

Dagger 2 完全解析（五），Kotlin 中使用 Dagger 2

最帅的 JohnnyShieh 2017-08-22 20:05



阅读本文大概需要 5 分钟。

Google I/O 2017 大会上宣布 Kotlin 成为 Android 开发一级语言，kotlin 语言作为 Google 的亲儿子，前途一片光明，而且 Kotlin 加入了很多现代语言特性。最近把我之前写的一个开源项目重构为 Kotlin 版本 — Kotlin 版本的 Gank app，以此来学习 kotlin 语言。其中在 Kotlin 语言中使用 Dagger 2 的写法与 Java 中有些变化，所以写下这篇文章记录下如何在 Kotlin 语言中使用 Dagger 2。

本文中的 Kotlin 版本为 1.1.2-5

1. Dagger 2 依赖

引入方式与之前有所区别，Dagger 2 根据注解在编译时生成静态代码，所以需要注解处理器，而 Kotlin 中注解处理工具为 Kotlin Annotation processing tool (kapt)。

build.gradle

```
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply plugin: 'kotlin-kapt' // 使用 kapt 注解处理工具
```

```
...

dependencies {
    compile "com.google.dagger:dagger:$dagger_version"
    kapt "com.google.dagger:dagger-compiler:$dagger_version" // kapt
}
```

2. Inject 注解

- 标记目标类构造函数，以此为 Component 提供依赖实例

```
// 标注 constructor 构造方法
class Car @Inject constructor() { .. }

class Car {
    @Inject constructor(name: String)
}
```

- 标注成员属性，确定需要注入的依赖

```
// 标注 setter
class Man {
    lateinit var car: Car
    @Inject set
}
```

直接标注属性 `@Inject lateinit var car: Car`，编译时会报错

3. Module 和 Component

Module 和 Component 的写法和 Java 一样，只是 Kotlin 语法不一样

```
@Module
class CarModule {
    @Provides
    fun provideCar() = Car()
}

@Component(modules = arrayOf(CarModule::class))
interface ManComponent {
    fun inject(man: Man)
}
```

4. Scope 作用域

自定义的作用域的语法也变了，下面是常见的两个自定义 Scope：

```
// 主要是注解的声明语法不同
@Scope
@Retention(AnnotationRetention.RUNTIME)
@MustBeDocumented
annotation class ActivityScope

@Scope
@Retention(AnnotationRetention.RUNTIME)
@MustBeDocumented
annotation class FragmentScope
```

5. 总结

kapt 编译时生成的代码和之前一样，Component 的使用方式没有改变，需要注意就是 Inject 标注成员属性时，需要标注 setter。

如果想看更具体的代码，可以前往我开源的 Gank - kotlin 项目：
<https://github.com/JohnnyShieh/Gank/tree/kotlin>

到这里，Dagger 2 完全解析系列就结束了，有什么疑问欢迎在下面留言。

END

一个白日做梦的工程师！



不只有技术，还有咖啡和彩蛋！

个人博客：<http://johnnyshieh.me>

[阅读原文](#)