플리케이션의 모습을 미리 보실 수 있습니다.

\*테스트를 진행후 AFK를 만들예정이면 .expo폴더를 지워준다.(connect 오류 방지)



======================================================================

AFK로 만들기

아래 명령어들을 터미널에 순서대로 실행시켜 줍니다.

npm install -g expo-cli (expo-cli 설치) (codeview로작성)



node -v // node버전 확인 (생략가능) (codeview로작성)

npm cache clean -f // 캐시삭제 (codeview로작성)



npm install -g n //nodejs 버전관리 플러그인 (codeview로작성)



n lts // lts버전 (codeview로작성)

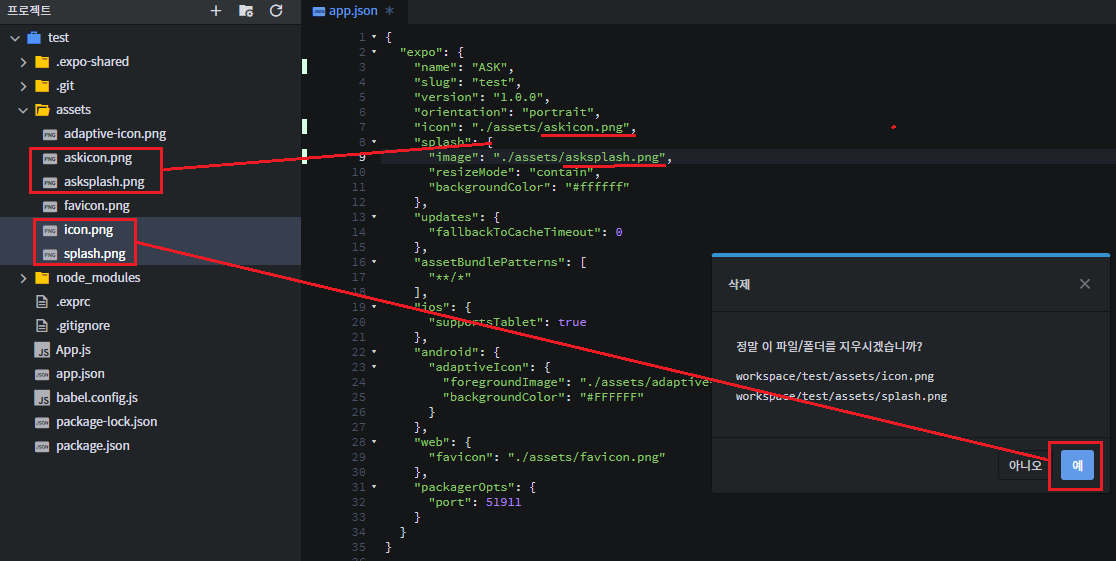


expo install expo-permissions



app.json을 실행합니다.

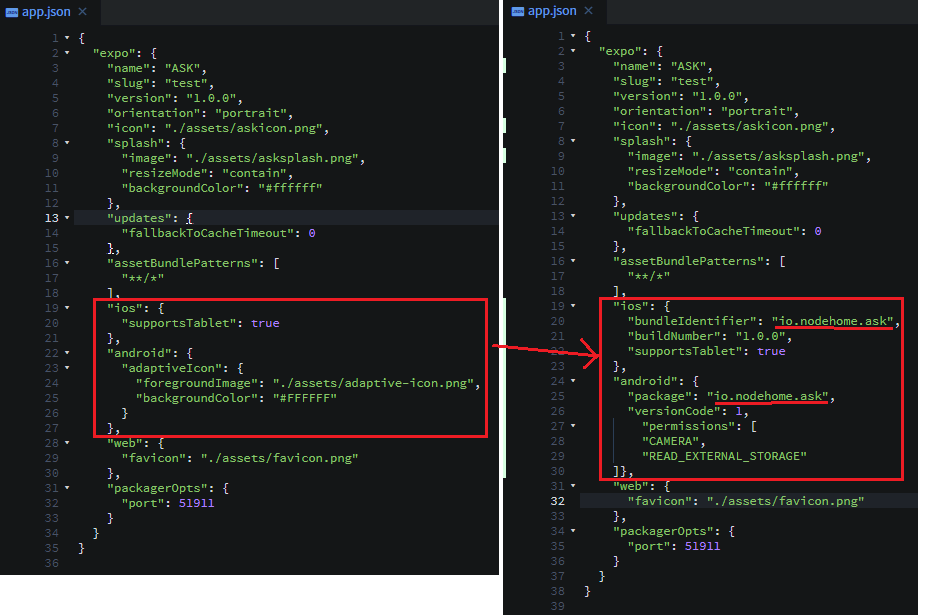
기존의 icon.png와 splash.png를 삭제 후 우리가 원하는 이미지로 저장하고 내용을 아래 사진과 같이 바꿔줍니다.



icon.png는 앱 모양이 될 사진이며 권장사이즈는 1024 \* 1024입니다. 사이즈가 너무작거나 너무크면 빌드할때 애러가 발생됩니다.

splash.png는 앱을 터지하고 나서 잠시동안 나타나는 화면을 뜻합니다. 권장사이즈는 1242 \* 2436 입니다.

두 이미지 모두 PNG만 가능하며, JPG로 업로드했을때 빌드시 애러가 발생됩니다.



ios와 안드로이드 부분을 아래 코드로 교체해줍니다.

"ios": {

"bundleIdentifier": "원하는 패키지명(ex com.javalec.ex)",

"buildNumber": "1.0.0",

"supportsTablet": true

},

"android": {

"package": "원하는 패키지명(ex com.javalec.ex)",

"versionCode": 1,

"permissions": [

"CAMERA",

"READ\_EXTERNAL\_STORAGE"

]},

(안드로이드의 경우 카매라와 앨범 접근 권한까지만 설정 진행함)

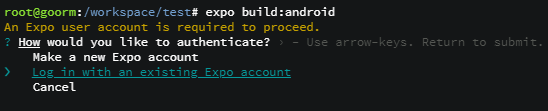
이제 모든 코드 작업은 끝났고 afk로 변환하기 위한 빌드만 진행하면 됩니다.

빌드를 하기 위해선 expo 회원가입이 먼저 필요합니다.

<https://expo.io/>

signup을통해 간단하게 회원가입 진행

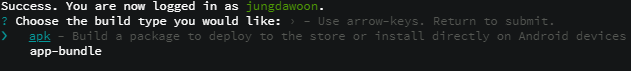
expo build:android



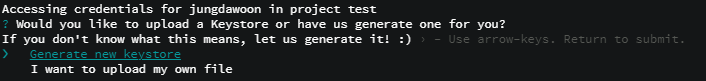
방향키로 이미 아이디가 있다는 선택지를 가고 엔터를 쳐줍니다.



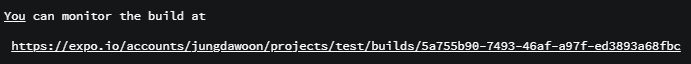
아이디 비번 입력 후 엔터



apk를 만드는것이므로 apk선택 후 엔터



진행중에

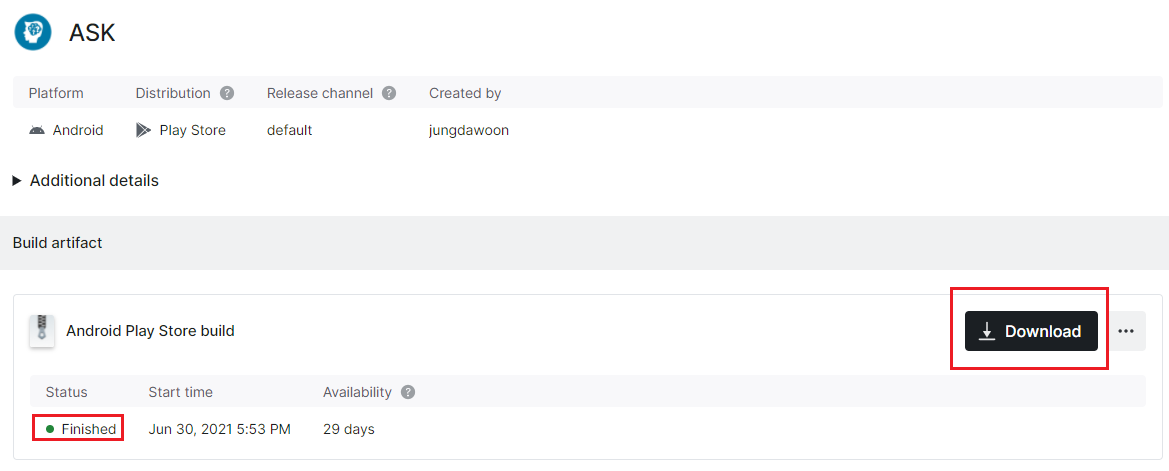


이런식으로 링크가 나오는데 여기서 진행과정을 확인할 수 있으며, 완료되면 apk다운로드가 가능하다.

마지막으로 generate new keystore 선택후 기다려주기만 하면 됩니다. (약20분~30분 걸렸음.)

<https://expo.io/accounts/jungdawoon/projects/test/builds>

에서 본인의 프로젝트를 선택한 후



다운로드하게되면 어플을 다운받을 수 있는 apk가 완성됩니다.