

# Part5. 웹뷰 앱 배포부터 업데이트까지 모든 것

웹뷰로 만든 앱 출시하기  
**학습 목표 및 구성**

## 학습 목표

01

Fastlane을 이용하여 React Native CLI Android 앱을 배포할 수 있다

02

Fastlane을 이용하여 React Native CLI iOS 앱을 배포할 수 있다

03

EAS를 이용하여 Expo Android 앱을 배포할 수 있다

04

EAS를 이용하여 Expo iOS 앱을 배포할 수 있다

# 배포 자동화의 필요성

- 복잡한 배포 프로세스

인증서 관리

앱 빌드

빌드 업로드

메타데이터 등록

리뷰 제출

# 배포 자동화 도구

**React Native CLI**

Fastlane

**Expo**

EAS  
(Expo Application Services)

웹뷰로 만든 앱 출시하기 (React Native CLI)

# Fastlane 설치하기

## 학습 목표

01

Fastlane이 무엇인지 이해한다

02

Fastlane을 설치 할 수 있다

# Fastlane

- iOS 및 Android 애플리케이션의 빌드, 테스트, 릴리즈, 배포 과정을 자동화하는 오픈 소스 툴
- 코드 서명 및 프로비저닝 프로파일 관리
- 빌드 및 테스트 자동화
- 스크린샷 생성 및 업로드
- 앱 배포
- 버전 관리 및 릴리스 노트 작성
- 통합 및 확장 가능성

Automate your development and release process

*fastlane* is an open source platform aimed at simplifying Android and iOS deployment.  
*fastlane* lets you automate every aspect of your development and release workflow.

 AUTOMATE SCREENSHOTS  
Automatically generate localized screenshots for the app store  
[LEARN MORE](#)

 BETA DEPLOYMENT  
Easily distribute beta builds to testers  
[LEARN MORE](#)

 APP STORE DEPLOYMENT  
Publish a new release to the app store in seconds  
[LEARN MORE](#)

 CODE SIGNING  
Reliably and consistently code sign your app—no more headaches  
[LEARN MORE](#)

# Fastlane 설치

- Ruby 설치
  - react native cli 셋업 시에 이미 설치했음
- Bundler 이용
  - Gemfile에 gem "fastlane" 추가
  - bundle install
  - bundle exec fastlane [lane]
- Reference
  - <https://docs.fastlane.tools/getting-started/ios/setup/>
  - <https://docs.fastlane.tools/getting-started/android/setup/>

웹뷰로 만든 앱 출시하기 (React Native CLI)

**Fastlane로 Android 앱 배포하기**

# 학습 목표

01

Fastlane을 이용하여 안드로이드 배포 자동화를 할 수 있다

## 앱 패키지 이름 변경

- 중복된 패키지 이름을 피하기 위해서 안드로이드 패키지 명 변경
- com.airecorderapp -> com.daehoon.airecorderapp

# 구글 플레이 개발자 계정

- 구글 플레이 개발자 계정 (유료: \$25)
- Reference
  - <https://play.google.com/apps/publish/signup>
  - <https://support.google.com/googleplay/android-developer/answer/6112435?hl=ko>

The screenshot shows the initial sign-up screen for the Google Play Console. At the top, there's a logo and the text "Google Play Console". Below it, a heading says "시작하려면 계정 유형을 선택하세요." (Select your account type to get started). A link "누가 계정을 사용할 것인가요? [계정 유형 선택에 관해 자세히 알아보기](#)" (Who will use this account? [Learn more about choosing an account type](#)) is provided. The main section is titled "기관/단체" (Organization/Institution) with the sub-instruction "조직 또는 비즈니스의 계정을 만들려면 선택하세요. 조직을 인증해야 합니다." (Select if you want to create an organization or business account. Your organization must be verified). A dropdown menu labeled "조직 유형 선택" (Select organization type) is open. To the right of the dropdown is a decorative icon of colorful geometric shapes. Below this section is a "개인" (Individual) section with the same verification instruction and a decorative icon of a person and a smartphone.

## 배포 시나리오

- 내부 테스팅 트랙으로 배포 자동화
- 수동으로 프로덕션으로 트랙으로 출시

# Overview

- 구글 플레이에 안드로이드 앱 생성하기
- 안드로이드 프로덕션 앱 빌드
  - 안드로이드 키 생성하기
    - keytool -genkeypair -v -storetype PKCS12 -keystore my-upload-key.keystore -alias my-key-alias -keyalg RSA -keysize 2048 -validity 10000
  - 키 정보 업데이트
    - android/app/build.gradle
  - 빌드
    - npx react-native build-android --mode=release
  - Reference
    - <https://reactnative.dev/docs/signed-apk-android#generating-an-upload-key>

# Overview

- 내부 테스팅 버전 만들기
- 프로덕션 버전으로 승급 시키고, 드래프트 상태로 만들기
  - fastlane init 명령어 에러 방지
- 카메라 퍼미션 관련 프라이버시 정책 업데이트
  - 앱 콘텐츠 -> 개인정보처리방침 url 추가
- Fastlane에서 사용할 서비스 키 발급 받기
  - [https://developers.google.com/android-publisher/getting\\_started?hl=ko](https://developers.google.com/android-publisher/getting_started?hl=ko)
- Fastlane 초기화
  - bundle exec Fastlane init
- 배포 자동화를 위한 Fastfile 코드 수정

웹뷰로 만든 앱 출시하기 (Expo)

**EAS로 Android 앱 배포하기**

# 학습 목표

01

EAS를 이용하여 안드로이드 배포 자동화를 할 수 있다

# Expo Application Services (EAS)

- Expo 앱 개발을 위한 클라우드 기반 도구
- **EAS Build** : Android와 iOS 앱 빌드를 클라우드에서 수행하도록 지원
- **EAS Submit** : 빌드된 애플리케이션을 앱 스토어(Apple App Store, Google Play)에 제출하는 과정을 자동화
- **EAS Update**: 앱 스토어를 거치지 않고 사용자에게 업데이트 제공
- **EAS Credentials**: 빌드와 배포 과정에서 필요한 인증서나 키 등을 관리

## 배포 시나리오

- 내부 테스팅 트랙으로 배포 자동화
- 수동으로 프로덕션으로 트랙으로 출시

# 구글 플레이 개발자 계정

- 구글 플레이 개발자 계정 (유료: \$25)
- Reference
  - <https://play.google.com/apps/publish/signup>
  - <https://support.google.com/googleplay/android-developer/answer/6112435?hl=ko>

The screenshot shows the Google Play Console sign-up interface. At the top, there's a logo and the text "Google Play Console". Below it, a message says "시작하려면 계정 유형을 선택하세요." (Select your account type to get started). A link "누가 계정을 사용할 것인가요? 계정 유형 선택에 관해 자세히 알아보기" (Who will use the account? Learn more about selecting account type) is provided. The main section is titled "기관/단체" (Organization/Company). It asks "조직 또는 비즈니스의 계정을 만들려면 선택하세요. 조직을 인증해야 합니다." (Select if you want to create an organization or business account. Your organization needs to be verified). A dropdown menu is open, showing "조직 유형 선택" (Select organization type). To the right of the dropdown is a decorative icon of colorful geometric shapes. Below this section is a "개인" (Individual) section with a similar verification message and a decorative icon of a person and a smartphone.

## 앱 패키지 이름 변경

- 중복된 패키지 이름을 피하기 위해서 안드로이드 패키지 명 변경
- com.daehoon.airecorderapp
- app.json 파일 수정

# Overview

- EAS 초기화
  - npm install -g eas-cli
  - eas login
  - eas init
  - <https://docs.expo.dev/tutorial/eas/configure-development-build/#install-eas-cli>
- 구글 플레이에 안드로이드 앱 생성하기
- 안드로이드 프로덕션 앱 빌드
  - 안드로이드 키 생성 및 키 정보는 EAS가 자동으로 관리
  - 빌드
    - eas build --platform android
  - <https://docs.expo.dev/tutorial/eas/android-production-build/>
- 내부 테스팅 버전 만들기
- 카메라 퍼미션 관련 프라이버시 정책 업데이트
  - 앱 콘텐츠 -> 개인정보처리방침 url 추가

# Overview

- EAS에서 사용할 서비스 키 발급 받기
  - React Native CLI 클립 참고
  - [https://developers.google.com/android-publisher/getting\\_started?hl=ko](https://developers.google.com/android-publisher/getting_started?hl=ko)
- eas.json에 서비스 키 등록
- 배포 자동화 명령어
  - eas build --platform android --auto-submit

웹뷰로 만든 앱 출시하기 (React Native CLI)

**Fastlane로 iOS 앱 배포하기**

# 학습 목표

01

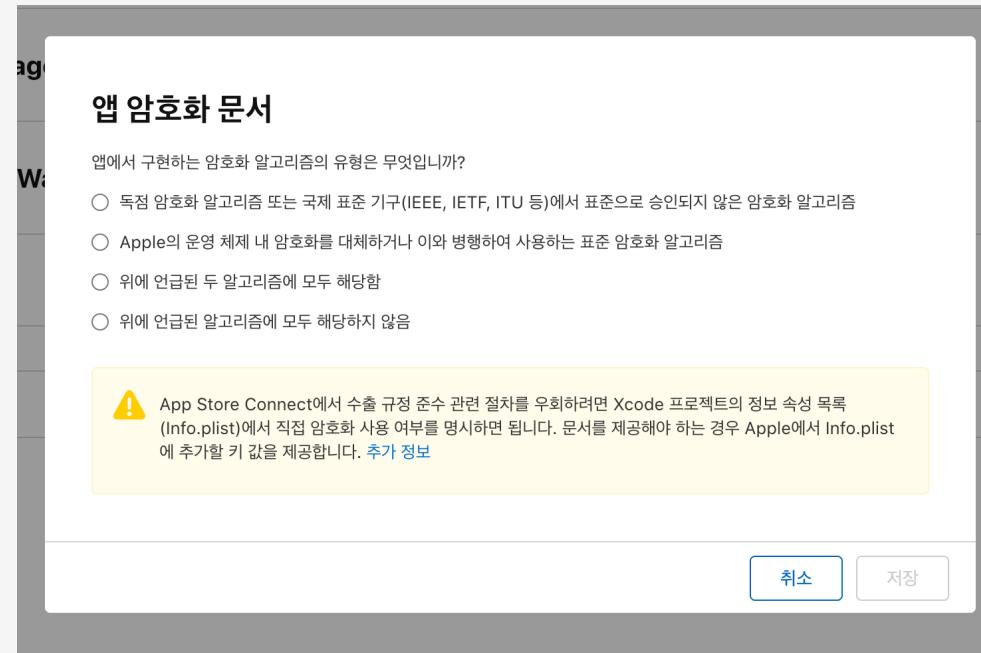
Fastlane을 이용하여 iOS 앱 배포 자동화를 할 수 있다

## 앱 번들 ID 변경

- 중복된 앱 번들 ID를 피하기 위해서 변경
- org.reactjs.native.example.AiRecorderApp -> com.daehoon.airecorderapp

# 사전 작업

- 앱 아이콘 설정 필요
- 수출 규정 준수 관련 절차 우회
  - <key>ITSApplUsesNonExemptEncryption</key>  
<false/>



# 애플 개발자 계정

- Apple Developer Program 가입 (연 129,000원)
- Reference
  - <https://developer.apple.com/kr/programs/>

# Overview

- App Store Connect에 신규 앱 생성하기
- Fastlane 초기화
  - bundle exec fastlane init
- 배포용 인증서 발급
  - bundle exec fastlane match init
  - bundle exec fastlane match Appstore
- AppStoreConnect 인증하기
  - App Store Connect API key 이용
  - <https://docs.fastlane.tools/getting-started/ios/authentication/>
- 배포 자동화 명령어
  - Bundle exec fastlane release

웹뷰로 만든 앱 출시하기 (Expo)

EAS로 iOS 앱 배포하기

## 학습 목표

01

EAS을 이용하여 iOS 앱 배포 자동화를 할 수 있다

## 앱 번들 ID 변경

- 중복된 앱 번들 ID를 피하기 위해서 변경
- com.daehoon.airecorderapp
- app.json 수정

## 사전 작업

- 앱 아이콘 변경 – assets/icon.png 파일 변경

# 애플 개발자 계정

- Apple Developer Program 가입 (연 129,000원)
- Reference
  - <https://developer.apple.com/kr/programs/>

# Overview

- EAS 초기화
  - npm install -g eas-cli
  - eas login
  - eas init
  - <https://docs.expo.dev/tutorial/eas/configure-development-build/#install-eas-cli>
- 배포용 인증서 발급
  - eas credentials
- AppStoreConnect 인증하기
  - EAS가 자동으로 처리
- 배포 자동화 명령어
  - eas build --platform ios --auto-submit

웹뷰로 만든 앱 출시하기

학습 리뷰

# 앱 배포 자동화 도구

- **React Native CLI – Fastlane**

- iOS 및 Android 애플리케이션의 빌드, 테스트, 릴리즈, 배포 과정을 자동화하는 오픈 소스 툴
- 코드 서명 및 프로비저닝 프로파일 관리
- 빌드 및 테스트 자동화
- 스크린샷 생성 및 업로드
- 앱 배포
- 버전 관리 및 릴리스 노트 작성
- 통합 및 확장 가능성

- **Expo – Expo Application Services**

- Expo 앱 개발을 위한 클라우드 기반 도구
- EAS Build : Android와 iOS 앱 빌드를 클라우드에서 수행하도록 지원
- EAS Submit : 빌드된 애플리케이션을 앱 스토어(Apple App Store, Google Play)에 제출하는 과정을 자동화
- EAS Update: 앱 스토어를 거치지 않고 사용자에게 업데이트 제공
- EAS Credentials: 빌드와 배포 과정에서 필요한 인증서나 키 등을 관리

# 인증서 관리

- React Native CLI
  - Android: keytool 명령어
  - iOS: fastlane match
- Expo – Expo Application Services
  - Android: EAS 내부 관리
  - iOS: eas credentials

웹뷰로 만든 앱 업데이트 하기

# 학습 목표 및 구성

## 학습 목표

01

웹 프로젝트를 호스팅하여 웹앱의 콘텐츠를 즉시 업데이트 할 수 있다

02

스토어(App Store, Play Store)에 앱을 제출하여 앱을 업데이트 할 수 있다

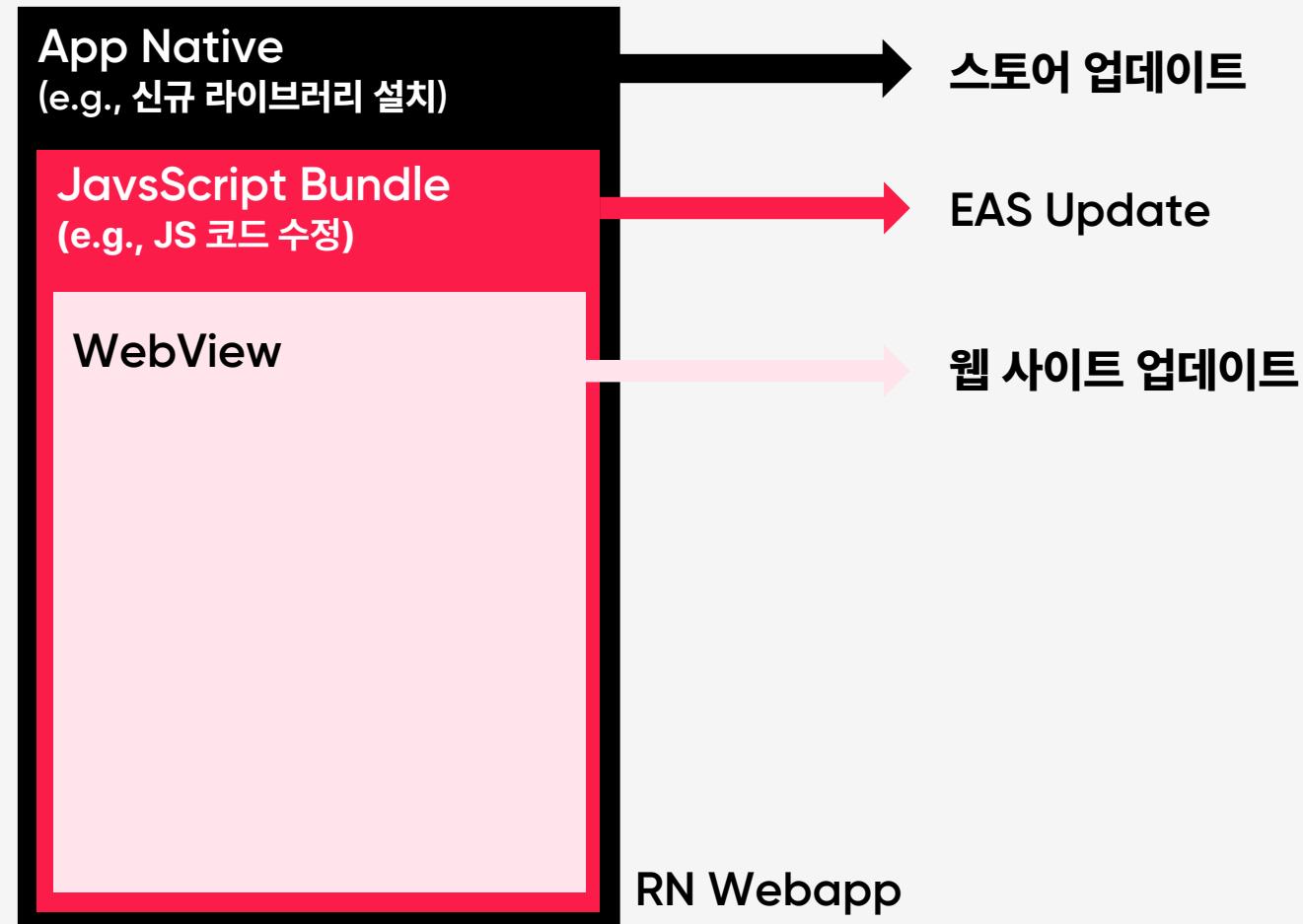
03

EAS Update를 이용하여 스토어 제출 없이 앱을 업데이트 할 수 있다

04

업데이트 알림을 사용자에게 보낼 수 있다

# 웹앱 업데이트 종류



웹뷰로 만든 앱 업데이트 하기 (웹)

# 웹 사이트 업데이트를 통한 실시간 업데이트

# 학습 목표

01

NextJS로 만든 웹 프로젝트를 온라인에 배포 할 수 있다

# Vercel을 이용한 NextJs 프로젝트 배포

- 웹앱을 서비스하기 위해서는 웹 사이트를 온라인에 호스팅 해야함
- Vercel
  - NextJs에 개발팀에서 만든 호스팅 서비스
  - <https://vercel.com/>
- 배포 명령어
  - npm i -g vercel
  - vercel login
  - vercel --prod
  - 환경 변수 설정

웹뷰로 만든 앱 업데이트 하기 (앱)

**Fastlane으로 앱 업데이트 배포하기**

## 학습 목표

01

배포한 iOS 및 안드로이드 앱을 설치할 수 있다

02

iOS 및 안드로이드 앱을 업데이트 할 수 있다 (React Native CLI, Expo)

# iOS 앱 설치 및 업데이트 방법

- 정식 출시 전에는 TestFlight 앱을 통해서 앱을 설치 할 수 있음
  - 앱 스토어에서 설치 가능
- 테스터 등록
  - AppStoreConnect -> TestFlight -> 내부 테스팅 그룹 추가 -> 테스터 등록 -> TestFlight 설치 -> 초대 이메일 수락
- 업데이트
  - React Native CLI
    - bundle exec fastlane release
  - Expo
    - eas build --platform ios --auto-submit

# Android 앱 설치 및 업데이트 방법

- 정식 출시 전에는 테스터를 추가해서 설치 가능
- 테스터 등록
  - Google Play Console -> 내부 테스트 -> 테스터 -> 테스터 등록 -> 초대 링크를 통해 참여 수락
- 업데이트
  - React Native CLI
    - bundle exec fastlane beta
  - Expo
    - eas build --platform android --auto-submit

웹뷰로 만든 앱 업데이트 하기 (앱)

# EAS Update 소개 및 설치

# AppCenter Retirement

- AppCenter: Code Push 운영 서버
- AppCenter 종료 후 Code Push 사용 방법이 불명확한 상태
- 대체 서비스로 EAS Update 소개

## ☞ Overview

### Important

Visual Studio App Center is scheduled for retirement on **March 31, 2025**. While you can continue to use Visual Studio App Center until it is fully retired, there are several recommended alternatives that you may consider migrating to.

[Learn more about support timelines and alternatives.](#)

## 학습 목표

01

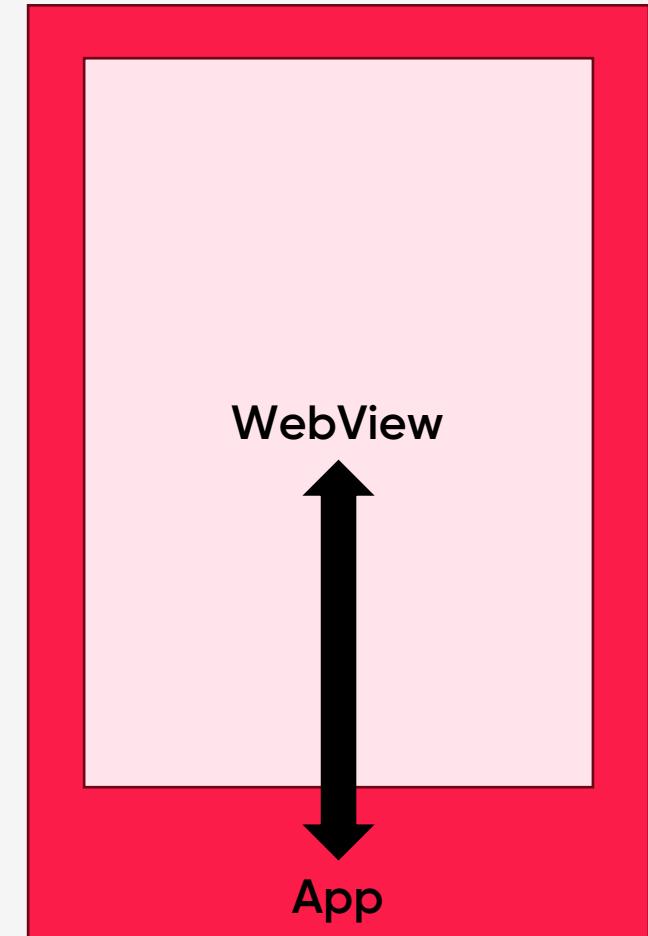
EAS Update가 무엇인지 이해한다

02

React Native CLI 프로젝트에 EAS Update를 설치할 수 있다

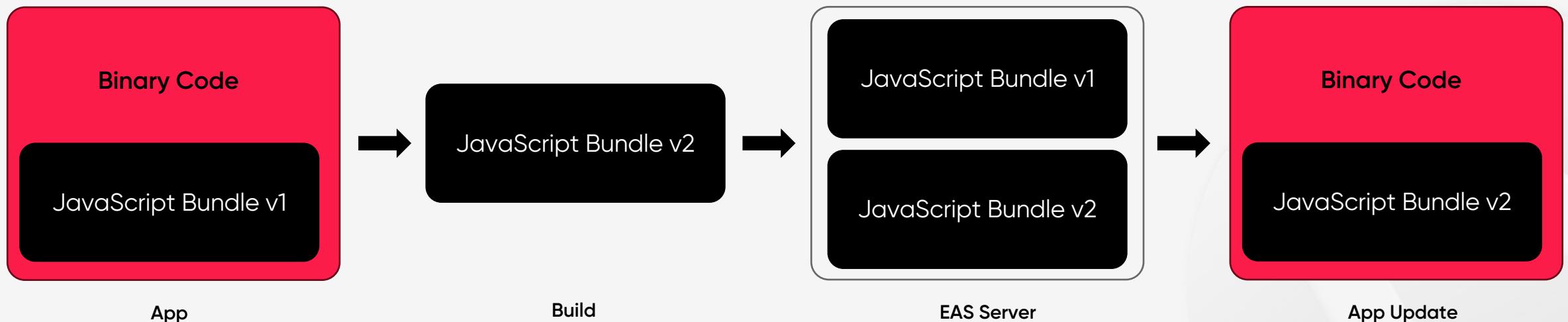
## 스토어 제출 없이 업데이트

- 웹뷰가 아닌 앱에 업데이트를 하려면 스토어에 제출이 필요
- 앱 업데이트는 리뷰 과정에 시간이 걸림
- EAS Update는 스토어 제출 없이 앱을 업데이트 할 수 있음



# EAS Update

- Expo Application Services (EAS)의 한 기능으로, 애플리케이션을 앱 스토어에 다시 배포하지 않고 원격으로 업데이트
- 실시간으로 사용자에게 최신 기능, 버그 수정 및 콘텐츠를 제공 가능
- JavaScript 부분 코드만 업데이트 가능! (바이너리 부분 업데이트는 불가능)



# EAS CLI 설치 (React Native CLI)

- EAS 서비스를 실행하기 위한 애플리케이션
- 설치 방법
  - Expo 회원 가입: <https://expo.dev/>
  - npm install -g eas-cli
  - eas login

# Expo Module 설치 (React Native CLI)

- React Native CLI에서 Expo Update를 사용하려면 Expo Module 설치 필요
- 설치 방법
  - `npx install-expo-modules@latest`
  - `bundle exec pod install`
- 참고 링크
  - <https://docs.expo.dev/bare/installing-expo-modules/>

# EAS Update 설치 (React Native CLI)

- 방법
  - npx expo install expo-updates
  - bundle exec pod install
  - eas update:configure
  - 바이너리 코드 설정 -> 다른 클립에서 소개
- 참고 링크
  - <https://docs.expo.dev/bare/installing-updates>

# EAS Update 설치 (Expo)

- 방법
  - `npx expo install expo-updates`
  - `eas update:configure`
- 참고 링크
  - <https://docs.expo.dev/eas-update/getting-started/>

웹뷰로 만든 앱 업데이트 하기 (앱)

**EAS Update** 셋업

**Android (React Native CLI)**

# EAS Update Android 설정 (React Native CLI)

- 방법
  - 안드로이드 네이티브 파일 수정
  - 업데이트 채널 설정
- 참고 링크
  - <https://docs.expo.dev/bare/installing-updates/#android>
  - <https://docs.expo.dev/eas-update/updating-your-app/#configuring-the-channel-manually>

웹뷰로 만든 앱 업데이트 하기 (앱)

# EAS Update 셋업

## iOS (React Native CLI)

# EAS Update iOS 설정 (React Native CLI)

- 방법
  - iOS 네이티브 파일 수정
  - 업데이트 채널 설정
- 참고 링크
  - <https://docs.expo.dev/bare/installing-updates/#ios>
  - <https://docs.expo.dev/eas-update/updating-your-app/#configuring-the-channel-manually>

웹뷰로 만든 앱 업데이트 하기 (앱)

# EAS Update 셋업

## 업데이트 스크린 만들기

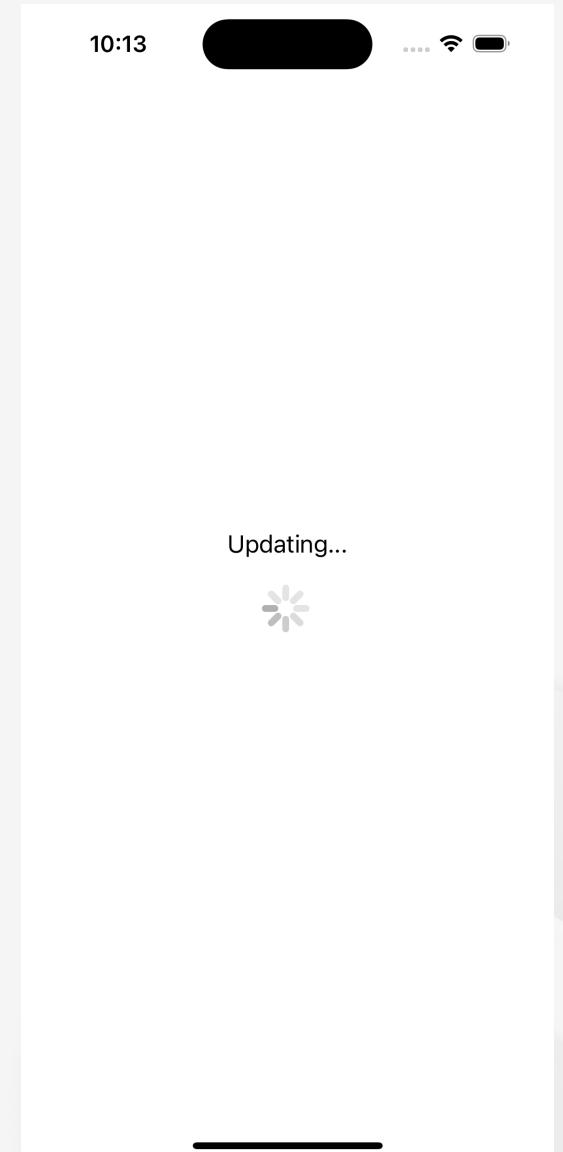
# EAS Update 시나리오

- **기본값**

- 앱이 켜지면 백그라운드에서 업데이트를 다운로드
- 사용자가 앱을 다시 켜면 업데이트가 적용
- 문제점 – 앱을 다시 켰을 때 업데이트가 적용되어서, 잘못된 버전의 앱을 사용자가 사용할 수 있음

- **변경 사항**

- 앱이 켜지면 업데이트를 다운받고 즉시 적용
- 업데이트 다운로드 받는 동안 스크린 보여주기



웹뷰로 만든 앱 업데이트 하기 (앱)

# EAS Update 셋업 업데이트 로직 구현

# EAS Update 수동 업데이트

- 업데이트 확인 시점 수정
  - React Native CLI
    - iOS: /Supporting/Expo.plist 수정 (EXUpdatesCheckOnLaunch)
    - Android: main/AndroidManifest.xml 수정 (EXPO\_UPDATES\_CHECK\_ON\_LAUNCH)
  - Expo
    - app.json 수정 (updates. checkAutomatically)
- 업데이트 관련 함수
  - 업데이트 확인: Updates.checkForUpdateAsync();
  - 업데이트 다운로드: Updates.fetchUpdateAsync();
  - 업데이트 적용: Updates.reloadAsync();

웹뷰로 만든 앱 업데이트 하기 (앱)

**EAS Update로 앱 업데이트 배포하기**

# 앱 업데이트 배포하기

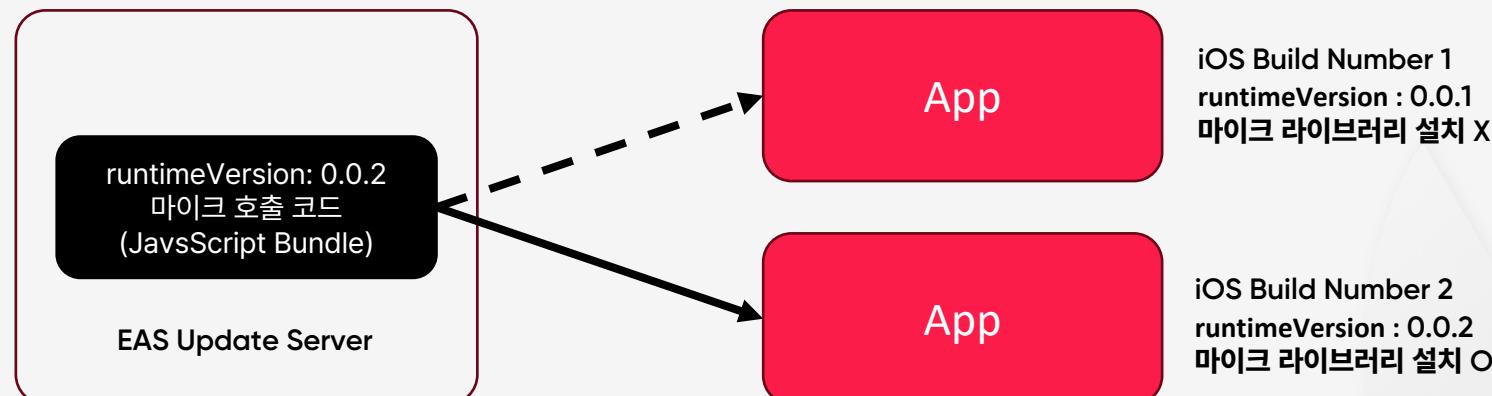
- EAS Update 업데이트 로직 구현 후 스토어에 앱 업데이트 제출 필요!
  - React Native CLI
    - 네이티브 버전 업데이트
    - iOS: bundle exec fastlane release
    - Android: bundle exec fastlane beta
  - Expo
    - app.json에서 ios.buildNumber, android.versionCode 업데이트
    - iOS: eas build --platform ios --auto-submit
    - Android: eas build --platform android --auto-submit

# 앱 업데이트 배포하기

- 채널 생성
  - React Native CLI
    - eas channel:create CHANNEL\_NAME
  - Expo
    - 자동 생성됨
- 업데이트 제출
  - eas update --branch BRANCH\_NAME
  - BRANCH\_NAME은 CHANNEL\_NAME과 동일하게 설정하면 자동으로 채널과 연결됨
  - Branch?
    - 업데이트를 생성하는 장소
    - 예) v1, v2
  - Channel?
    - 브랜치의 업데이트를 제공하는 곳
    - 예) production, preview

# 바이너리 변경 사항이 있으면?

- EAS Update는 JS 변경 사항만 반영 가능
- 바이너리 부분에 변경이 있을 때는 스토어 업데이트 필요
  - 예) 라이브러리 추가로 인한 pod 내용 변경
- 버전 업데이트
  - 네이티브 버전 코드 업데이트 (앱 배포 때 버전 업데이트 하던 것)
  - **runtimeVersion 전략에 따라서 버전 업데이트 필요**
    - 버전 업데이트 하지 않으면, 이전 바이너리에 업데이트가 적용될 수 있음
  - 참고 링크: <https://docs.expo.dev/eas-update/runtime-versions/>



웹뷰로 만든 앱 업데이트 하기 (앱)

# 웹뷰 리소스 업데이트 후 푸시 알림 전송 기능 구현하기 1

## 학습 목표

01

React Native CLI 프로젝트에서 푸시 알림 기능을 구현 할 수 있다

02

Firebase를 이용하여 푸시 알림 전송 API를 구현할 수 있다

# 푸시 알림 토큰 가져오기 (React Native CLI)

- 토큰
  - 푸시 메시지를 전송할 주소
  - Firebase Cloud Messaging 서비스를 이용하여 토큰 가져오기
- Firebase Project 생성
- Firebase 관련 라이브러리 설치
  - <https://rnfirebase.io/>
  - <https://rnfirebase.io/messaging/usage>
- 관련 함수
  - messaging().requestPermission()
  - PermissionsAndroid.request()
  - messaging().getToken()

# 업데이트 알림 전송 과정

- **토큰 업로드 API**
  - 앱에서 FCM 토큰을 Firestore에 저장
- **업데이트 알림 API**
  - Firestore에서 저장된 FCM 토큰 가져오기
  - Firebase Messaging 이용하여 푸시 알림 전송하기
- **Firebase Cloud Functions 이용하여 HTTP API 제작**
  - 서비스 환경에서 자동으로 실행되는 백엔드 코드를 작성하고 배포할 수 있는 서비스
  - <https://firebase.google.com/docs/functions/get-started>

웹뷰로 만든 앱 업데이트 하기 (앱)

웹뷰 리소스 업데이트 후 푸시 알림

전송 기능 구현하기 2

## 학습 목표

01

Expo 프로젝트에서 푸시 알림 기능을 구현 할 수 있다

02

Firebase를 이용하여 푸시 알림 전송 API를 구현할 수 있다

# 푸시 알림 토큰 가져오기 (Expo)

- 토큰
  - 푸시 메시지를 전송할 주소
  - Expo 서비스를 이용하여 토큰 가져오기
- 라이브러리 설치
  - `npx expo install expo-notifications expo-device expo-constants`
- Firebase 프로젝트 초기화 및 Credential 설정
- 관련 함수
  - `Notifications.setNotificationChannelAsync`
  - `Notifications.getPermissionsAsync`
  - `Notifications.requestPermissionsAsync`
  - `Notifications.getExpoPushTokenAsync`
- 참고 링크
  - <https://docs.expo.dev/push-notifications/push-notifications-setup/>

# 업데이트 알림 전송 과정

- **토큰 업로드 API**
  - 앱에서 Expo 토큰을 Firestore에 저장
- **업데이트 알림 API**
  - Firestore에서 저장된 Expo 토큰 가져오기
  - Expo Server SDK 이용하여 푸시 알림 전송하기
  - <https://docs.expo.dev/push-notifications/sending-notifications/>
- **Firebase Cloud Functions 이용하여 HTTP API 제작**
  - 서비스 환경에서 자동으로 실행되는 백엔드 코드를 작성하고 배포할 수 있는 서비스
  - <https://firebase.google.com/docs/functions/get-started>

웹뷰로 만든 앱 업데이트 하기

학습 리뷰

# 웹앱 업데이트

- 웹 사이트 업데이트
  - 웹앱을 서비스하기 위해서는 웹 사이트를 온라인에 호스팅 해야함 (Vercel)
- 앱 업데이트
  - App Native 변경 사항이 있을 때 스토어를 통해서 업데이트
  - 앱 리뷰 과정이 있어서 업데이트 적용까지 시간이 걸림
  - React Native CLI – Fastlane을 이용한 배포 자동화
  - Expo – EAS를 이용한 배포 자동화
- JS 번들 업데이트
  - JS 번들 부분에만 변경 사항이 있을 때
  - 리뷰 없이 바로 적용 가능
  - EAS Update 이용
  - App Native 변경 있을 시에 이전 앱 버전에 업데이트가 제공되지 않도록 runtime version을 업데이트 해줘야 함
- 업데이트 알림 전송
  - React Native CLI – Firebase Cloud Messaging
  - Expo – EAS Notifications
  - API – Firestore, Firebase Cloud Functions

**감사합니다!**