

# 빌드 변형과 플레이버란 무엇인가요?

Build Variants / Flavors

다이스

# 왜 빌드 변형이 필요할까?

- 하나의 코드베이스로 다양한 앱 버전 운영 가능
- 광고 여부, 기능 제한, 테스트 환경 구분 등

QA - 내부 테스트 버전

유료 앱 vs 무료 앱

지역별 기능 제한

# Build Variant = Build Type x Product Flavor

	Free	Paid
Debug	freeDebug	paidDebug
Release	freeRelease	paidRelease


# Build Type: 앱을 어떻게 빌드할 것인가?

Debug: 개발용 (logging, Test, Suffix)

Release: 배포용 (서명, 최적화, 난독화)

외에도 Staging, Internal 등 커스텀 타입도 가능

# Build Type별 App Name 다르게 표시하는 방법

```
buildTypes {  
    release {  
          
        resValue(type: "string", name: "app_name", value: "보따리")  
    }  
  
    debug {  
          
        applicationIdSuffix = ".dev"  
        resValue(type: "string", name: "app_name", value: "보따리 (Dev)")  
    }  
}
```

# Build Type별 App Name 다르게 표시하는 방법



보따리



보따리 (Dev)

# Product Flavors: 앱의 기능이나 내용은?

광고 여부, 유료 기능, 지역 구분

App ID, Version Name, 리소스도 Flavor별 분기 가능

Free : 광고 포함, 제한 기능

Paid : 프리미엄 기능 제공

Kr : 한국 전용 기능 제공

US : 미국 전용 정책 적용

# Flavor 구성 예시

```
productFlavors {  
    create( name: "free") {  
        applicationIdSuffix = ".free"  
        versionNameSuffix = "-free"  
    }  
    create( name: "paid") {  
        applicationIdSuffix = ".paid"  
        versionNameSuffix = "-paid"  
    }  
}
```

freeDebug (default) ▾

freeDebug (default)

freeRelease

paidDebug

paidRelease



# Flavors 사용 예시

Flavor에 따른 코르 흐름 변경

```
if (BuildConfig.FLAVOR == "free") {  
    showAds()  
} else {  
    hideAds()  
}
```

# Flavors 사용 예시

## 리소스 파일 분리

Flavor Directory를 분리해 문자열, 이미지, 스타일 등을 다양화할 수 있음

```
src/  
├── free/  
│   └── res/values/strings.xml    // 광고 포함  
├── paid/  
│   └── res/values/strings.xml    // 광고 제거
```

# 왜 사용해야 할까?

- 코드 중복 없이 다양한 앱 제공
- 유료/무료 기능 제어
- 자동화된 빌드 처리 (buildVariant에 따른 자동 설정)

# 마무리 및 주의점

- Internal, mockApi, beta, qa 등 자유롭게 조합 가능
- 단, 변형이 너무 많아지면 유지보수 어려움
- 팀 상황에 맞는 구조로 최소한의 조합 추천