Jetpack Hilt 训练营-讲义

Hilt 的使用

基本使用

用 `@Inject` 对变量进行注解，表示「这个变量要用依赖注入的方式来加载」：

```
1 @Inject lateinit var user: User
```

用 `@Inject` 对依赖的类的构造函数进行注解，表示「当要提供这个类的对象的时候，调用这个函数来创建对象」：

```
1 data class User constructor(var id: Int, var name: String, var moo
  d: String) {
2   @Inject constructor() : this(1, "扔物线", "毫无波澜")
3 }
```

然后，要记得用 `@AndroidEntryPoint` 指出要被注入依赖的组件类：

```
1 @AndroidEntryPoint
2 class MainActivity : AppCompatActivity() {
3   ...
4 }
```

以及，用 `HiltAndroidApp` 来宣布你的 Application 要使用 Hilt：

```
1 @HiltAndroidApp
2 class HiltApp : Application()
```

作用域（范围内共享数据）

额外的作用域注解：

```
1 @ActivityScoped
```

```
data class User constructor(var id: Int, var name: String, var mood: String) {
  @Inject constructor() : this(1, "扔物线", "毫无波澜")
}
```

更多提供依赖的方式

类型绑定

`@Binds`：

```
@Module
@InstallIn(ActivityComponent::class)
abstract class HiltModule {
  @Binds abstract fun bindAny(user: User): Any
}
```

直接提供代码

`@Provides`：

```
@Module
@InstallIn(ActivityComponent::class)
object HiltModule2 {
  @Provides
  fun provideUser(): User {
    val user = User()
    user.name = "丢物线"
    user.mood = "难过"
    return user
  }
}
```

Hilt 有什么用

什么是依赖注入

- 依赖：一个类里面有变量，这些变量就是这个类的依赖，或者说，这个类依赖这些变量。
- 依赖注入：内部依赖的值不由你类自己来提供，而是由外部来交给这个类。

依赖注入有什么用

- 依赖注入的作用：自动加载。
- 自动加载的作用：数据共享。
- 不共享的数据，需要依赖注入吗？
    - 多数不需要，有一种例外：多处被使用，并且加载方式一致的。作用是「省事」。

Hilt 的作用

更方便地使用依赖注入。

「那我如果不需要依赖注入呢？」——听完课了怎么可能还这么说，再去听一遍课吧。

Dagger

Dagger 为什么不好用

1. 因为它完备和灵活的功能，导致了一些上手成本。——可以被 Hilt 解决。
2. 依赖关系无法追踪。——可以被新版 Android Studio 解决。
3. 很多人不懂依赖注入。——需要自己解决（听完课的可以自动解决）。

Hilt 会有 Dagger 的问题吗？

不会的。但是有前提：

- 你要用新版 Android Studio
- 你要听我的，不要乱用，先想明白再用

Dagger、Koin 和 Hilt 对比

- 易用性：Hilt = Koin > Dagger
- 性能：Dagger = Hilt > Koin
- 依赖图的安全性：Dagger = Hilt > Koin
- 依赖关系的追踪：Dagger = Hilt > Koin
- 编译时性能： Koin > Dagger = Hilt
- 运行时性能：Dagger = Hilt > Koin

依赖注入和视图绑定

依赖注入和视图绑定的区别：

- 依赖注入：我要，怎么来的我不管；
- 视图绑定：我要，而且我很明确需要这么这么加载，但我希望你把这件事做了。