

Android - Kodein - Kotlin Workshop

Abhängigkeiten im Application deklarieren

```
class MyApp : Application(), KodeinAware {  
    override val kodein by Kodein.lazy { /* bindings */ }  
}
```

Lazy Kodein benutzen

```
class MainActivity : Activity() {  
  
    val kodein = LazyKodein(appKodein)  
    val diceProvider: () -> Dice by kodein.provider()  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        val random: Random = kodein().instance()  
    }  
}
```

Android App mit Kodein aufsetzen

Beispiel an einem Fragment:

```
class MyFragment : CustomFragment(), FragmentInjector {  
    override val injector: KodeinInjector = KodeinInjector()  
  
    private val logTag: String by instance("log-tag")  
    private val app: Application by injector.instance()  
  
    override fun provideOverridingModule() = Kodein.Module {  
        bind<MyFragment>() with instance(this@MyFragment)  
        bind<String>("log-tag", overrides = true) with  
            instance("MyFragment") /*Wird überschrieben von  
            MyApplication*/  
    }  
  
    override fun onCreate(savedInstanceState: Bundle) {  
        super.onCreate(savedInstanceState)  
        initializeInjector()  
        Log.i(logTag, "Calling onCreate from MainActivity in  
            ${app.applicationInfo.className}")  
    }  
  
    override fun onDestroy() { destroyInjector()  
        super.onDestroy() }  
}
```

Activity Scope

Binding:

```
val kodein = Kodein {  
    bind<Logger>() with scopedSingleton(androidActivityScope) {  
        LogManager.getNamedLogger(it.localClassName) }  
}
```

Abruf

```
val logger: Logger = kodein.with(getActivity()).instance()
```

Auto activity scope

Binding:

```
val kodein = Kodein {  
    bind<Logger>() with autoScopedSingleton(androidActivityScope) {  
        LogManager.getNamedLogger(it.localClassName) }  
}
```

Abruf:

```
val logger: Logger = kodein.instance()
```

Lifecycle Manager Registrieren: (ist zwingend nötig, damit autoScoped für androidActivityScope funktioniert!)

```
class MyApplication : Application { override fun onCreate() {  
    registerActivityLifecycleCallbacks(  
        androidActivityScope.lifecycleManager ) } }
```