# 토이 프로젝트

# 리팩토링을 진행하며 배운 점

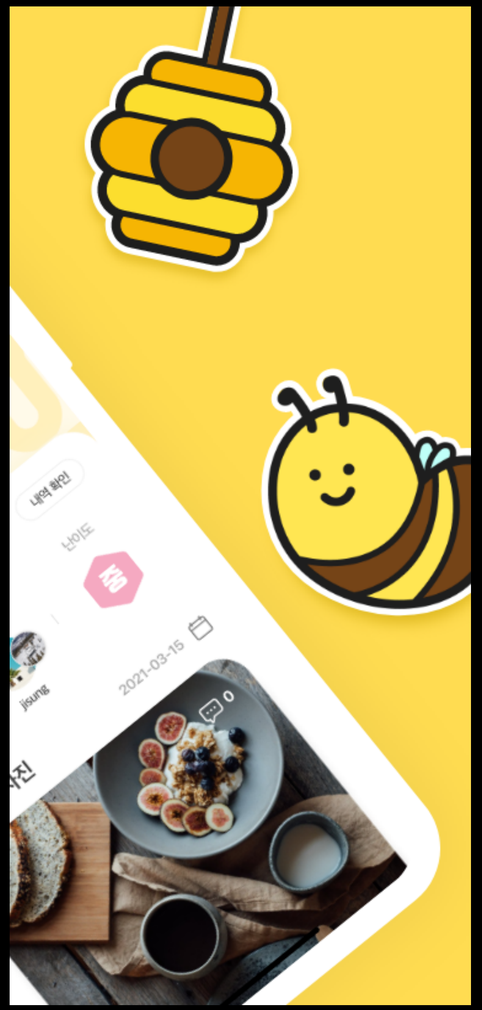
안드로이드 2팀 박규림

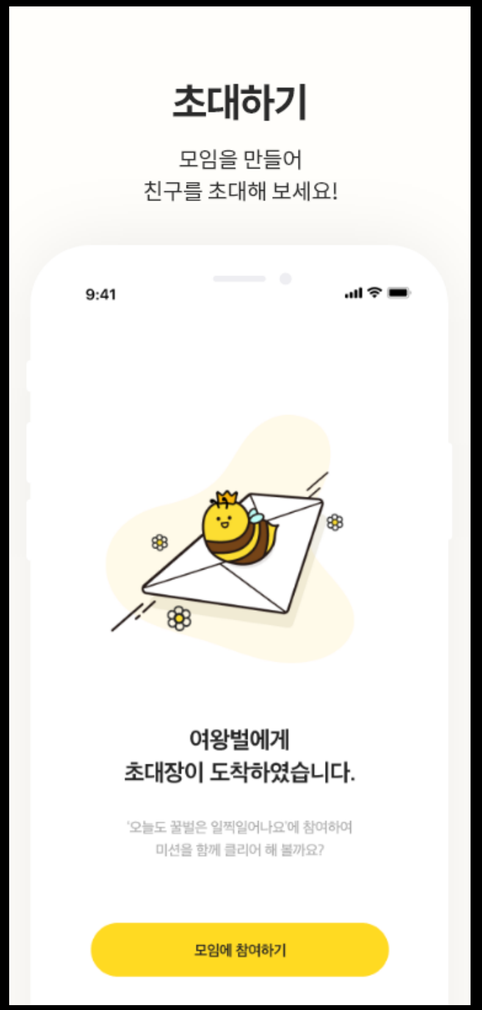
# 목차

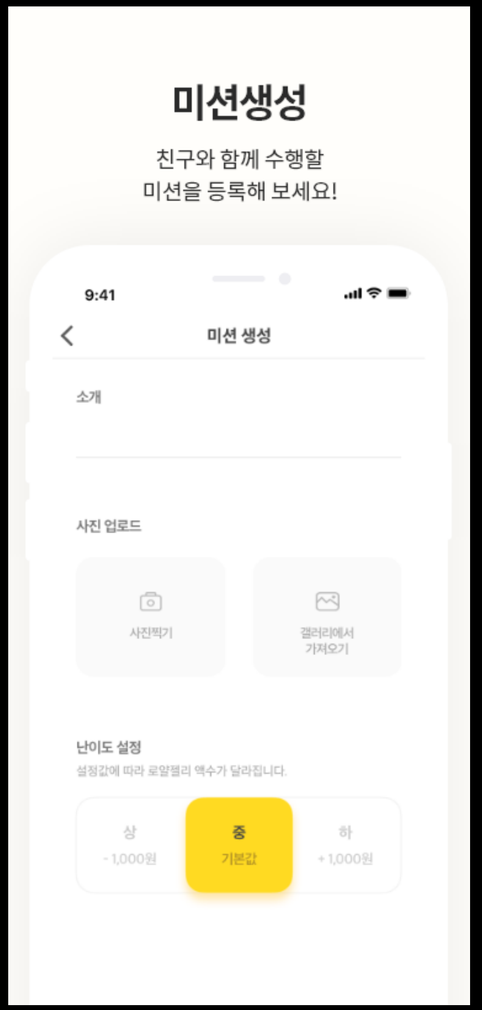
- 토이프로젝트 소개
- 리팩토링 목표
- 리팩토링을 통해 배운 점

# 토이 프로젝트 간단 소개


9:41
오늘도 꿀벌은
일찍 일어나요
누적 로얄젤리
미션 시간
MORNING BEES
MORNING BEES


내역 확인
난이도
중
2021-03-15


초대하기
모임을 만들어
친구를 초대해 보세요!
9:41
여왕벌에게
초대장이 도착하였습니다.
'오늘도 꿀벌은 일찍일어나요'에 참여하여
미션을 함께 클리어 해 볼까요?
모임에 참여하기


미션생성
친구와 함께 수행할
미션을 등록해 보세요!
9:41
미션 생성
소개
사진 업로드
사진찍기
갤러리에서
가져오기
난이도 설정
설정값에 따라 로얄젤리 액수가 달라집니다.
상
- 1,000원
중
기본값
하
+ 1,000원

# 프로젝트 진행 전, 지식 상태

비동기 서버 통신...?

Glide...?

JSON...?

제플린...?

```
584             private const val RC_SIGN_IN = 9001
585             private const val RC_GET_TOKEN = 9002
586             private const val REQUEST_LOGOUT = 1004
587         }
588     }
```

```
892             private const val RELOAD = 120
893             private const val FINISH = 121
894             private const val TAG = "MainActivity"
895         }
896     }
```

# 이해하기 어려움

# 비동기 서버 요청 & 응답 코드

```
override fun requestSignInApi(signInRequest: SignInRequest) {
    service.signIn(signInRequest).enqueue(object : Callback<SignInResponse> {
        override fun onFailure(call: Call<SignInResponse>, t: Throwable) {
            Dlog().d(t.toString())
        }

        override fun onResponse(
            call: Call<SignInResponse>,
            response: Response<SignInResponse>
        ) {
            val i = response.code()
            Dlog().d(response.code().toString())
            when (i) {
                200 -> {
                    val signInResponse: SignInResponse = response.body()!!
                    when (signInResponse.type) {
                        1 -> {
                            GlobalApp.prefs.accessToken = signInResponse.accessToken
                            GlobalApp.prefs.refreshToken = signInResponse.refreshToken

                            requestMeApi()
                        }

                        0 -> {
                            gotoSignUp(signInRequest)
                        }
                    }
                }

                400 -> {
                    showToast { "다시 로그인해주세요." }
                    signOut()
                }

                500 -> {
                    val jsonObject = JSONObject(response.errorBody()!!.string())
                    val message = jsonObject.getString("message")
                    showToast { message }
                }
            }
        }
    })
}
```

하드 코딩의 흔적...

```
override fun requestSignInApi(signInRequest: SignInRequest) {
    service.signIn(signInRequest).enqueue(object : Callback<SignInResponse> {
        override fun onFailure(call: Call<SignInResponse>, t: Throwable) {
            Dlog().d(t.toString())
        }

        override fun onResponse(
            call: Call<SignInResponse>,
            response: Response<SignInResponse>
        ) {
            val i = response.code()
            Dlog().d(response.code().toString())
            when (i) {
                200 -> {
                    val signInResponse: SignInResponse = response.body()!!
                    when (signInResponse.type) {
                        1 -> {
                            GlobalApp.prefs.accessToken = signInResponse.accessToken
                            GlobalApp.prefs.refreshToken = signInResponse.refreshToken

                            requestMeApi()
                        }

                        0 -> {
                            gotoSignUp(signInRequest)
                        }
                    }
                }

                400 -> {
                    showToast { "다시 로그인해주세요." }
                    signOut()
                }

                500 -> {
                    val jsonObject = JSONObject(response.errorBody()!!.string())
                    val message = jsonObject.getString("message")
                    showToast { message }
                }
            }
        }
    })
}
```

가독성 점수 -100점

# 리팩토링 계획 💭

MVVM 패턴 + Data Binding

Repository 패턴을 사용

Manager 사용

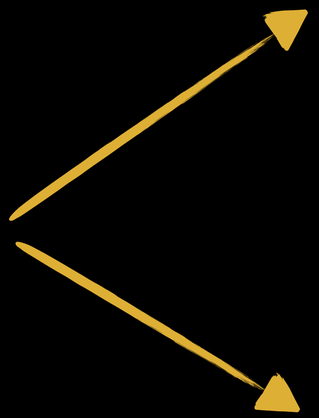
Kotlin 확장 함수 사용


# MVVM 패턴 + Data Binding

# MVC 패턴

Activity.kt

```
584          private const val RC_SIGN_IN = 9001
585          private const val RC_GET_TOKEN = 9002
586          private const val REQUEST_LOGOUT = 1004
587      }
588  }
```



# MVVM 패턴

Activity.kt

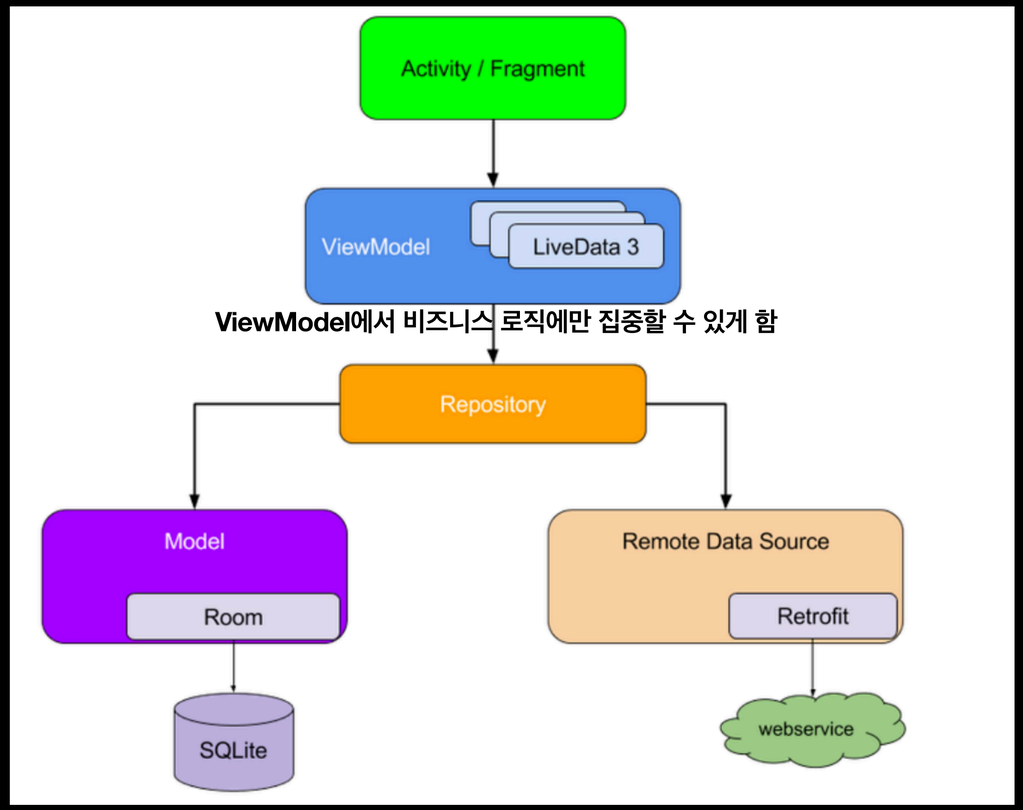
```
139      companion object {
140          private const val TAG = "LoginActivity"
141          private const val RC_SIGN_IN = 9001
142      }
143  }
```

ViewModel.kt

```
227          const val JOIN_BEE_API = 4
228          const val ME_API = 5
229      }
230  }
```

# Repository 패턴을 사용해보자 🖇️

# Repository 패턴?




```
open class BaseRepository {

    suspend fun <T : Any> safeApiCall(call: suspend () -> Response<T>, error: String): Output<T>? {

        try {
            val response : Response<T> = call.invoke()
            return Output.Success(response.body()!!)
        } catch (throwable: Throwable) {
            return when (throwable) {
                is IOException -> Output.NetworkError

                is HttpException -> {
                    val code : Int = throwable.code()
                    val errorResponse : ErrorResponse? = convertErrorBody(throwable)
                    Output.Error(code, errorResponse)
                }

                else -> {
                    Output.Error( code: null,  error: null)
                }
            }
        }
    }
```

# ViewModel

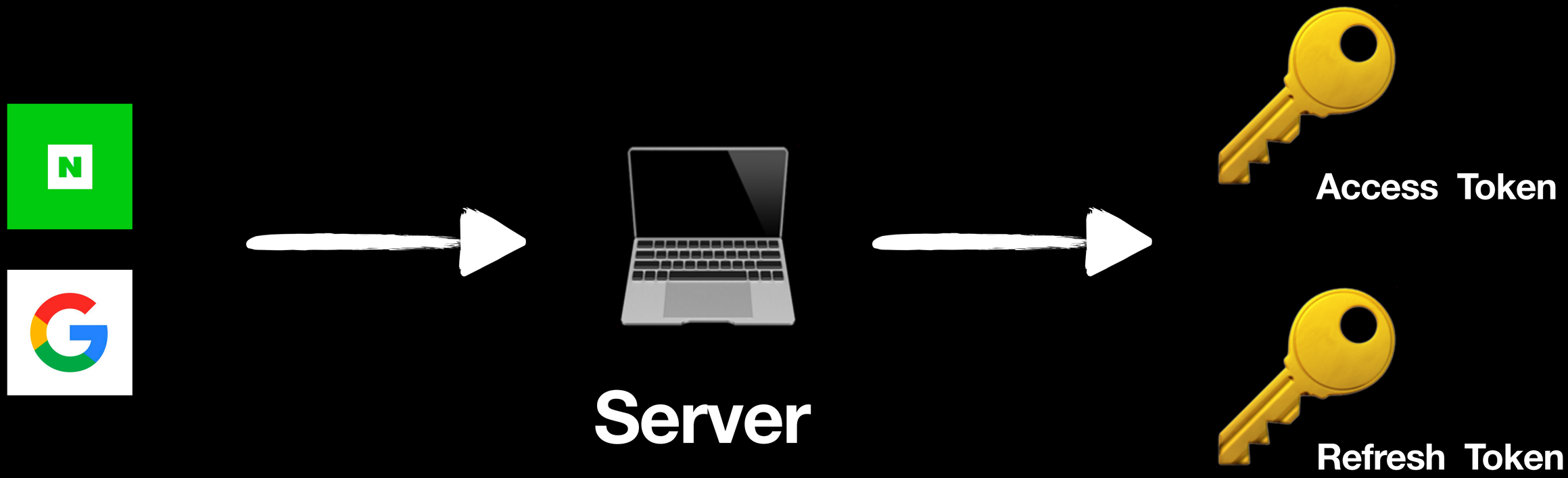
```
private fun requestJoinBeeApi() {

    val joinBeeRequest = JoinBeeRequest(
        GlobalApp.prefsBeeInfo.beeId,
        GlobalApp.prefs.userId,
        GlobalApp.prefsBeeInfo.beeTitle
    )

    scope.launch { this: CoroutineScope
        when (val joinBeeResponse : Output<Void>? = mRepository.requestJoinBeeApi(joinBeeRequest)) {
            is Output.Success -> {
                _mainActivityChangeEvent.value = Unit
            }

            is Output.Error -> showGenericError(joinBeeResponse.error, JOIN_BEE_API)
            is Output.NetworkError -> showNetworkError()
        }
    }
}
```

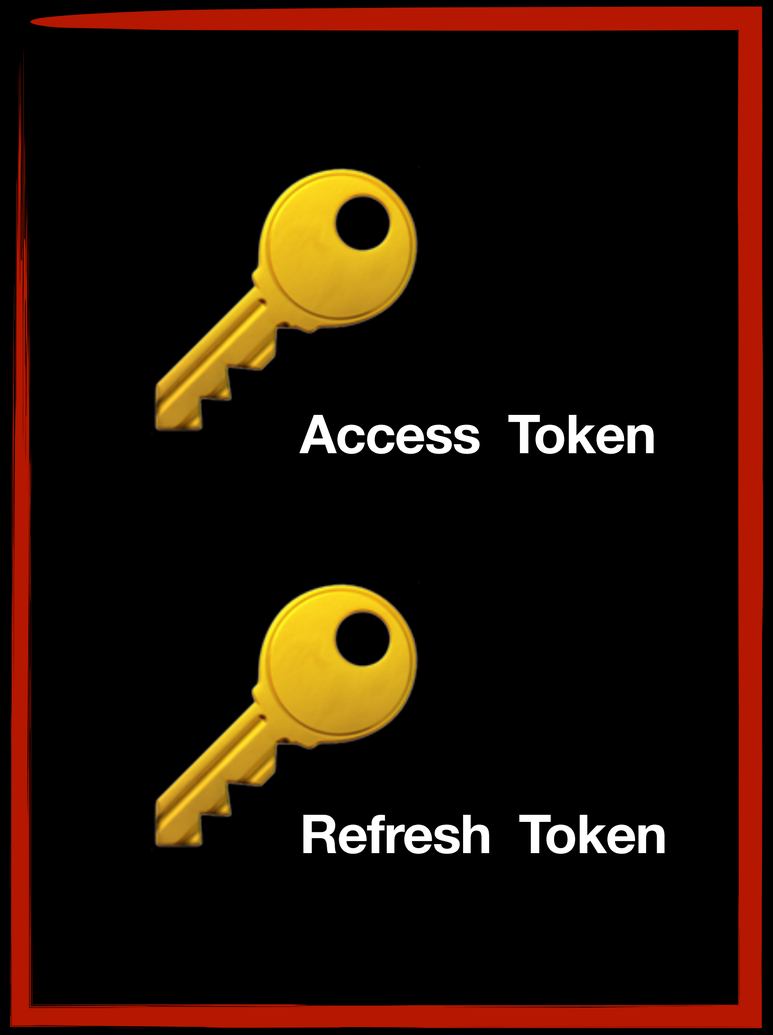
# API 요청 담당 Repository

```
suspend fun requestJoinBeeApi(joinBeeRequest: JoinBeeRequest) : Output<Void>? {
    return safeApiCall(
        call = {service.joinBee(accessToken, joinBeeRequest)},
        error = "Join Bee API Error fetching"
    )
}
```

# Manager 사용 

# SNS Login Flow






Intent

# 당연하게 따라오는 문제점

가입한 유저

가입하지 않은 유저


무조건

다시 로그인 !

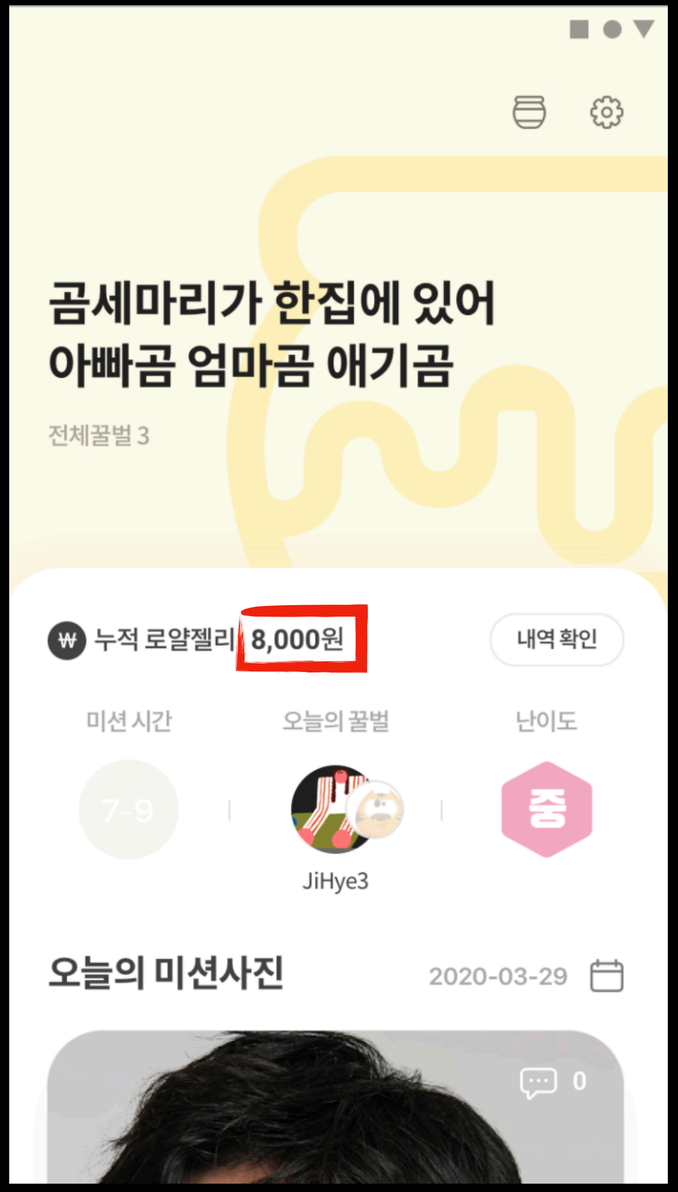
# 로그인 Manager

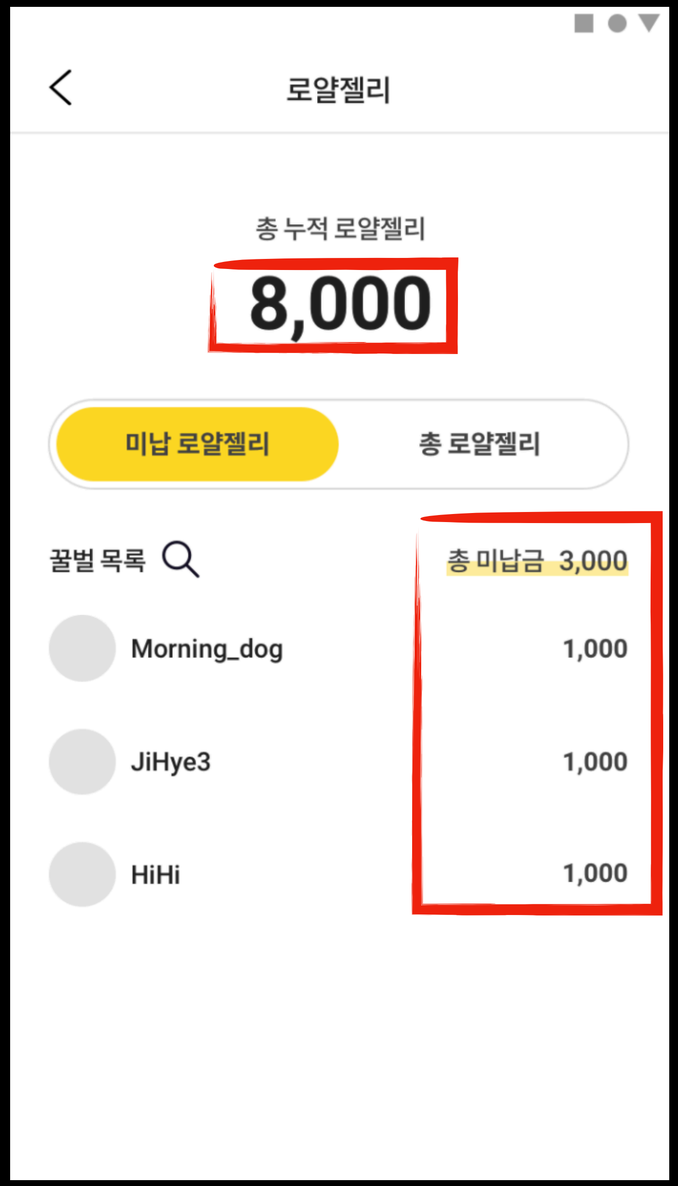
```
object NaverLoginManager : OAuthLoginHandler() {

    private val appContext : Context = AppResources.getContext()

    private val naverLoginInstance : OAuthLogin! = OAuthLogin.getInstance().apply { this: OAuthLogin!
        init (
            appContext,
            AppResources.getStringResId(
            AppResources.getStringResId(
            AppResources.getStringResId(
        )
    }

    override fun run(success: Boolean) {
        if (success) {
            val accessToken : String! = naverLoginInstance.getAccessToken(
                appContext
            )
            GlobalApp.prefs.accessToken = accessToken
            GlobalApp.prefs.provider = "naver"

        } else {
            val errorCode : String! = naverLoginInstance.getLastErrorCode(
                appContext
            ).code
            val errorDesc : String! = naverLoginInstance.getLastErrorDesc(
                appContext
            )
            Dlog().d("errorCode:$errorCode, errorDesc:$errorDesc")
        }
    }
```

SharedPreferences

```
class MorningBessRepository : BaseRepository() {
    private val service : MorningBeesService = NetworkModule.morningBeesService

    private var accessToken : String = GlobalApp.prefs.accessToken
    private var refreshToken : String = GlobalApp.prefs.refreshToken

    suspend fun requestMeApi(): Output<MeResponse>? {
        return safeApiCall(
            call = {service.me(accessToken)},
            error = "Me API Error Fetching"
        )
    }
```

# Kotlin 확장 함수 사용



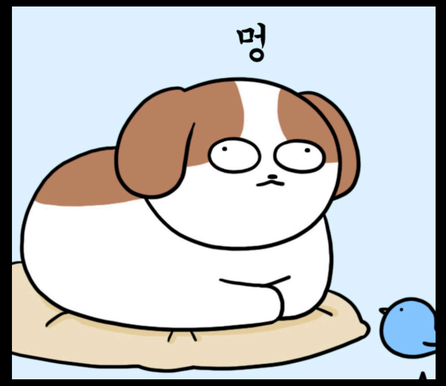

## Kotlin 확장 함수

```
fun Int.getPriceAnnotation(): String {
    return DecimalFormat( pattern: "###,###").format( obj: this)
}
```

## 실제 사용 예시

```
maxPaymentSeekBar.text = partPaymentList.penalty.getPriceAnnotation()

totalPenalty = beeInfoResponse.totalPenalty.getPriceAnnotation()
```

# 배운 점




남들이 이해하기 힘든 코드는

작성자인 나도 이해하기 힘들다 🥲

코드의 **분업화**를 잘하자!

**중복 코드**를 최대한 없애자!

**가독성**을 높여 실수를
**줄일 수 있는 코드**를 작성하자!

# 감사합니다