droidcon

Migrating the UI

Catalin

@catalin_ste

in/catalins

droidcon

Module 2

Module Intro and Project Setup

Module Intro and Project Setup

In this module you will learn:

- Project discovery and setup
- Imperative vs Declarative paradigm
- Enable Compose in our project
- Migrate a section of a screen
- Migrate a complete screen
- Implement an XML layout in Compose
- How state works in Compose
- Migrate a RecyclerView

droidcon

App Demo

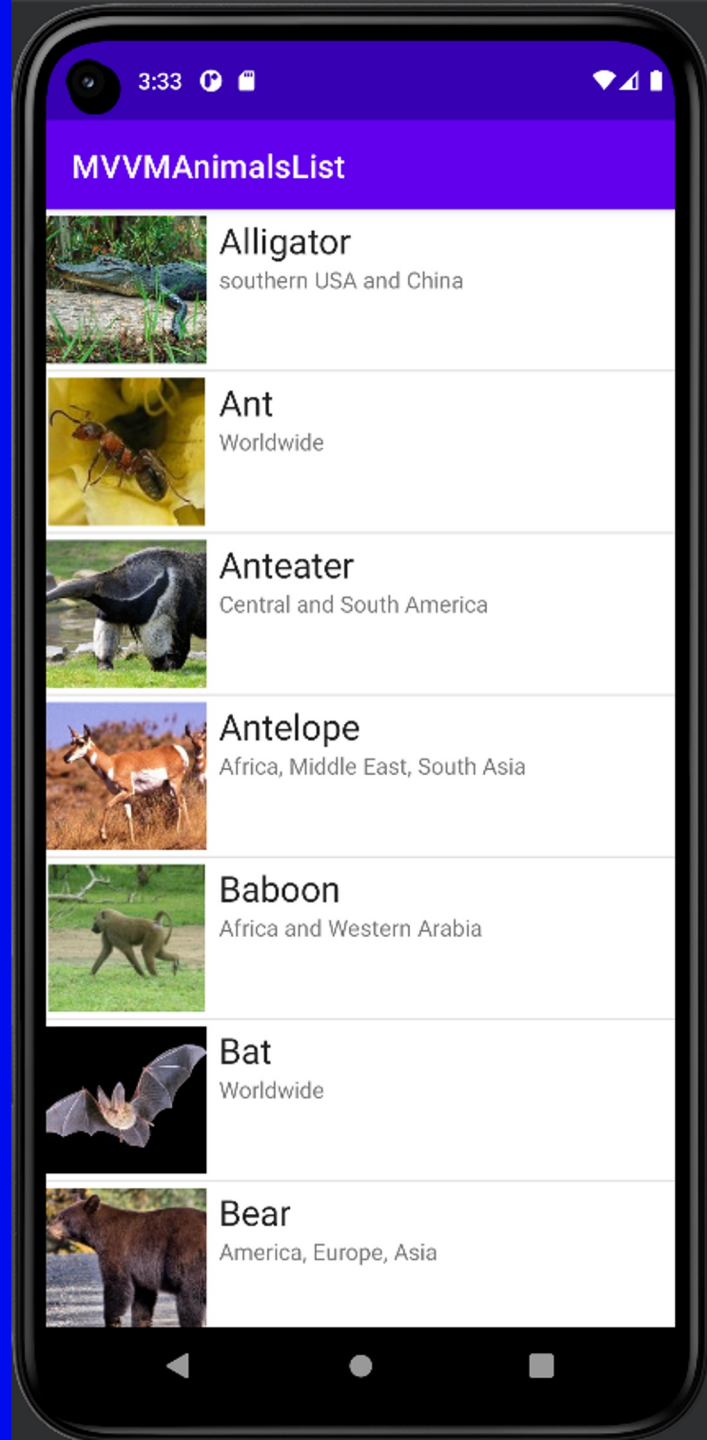
2 screens

droidcon

App Demo

2 screens

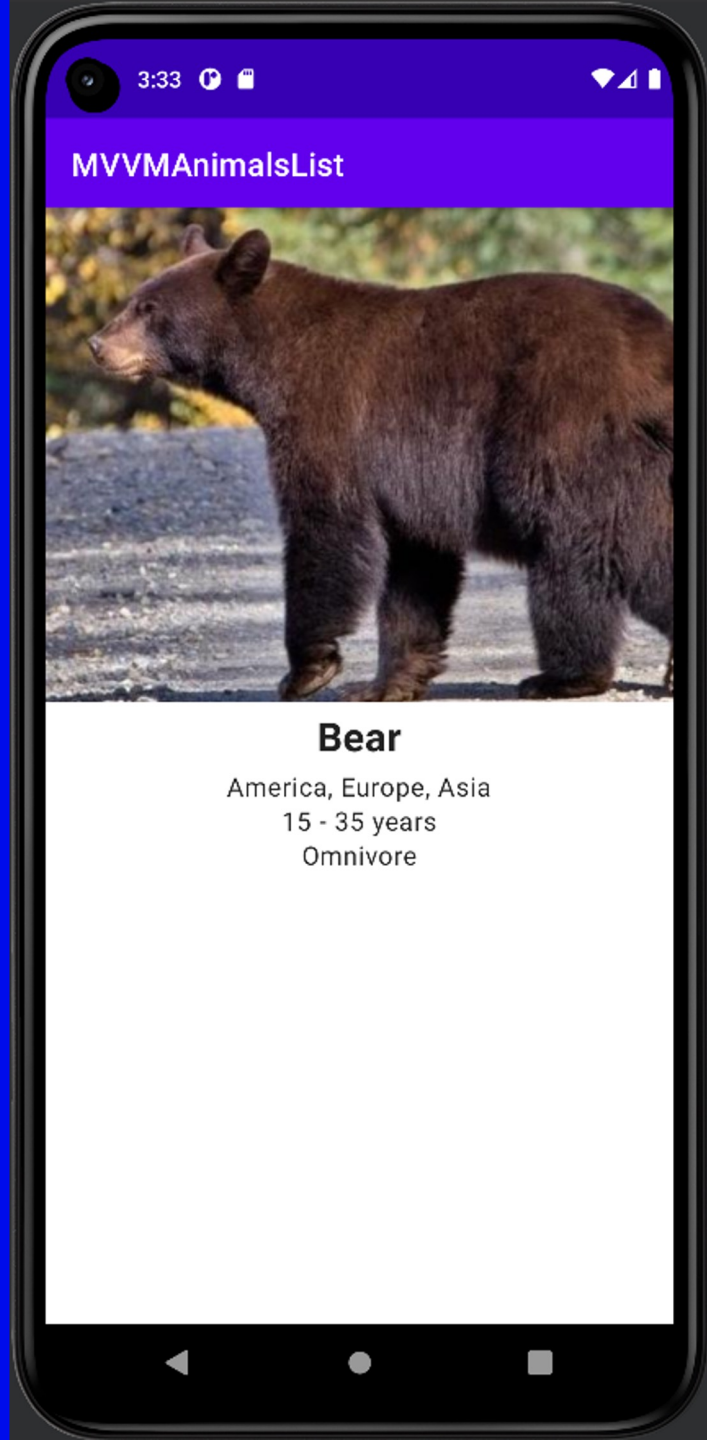
- list of animals



App Demo

2 screens

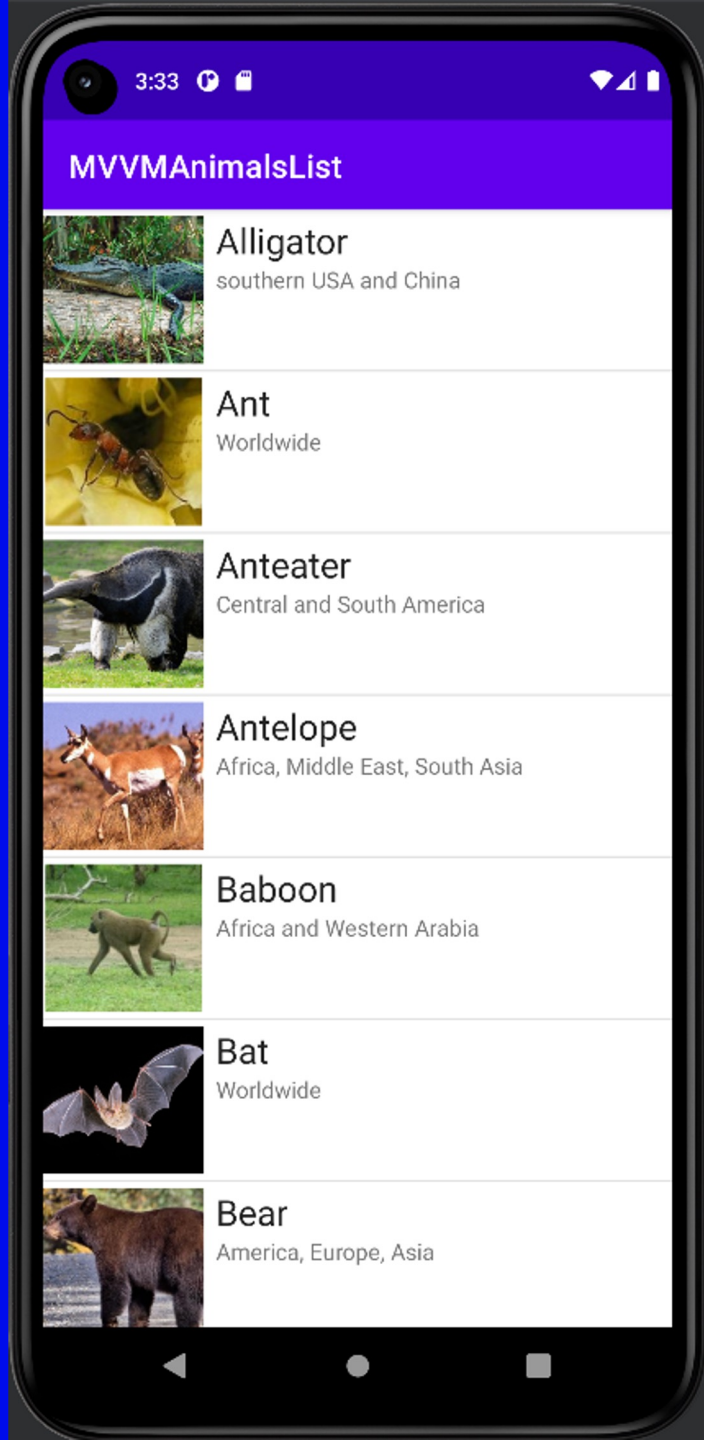
- list of animals
- animal details



App Demo

2 screens
- list of animals
- animal details

Behind the scenes
- Activity / Fragment
- MVVM architecture
- Retrofit
- Glide
- ViewBinding
- Hilt
- XML Navigation
- Animations



droidcon

Imperative vs Declarative

Catalin

@catalin_ste

in/catalins

Imperative vs Declarative

Imperative: tree of widgets

- Walk the tree to update a certain View
- Add / remove children manually
- Add user interaction (.onClick)
- Update UI based on observable state

Declarative: composable functions

- State update triggers changes
- Only redraw composables that change
- Fast, idempotent and free of side effects*

Imperative vs Declarative

Add Composable functions to existing XML
Add XML to Composable functions
Mix Fragments with Composable screens
Mix widget trees with state based Composable functions
Theming works differently
Gradual approach to migration

droidcon

droidcon

Add Compose to an existing project

Catalin

@catalin_ste

in/catalins

Add Compose to an existing project

Add Compose version in build.gradle project file
Enable Compose as a build feature
Set composeOptions
Add all required dependencies

droidcon