

ANDROID KOTLIN - INTERVIEW SECRETS

```
object AppConstant {
    const val baseUrl="https://jsonplaceholder.typicode.com/"
}

interface RetrofitApi {

    @GET("posts")
    suspend fun getPost() : List<Post>
}

object RetrofitBuilder {

    private val retrofit:Retrofit by lazy {
        Retrofit.Builder()
            .baseUrl(AppConstant.baseUrl)
            .addConverterFactory(GsonConverterFactory.create())
            .build()
    }
    val retrofitApi:RetrofitApi by lazy {
        retrofit.create(RetrofitApi::class.java)
    }
}

class PostRepository {

    fun getPost(): Flow<List<Post>> = flow {
        val postList = RetrofitBuilder.retrofitApi.getPost()
        emit(postList)
    }.flowOn(Dispatchers.IO)
}

class PostViewModel(private val postRepository: PostRepository) : ViewModel() {

    val postData: MutableLiveData<List<Post>> = MutableLiveData()

    // Way: 1
    fun getPost() {
        viewModelScope.launch {
            postRepository.getPost()
                .catch { e ->
                    Log.d("main", "getPost: ${e.message}")
                }
                .collect { postData1 ->
                    postData.value = postData1
                }
        }
    }
}
```

ANDROID KOTLIN - INTERVIEW SECRETS

```
// Way: 2
val postData1: LiveData<List<Post>> = liveData {
    postRepository.getPost()
        .catch { }
        .collect { postData ->
            emit(postData)
        }
}

// Way: 2
val postData2: LiveData<List<Post>> = postRepository.getPost()
    .catch { }
    .asLiveData()
}

class PostViewModelFactory(private val postRepository: PostRepository)
    : ViewModelProvider.Factory {
    override fun <T : ViewModel?> create(modelClass: Class<T>): T {
        return PostViewModel(postRepository) as T
    }
}

class MainActivity : AppCompatActivity() {
    private val TAG = "main"
    private lateinit var recyclerView: RecyclerView
    private lateinit var postAdapter: PostAdapter
    private lateinit var postViewModel: PostViewModel

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        initView()

        val postViewModelFactory = PostViewModelFactory(PostRepository())
        postViewModel = ViewModelProvider(this, postViewModelFactory)
            [PostViewModel::class.java]
        postViewModel.getPost()

        postViewModel.postData.observe(this, Observer {
            Log.d(TAG, "onCreate: ${it[0].body}")
            postAdapter.setPostData(it as ArrayList<Post>)
            progressBar.visibility = View.GONE
            recyclerView.visibility = View.VISIBLE
        })
    }
}
```

ANDROID KOTLIN - INTERVIEW SECRETS

```
private fun initUi() {  
    recyclerView = findViewById(R.id.recyclerView)  
    postAdapter = PostAdapter(this, ArrayList())  
    recyclerView.apply {  
        setHasFixedSize(true)  
        layoutManager = LinearLayoutManager(this@MainActivity)  
        adapter = postAdapter  
    }  
}
```

-

ANDROID KOTLIN - INTERVIEW SECRETS

ANDROID KOTLIN - INTERVIEW SECRETS