# ViewModelProvider

| **Java**

```
public class ViewModelProvider
extends Object
```

java.lang.Object
  ↳ androidx.lifecycle.ViewModelProvider

An utility class that provides `ViewModels` for a scope.

Default `ViewModelProvider` for an `Activity` or a `Fragment` can be obtained by passing it to [ViewModelProvider(ViewModelStoreOwner)](https://developer.android.com/reference/androidx/lifecycle/ViewModelProvider?authuser=1#ViewModelProvider(androidx.lifecycle.ViewModelStoreOwner))
.

## Summary

### Nested classes

| | |
|---|---|
| class | [ViewModelProvider.AndroidViewModelFactory](https://developer.android.com/reference/androidx/lifecycle/ViewMo... authuser=1) |
| | [ViewModelProvider.Factory](https://developer.android.com/reference/androidx/lifecycle/ViewMo... may create [AndroidViewModel](https://developer.android.com/reference/androidx/lifecycle/Android\ [ViewModel](https://developer.android.com/reference/androidx/lifecy... have an empty constructor. |
| interface | [ViewModelProvider.Factory](https://developer.android.com/reference/androidx/lifecycle/ViewMo... Implementations of `Factory` interface are responsible to instantiate V |
| class | [ViewModelProvider.NewInstanceFactory](https://developer.android.com/reference/androidx/lifecycle/ViewMo... authuser=1) |

Simple factory, which calls empty constructor on the give class.

## Public constructors

### ViewModelProvider
(https://developer.android.com/reference/androidx/lifecycle/ViewModelProvider?
authuser=1#ViewModelProvider(androidx.lifecycle.ViewModelStoreOwner))
(**ViewModelStoreOwner** (https://developer.android.com/reference/androidx/lifecycle/ViewModelStore
owner)

Creates `ViewModelProvider`.

### ViewModelProvider
(https://developer.android.com/reference/androidx/lifecycle/ViewModelProvider?
authuser=1#ViewModelProvider(androidx.lifecycle.ViewModelStoreOwner,%20androidx.lifecycle.ViewMo
(**ViewModelStoreOwner** (https://developer.android.com/reference/androidx/lifecycle/ViewModelStore
owner, **ViewModelProvider.Factory**
(https://developer.android.com/reference/androidx/lifecycle/ViewModelProvider.Factory?authuser=1) f

Creates `ViewModelProvider`, which will create `ViewModels` via the given `Factory` and retain them in
`ViewModelStoreOwner`.

### ViewModelProvider
(https://developer.android.com/reference/androidx/lifecycle/ViewModelProvider?
authuser=1#ViewModelProvider(androidx.lifecycle.ViewModelStore,%20androidx.lifecycle.ViewModelPro
(**ViewModelStore** (https://developer.android.com/reference/androidx/lifecycle/ViewModelStore?authu
**ViewModelProvider.Factory**
(https://developer.android.com/reference/androidx/lifecycle/ViewModelProvider.Factory?authuser=1) f

Creates `ViewModelProvider`, which will create `ViewModels` via the given `Factory` and retain them in

## Public methods

| `<T extends ViewModel` (https://developer.android.com/reference/androidx/lifecycle/ViewModel? (https://developer.android.com authuser=1) `> T` | `get` authuser=1#get(java.lang.Clas `(Class<T> modelClass)` Returns an existing ViewModel fragment or an activity), associ |
|---|---|
| `<T extends ViewModel` (https://developer.android.com/reference/androidx/lifecycle/ViewModel? (https://developer.android.com authuser=1) `> T` | `get` authuser=1#get(java.lang.Strin `(String key, Class<T> m` |

**Inherited methods**

| | |
|---|---|
| `Object` | `clone()` |
| `boolean` | `equals(Object arg0)` |
| `void` | `finalize()` |
| `final Class<?>` | `getClass()` |
| `int` | `hashCode()` |
| `final void` | `notify()` |
| `final void` | `notifyAll()` |
| `String` | `toString()` |
| `final void` | `wait(long arg0, int arg1)` |
| `final void` | `wait(long arg0)` |
| `final void` | `wait()` |

# Public constructors

## ViewModelProvider

`public ViewModelProvider (`**`ViewModelStoreOwner`**` (https://developer.android.com/reference/a`

Creates `ViewModelProvider`. This will create `ViewModels` and retain them in a store of the given `ViewModelStoreOwner`.

This method will use the **default factory**
 (https://developer.android.com/reference/androidx/lifecycle/HasDefaultViewModelProviderFactory?
authuser=1#getDefaultViewModelProviderFactory())
if the owner implements **`HasDefaultViewModelProviderFactory`**

(https://developer.android.com/reference/androidx/lifecycle/HasDefaultViewModelProviderFactory?
authuser=1)
. Otherwise, a **_ViewModelProvider.NewInstanceFactory_**
(https://developer.android.com/reference/androidx/lifecycle/ViewModelProvider.NewInstanceFactory?
authuser=1)
will be used.

---

### Parameters

---

| owner | ViewModelStoreOwner |
|---|---|

---

## ViewModelProvider

```
public ViewModelProvider (ViewModelStoreOwner (https://developer.android.com/reference/a
                          ViewModelProvider.Factory (https://developer.android.com/reference/androi
```

Creates `ViewModelProvider`, which will create `ViewModels` via the given `Factory` and
retain them in a store of the given `ViewModelStoreOwner`.

---

### Parameters

---

| owner | ViewModelStoreOwner: a ViewModelStoreOwner whose **ViewModelStore** (https://developer.android.com/reference/androidx/lifecycle/ViewMoc authuser=1) will be used to retain ViewModels |
|---|---|
| factory | ViewModelProvider.Factory: a Factory which will be used to in: new ViewModels |

---

## ViewModelProvider

```
public ViewModelProvider (ViewModelStore (https://developer.android.com/reference/android
                          ViewModelProvider.Factory (https://developer.android.com/reference/androi
```

Creates `ViewModelProvider`, which will create `ViewModels` via the given `Factory` and retain them in the given `store`.

---

**Parameters**

| | |
|---|---|
| `store` | `ViewModelStore`: `ViewModelStore` where ViewModels will be stored. |
| `factory` | `ViewModelProvider.Factory`: factory a `Factory` which will be used to instantiate new `ViewModels` |

# Public methods

## get

```
public T get (Class<T> modelClass)
```

Returns an existing ViewModel or creates a new one in the scope (usually, a fragment or an activity), associated with this `ViewModelProvider`.

The created ViewModel is associated with the given scope and will be retained as long as the scope is alive (e.g. if it is an activity, until it is finished or process is killed).

---

**Parameters**

| | |
|---|---|
| `modelClass` | `Class`: The class of the ViewModel to create an instance of it if it is not present. |

---

**Returns**

| | |
|---|---|
| T | A ViewModel that is an instance of the given type T. |

## get

```
public T get (String key,
              Class<T> modelClass)
```

Returns an existing ViewModel or creates a new one in the scope (usually, a fragment or an activity), associated with this `ViewModelProvider`.

The created ViewModel is associated with the given scope and will be retained as long as the scope is alive (e.g. if it is an activity, until it is finished or process is killed).

## Parameters

| | |
|---|---|
| `key` | `String`: The key to use to identify the ViewModel. |
| `modelClass` | `Class`: The class of the ViewModel to create an instance of it if it is not present. |

## Returns

| | |
|---|---|
| T | A ViewModel that is an instance of the given type T. |

Last updated 2021-02-24 UTC.