JÖNKÖPING UNIVERSITY

School of Engineering

ANDROID VIEWMODEL

Peter Larsson-Green

Jönköping University

Spring 2021

WHY?

Handling runtime configuration changes is hard.

- `activity.onSaveInstanceState()`
  `activity.onRestoreInstanceState()`
  - Can't store object references.
- `activity.onRetainNonConfigurationInstance()`
  `activity.getLastNonConfigurationInstance()`
  - Casting needed.
  - Deprecated in API level 15.
- Retained fragment (model fragment).
  - Casting needed.
- `ViewModel`
  - Implementation built on retained fragments…

BASIC EXAMPLE `ViewModel` (Activity)

Add a Kotlin extension to your Gradle file.

```
implementation "androidx.activity:activity-ktx:1.1.0"
```

Create your own class inheriting from `ViewModel`.

```
class MyViewModel : ViewModel() {
  val serverConnection = ServerConnection("my-server.com")
}
```

Obtaining an instance in an `Activity`.

```
val viewModel: MyViewModel by viewModels()
viewModel.serverConnection.someMethod()
```

```
val viewModel = ViewModelProvider(theActivity).get(MyViewModel::class.java)
```

BASIC EXAMPLE `ViewModel` (Fragment)

Add a Kotlin extension to your Gradle file.

```
implementation "androidx.fragment:fragment-ktx:1.2.5"
```

Create your own class inheriting from `ViewModel`.

```
class MyViewModel : ViewModel() {
  val serverConnection = ServerConnection("my-server.com")
}
```

Obtaining an instance in a `Fragment`.

```
val viewModel: MyViewModel by viewModels()
viewModel.serverConnection.someMethod()
```

```
val viewModel: MyViewModel by activityViewModels()
```

```
val viewModel = ViewModelProvider(theFragment).get(MyViewModel::class.java)
```

onCleared()

Can be used to clean up (e.g. stop listening for events).

```
class MyViewModel : ViewModel() {
  val serverConnection = ServerConnection("my-server.com")
  init {
    serverConnection.onNewMessage {
      // Received a new message from server.
    }
  }
  override fun onCleared() {
    serverConnection.close()
  }
}
```

NOTIFYING THE GUI

Use `LiveData` (and `MutableLiveData`).

- Built on the Observer Pattern.
  - Observers only invoked if the activity/fragment is not stopped.

```
class MyViewModel : ViewModel() {
  val serverConnection = ServerConnection("my-server.com")
  val lastMessage = MutableLiveData<String?>(null)
  init {
    serverConnection.onNewMessage {
      lastMessage.value = it
    }
  }
}
```

```
val viewModel: MyViewModel by viewModels()
viewModel.lastMessage.observe(theActivity){
  val theValue = it
  // Update the GUI.
}
```

AndroidViewModel

A `ViewModel` with a `Context`.

- Can call `getApplication()` to get the `Context`.

```
class MyViewModel : AndroidViewModel() {
  val db = Room.databaseBuilder(
    getApplication(),
    MyDatabase::class.java,
    "my-db"
  ).build()
}
```