



AWSKRUG Frontend

React Native WebView 활용해서 Hybrid Application 만들기 (+CI/CD)

Changhyeon Yoon

Software Engineer

2023.07.12 19:30 ~ 21:30

서울 서초구 강남대로 465길 교보타워 11층 B구역 당근마켓

Changhyeon Yoon

전) LIKELION NFT Division - Frontend Developer

현) Granter - Co-Founder / Software Engineer (Frontend, DevOps)

Google Developer Student Clubs – Hanyang Univ 1st Lead /
Founder

GopherCon Korea 2023 – STAFF

Next.js Korea User Group – Organizer

AWS Korea User Group – Frontend, Beginners, Pangyo

Github : <https://github.com/ChanghyeonYoon>

LinkedIn : <https://linkedin.com/in/yooniversedev>

Blog : <https://changhyeon.net>



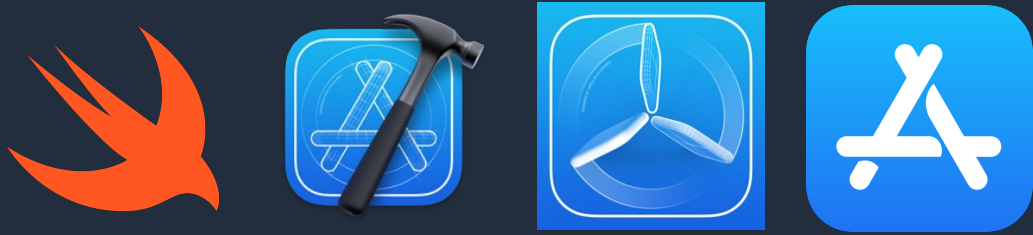
Table of Contents

1. Native App vs Hybrid App
2. Pros and Cons for Developer
3. Pros and Cons for User
4. Service Perspective
5. Develop Guidelines
6. CI / CD
7. Apple App Store Review
8. Q&A

Native App vs Hybrid App

Native App

iOS

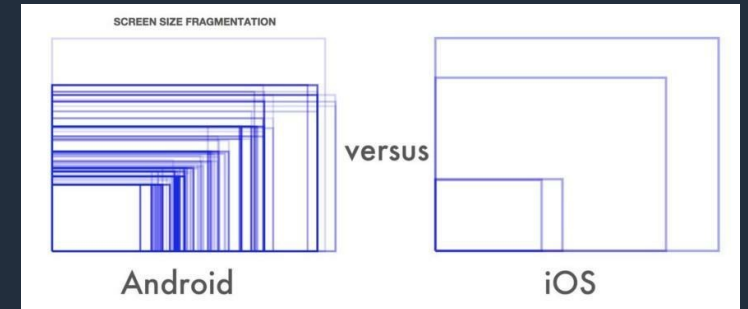


- 귀한 iOS 개발자
- 폐쇄적인 개발환경
- 까다로운 App Store Review

Android



- 다양한 기기 및 제조사
- 다양한 제조사



Cross Platform Framework

React Native



- 웹 개발자가 가장 작업하기 쉬움
- 자칫하면 프로젝트가 매우 무거워짐

Flutter



- 언어의 장벽 (Dart)
- 국내시장 및 커뮤니티는 아직...

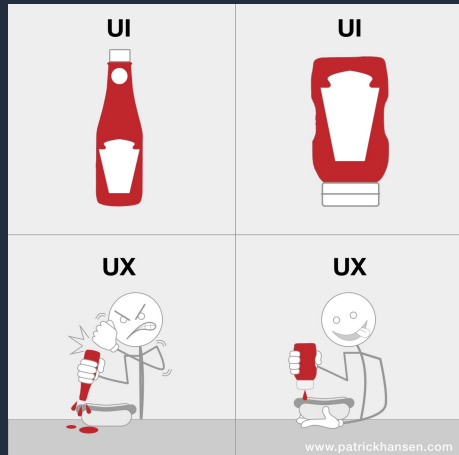
Hybrid App

하이브리드 앱이란?

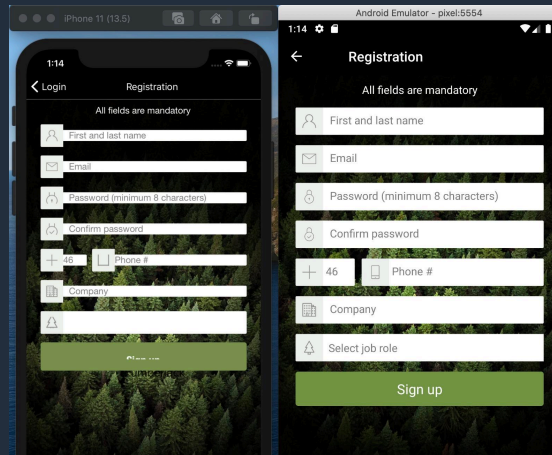
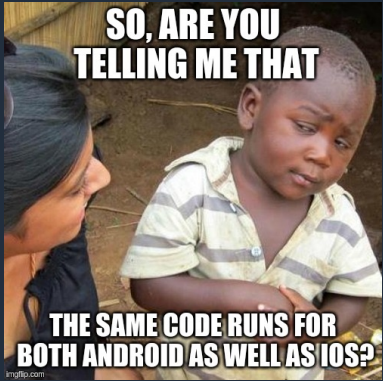
- 네이티브 앱과 웹 앱의 장점을 합친 앱
- 화면이나 기능은 웹 + 하드웨어 기능을 위한 네이티브 앱

Pros and Cons for Developer

Cons for Developer

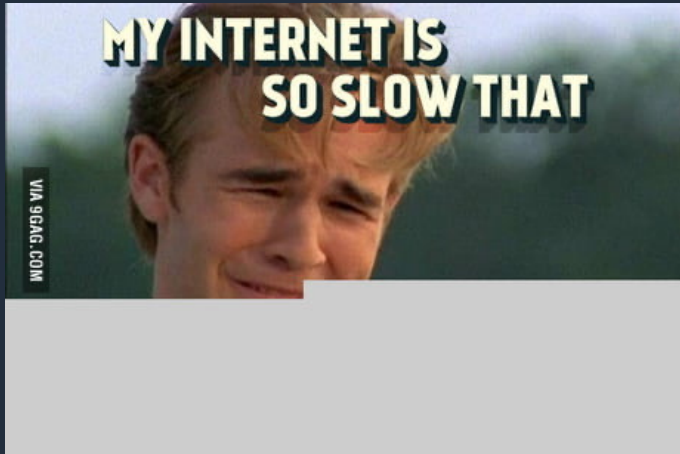


Pros for Developer



Pros and Cons for User

Cons for User



Pros for User

자동 업데이트

가벼운 앱 사이즈

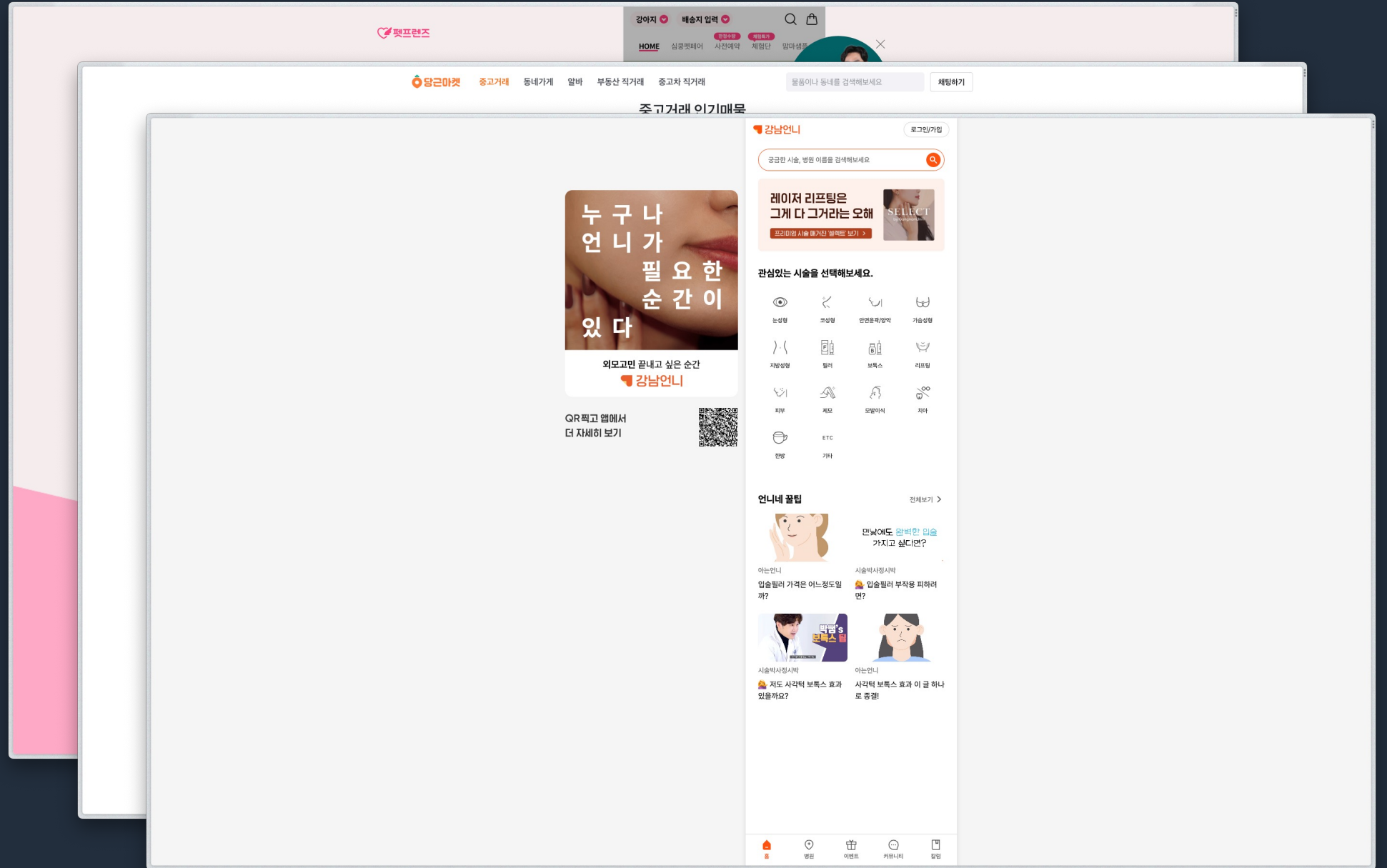
복불이 가능하다!!!!

PC와 동일한 경험

Service Perspective



Service Perspective



Develop Guidelines



SplashScreen

```
<View
  style={{
    flex: 1,
    zIndex: 0,
  }}>
  <WebView
    style={[
      styles.f1,
      {
        position: 'absolute',
        width: windowWidth,
        height: windowHeight,
      },
    ]}
    source={{uri: host}}
    ref={ref}
    onMessage={webToNative}
    onContentProcessDidTerminate={() => {
      ref.current?.reload();
    }}
    onLoad={() => {
      setTimeout(() => {
        setLoading(false);
      }, 1500);
    }}
  />
</View>
```

```
import React from 'react';
import {Dimensions, Image, StyleSheet, Animated} from 'react-native';
import Lottie from 'lottie-react-native';
import {Colors} from 'react-native/Libraries/NewAppScreen';
import SplashScreen from 'react-native-splash-screen';
import {SafeAreaProvider} from 'react-native-safe-area-context';
import WebViewPage from '../webView';

function App(): JSX.Element {
  const [loading, setLoading] = React.useState(true);
  const [splash, setSplash] = React.useState(true);
  const lottieRef = React.useRef<Lottie>(null);
  const fadeAnim = React.useRef(new Animated.Value(1)).current;
  const {width, height} = Dimensions.get('window');

  React.useEffect(() => {
    SplashScreen.hide();
  }, []);

  React.useEffect(() => {
    if (!loading) {
      Animated.timing(fadeAnim, {
        toValue: 0,
        duration: 1500,
        useNativeDriver: true,
      }).start(() => {
        setSplash(false);
      });
    }
  }, [loading, fadeAnim]);

  return (
    <SafeAreaProvider style={{[Colors.lighter, styles.f1]}}>
      {splash && (
        <Animated.View
          style={[
            styles.f1,
            {
              position: 'absolute',
              zIndex: 10,
              width,
              height,
              opacity: fadeAnim,
            },
          ]}
        >
          <Image
            source={require('./background.png')}
            style={[styles.f1, {width}]}
          />
          <Lottie
            style={{
              bottom: 0,
              width,
              position: 'absolute',
              zIndex: 10,
            }}
            ref={lottieRef}
            source={require('./lottie.json')}
            autoPlay
            loop
          />
        </Animated.View>
      )}
      <WebViewPage loading={loading} setLoading={setLoading} />
    </SafeAreaProvider>
  );
}
```

Web - App 통신

```
const getLocation = () => {
  Geolocation.getCurrentPosition(
    position => {
      setLocation(position);
      const payload = {
        lat: position.coords.latitude,
        lng: position.coords.longitude,
        speed: position.coords.speed,
      };
      ref?.current?.postMessage(
        JSON.stringify({type: 'SET_LOCATION', payload}),
      );
    },
    error => {
      console.log(error.code, error.message);
      if (error.code === 1) {
        Alert.alert(
          '위치 정보를 사용할 수 없습니다.',
          '알림 기능을 켜주세요. > 설정 > 알림 > AppName 에서 알림 권한을 허용해주세요.',
          [
            {
              text: '닫기',
            },
            {
              text: '설정으로',
              onPress: () => {
                Linking.openSettings().then(console.log);
              },
            },
          ],
        );
      }
    },
    {enableHighAccuracy: true, timeout: 15000, maximumAge: 10000},
  );
};
```

```
const webToNative = (event: WebViewMessageEvent) => {
  const {data: res} = event.nativeEvent;
  try {
    const {type, payload} = JSON.parse(res);
    if (!type || !payload) {
      return;
    }
    switch (type) {
      case 'GET_LOCATION':
        getLocation();
        break;
      case 'SEND_MESSAGE':
        sendSMS.send(
          {
            body: payload.message,
            recipients: [payload.num],
            // @ts-ignore
            successTypes: ['all'],
            allowAndroidSendWithoutReadPermission: true,
          },
        ),
        completed => {
          if (completed) {
            Alert.alert('전송 완료', '문자가 전송되었습니다.', [
              {
                text: '닫기',
                onPress: () => {
                  ref?.current?.goBack();
                },
              },
            ],
          );
        },
        break;
      default:
        break;
    }
  } catch (e) {
    console.log(e);
  }
};
```

```
const useLocation = () => {
  const receiver = React.useRef<Window | Document>();
  const [location, setLocation] = React.useState<Location>(INIT_LOCATION);
  React.useEffect(() => {
    if (typeof window === "undefined") return;
    const host = navigator.userAgent.toLowerCase().match(/(ip.*applewebkit?!.*(version|crios)/));
    receiver.current = host ? window : document;
  }, []);

  // 이 함수 호출하면 웹뷰에서 현재 위치를 받아옴
  const updateLocation = React.useCallback(() => {
    if (!receiver.current) return;
    receiver.current?.ReactNativeWebView?.postMessage(
      JSON.stringify({
        type: "GET_LOCATION",
        payload: "GET_LOCATION",
      }),
    );
  }, [receiver]);

  // EventListener 등록
  React.useEffect(() => {
    if (!receiver.current) return;
    const eventListener = (event: any) => {
      const { type, payload } = JSON.parse(event.data);
      if (type === "SET_LOCATION") {
        console.log("LOCATION IS UPDATED!!", payload);
        setLocation(payload);
      }
    };
    receiver.current.addEventListener("message", eventListener);
    return () => {
      receiver?.current?.removeEventListener("message", eventListener);
    };
  }, [receiver]);

  return {
    location,
    updateLocation,
  };
};
```

Fallback

```
<View
  style={{
    flex: 1,
    zIndex: 0,
  }}>
  <WebView
    style={[
      styles.fl1,
      {
        position: 'absolute',
        width: windowWidth,
        height: windowHeight,
      },
    ]}
    source={{uri: host}}
    ref={ref}
    onMessage={webToNative}
    onContentProcessDidTerminate={() => {
      ref.current?.reload();
    }}
    onLoad={() => {
      setTimeout(() => {
        setLoading(false);
      }, 1500);
    }}
  />
</View>
```

CI / CD



AWS Amplify



간편한 사용 – 레포지토리와 연동만 하면 알잘딱깔센

확장성, 가용성 – 완전 관리형 서비스로 자체적으로 모니터링,
로드밸런싱, 보안, CDN, SSL 등등 알아서 함

프리티어 – 12개월 제한된 리소스 안에서 무료 사용 가능

자체 도메인 제공 – 도메인을 구매하지 않아도 자체 생성된 도메인 제공

AWS App Runner



간편한 배포 – 컨테이너 이미지만 업로드 하면 알잘딱깔센

확장성, 가용성 – 컨테이너 완전 관리형 서비스로 자체적으로 모니터링,
로드밸런싱, 보안, CDN 등등 알아서 함

저렴한 비용 – 딱 쓴만큼만 내면 됨!

결론 -> AWS Amplify와 EC2에 직접 CI/CD 세팅하는 것의 중간 녀석

Vercel



간편한 사용 – 레포지토리와 연동만 하면 알잘딱깔센

확장성, 가용성 – 완전 관리형 서비스로 자체적으로 모니터링,
로드밸런싱, 보안, CDN, SSL 등등 알아서 함

저렴 할수도 있는 비용 – 깃헙 조직을 사용하지 않거나 유저가 적다면
무료일지도...PRO플랜은 다소 부담스러운 가격

자체 도메인 제공 – 도메인을 구매하지 않아도 자체 생성된 도메인 제공

Apple App Store Review

악명높은 App Store Review...

기존 웹이랑 다르게 뭐냐?

권한 받는 이유 상세하게 써라

유저의 네트워크가 불안정 한 경우 핸들링 처리 해야한다

커뮤니티가 있다면 숨김, 차단, 신고기능, 탈퇴 넣어라

소셜로그인 넣을거면 애플 꼭 넣어라

Apple로 시작하기면 원터치 로그인 되어야지 추가적인 정보 왜받냐

개발, 테스트, 준비중 등등 테스트성 단어

심사 중인데 배포 해서 깜빡거림

스토어 이미지 우리 기준에 안맞음

이 외 다수....

Q&A



Contact

KakaoTalk : hyeon0411

Email : hyeon120411@gmail.com

Github : <https://github.com/ChanghyeonYoon>

LinkedIn : <https://linkedin.com/in/yooniversedev>

Blog : <https://changhyeon.net>



Thank you!

