Join the discussion @ p2p.wrox.com

Wrox Programmer to Programmer™

Expert Android® Studio

Murat Yener, Onur Dundar

EXPERT ANDROID® STUDIO

EXPERT Android® Studio

EXPERT
Android® Studio

Murat Yener

Onur Dundar

wrox™
A Wiley Brand

Expert Android® Studio

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-119-08925-4
ISBN: 978-1-119-11071-2 (ebk)
ISBN: 978-1-119-11073-6 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2016947909

To Nilay, Burak, Semra, and Mustafa Yener, for all your support and the time I needed to write this book.

—Murat

To Canan and my entire family: Aysel, Ismail, Ugur, Umut, Aysun, and Murat.

—Onur

ABOUT THE AUTHORS

MURAT YENER is a code geek, open source committer, Java Champion, and Google Developer Expert on Android, who is working at Intel as an Android developer. He is the author of *Professional Java EE Design Patterns* (Wrox, 2015). He has extensive experience with developing Android, Java, web, Java EE, and OSGi applications, in addition to teaching courses and mentoring. Murat is an Eclipse committer and one of the initial committers of the Eclipse Libra project.

Murat has been a user group leader at GDG Istanbul since 2009, organizing, participating, and speaking at events. He is also a regular speaker at major conferences such as DroidCon, JavaOne, and Devoxx.

LinkedIn: `www.linkedin.com/in/muratyener`

Twitter: `@yenerm`

Blog: `www.devchronicles.com`

ONUR DUNDAR started his professional career in software engineering at Intel Corporation as a Software Application Engineer working on mobile platforms, tablets, and phones with MeeGo and Android, and later on IoT platforms such as IPTV, Intel's Galileo, and Edison. He is the author of *Home Automation with Intel Galileo* (Packt Publishing, 2015). Onur presents training sessions on Android application development, so he was keen to author this book.

LinkedIn: `www.linkedin.com/in/odundar`

Twitter: `@odunculuk`

Amazon author page: `www.amazon.com/Onur-Dundar/e/B00V0VOIGA`

ABOUT THE TECHNICAL EDITOR

XAVIER HALLADE is Application Engineer at Intel Software and Services Group in France. He's been working on a wide range of Android frameworks, libraries, and applications. Xavier is a Google Developer Expert in Android, with a focus on the Android NDK, and actively contributes in the Android community, writing articles, helping developers on Stack Overflow, and giving talks around the word.

CREDITS

PROJECT EDITOR
Tom Dinse

TECHNICAL EDITOR
Xavier Hallade

PRODUCTION EDITOR
Barath Kumar Rajasekaran

COPY EDITOR
Nancy Rapoport

MANAGER OF CONTENT DEVELOPMENT & ASSEMBLY
Mary Beth Wakefield

PRODUCTION MANAGER
Kathleen Wisor

MARKETING MANAGER
Carrie Sherrill

PROFESSIONAL TECHNOLOGY & STRATEGY DIRECTOR
Barry Pruett

BUSINESS MANAGER
Amy Knies

EXECUTIVE EDITOR
Jody Lefevere

PROJECT COORDINATOR, COVER
Brent Savage

PROOFREADER
Nancy Bell

INDEXER
Johnna VanHoose

COVER DESIGNER
Wiley

COVER IMAGE
Leo Blanchette/Shutterstock

ACKNOWLEDGMENTS

I WANT TO THANK MY COAUTHOR, ONUR DUNDAR, for all his hard work and for his efforts to keep me on schedule. Without him, this book wouldn't be half as good.

I am grateful to my team and colleagues, in particular John Wei and Sunil Tiptur Nataraj, who gave me the time and flexibility to work on this book, and Angus Yeung for his support. My thanks also go to my Google Developer Relations contacts, Uttam Tripathi, Martin Omander, Baris Yesugey, and others who supported me in all ways possible. I want to thank Alex Theedom for covering for me on my Java EE–related responsibilities while I was deeply buried in Android. No words are enough to thank Jim Minatel and Tom Dinse, who patiently worked on all the details while keeping most of the stresses away from us. And thanks, of course, to everyone at Wrox/Wiley who got this book on the shelves.

I must thank three important people who are responsible for who I am in my professional life.

First, thanks to my dad, Mustafa Yener, for giving me my first computer, a C64, at an early age while I was asking for slot cars. I wrote my very first code on that computer.

Second, thanks to my thesis advisor, Professor Mahir Vardar, for the early guidance I needed to start my career.

Finally, thanks to my lifetime mentor and friend, Naci Dai, who taught me almost everything I know about being a professional software developer.

—Murat Yener

I WANT TO THANK MURAT YENER for taking me on this journey and enabling me to share in the fun of writing this book. With Murat's experience and knowledge, we delivered a high quality book for developers and engineers.

I am thankful to Oktay Ozgun for his wisdom and advice throughout my career and life. His guidance has helped me become a passionate engineer and a better person.

I also want to thank Professor Arda Yurdakul, who was generous with her experience and guidance as she encouraged me to learn more about software and computers and to become a better engineer.

In my final year at Bogazici University, Professor Cem Ersoy, Professor Alper Sen, and Dr. Hande Alemdar provided great assistance. I would also like to thank the folks at Intel—Steve Cutler, Andrew John, Brendan LeFoll, Todor Minchev, Peter Rohr, Rami Radi, Alex Klimovitski, and Marcel Wagner—for their great support and for the opportunities they provided to help me learn and develop in my engineering career.

I want to thank Professor Gurkan Kok, for the interesting stuff I am working on right now.

Thanks to all the Wiley/Wrox editors for their help with this book.

Finally, my special thanks to Ali Caglar Ozkan for motivating me (and taking great photos) after my first book, along with with Aytac Yurdakurban and Engin Efecik, for supporting my continuing interest in writing books.

—Onur Dundar

CONTENTS

FOREWORD

"A bad workman always blames his tools."

When my brother and I were growing up in leafy, suburban England, my father relied on that old proverb. My brother and I often found excuses to avoid the admittedly small number of chores we were asked to do, and we placed the blame for not completing the task at hand on an inanimate object that couldn't answer back. This proverb was my father's standard response to our excuses, and it immediately negated our protestations.

As I've gotten older, I have learned that there is far more wisdom in this old proverb than merely getting young boys to complete their chores. It is not just about ensuring that you have the right tools for the task at hand, but it is also about having the knowledge of how to use them effectively that is key to being productive. If you think about the standard tools in a handyman's toolkit, it is pretty clear to most people that a hammer is not the most appropriate tool to remove a screw from a piece of wood, but as tools become more complex and refined, these distinctions become less clear.

In the Android development world, the de-facto standard development tool is Android Studio, not least because it is under extremely active development by Google—the same organization responsible for Android itself. Since the early preview versions arrived in May 2013, the feature set has grown quite considerably and continues to do so at an impressive rate. If we also consider that Android Studio is built on the foundations of IntelliJ IDEA—which is already an extremely feature-rich development environment—then it should be pretty clear that any analogies with hammers or screwdrivers are going to break down rather quickly. Rather than comparing Android Studio with individual tools, it is, perhaps, better to consider it as the entire toolbox, which contains lots of individual tools that can sometimes be used individually, sometimes be used together, but, when used effectively, can simplify and speed up many of our everyday development tasks—including the really mundane or repetitive ones that we all hate!

Modern software development is so much more than simply writing code, and this is especially true on Android. The main logic of your app may be written in Java. You also have resources (which are largely XML-based) such as vector drawables (which incorporate SVG path data into that XML), build files (which are groovy/grade files), and test source code (which is Java with test domain–specific dialects such as Espresso, Fest, or Hamcrest). This is before you start considering frameworks that change the syntax and flow of your code, such as Rx, and even alternate languages that are gaining traction, such as Kotlin. Mastery of all of this can be hard. The ability to "context switch" between different components, languages, frameworks, and dialects is made much easier by basic features such as code highlighting and pre-compilation, which show errors inline as you code. But we are so used to these that we hardly notice them, and because they have become second nature to us, context switching itself becomes second nature.

While using the tools available until they become second nature is important, a prerequisite for that is actually knowing what tools there are and how to use them effectively. That is where this book comes in. Murat and Onur have provided a guide to Android Studio and its many facets that will be of great value to both the novice and the seasoned Android Studio user alike.

Mark Allison
June 2016

INTRODUCTION

NO MATTER HOW GOOD YOU ARE AT WRITING CODE, without proper knowledge of Integrated Development Environments (IDEs), you will face many obstacles. This book covers Google's Android Studio, the official tool for developing Android applications. Each chapter focuses on a specific topic, progressing from the basics of how to use the IDE to advanced topics such as writing plugins to extend the capabilities of the IDE.

WHO THIS BOOK IS FOR

This book is for developers with any level of experience. Whether you are new to Android or a seasoned Android developer who used Eclipse-based ADT before, this book will bring you to a level where you can unleash your true development potential by making use of Android Studio's tools.

WHAT THIS BOOK COVERS

This book not only covers features of Android Studio essential for developing Android apps but also touches on topics related to the whole development cycle. The following are just a few examples of the topics covered that are basic to Android Studio or that extend its capabilities:

- Sharing and versioning your code with Git
- Managing your builds with Gradle
- Keeping your code maintainable and bug free with testing
- Controlling the whole build and test cycle with Continuous Integration
- Writing plugins for Android Studio to extend its capabilities and add desired custom features
- Using third-party tools with Android Studio to improve the development process

HOW THIS BOOK IS STRUCTURED

Each chapter focuses on a specific topic related to Android Studio or an accompanying tool by explaining why it is needed and how it is used or configured. Some chapters provide code samples to demonstrate the use case or provide an example for the topic.

- **Chapter 1: Getting Started:** Installing and setting up your development environment. Creating an emulator for running your projects.
- **Chapter 2: Android Studio Basics:** Beginning with Android Studio, creating a new project, building your project, and migrating projects to Android Studio.

- **Chapter 3: Android Application Development with Android Studio:** Structure of Android Studio projects. How to use assets, XML files and the Android Manifest. Creating and working with modules.
- **Chapter 4: Android Studio In Depth:** Deep dive into Android Studio, explaining menus, editors, views, and shortcuts. How to use live templates and refactoring. How to build your projects and sign apks.
- **Chapter 5: Layouts with Android Studio:** How to use layouts with Android Studio. Explanation of previews and tools for UI development. Managing external dependencies. How to use and organize assets.
- **Chapter 6: Android Build System:** How to use and configure Gradle effectively. Writing plugins for Gradle
- **Chapter 7: Multi-Module Projects:** Adding modules in your project. How to create and work with Phone/Tablet, Library, Wear, TV, Glass, Auto, and Cloud modules.
- **Chapter 8: Debugging and Testing:** Debugging Android code with ADB. Learn details of the Android Devices Monitor, Android virtual devices, Lint, and testing your code.
- **Chapter 9: Using Source Control:** GIT: How to share your project and enable version control by using Git.
- **Chapter 10: Continuous Integration:** Automating your builds, tests, and releases using continuous integration servers.
- **Chapter 11: Using Android NDK with Android Studio:** Installing and using Android NDK for building projects with C/C++ code.
- **Chapter 12: Writing Your Own Plugins:** Writing your own plugins to extend the capabilities of the IntelliJ platform. Interacting with UI, editor, and adding your actions.
- **Chapter 13: Third-Party Tools:** Other accompanying tools that can help and speed the development lifecycle.

WHAT YOU NEED TO USE THIS BOOK

Any modern computer with an operating system that is supported by Android SDK and Android Studio is sufficient to use Android Studio, build Android apps, and run the samples given in this book. You need to install appropriate Android SDK, Android Studio, and Java Virtual Machine (JVM) for your OS. Some chapters require additional tools or frameworks to be installed such as Android NDK. You can find more information on exact hardware requirements needed in Chapter 1.

WHY WE WROTE THIS BOOK

In November 2007, Google released a preview version of Android SDK to allow developers to start playing with the new mobile operating system. Roughly two years later, in October 2009, ADT (Android Developer Tools) a plugin set for Eclipse, was released to the public.

As a Google I/O 2009 attendee, I (Murat) was lucky enough to have an Android device and was probably one of the earliest developers to download and install the plugins to my Eclipse. As years passed, we both followed the same passion to download and try new stuff released with new ADT versions.

At the time, I was an Eclipse committer who knew how to write plugins, extend the IDE's capabilities, and introduce the behavior and functionality I needed. So with each release of ADT, I was more and more excited to see what had been done with the tools.

On May 2013, at Google I/O, roughly four years after our love-hate relationship with ADT started, Google announced Android Studio, which soon became the official, supported IDE for Android development. ADT was never perfect. but it was familiar. Like many other developers, we knew all the shortcuts, how things work, what to do when something was not working, workarounds, and how the projects were structured. More significantly, we were able to write our own plugins or inspect ADT plugins to see why something went wrong. However, with the release of Android Studio, suddenly we were all in a new platform that we knew very little about.

We resisted switching to Android Studio for a while, but finally gave it a try. Suddenly, Android, a platform we were long familiar with, was a stranger. The new project structure was very different because of the changes introduced by IntelliJ and Gradle. To adopt IntelliJ, we decided to follow IntelliJ shortcuts instead of using IntelliJ shortcut mapping for Eclipse shortcuts, which made the situation even worse. We were barely able to search for a file or piece of code, navigate through menus, right-click to create files, or even generate some basic getters and setters. We went from being experts with ADT to beginners with Android Studio.

We had finally had enough! We were experienced developers, but struggled with Android Studio and were not able to show our skills. So we started following IntelliJ talks, pinning the IntelliJ shortcut cheat sheet in our cubicles, reading IntelliJ plugin code, and forcing ourselves to use Android Studio in our daily work.

This book is the summary of the lessons we learned walking unaided on this difficult path . This book is what we needed for ourselves when we were switching from Eclipse-based ADT to IntelliJ-based Android Studio. This is why we believe any developer, whether an Android newbie or a seasoned Android developer who used to work on ADT, will find this book useful for developing his or her knowledge of the tools that are actually there to support his or her coding skills.

Quoting Alex Theedom, co-author of my previous book: “Every chapter that we wrote has this goal: Write content that we would like to read ourselves.” We followed the same goal with Onur and the result is the book you are holding in your hands.

We hope that you enjoy reading this book as much as we enjoyed writing it.

> **NOTE** *Be sure to read our blog at* `http://www.devchronicles.com/2016/06/expert-android-studio-book-updates.html` *to see the changes announced at Google I/O 2016.*

CONVENTIONS

To help you get the most from the text and keep track of what’s happening, we’ve used a number of conventions throughout the book.

> **WARNING** *Boxes like this one hold important, not-to-be forgotten information that is directly relevant to the surrounding text.*

> **NOTE** *Notes, tips, hints, tricks, and asides to the current discussion are offset and placed in italics like this.*

As for styles in the text:

- We *highlight* new terms and important words when we introduce them.
- We show keyboard strokes like this: Ctrl+A.
- We show file names, URLs, and code within the text like so: `persistence.properties`.
- For code:

```
We use a monofont type for code examples.
We use bold to emphasize code that is of particular importance in
the current context.
```