

크로스플랫폼(Flutter), 레인보우 개발기

글. 김동희 CBS 디지털콘텐츠국 플랫폼개발부 엔지니어

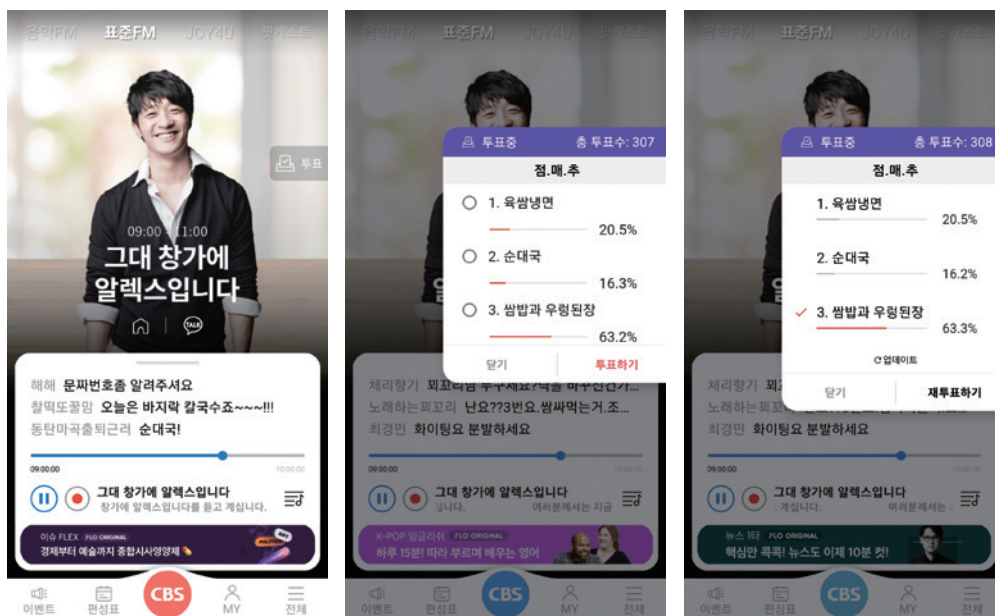


CBS는 올해 6월, 새로운 라디오 스트리밍 애플리케이션 레인보우를 새롭게 출시했다. 2006년 처음 ‘CBS 레인보우’를 배포한 후 15년 만의 새 단장이다. 새로운 ‘CBS 레인보우’는 청취자의 재미와 편의를 위해 차별화된 기능을 추가했다. 그중 몇 가지를 소개하고자 한다.

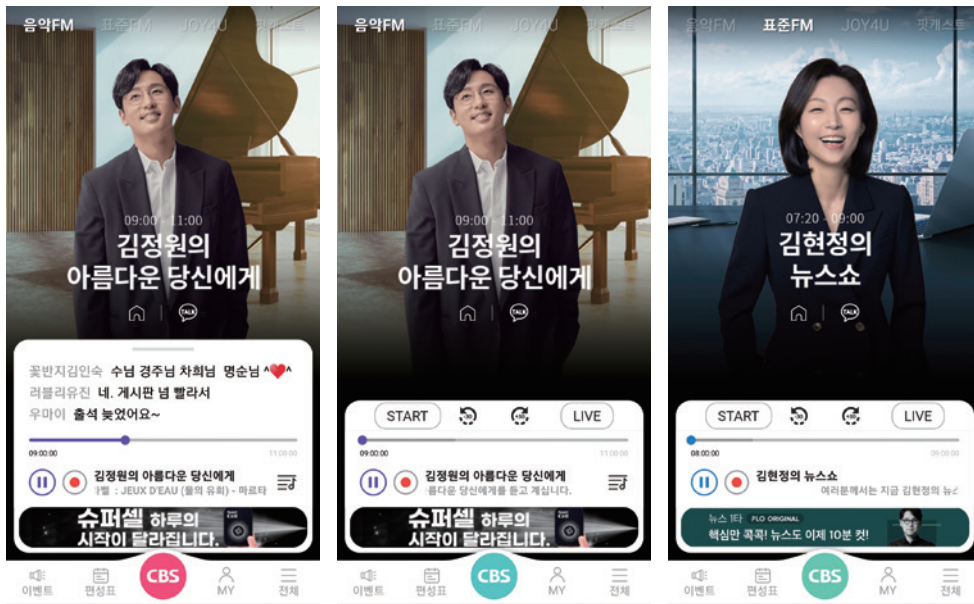
투표 기능

실시간으로 진행자와 청취자가 투표하며 소통할 수 있는 기능이다.

제작진이 투표를 등록하면 현재 레인보우를 청취하고 있는 사용자들에게 투표 팝업이 나타난다. 사용자들은 이 팝업을 통해 투표에 참여할 수 있다. 항목별 참여 수가 실시간으로 나타나고, 함께 결과를 보며 자유롭게 소통할 수 있다.



투표 이미지



타임머신 이미지

타임머신 기능

생방송에 늦게 입장한 청취자가 해당 방송을 처음부터 들을 수 있게 해주는 기능이다.

CBS 레인보우 앱 플레이어 영역에 있는 진행바를 왼쪽으로 밀어 방송을 처음부터 들을 수 있다. 타임머신 기능이 활성화되면 라이브 요소(실시간 채팅, 실시간 선곡표, 보이는 라디오 등)는 차단되어 기능이 제한된다. 청취자는 원할 시 언제든지 'LIVE' 버튼을 클릭하여 다시 실시간 라디오를 청취할 수 있다.

프로그램 홈

- 프로그램 홈을 통해 제작진/진행자가 공지할 내용이나 공유하고 싶은 사진들을 청취자들에게 전달할 수 있다.
- 청취자는 원하는 신청 곡이나 사연을 실시간으로 등록할 수 있고, 다른 청취자의 사연들을 보며 댓글을 남길 수 있다.

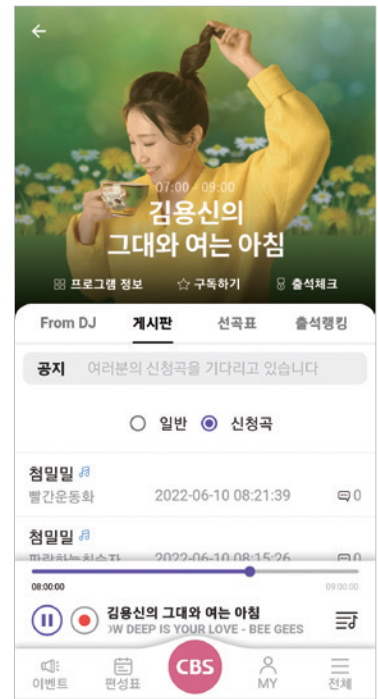
이외에도 제공되는 다양한 기능들이 많으니 사용해보시길 권해드립니다.

CBS 레인보우 홍보는 여기서 마치고,

크로스플랫폼이란 무엇이며, 플랫폼개발부가 많은 크로스플랫폼 중 'Flutter'를 선택하게 된 이유와 장단점을 설명하고자 한다.

애플리케이션 개발 방법

Play store(Android), App store(iOS)에 배포할 앱을 만드는 방법은 크게 4가지 정도로 분류된다. 네이티브, 웹앱, 하이브리드, 크로스플랫폼이 있다. 방법마다 장단점이 뚜렷하게 존재하고, 지금도 앱 개발을 위한 다양한 방법들이 시도되고 있다. 이 중 크로스플랫폼에 대해 알아보자.



프로그램 홈 이미지

크로스플랫폼이란?

크로스플랫폼이란, 하나의 소스로 두 개 이상의 플랫폼(운영체제)에서 실행 가능한 소프트웨어를 말한다. 대표적으로 리액트 네이티브(React Native), 자마린(Xamarin), 일렉트론(Electron) 그리고 플러터(Flutter)를 들 수 있다.

React Native는 모바일 애플리케이션을 만들기 위한 JavaScript 기반 프레임워크이다. 2015년 Facebook에서 오픈 소스 프로젝트로 처음 출시하여 운영되고 있고, 리액트 네이티브로 개발된 대표적인 앱은 월마트, 스카이프 등이 있다.

주목할 만한 점은 Airbnb가 애플리케이션 개발 프레임워크로 React Native를 채택했다가 추가 개발을 포기하고 안드로이드, IOS 네이티브로 전환했다는 점이다. 왜 Airbnb가 막대한 비용을 들여가며 네이티브로의 전환을 선택했는지 '크로스플랫폼의 단점'에서 설명할 예정이다.

Xamarin은 .NET, C#을 베이스로 한 IOS, Android, 더 나아가 Windows 용 애플리케이션까지 개발할 수 있는 오픈소스 프레임워크이다. Xamarin으로 만들어진 대표적인 앱은 스토리오-비디오 스토리 크리에이터, 온라인 기금 마련 플랫폼인 저스트기빙 등이 있다.

Electron은 OpenJS Foundation에서 개발한 Javascript 기반 프레임워크이다. 데스크톱 애플리케이션을 만드는데 최적화되어 있으며 VSCode, Slack, Discord 등 세계적으로 유명한 프로그램들이 Electron으로 개발되었다. 완성도가 뛰어나 성능과 개발 편의성은 검증되었다고 볼 수 있다. 하지만 구동에 필요한 기본 용량이 100MB 이상이라 모바일 애플리케이션용으로는 잘 사용되지 않는다.

Flutter는 Dart를 기반으로 한 Google의 오픈소스 프레임워크이다. 단일 코드 베이스를 활용하여 Android, IOS, Web, Windows 등 다양한 플랫폼에 고성능 애플리케이션을 빌드할 수 있도록 지원한다. 특히, 개발에 많은 시간이 필요한 UI 요소를 고성능 위젯 라이브러리를 제공함으로써 편리한 개발 환경을 지원한다.

이처럼, 개발 편의성을 갖춘 다양한 프레임워크들이 존재한다. 개발자들은 개발 목표에 따라 검증된 프레임워크를 선택하여 멀티 플랫폼을 위한 애플리케이션을 빠르게 제작할 수 있다.

Flutter를 선택하게 된 배경

Flutter는 2017년 5월, Google에서 출시한 모바일 / 웹 / 데스크톱 애플리케이션 개발을 위한 프레임워크이며 2011년 10월에 Google에서 공개한 'Dart'라는 다소 생소한 언어를 사용한다. 새로운 언어를 학습해야 함에도 Flutter를 선택하게 된 배경은 아래와 같다.

모바일과 PC

Flutter는 하나의 프로젝트로 모바일, 데스크톱을 동시에 개발할 수 있는 것이 가장 큰 장점이다.

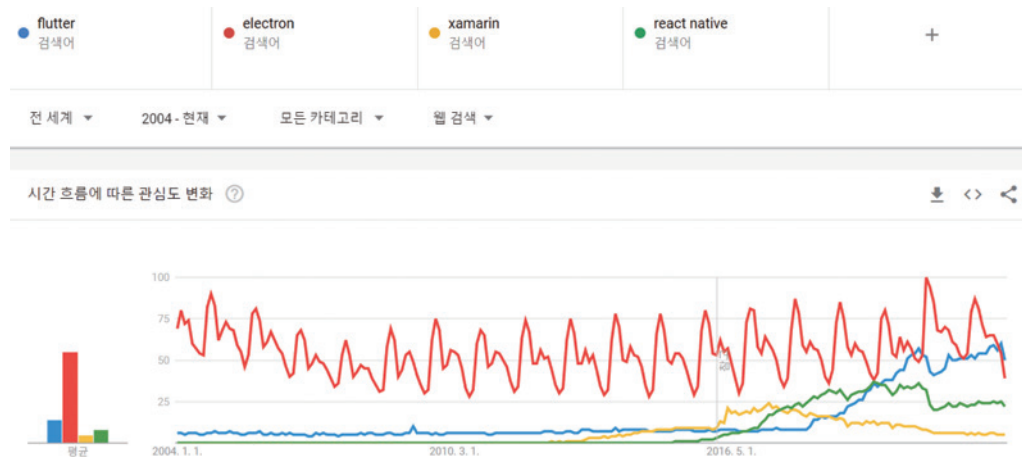
CBS 라인보우는 2016년 Android, IOS, Windows 버전으로 배포되었다. Android는 Java, IOS는 Object-C 그리고 Windows는 C++로 개발되었다. 플랫폼마다 다른 언어로 개발된 라인보우는 새로운 기능을 추가하거나 문제가 발생했을 때 각기 개발된 언어로 문제를 해결해야 했다. 한정적인 개발 인력이 감당하기에는 어려운 일이었다. 이와 반대로 크로스플랫폼은 소수의 개발 인원이 하나의 소스 코드를 공유하고 발전시켜 나갈 수 있다는 점에서 효율적이다.

구글의 지원

Flutter는 Google이 개발하여 출시했다. Facebook의 React Native와 비교했을 때, 개발사의 지원이 적극적일 것으로 판단했다.

실제로 2021년 5월 처음 Flutter로 개발한 CBS 노컷뉴스 앱을 배포했고 Flutter는 Version 1.0.0을 시작으로 현재 v3.0.0까지 빠르게 업데이트되었다. 이처럼 구글 'Flutter'팀의 지속적인 서포트로 더 안정적인 앱 개발을 이어올 수 있었다.

위 Google Trends와 Stack Overflow 조사에서도 확인할 수 있듯이, Flutter의 높은 안정성과 지속적인 업데이트는 많은 개발자의 관심과 검색량 증가로 이어지고 있다고 생각한다.



대표적인 크로스플랫폼 관심도 변화

Most Loved, Dreaded, and Wanted Other Frameworks, Libraries, and Tools

Loved	Dreaded	Wanted
.NET Core	77.2%	
Torch/PyTorch	77.1%	
Flutter	75.4%	
Pandas	74.7%	
TensorFlow	73.0%	
Node.js	72.1%	
Apache Spark	63.5%	
React Native	62.5%	
Ansible	61.7%	
.NET	61.0%	
Unity 3D	60.9%	
Unreal Engine	59.5%	
Hadoop	53.3%	
Xamarin	48.3%	
CryEngine	42.7%	
Puppet	38.2%	
Cordova	36.7%	
Chef	33.3%	

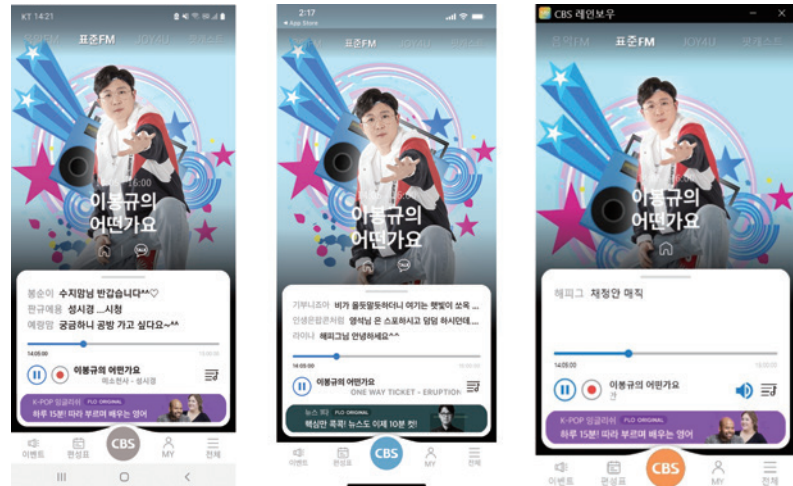
% of developers who are developing with the language or technology and have expressed interest in continuing to develop

Flutter에 대한 선호도와 인기를 증명하는 조사 결과

편리한 UI 개발

애플리케이션을 네이티브로 개발하다 보면 서로 비슷한 UI를 만들기 위해 상당한 시간이 소요된다. Flutter는 다양한 Widget 라이브러리를 제공하며 같은 UI와 비즈니스 로직을 모든 플랫폼에 적용할 수 있다.

현재 배포된 Android, IOS와 배포 예정인 Windows 애플리케이션의 사진을 보면 하나의 코드로 구현된 화면의 완성도가 매우 높다는 것을 확인할 수 있다.



왼쪽부터 Android, IOS, Windows 레인보우 화면

‘핫 리로드’

Flutter는 ‘핫 리로드’ 기능을 제공한다. 소스 코드를 수정하면 실행 중인 Dart 가상머신(VM)이 수정 사항을 실시간으로 다시 그려서 Simulator나 On Device 화면에 바로 반영시킨다. 실제로 개발을 진행해봤다면 작은 수정에도 결과를 확인하는데 있어 꽤 오랜 시간을 기다려야 했던 경험이 있을 것이다. 이런 불편함을 해소해 주는 기능인 ‘핫 리로드’를 경험해 보시길 추천한다.

효율적인 유지보수

제한된 소프트웨어 개발 인력은 크로스플랫폼으로 전향하는 가장 큰 계기가 됐다. 담당자가 부재중이어도 부서 내에서 문제를 파악해 빠르게 수정하고 배포하는 것이 가능했다. 애플리케이션은 초기 개발보다 유지보수에 더 많은 시간과 노력이 필요하다. 코드 재사용과 외부 패키지를 사용하여 새로운 기능을 빠르게 적용해 볼 수 있다는 건 큰 장점이다.

Flutter의 단점

Third Party 플러그인

Flutter는 Third party 플러그인으로 편리한 개발이 가능하지만, Flutter 버전이 상승할 때 문제가 발생하기도 한다. 최신화된 Flutter 엔진을 사용하기 위해 버전을 올리면 기존에 사용하던 Third party에서 수많은 에러가 나타난다. Third party에서 신속하게 문제를 해결하고 새로운 버전을 올리지 않는 이상 Flutter 버전을 올리는 것이 어려워진다. 쉽게 설명해서

Third-Party 플러그인을 많이 사용하면 유지보수 시 상당한 시간이 걸릴 수 있다는 것이다. 플러그인을 직접 수정할 수 있다면 가장 좋겠지만, 다른 사람이 작성한 코드를 수정하고 안정적으로 동작시키는 것은 쉽지 않은 일이다.

따라서 우측 이미지의

계속된 업데이트와 추천을 많이 받은 플러그인 이미지

외부 플러그인처럼 개발자가 지속해서 업데이트를 진행하고 있는지 또는 개발자들의 추천을 많이 받았는지 등을 확인해 보는 것을 추천한다.

네이티브 의존성

Dart를 사용하여 Flutter를 개발할 때 Native 언어(Kotlin, Swift 등)에 대한 이해도가 없다면, 복잡한 기능 개발이 어려울 수 있다.

* Airbnb가 React Native에서 Android, iOS 네이티브 개발로 돌아선 이유도 깊어지는 네이티브 의존성 때문이었던 것으로 보인다. 회사가 거대해지고 사용자가 많아지면 더 복잡하고 정교한 기술, 개발이 필요하다. Airbnb 앱 개발 당시의 Flutter는 초기 버전이라 현재 배포된 Flutter version보다 제공하는 Native 코어 기능이 적었을 것이다. Airbnb는 필요한 Native 코어 작업을 위해 React Native와 Android, iOS 간의 연동을 반복해야 했고, 더 많은 작업과 노력이 필요했을 것이다. 개발, 유지보수의 편리함을 위해 크로스플랫폼을 선택했지만, 네이티브로 개발할 때보다 효율이 낮아지는 딜레마에 빠질 수 있었던 것이다.

레인보우는 Streaming CDN 업체에서 따로 제공해주는 Java 기반의 플레이어를 사용한다. 해당 플레이어는 CDN(스트리밍 서버) - Client(애플리케이션) 간 안정성이 보장된 상황이었다. 하지만 Java로 만들어졌기 때문에 Flutter에서 독립적으로 동작하지 않았고 Android와 연동 작업이 필요했다. Flutter에서 제공하는 Method Channel을 사용하면 연동이 어렵진 않지만, Android 네이티브 개발에 쓰이는 언어인 Java나 Kotlin에 대한 이해가 필요했다.

향후 과제

플랫폼 확장

Flutter는 데스크톱 애플리케이션 개발을 지원하기 위해 지속해서 업데이트하고 있다. CBS도 레인보우 Windows 버전을 올해 7월경 배포할 예정이다. Mac, Web 용 애플리케이션도 추가로 개발하여 배포를 고려하고 있다.

Android Hybrid 기능

과거의 안드로이드 CBS 레인보우는 모바일 기기의 FM 수신기로 수신된 라디오를 청취할 수 있었다. 당시 국내 스마트폰 제조사를 통해 JAVA SDK를 전달받아 개발을 진행했다. 하지만 새롭게 배포된 CBS 레인보우는 안정적인 배포에 초점을 뒀다. 당 기능은 개발 초기 단계에서 제외됐다. 현재 배포된 레인보우의 안정화 이후 하이브리드 기능도 추가되면 좋을 것 같다.

안정화, 유지보수 그리고 고도화

애플리케이션 배포 뒤 안정화를 위한 노력은 끝이 없다. CBS 레인보우는 1:1 문의를 통해 반복적으로 들어오는 오류 사항을 처리 중이다. 어느 정도 오류가 해결된 뒤 새로운 기능들을 추가하고 데이터 분석을 통해 더 나은 서비스를 위한 방법을 연구할 예정이다.

마치며

Flutter는 계속해서 발전하고 있고, 그만큼 쓰임도 많아지고 있다. 최근 'BMW My' 모바일 애플리케이션도 Flutter로 개발됐다. 연료 충전량, 도어락 제어, 경적, 환기 제어, 위치 등의 다양한 기능을 제공한다. Flutter가 데스크톱 계의 Electron처럼 더 많은 메이저 업계에서 사용되길 희망한다. 다양한 산업 분야에서 Flutter를 채택하고 많은 개발자가 풍성하고 유용한 플러그인들을 만들 날을 기대해본다. 📺