

Pro Kotlin Web Apps from Scratch

Building Production-Ready Web Apps
Without a Framework

—
August Lilleaas

Apress®

Pro Kotlin Web Apps from Scratch

Building Production-Ready Web Apps Without a Framework

August Lilleaas

Apress®

Pro Kotlin Web Apps from Scratch: Building Production-Ready Web Apps Without a Framework

August Lilleaas
Oslo, Norway

ISBN-13 (pbk): 978-1-4842-9056-9
<https://doi.org/10.1007/978-1-4842-9057-6>

ISBN-13 (electronic): 978-1-4842-9057-6

Copyright © 2023 by August Lilleaas

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Jonathan Gennick
Development Editor: Laura Berendson
Coordinating Editor: Jill Balzano

Cover Photo by George Tasios on Unsplash

Distributed to the book trade worldwide by Springer Science+Business Media LLC, 1 New York Plaza, Suite 4600, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub (<https://github.com/Apress>). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*Dedicated to my mother and father, who made it all happen,
and
in loving memory of Ann-Cecilie Saltnes (11.03.1980–04.07.2022)*

Table of Contents

About the Author	xv
About the Technical Reviewer	xvii
Acknowledgments	xix
Introduction	xxi
Part I: Up and Running with a Web App.....	1
Chapter 1: Setting Up a Development Environment.....	3
Get Started with IntelliJ IDEA Community Edition	3
Download and Run IntelliJ IDEA	4
Create a New Kotlin Project with IntelliJ IDEA.....	4
Kotlin Hello, World!	9
Kotlin Naming Conventions	9
Write Some Code	9
Run Your Code with IntelliJ IDEA	11
Run Your Code with Gradle	12
IntelliJ IDEA Tips and Tricks	13
Look Out for the Progress Bar	14
Remember to Refresh the Gradle Project in IntelliJ IDEA.....	15
Embrace Auto-completion in IntelliJ IDEA	16
Chapter 2: Setting Up the Web App Skeleton.....	19
Web Server Hello, World!.....	20
Choosing Ktor	20
Add the Ktor Library	20
Start a Web Server.....	21
Extracting to a Separate Function	22

TABLE OF CONTENTS

Using Lambdas	23
Named Arguments Instead of Magic Numbers and Booleans	24
Run Your Web App.....	25
Logging	26
Loggers on the Java Platform.....	26
Configure Logging in Your Web App.....	27
Write to the Log from Your Own Code.....	30
A Note on “Magic” XML Config Files	32
Useful Error Pages	33
Chapter 3: Configuration Files	37
Create a Configuration File	38
The Configuration File Anatomy.....	38
Reading Values from Your Configuration File.....	39
Make the Configuration Fail Early and Type-Safe	40
Store Configuration in a Data Class.....	41
Load Config into the Data Class.....	42
Kotlin Null Safety	42
Kotlin Platform Types.....	43
Use the Data Class in Your Web App.....	44
Use let Blocks to Avoid Intermediate Variables	45
Provide Different Default Values for Different Environments	46
Environment-Specific Config Files	46
Define the Web App Environment	47
Override Defaults with Environment-Specific Config	47
Secret Config Values	48
Don’t Store Secrets in Version Control	48
Logging Config on Web App Startup.....	51
Format the Output	52
Masking Secrets.....	52
Writing to the Log	53

Chapter 4: Decoupling Web Handlers from Specific Libraries	55
Your Own HTTP Response Abstraction.....	56
Representing HTTP Responses.....	56
Multiple Response Types	57
Convenience Functions for Headers.....	59
Case-Insensitive Headers.....	64
The fold Function.....	65
Destructuring Assignment	66
Connecting to Ktor	67
Ktor Route Mapping.....	67
Extension Functions	68
Functions Returning Functions.....	69
Map Headers	70
Map TextWebResponse and Status Code.....	71
Map JsonResponse	72
Using the New Mapping	74
A Note on Overengineering	75
Part II: Libraries and Solutions.....	77
Chapter 5: Connecting to and Migrating SQL Databases	79
Connecting to a SQL Database.....	80
The Connection Pool.....	80
Installing the Database Driver	81
Setting Up H2.....	82
Updating WebappConfig	82
Set Up the Connection Pool	84
Creating a Connection	85
The Initial Schema	86
Installing Flyway.....	86
Running Flyway During Startup.....	86
Creating the Initial Schema	88

TABLE OF CONTENTS

Managing Schema Changes	89
Don't Edit Migrations After Run	89
Adding More Schema	90
Adding Non-nullable Columns	90
Backward-Compatible Migrations	92
Insert Seed Data	93
Repeatable Migrations.....	93
Updating Repeatable Migrations	94
Handling Failed Migrations	95
Failed Migrations Locally.....	95
Failed Migrations in Production.....	95
Rerunning Failed Migration	96
Manually Performing Migration	96
Chapter 6: Querying a SQL Database	97
Setting Up Querying	97
Use SQL Directly	98
Installing Kotliquery.....	99
Mapping Query Results	99
Execute SQL Queries.....	100
Creating a Session.....	100
Querying for Single Rows.....	101
Querying for Multiple Rows	101
Inserting Rows.....	103
Updating and Deleting	104
Positional vs. Named Parameters.....	104
Additional Operations	105
Querying from Web Handlers	105
Creating a Helper Function	105
A Note on Architecture.....	107
Avoid Long-Running Connections.....	107

Maps vs. Data Classes	108
Passing Around Maps	108
Passing Individual Properties	109
Mapping to a Data Class	110
Database Transactions	113
Creating Transactions	113
Transactions in Web Handlers	114
Type-Safe Transactional Business Logic	115
Nested Transactions	116
Chapter 7: Automated Tests with junit 5	119
Setting Up Your Environment	119
Adding junit 5 and kotlin.test	120
Writing a Failing Test	121
Running a Failing Test	121
Making the Test Pass	123
Writing Web App Tests	125
Setting Up the Basics	125
Writing a Failing Test	126
Making the Test Pass	128
Avoiding Leaky Tests	129
Leaky Tests	129
Avoiding Leaks with Transactions	130
Avoiding Leaks with Relative Asserts	133
Test-Driven Development	135
Design and Verification	135
Writing a Failing Test	135
Making the Test Pass	137
Notes on Methodology	137
What About Front-End Tests?	137
Real Database Writes vs. Mocks	138
Unit vs. Integration Tests	139

TABLE OF CONTENTS

Chapter 8: Parallelizing Service Calls with Coroutines	141
Preparing Your Web App.....	141
Implementing a Fake Service.....	142
Adding a Second Server	142
Understanding Coroutines.....	144
The delay Coroutine.....	144
The Problem with Threads.....	144
Suspending Instead of Locking	145
Coroutine Contexts	145
Coroutines in Ktor.....	147
Parallelizing Service Calls.....	148
Adding Dependencies.....	148
Performing Parallelized Calls	149
Handling Race Conditions.....	151
Adding to Ktor.....	151
Mixing Coroutines and Blocking Calls.....	152
Wrapping Blocking Calls.....	152
Coroutine Internals.....	153
Kotlin vs. KotlinX.....	153
Comparing with Arrow	154
Using Low-Level Continuations	157
Java Platform Implementation Details	158
Chapter 9: Building Traditional Web Apps with HTML and CSS	161
Patterns and Organization.....	162
Generating HTML	162
Using the kotlinx.html DSL	162
Using the Ktor HTML DSL	163
The Power of the DSL	164
Operator Overloading.....	165
Adding Custom Tags	166

CSS and Assets	167
Serving Static Files.....	167
Instant Reloading.....	170
Reusable Layouts.....	171
Adding a Layout.....	172
Using the Layout.....	173
Adding WebResponse for HTML	174
Implementing HtmlWebResponse	174
Extension Function Precedence	175
Responding with HtmlWebResponse.....	176
A Note on Abstractions	178
User Security.....	178
Password Hashing.....	178
Using Bcrypt for Hashing.....	179
Updating Your Migrations	182
Adding a Login Form	183
Wiring Up Your Web App	183
Different Types of Sessions	185
Adding a New Ktor Extension Function	186
Configuring Session Cookies.....	187
Logging In.....	189
Protecting Routes with Authentication	191
Logging Out	194
Chapter 10: Building API-Based Back Ends.....	197
Handling API Calls	197
Parsing Input	197
Validating and Extracting Input.....	198
Avoid Automatic Serialization.....	200
Internal vs. External APIs.....	202

TABLE OF CONTENTS

Single-Page Apps.....	202
Hosting Alongside API.....	202
Hosting Separately with CORS	204
Authenticating Users	206
Native Apps.....	207
Using Cookies for Authentication.....	207
Using JWTs for Authentication	207
Performing Authentication.....	211
Chapter 11: Deploying to Traditional Server Based Environments	213
Packaging as Self-Contained JAR Files	213
What's a JAR File.....	213
Self-Contained JAR Files	214
Packaging with Gradle.....	215
Building the JAR File	215
Executing the JAR File.....	216
Packaging for Production.....	217
Building Docker Images.....	217
Running Docker Images Locally	219
Deploying to Production	220
Chapter 12: Building and Deploying to Serverless Environments	223
Detaching Your Code.....	224
Separating Handlers from Ktor.....	224
Basic Structure.....	225
Detached Web Handlers	227
Run Web Handlers on AWS Lambda.....	227
Making AWS Lambda Web Handlers.....	227
Deploying to AWS Lambda.....	229
Calling Your Existing Handlers.....	237
Improving Performance	243
Performance and Cold Starts	244
Initialization Time vs. Execution Time.....	244

Lazy Loading.....	245
Initializing Correctly.....	246
Migrations and H2.....	248
Java Runtime Flags.....	248
GraalVM, Kotlin/JS, and Kotlin Native.....	249
Chapter 13: Setup, Teardown, and Dependency Injection with Spring Context....	251
Why Spring and Dependency Injection?	252
Setting Up Spring Context.....	253
Adding Your Data Source	254
lateinit in Kotlin.....	256
Starting Your Web App.....	257
Extended Usage	258
Chapter 14: Enterprise Authentication Using Spring Security.....	261
Preparing Your Web App.....	261
Setting Up Embedded Jetty	261
Initializing the Servlet.....	264
Adding Ktor to Your Servlet.....	265
Using Spring Security	267
Setting Up the Servlet Filter	267
Configuring Spring Security	269
Authenticating Users	271
Filtering Requests.....	273
Accessing the Logged-In User.....	274
Part III: Tools of the Trade.....	277
Chapter 15: Choosing the Right Library.....	279
What's a Library?.....	279
Popularity Is Largely Irrelevant	280
Documentation and Stability Are Important	281

TABLE OF CONTENTS

Avoid Following the “X Way”	281
Don’t Shy Away from Platform Libraries	282
Boundaries and Layers	283
Chapter 16: An Assortment of Kotlin Tricks	285
Delegation (by lazy).....	285
Inline Functions.....	286
Reified Generics.....	288
Contracts.....	289
And So Much More.....	290
Appendix A: Using Jooby Instead of Ktor	293
Appendix B: Using Hoplite Instead of Typesafe Config	303
Appendix C: Using Spek Instead of junit 5.....	307
Index.....	313

About the Author



August Lilleaas has been building web apps, user interfaces, and real-time systems since 2004 and mobile apps since the app stores opened in the late 2000s. After picking up Clojure in 2012, he left the frameworks and ORMs behind and started to build web apps from scratch and has shipped to production using Clojure, Groovy, Node.js, Elixir, and Kotlin. He has worked as a consultant for a decade and is now an independent contractor and startup founder.

About the Technical Reviewer



Andres Sacco has been a professional developer since 2007, working with a variety of languages, including Java, Scala, PHP, Node.js, and Kotlin. Most of his background is in Java and the libraries or frameworks associated with it, like Spring, JSE, iBATIS, Hibernate, and Spring Data. He is focused on researching new technologies to improve the performance, stability, and quality of the applications he develops.

In 2017 he started to find new ways to optimize the transference of data between applications to reduce the cost of infrastructure. He suggested some actions, some of them applicable in all the microservices and others in just a few of them. As a result of these actions, the cost was reduced by 55%. Some of these actions are connected directly with the bad use of the databases.

Acknowledgments

Thanks to my fiancée, Tina, who believes in me and lets me know. Thanks to Jonathan Gennick for giving me the opportunity to become a technical writer. Thanks to everyone else at Apress for being a pleasure to work with. Thanks to Marius Mårnes Mathisen for catapulting my career way back when, despite my complete lack of programming education and professional experience. Thanks to Kolbjørn Jetne for further catapulting my career, despite my complete lack of consulting experience. Thanks to Magnar Sveen, Christian Johansen, and Alf Kristian Støyle for all the discussions, input, and inspiration through the years. A special thanks to Finn Johnsen, who helped form most of the patterns in this book and was a vital fellow traveler in my initial quest for escaping the framework and bridging the gap between the old and the new.

Finally, thanks to my son, Tobias, and my daughter, Lone, who are awesome. And, at my daughter's request/demand, "Pappa er en bok."

Introduction

Welcome to *Pro Kotlin Web Apps from Scratch!* In this book, you'll learn how to build professional and production-grade web apps, completely from scratch, without the use of big and unwieldy frameworks.

My personal web app journey started with frameworks, but when I learned more about what goes on under the hood and grew weary of fighting framework bugs and limitations, I started teaching myself how to build web apps from scratch instead. As it turns out, frameworks aren't a requirement!

You're in good hands when you're building from scratch, thanks to modern programming languages like Kotlin and amazing third-party open source libraries. Back in the day, you needed thousands of lines of boilerplate and XML configuration to wire up a framework-less web app. No wonder people preferred frameworks! Nowadays, though, all you need is a couple of handfuls of explicit code, completely free of bloat and magic, that only does what you tell it to.

In Part I, you'll set up a web app skeleton, completely from scratch. This code base forms the basis for Part II, where you'll learn a handful of patterns and practical solutions that build on the skeleton from Part I. Part III covers how to choose the right library and Kotlin tips and tricks that I didn't get to cover in the preceding parts. Finally, three appendixes explain how to replace some of the libraries chosen in Parts I and II, to demonstrate that you have free rein to choose different libraries than me and still write pro Kotlin web apps from scratch.

I've deliberately kept the chapter about Kotlin tips and tricks in Part III as short as I've been able to manage. Instead, you'll learn Kotlin tips and tricks in Parts I and II, and you'll learn how to build web apps from scratch alongside explanations of the various Kotlin language constructs you're using.

CHAPTER 1

Setting Up a Development Environment

In this chapter you'll set up a development environment for Kotlin. You'll install IntelliJ IDEA, a Java Development Kit (JDK), you'll write a small Kotlin program, and you'll compile and run it using the Gradle build system.

This chapter explains how to do everything using IntelliJ IDEA. For the other chapters in this book, though, using IntelliJ IDEA is not a requirement. For example, both the Eclipse IDE (integrated development environment) and Visual Studio Code have Kotlin plugins available, made by the Kotlin community. If you choose to use another editor, make sure that it supports *auto-completion* and *automatic imports*. Extension functions are prevalent in Kotlin, and you must import them before you can use them. You'll avoid lots of tedious manual work if your editor can search and match all available extension functions (auto-completion) and automatically add an import statement when you select one of the auto-completed options (automatic imports).

Get Started with IntelliJ IDEA Community Edition

You'll need an editor to write your code and a way to compile and run your code. IntelliJ IDEA gives you both.

IntelliJ IDEA is the canonical integrated development environment (IDE) for static languages on the Java platform. The free community edition of IntelliJ IDEA has everything you need, both for this book and for production-grade Kotlin development. Additionally, JetBrains makes both Kotlin and IntelliJ IDEA, making IntelliJ IDEA uniquely well suited for Kotlin work.

Download and Run IntelliJ IDEA

You can download and use IntelliJ IDEA Community for free from JetBrains. Go to the IntelliJ IDEA download page, shown in Figure 1-1, at the address www.jetbrains.com/idea/download/ and download the community edition from there. JetBrains provides a .dmg file for macOS, an .exe file for Windows, and a tarball with a shell script in `bin/idea.sh` for Linux.

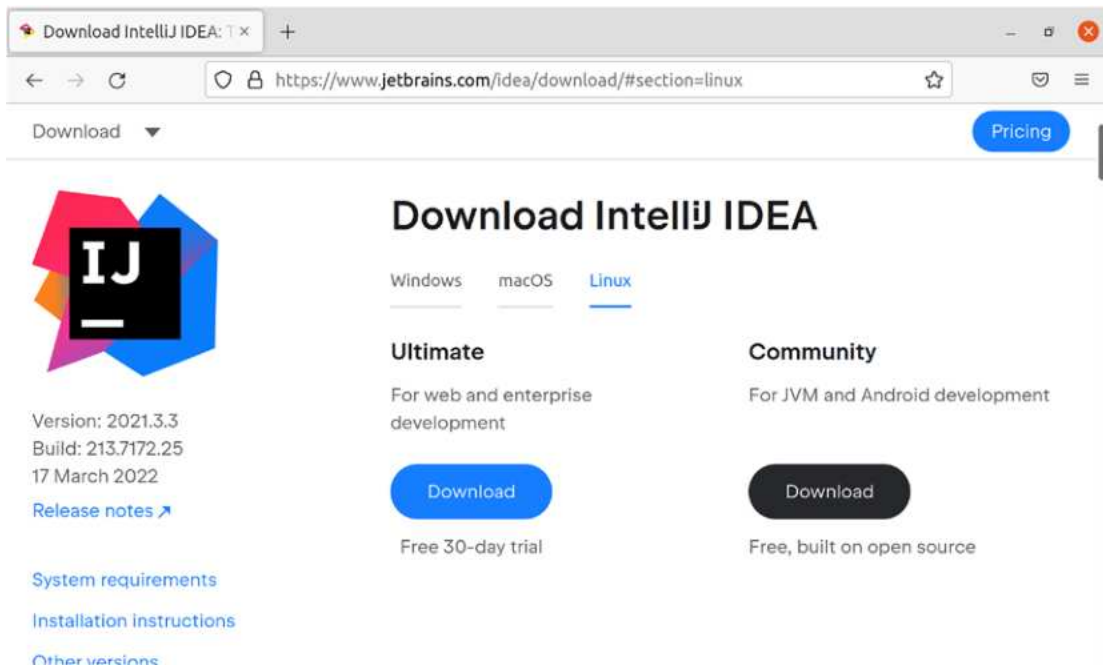


Figure 1-1. Download IntelliJ IDEA Community Edition for your platform

For more details on how to install and run IntelliJ IDEA, refer to the JetBrains website. For example, the download page has a link with installation instructions in the left sidebar, as seen in Figure 1-1.

Create a New Kotlin Project with IntelliJ IDEA

There are many ways to create a new Kotlin project. For this book, you'll be using IntelliJ IDEA to do it. It's nice to have a GUI to hold your hand through the process if you're not familiar with Kotlin and/or the JVM.

When you launch IntelliJ IDEA, you should see the splash screen shown in Figure 1-2.

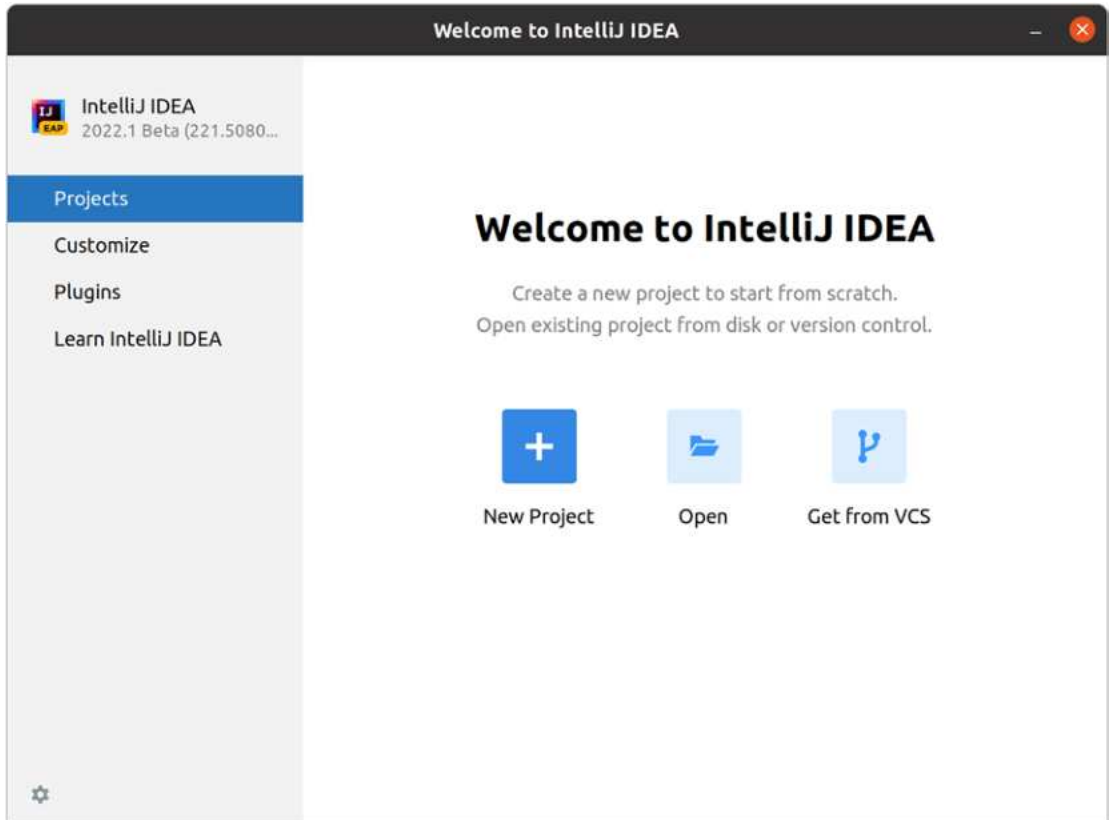


Figure 1-2. *Your first taste of IntelliJ IDEA*

Click “New Project” to create a new Kotlin project using the built-in wizard in IntelliJ IDEA, shown in Figure 1-3.

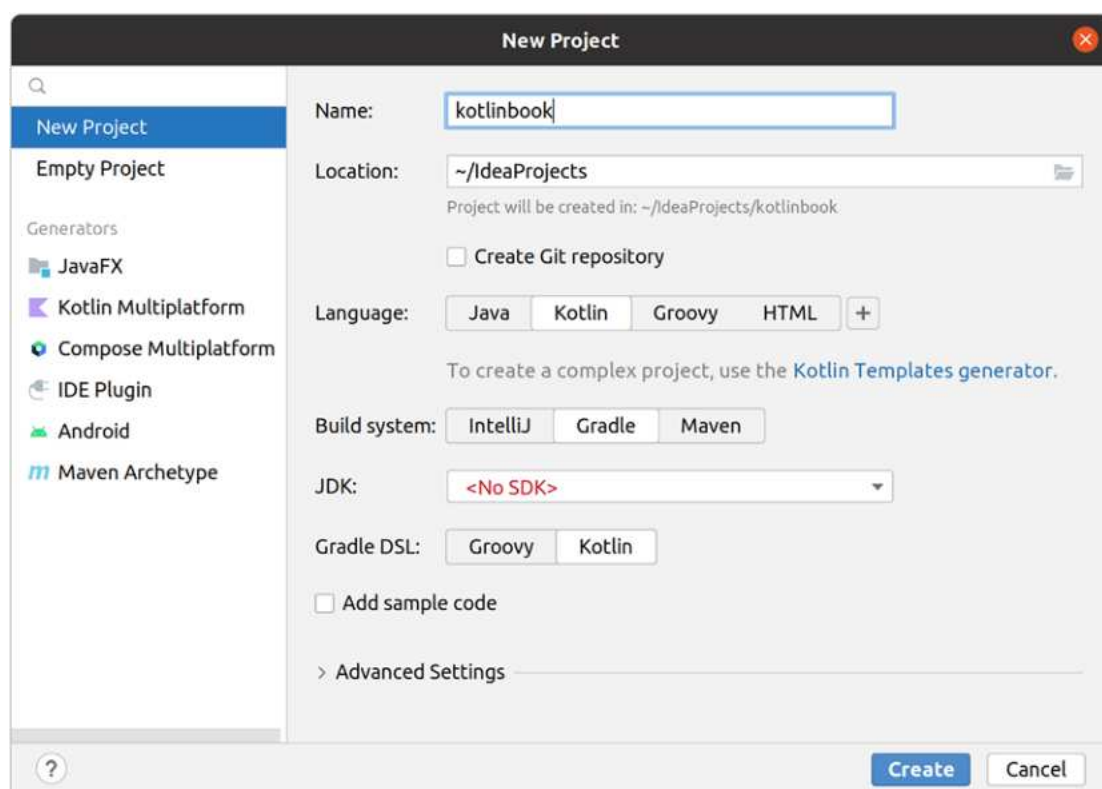


Figure 1-3. *The New Project screen in IntelliJ IDEA*

Use your good taste and judgment to name the project, and adjust the options as follows:

- Make sure you select “New Project” in the left sidebar.
- Set *Language* to *Kotlin*.
- Set *Build system* to *Gradle*.
- Set *Gradle DSL* to *Kotlin*.
- Uncheck *Add sample code*.

You need a JDK, and the easiest way to get one is to have IntelliJ IDEA download one for you. Click the red “<No SDK>” dropdown in the New Project dialog shown in Figure 1-3 and choose an appropriate JDK from the popup that appears, shown in Figure 1-4. JDK 17 or JDK 11 is a good choice, as those are stable Long-Term Support (LTS) releases.



Figure 1-4. Use IntelliJ IDEA to download a JDK

There are multiple vendors available. They are all full-fledged JDKs that adhere to the JDK specifications. For the most part, it doesn't matter which one you choose. The differences lie in a handful of edge cases and different defaults. For most web apps, you'll be able to switch between vendors and not notice any differences. I personally prefer Amazon Corretto, mostly out of habit. But I've used other vendors in real-world web apps, such as Azul Zulu. I tend to choose whichever vendor is most convenient on the platform I use, such as Azul Zulu on Azure, Amazon Corretto on AWS, and so on.

Make sure you choose a JDK of version 11 or newer. Versions 11 and 17 are both Long-Time Support (LTS) releases, which means they'll receive security updates longer than other versions. And later in this book, you'll use libraries that require at least version 11 to work.

You can also choose an existing JDK if you have one installed. Alternatively, you can manage JDK installations yourself, by using something like `sdkman` (<https://sdkman.io/>) on macOS and Linux and `Jabba` (<https://github.com/shyiko/jabba>) on Windows.

Tip You can have bad luck and choose a JDK version that's incompatible with the Gradle version used by IntelliJ IDEA. For example, the 2021 version of IntelliJ uses Gradle 7.1, which only works with JDK versions lower than 16. If this happens, delete your project and start over, using a different JDK or Gradle version.

After IntelliJ IDEA finishes downloading the JDK, click the "Create" button at the bottom of the New Project dialog shown in Figure 1-3, and your brand-new project appears, as shown in Figure 1-5.

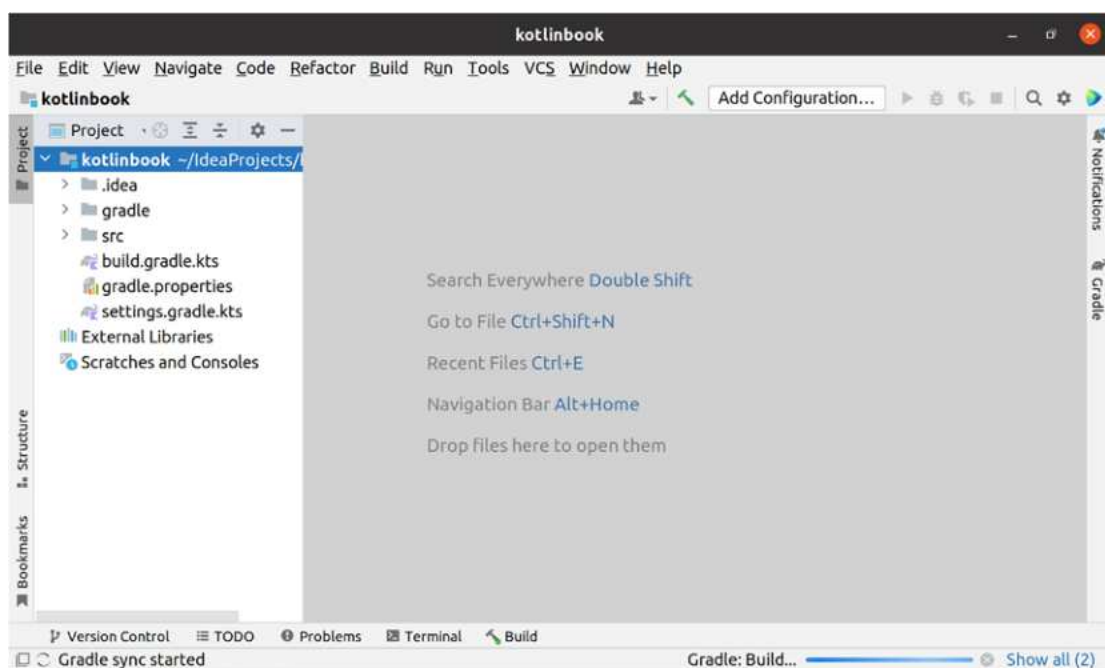


Figure 1-5. IntelliJ IDEA after it has finished with project initialization

With this setup, IntelliJ IDEA won't create sample code or pre-generated files, other than a Gradle build system skeleton. All you have now is a way to compile and run Kotlin code. And that's all you need in this book. There's no framework with tens of thousands of lines of code that needs to run first to get anything done.

Tip Gradle is a build system for the Java platform. This is where you add third-party code as dependencies, configure production builds, set up automated testing, and more. Maven is also an excellent choice for building production-grade Kotlin web apps. Most examples you'll see in books and online documentation use Gradle, though, and I've only used Gradle in real-world Kotlin apps, so that's why I use it in this book.



Lituz.com

**To'liq qismini
Shu tugmani
bosish orqali
sotib oling!**