

# M.ADI 성과 공유 시나리오

김인규 ( 리눅스AV1셀 )

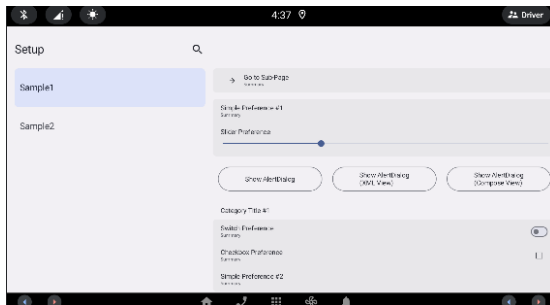
Exported on 09/12/2024

## Table of Contents

1	< M.ADI Phase #1> .....	3
1.1	결론: 개발자 입장에서 각 앱별 custom한 기능 개발에 집중할 수 있도록 개발 infra 제공 및 개발 산출물에 대한 배포 검증에 대한 DevOps 기반 자동화 .....	3
2	< M.ADI Phase #2> .....	4
2.1	결론: 개발자 입장에서 각 앱별 custom 한 기능 개발을 "최소화"하고 M.GPT 기반 Code Review 기능 도입으로 Code의 품질 강화 .....	4

## 1 < M.ADI Phase #1>

1. Android Studio IDE 설치
2. Android Studio IDE 상에서 M.ADI Project Wizard Plugin 설치
3. Plugin 상에서 GAT, MAT, SAT라는 표준화 Architecture Design 된 Application Template Code가 3개가 설치 가능 하다는 것을 보여줌
4. 이 중에 SAT를 선택 후 설치
5. 설치 시에, Gradle Sync 과정에서 아래의 Plugin들이 자동으로 다운로드 되는 것을 보여줌으로써 빠르고(M.ADI Maven을 통해) 쉽게 환경 설정 가능 하다는 것을 보여줌
  - Basic GUI Component
  - 다국어 자동생성 App Plugin
  - Multi-Platform 공용화를 위한 Publish App Plugin
  - 발 신뢰성을 위한 정적 코드 검증을 위한 Ktlint 및 BuildExt App Plugin
6. 이후 SAT를 빌드 + 실행한 화면을 보여줌
  - 앱 실행 후 좀 만져 본다.



7. 각 App별 Customization을 통해 자신의 App을 개발 함을 표현한다.
8. 개발 시, PR 생성 및 App. CI/CD 기반으로 build 검증 및 정적 코드 검증 자동 수행을 보여준다.

### 1.1 결론: 개발자 입장에서 각 앱별 custom한 기능 개발에 집중할 수 있도록 개발 infra 제공 및 개발 산출물에 대한 배포 검증에 대한 DevOps 기반 자동화

## 2 < M.ADI Phase #2>

1. 여러 앱에서 개발 생산성 강화와 공용화를 위해 사용 할 수 있는 공용 App Framework 라이브러리를 개발해 사용
  - 각 App들은 자신의 Biz Logic 개발에 Focus를 하고,  
최대한 그 이외 공용 부분과 GUI, App 구성요소들을 공용화 라이브러리로 만듦으로써 개발 및 유지 보수 비용을 절감하고 App의 공통된 동작성을 보장
2. 개발 신뢰성 증대를 위한 M.GPT Code Reviewer 개발
  - Code 품질 강화

### 2.1 결론: 개발자 입장에서 각 앱별 custom 한 기능 개발을 "최소화"하고 M.GPT 기반 Code Review 기능 도입으로 Code의 품질 강화