

Preface

Kotlin coroutines and flows allow developers to do asynchronous programming in Android using simple, modern, and testable code.

This book focuses on coroutines and flows using hands-on learning. You will begin with the basics of asynchronous programming, including an overview of coroutines and flows while also integrating them into your Android projects. You'll understand how to manage cancelations and exceptions, and then explore how to test your coroutines and flows.

By the end of this book, you will be able to use Kotlin coroutines and flows to simplify asynchronous programming in Android.

Who this book is for

This book is for Android developers who want to build high-quality apps using coroutines and flows and level up their Android development skills. Beginners with basic knowledge of Android development and Kotlin will also find this book useful.

What this book covers

[Chapter 1](#), Introduction to Asynchronous Programming in Android, visits asynchronous programming in Android and shows the various ways it is being done now. Toward the end, the new recommended ways of coroutines and flows will be introduced.

[Chapter 2](#), Understanding Kotlin Coroutines, introduces Kotlin coroutines and shows how they can be used for asynchronous programming in

Android. It will demonstrate how to create coroutines, and discuss coroutine builders, scopes, dispatchers, contexts, and jobs.

[Chapter 3](#), [Handling Coroutine Cancelations and Exceptions](#), discusses coroutine cancelations, and how you can manage coroutine cancelations, timeouts, and exceptions properly.

[Chapter 4](#), [Testing Kotlin Coroutines](#), describes how you can test Kotlin coroutines in Android.


[Chapter 5](#), [Using Kotlin Flows](#), covers the basics of Kotlin flows and how they can be used for asynchronous programming in Android. It continues with creating flows with flow builders. It will also discuss flow operators, buffering and combining flows, and StateFlow and SharedFlow.

[Chapter 6](#), [Handling Flow Cancelations and Exceptions](#), explores how to manage cancelations, completion, and exceptions in your flows.

[Chapter 7](#), [Testing Kotlin Flows](#), provides details on how you can test the flows in your Android project.

To get the most out of this book

You will need to have basic Android development skills and knowledge of using Kotlin.


You will need to have the latest version of Android Studio. You can download the latest version at <https://developer.android.com/studio> . For an optimal experience, the following specifications are recommended:


- Intel Core i5 or equivalent or higher
- 4 GB RAM minimum
- 4 GB available space

Software/hardware covered in the book	Operating system requirements
Android Studio	Windows, macOS, or Linux

If you are using the digital version of this book, we advise you to type the code yourself or access the code from the book's GitHub repository (a link is available in the next section). Doing so will help you avoid any potential errors related to the copying and pasting of code.

Download the example code files

You can download the example code files for this book from GitHub at <https://github.com/PacktPublishing/Simplifying-Android-Development-with-Coroutines-and-Flows/> . If there's an update to the code, it will be updated in the GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at <https://github.com/PacktPublishing/> . Check them out!

Conventions used

There are a number of text conventions used throughout this book.

Code in text: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: “**runOnUiThread** will perform the **displayText(text)** function on the main UI thread.”

A block of code is set as follows:

```
lifecycleScope.launch(Dispatchers.IO) {  
  
    val fetchedText = fetchText()
```

```
        withContext(Dispatchers.Main) {  
  
            displayText(fetchedText)  
  
        }  
  
    }
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
private fun fetchTextWithThread() {  
  
    Thread {  
  
        // get text from network  
  
        val text = getTextFromNetwork()  
  
        runOnUiThread {  
  
            // Display on UI  
  
            displayText(text)  
  
        }  
  
    }.start()  
  
}
```

Any command-line input or output is written as follows:

```
java.lang.IllegalStateException: Module with the Main  
dispatcher had failed to initialize. For tests  
Dispatchers.setMain from kotlinx-coroutines-test  
module can be used
```

Bold: Indicates a new term, an important word, or words that you see on-screen. For instance, words in menus or dialog boxes appear in **bold**.

Here is an example: “In Android Studio, the **Editor** window identifies the suspending function calls in your code with a gutter icon next to the line number.”


TIPS OR IMPORTANT NOTES

Appear like this.


Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, email us at customercare@packtpub.com and mention the book title in the subject of your message.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit www.packtpub.com/support/errata  and fill in the form.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at copyright@packt.com with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit authors.packtpub.com 

Share Your Thoughts

Once you’ve read *Simplifying Android Development with Coroutines and Flows*, we’d love to hear your thoughts! Please [click here to go straight to](#)

[the Amazon review page](#)  for this book and share your feedback.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

[Support](#) [Sign Out](#)

©2022 O'REILLY MEDIA, INC. [TERMS OF SERVICE](#) [PRIVACY POLICY](#)