Похоже это используется в SunFlowers

Hilt и AssistedInject работают вместе в Hilt v2.28-alpha times - версия ViewModel

// AssistedInject and Hilt working together in v2.28-alpha times

// Example of a ViewModel using AssistedInject injected in a Fragment by Hilt

// As AssistedInject isn't part of Dagger yet, we cannot use in

// conjuction with @ViewModelInject

// Пример ViewModel, используя AssistedInject встроенные в фрагменте, используя здоровье

// Поскольку AssistedInject еще не является частью Dagger, мы не можем использовать его в

// соединении с @ViewModelInject

data class MyInitParams(private val neededId: String)

class MyViewModel @AssistedInject constructor(

... // Другие зависимости, внедренные Hilt

... // Other dependencies injected by Hilt

// com.squareup.inject.assisted.Assisted annotation

@Assisted private val initParams: MyInitParams

) : ViewModel() {

... // ViewModel logic

@AssistedInject.Factory

interface AssistedFactory {

fun create(initParams: MyInitParams): MyViewModel

}

companion object {

fun provideFactory(

assistedFactory: AssistedFactory,

initParams: MyInitParams

): ViewModelProvider.Factory = object : ViewModelProvider.Factory {

override fun <T : ViewModel?> create(modelClass: Class<T>): T {

return assistedFactory.create(initParams) as T

}

}

}

}

// This code uses "com.squareup.inject:assisted-inject-annotations-dagger2:0.5.2"

// AssistedInject puts all assisted bindings in the same module.

// We need to make a decision about where to install it.

// In this case, as we only need it in fragments, we install it there.

// В этом коде используется "com.squareup.inject: assisted-inject-annotations-dagger2: 0.5.2"

// AssistedInject помещает все вспомогательные привязки в один и тот же модуль.

// Нам нужно решить, где его установить.

// В этом случае, поскольку он нам нужен только фрагментарно, мы устанавливаем его туда.

@InstallIn(FragmentComponent::class)

@AssistedModule

@Module(includes = [AssistedInject\_AssistedInjectModule::class])

// Needed until AssistedInject is incorporated into Dagger

interface AssistedInjectModule {}

@AndroidEntryPoint // Fragment injected by Hilt

class MyFragment : Fragment() {

private val args: MyFragmentArgs by navArgs()

// Inject the factory generated by AssistedInject

@Inject lateinit var myViewModelAssistedFactory: MyViewModel.AssistedFactory

// Initialize the ViewModel using ViewModelProvider.Factory

private val myViewModel: MyViewModel by viewModels {

MyViewModel.provideFactory(

myViewModelAssistedFactory, MyInitParams(args.neededId)

)

}

... // Fragment logic

}

// To disable checking for the InstallIn annotation in some modules,

// you need to enable the compiler option in your build.gradle file like this:

// Чтобы отключить проверку установочной аннотации в некоторых модулях,

// вам нужно включить опцию компилятора в файле build.gradle следующим образом:

android {

defaultConfig {

javaCompileOptions {

annotationProcessorOptions {

arguments["dagger.hilt.disableModulesHaveInstallInCheck"] = "true"

}

}

}

}

[**Vincent-Paing**](https://gist.github.com/vincent-paing)**прокомментировал [on 20 Jun](https://gist.github.com/manuelvicnt/437668cda3a891d347e134b1de29aee1" \l "gistcomment-3348447)**

|  |
| --- |
| Я обнаружил проблему с тем inline class, что если вы используете inline classв качестве @Assistedаннотированного параметра, процессор не обнаруживает этого и выдает ошибку, говоря, что ему нужен конструктор хотя бы с одним @Assistedпараметром |

[](https://gist.github.com/afTrolle)

[**afTrolle**](https://gist.github.com/afTrolle)**прокомментировал [on 6 Aug](https://gist.github.com/manuelvicnt/437668cda3a891d347e134b1de29aee1" \l "gistcomment-3406102)**

|  |
| --- |
| Спасибо за пример, отлично поработали! |

[](https://gist.github.com/isfaaghyth)

[**исфаагит**](https://gist.github.com/isfaaghyth)**прокомментировал [on 10 Aug](https://gist.github.com/manuelvicnt/437668cda3a891d347e134b1de29aee1" \l "gistcomment-3411669)**

|  |
| --- |
| спасибо, работает как шарм! |

[](https://gist.github.com/diousk)

[**Diousk**](https://gist.github.com/diousk)**прокомментировал [on 10 Oct](https://gist.github.com/manuelvicnt/437668cda3a891d347e134b1de29aee1" \l "gistcomment-3484215)**

|  |
| --- |
| Начиная с AssistedInject 0.6.0, сгенерированный класс AssistedInject\_AssistedInjectModuleтакже будет аннотирован @InstallIn. Нам все еще нужно, @Module(includes = [AssistedInject\_AssistedInjectModule::class])а не просто @Module? |

[](https://gist.github.com/Leeonardoo)

[**Leeonardoo**](https://gist.github.com/Leeonardoo)**прокомментировал [on 15 Oct](https://gist.github.com/manuelvicnt/437668cda3a891d347e134b1de29aee1" \l "gistcomment-3490979)**

|  |
| --- |
| Начиная с AssistedInject 0.6.0, сгенерированный класс AssistedInject\_AssistedInjectModuleтакже будет аннотирован @InstallIn. Нам все еще нужно, @Module(includes = [AssistedInject\_AssistedInjectModule::class])а не просто @Module?  Похоже, теперь вы можете просто использовать [**@module**](https://github.com/module) . Я все еще тестирую это |

[](https://gist.github.com/anilkumar416)

[**анилкумар416**](https://gist.github.com/anilkumar416)**прокомментировал**[**14 дней назад**](https://gist.github.com/manuelvicnt/437668cda3a891d347e134b1de29aee1#gistcomment-3542241)

|  |
| --- |
| Спасибо мне помогло! |