Installation instructions (to be completed before the workshop)

For this workshop, you will need:

- Weather API key (**time sensitive: it may take some time to activate**)
- Android Studio **Jellyfish 2023.3.1** installed & configured.
- JDK **17** installed & configured
- Cloned repository
- Additional iOS setup (macOS only)
  - Xcode 15 installed & configured. (I have Xcode 15.3)
  - KDoctor installed and checked
- **Setup verification done**

Please see sections below for details.

Questions?

Email: pamela.hill@jetbrains.com

Time-critical setup instructions (do **ASAP**):

1. Sign up for the Weather API.
   a. Navigate to https://home.openweathermap.org/users/sign_up.
   b. Fill in the form with your username, email address (the API key will be sent here), and password. Check the appropriate checkboxes and click on “Create account”.
   c. Next you will need to fill in the “How and where will you use our API?” dialog. I used “Education/Science” as “Purpose” and clicked the “Save” button.
   d. Go to your email and locate the OpenWeatherMap email that was sent to you. Click on the “Verify your email” button.
   e. The next email you will receive is the “API Instruction” in which you will receive your API key. **It may take some time to activate this key.**

Everyone: Android Studio setup

The instructions below describe the minimum setup to be able to run Android/Desktop. Please follow these instructions for macOS, Windows or Linux.

If you have a Mac and want to make sure the iOS components are set up properly, please also follow the Additional iOS setup section below. If not, you can skip that section and run only on Android/Desktop.

You can ignore all instructions related to Xcode and KDoctor if you’re not planning on running the app on iOS.

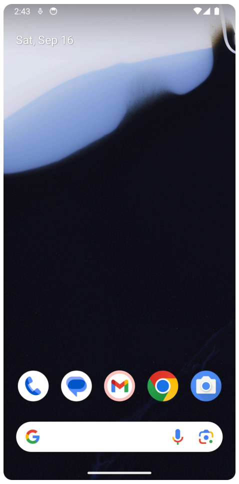
Pre-steps/knowledge:

1. You will need to install Android Studio **Jellyfish 2023.3.1**.
2. You will need to install the JDK **17** or higher (I used 18.0.2)**.**
3. If you're not sure whether your Mac has an Intel or Apple Silicon processor, follow the guide here.

Instructions:

1. Navigate to https://kotlinlang.org/docs/multiplatform-mobile-setup.html and follow the installation instructions.
2. In Android Studio, get the workshop repository from GitHub.
   - a. Click on the "Get from VCS" button (assuming no other projects are open).
   - b. In the "Get from Version Control" screen:
     - i. Set "URL" to: https://github.com/pahill/kmp-btb-workshop
     - ii. Set "Directory" to where you want the project to be saved (and take note where you're saving to).
     - iii. Click on the "Clone" button.
     - iv. Note this may take some time.

4. In Android Studio, set up an emulator with API 24 or above. Assuming you have a project open:
   - a. Navigate to "Tools" -> "Device Manager"
   - b. In the "Device Manager" panel, click on "+" for adding a new device.
   - c. In the dropdown that appears, select "Create Virtual Device".
   - d. In the "Select Hardware" dialog, select "Phone" for Category, and any phone that has the Play Store installed (the arrow icon ▷). I used Pixel 8. Click the "Next" button.
   - e. In the "System Image" panel, click on the download button next to the release name at the top of the list. The "SDK Component Installer" dialog box will appear and install the correct system image. I used "VanillaIceCream". This may take some time to download and install. Click the "Finish" button.
   - f. Back in the "System Image" screen, click the "Next" button.
   - g. In the "Android Virtual Device" screen, give your emulator a name in the "AVD Name" box, and click the "Finish" button.
   - h. To verify that you have set up the emulator correctly:
     - i. Go to "Tools" -> "Device Manager" again.
     - ii. Click the launch button (▶) corresponding to the emulator you created.
     - iii. The device will then boot up, and you'll be taken to the home screen. Your home screen might look something like this:



Additional iOS setup

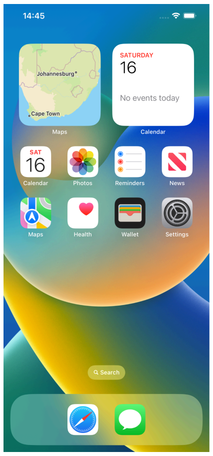
After you have installed and set up Android studio as described above, follow the instructions below to make sure iOS components are set up properly. Note: this is for macOS users only.

Pre-steps/knowledge:

1. If prompted to update your Ruby, follow the instructions here. I used RVM.
2. Select the default devices when prompted by Xcode what you want to develop for, like iOS and macOS. **Do not select visionOS.**

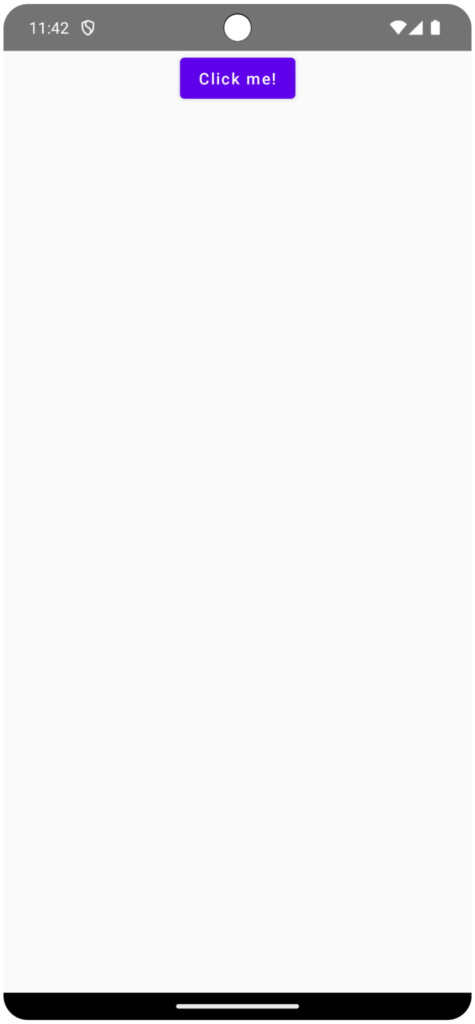
Installation steps:

1. Navigate to https://kotlinlang.org/docs/multiplatform-mobile-setup.html and follow the installation instructions, also follow the Xcode and KDoctor instructions. You may ignore KDoctor's warnings regarding the CocoaPods installation, we'll be using Direct Integration for simplicity.
2. In XCode, set up a simulator.
   a. Follow these instructions.
   b. Still in XCode, go to the menu “Xcode” -> “Open Developer Tool” -> “Simulator”.
   c. In Simulator, go to the menu “File” -> “Open Simulator”, then tick the simulator you'd like to launch. The simulator will boot and launch.
   d. Your home screen might look something like this:

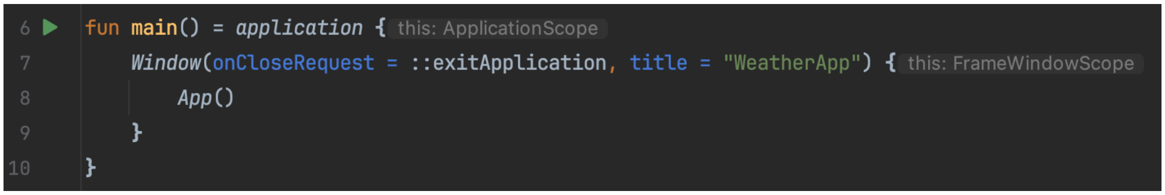


Everyone: Workspace verification

1. In Android Studio, navigate to “File” -> “Open”. In the file dialog, navigate to your local repository and then to the subdirectory /workspace-verification. The project should be compilable and runnable, but may need some time to finish syncing.
2. To run the project on an Android emulator, ensure that “composeApp” is selected in the “Run configurations” dropdown, that the emulator you created in the installation steps is selected, and click on the “Run composeApp” button (to the right). The app should start and the following screen should appear.

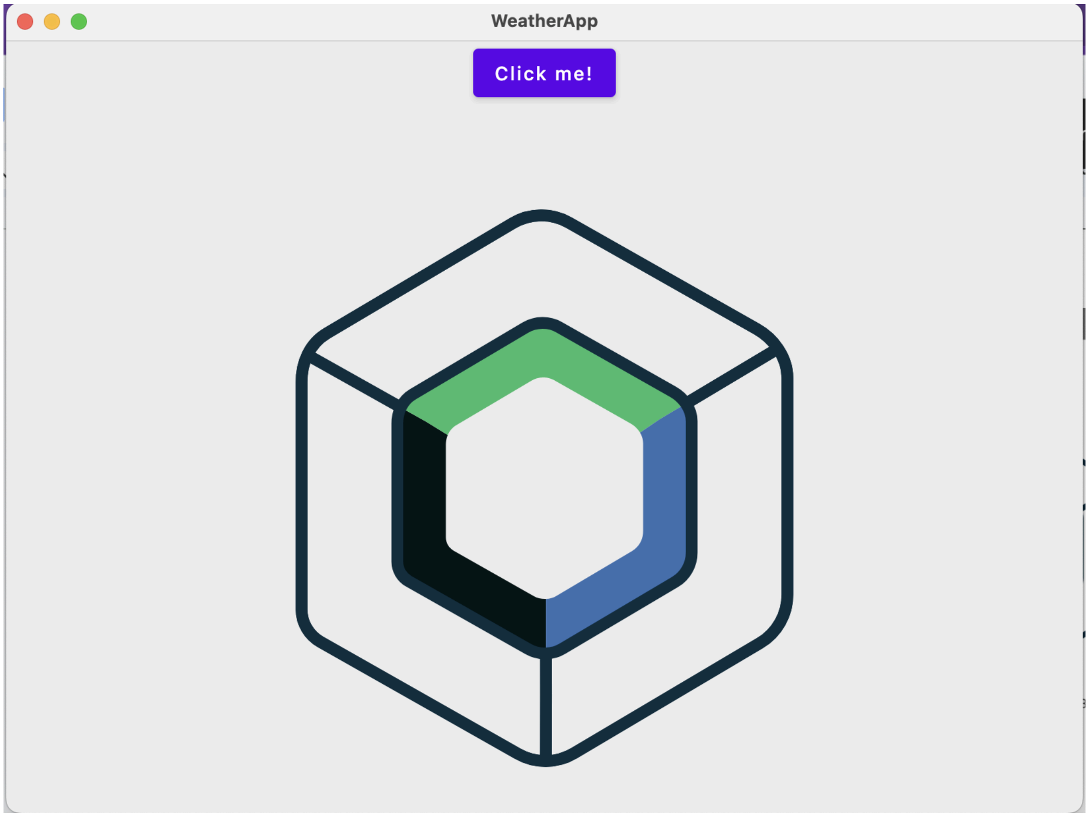


3. To run the project on Desktop, navigate to composeApp/src/desktopMain/kotlin/main.kt. In the code editor, there should be a green play gutter icon to run the application. Press the green gutter icon.

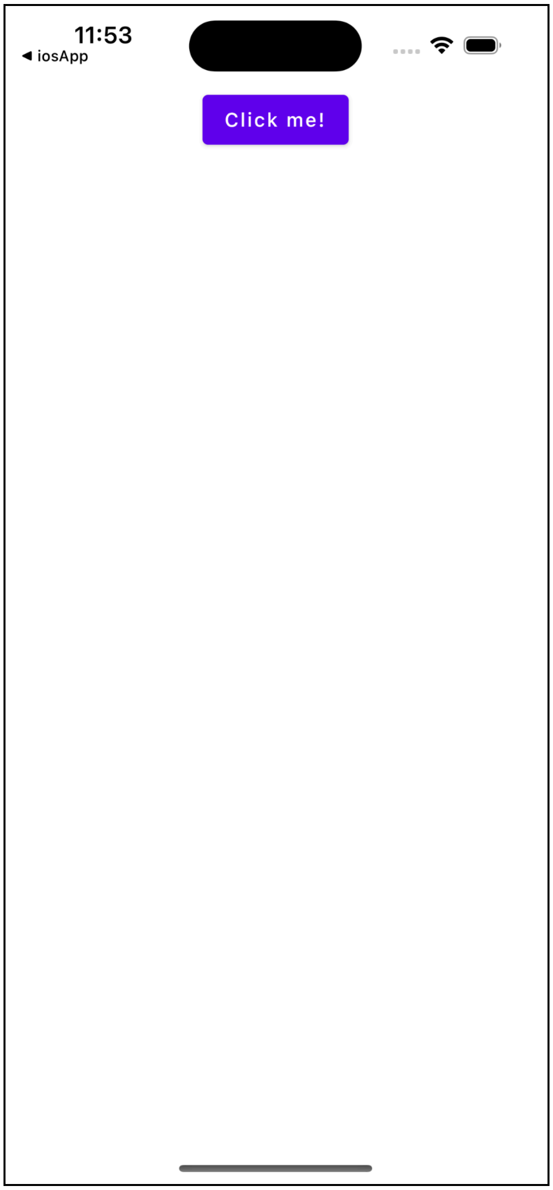


```kotlin
fun main() = application {
    Window(onCloseRequest = ::exitApplication, title = "WeatherApp") {
        App()
    }
}
```

The app should start and the following screen should appear as a desktop app.



4. **Close the desktop application before proceeding to the next step. There seems to be a bug in Android Studio that requires this action.**
5. To run the project on an iPhone simulator, ensure that “iosApp” is selected in the “Run configurations” dropdown, and click on the “Run ‘iosApp’” button (to the right). The app should start and the following screen should appear.



Troubleshooting:

- It might be that your iOS simulator isn't selected correctly in Android Studio.
  - Click on the drop down menu arrow next to iosApp.
  - Click on "Edit Configurations".
  - Check that the "Execution target" is the right simulator.
- If there is no gutter icon for your desktop application/you get an error, you can run it manually on the command-line using: ./gradlew composeApp:run.