

ViewModel APIs cheat sheet

v1.0 by @AndroidDev

This cheat sheet lists the ViewModel APIs in Jetpack. It includes the artifact where they can be found, the scope of the returned ViewModel instance, and an example of how to use them.

You can scope ViewModel instances to a `ViewModelStoreOwner`.

The owner can be an Activity, Fragment, Navigation graph, a destination of a Navigation graph, or your own custom owner.

For more use cases and to learn about ViewModel factories: goo.gl/architecture-viewmodel

Get an instance in Jetpack Compose

Compose - `lifecycle.lifecycle-viewmodel-compose`

= `viewModel()` `closest ViewModelStoreOwner`

```
@Composable
fun MyScreen(vm: ViewModel = viewModel()) { ... }
```

= `viewModel(vmStoreOwner)` `any ViewModelStoreOwner`

```
// Example of getting a ParentViewModel instance scoped to the Navigation graph
composable("myScreen") { backStackEntry ->
    val parentEntry = remember(backStackEntry) {
        navController.getBackStackEntry("parentNavigationRoute")
    }
    val parentViewModel = viewModel<ParentViewModel>(parentEntry)
    ...
}
```

Compose + Navigation + Hilt - `hilt.hilt-navigation-compose`

When using Hilt and Navigation together, use this API instead of `viewModel()`

= `hiltViewModel()` `closest ViewModelStoreOwner`

```
@Composable
fun MyScreen(vm: ViewModel = hiltViewModel()) { ... }
```

Core ViewModel API

ViewModel - `lifecycle:lifecycle-viewmodel-ktx`

ViewModelProvider `any ViewModelStoreOwner`

```
val vm = ViewModelProvider(anyViewModelStoreOwner)[MyViewModel::class.java]
```

The APIs and artifacts listed in this page follow this template

Library - artifact

ViewModel API

Scope of the returned ViewModel instance

Example code

Get an instance in an Activity

Activity - `activity.activity-ktx`

by `viewModels()`

Activity (ViewModelStoreOwner)

```
val vm: MyViewModel by viewModels()
```

Get an instance in a Fragment

Fragment - `fragment.fragment-ktx`

by `viewModels()`

Fragment (ViewModelStoreOwner)

```
val vm: MyViewModel by viewModels()
```

by `viewModels(ownerProducer)`

any ViewModelStoreOwner

```
// Example of getting a SharedViewModel instance scoped to the parent Fragment
val vm: SharedViewModel by viewModels(ownerProducer = { requireParentFragment() })
```

by `activityViewModels()`

Activity (ViewModelStoreOwner)

```
val vm: SharedViewModel by activityViewModels()
```

Fragment + Navigation - `navigation.navigation-fragment`

by `navGraphViewModels(graphId)`

NavGraph (ViewModelStoreOwner)

```
val vm: SharedViewModel by navGraphViewModels(R.id.nav_graph)
```

Fragment + Navigation + Hilt - `hilt.hilt-navigation-fragment`

When using Hilt and Navigation together, use this API instead of `navGraphViewModels()`. To get an instance not scoped to a NavGraph, you can keep using `viewModels()` or `activityViewModels()`.

by `hiltNavGraphViewModels(graphId)`

NavGraph (ViewModelStoreOwner)

```
val vm: SharedViewModel by hiltNavGraphViewModels(R.id.nav_graph)
```