

## Implementing Retrofit and Hilt Dependency Injection

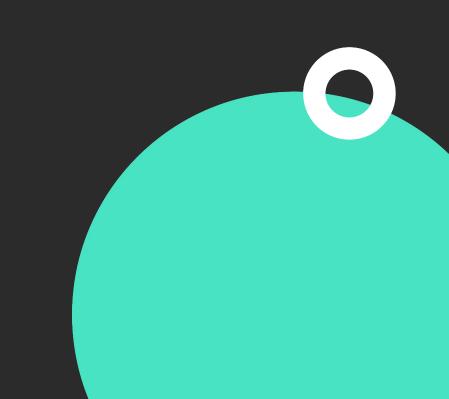


#### Catalin Stefan



@catalin\_ste





#### Section Overview: TODO

- •Add appear or fade animation to each bullet
  - Talk for each bullet as they appear on screen
    - Use only theme colors to highlight a text or code or insert any shapes
      - I am sure this fourth level is not required. Use it cautiously.
- •Use code snippet and highlight it just like done below
  - Display Toast on click of RecyclerView item
- •Learn to use @Embedded annotation in Room database
- •Explore the starter project
- •Press enter after this bullet and get default animation in next bullet
  - This bullet has a animation
  - Use @Preview with your Composable to check its preview
- •// Don't write below this bullet. Instead use a new slide.





#### Retrofit

- •Communicates with Google Books API
- •Retrieves JSON response
- •Parse JSON response into data classes
- •Makes data available using Kotlin Flows





## Hilt Dependency Injection

- Injects dependencies into ViewModel
- •Makes Retrofit repo available without tight coupling





## Connectivity Monitor

- •We will create a class to monitor network availability
- •Allows us to update the UI





## Section Summary: TODO

- •Add appear or fade animation to each bullet
  - Talk for each bullet as they appear on screen
    - Use only theme colors to highlight a text or code or insert any shapes
      - I am sure this fourth level is not required. Use it cautiously.
- •Use code snippet and highlight it just like done below
  - Display Toast on click of RecyclerView item
- •Learn to use @Embedded annotation in Room database
- •Explore the starter project
- •Press enter after this bullet and get default animation in next bullet
  - This bullet has a animation
  - Use @Preview with your Composable to check its preview
- •// Don't write below this bullet. Instead use a new slide.







# Adding ViewModel and Repository Layer in MVVM

Up Next



