Installation instructions (to be completed before the workshop)

For this workshop, you will need:

- Latest stable Android Studio installed & configured (I have Meerkat 2024.3.1)
- JDK **17** installed & configured
- Cloned repository (please pull again just before the workshop)
- Additional iOS setup (macOS only)
    - Xcode 16 installed & configured. (I have Xcode 16.2, **note that there is currently an issue with Xcode 16.3 and KMP/CMP**)
    - KDoctor installed and checked
- **Setup verification done**

Please see sections below for details.

Questions?

Email: pamela.hill@jetbrains.com

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 iPhone 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 your iPhone. Due to the nature of the app (using the camera), you cannot run the app on an iPhone simulator.

Pre-steps/knowledge:

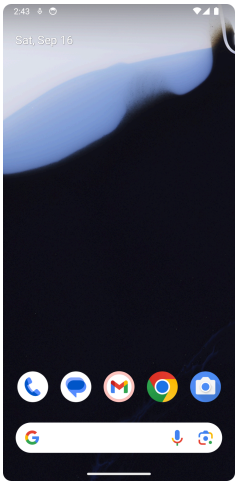
1. You will need to install the latest stable Android Studio.
2. You will need to install the JDK **17** or higher (I used 18.0.2)**.**
3. If you're not sure whether your Mac has an Intel or Apple Silicon processor, follow the guide here.

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/deep-dive-into-kmp.git
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” for API 35. 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.

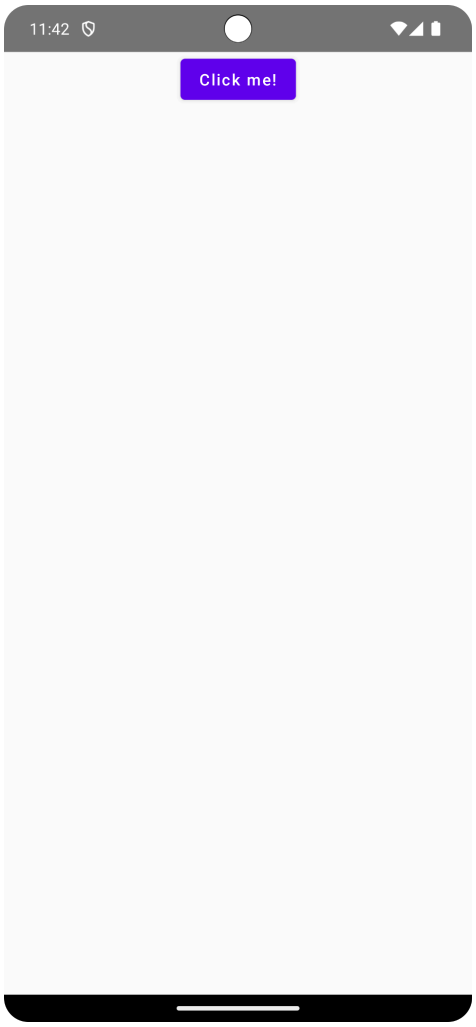
Installation steps:

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. Set up your phone:
    a. Enable developer mode on your iPhone by following these instructions.
    b. Set up Xcode:
        i. Add your Apple Account to the Accounts settings in Xcode:
            1. Choose Xcode > Settings to open Xcode's preferences window.
            2. Click the Accounts button at the top of the preferences window.
            3. Click the Add (+) button at the bottom of the list of accounts.
            4. Select Apple ID from the list of account types.
            5. Click the Continue button.
            6. Enter the email of your Apple ID.
            7. Click the Next button.
            8. Enter the password for your Apple ID.
            9. Click the Next button.
            10. A sheet will open with the title Apple ID Security.
            11. Click the Continue button to set up two factor authentication for signing in with this account.
        ii. Choose a valid team in your project's Signing & Capabilities pane:
            1. In the right-hand pane, click on "iosApp"
            2. Then select "iosApp" under Targets.
            3. Click on the "Signing & Capabilities" button to open the pane.
            4. Select your account under Team. In my case it says Pamela Hill (Personal Team).
    c. To pair a device with a physical connection:
        i. Connect the device to your Mac using **an appropriate cable**.
        ii. Unlock the device and follow any instructions that appear in Xcode or on the device to trust the computer/device.

Everyone: Workspace verification

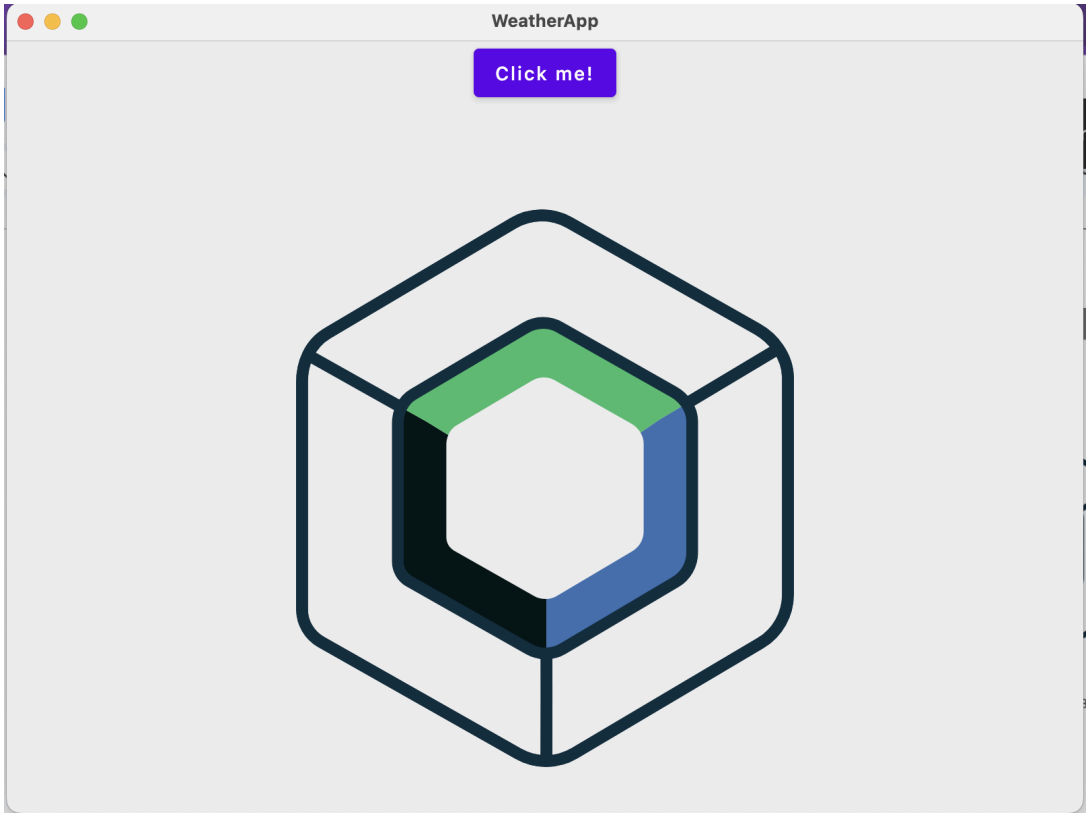
1. In Android Studio, navigate to "File" -> "Open". In the file dialog, navigate to your local repository and then to the subdirectory /workspace-verification. The project should be compilable and runnable, but may need some time to finish syncing.

2. To run the project on an Android emulator, ensure that “WorkspaceVerification.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.

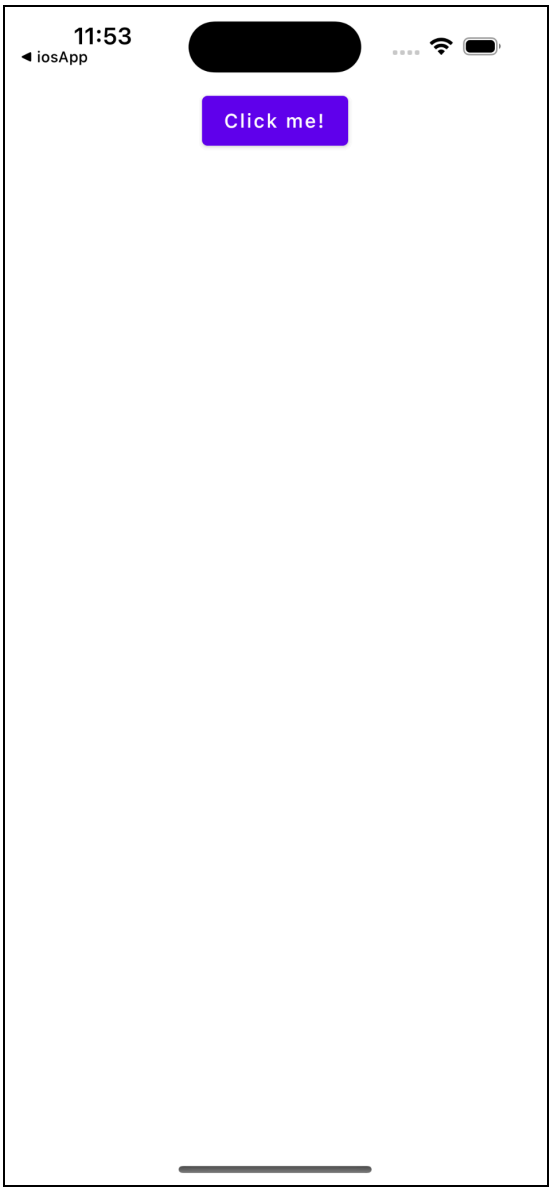


3. To run the project on Desktop, open the terminal in Android Studio, then execute the following command:
   `./gradlew composeApp:run`

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, 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:

- You may have to select your Xcode account as team in the Xcode project.
    1. Open the project in Xcode from Android Studio:
        a. Right-clicking on the .xcodeproj file.
        b. Click on the “Open In” menu option.
        c. Select Xcode.
    2. Setting the team in the Xcode project:
        a. In the right-hand pane, click on “iosApp”.
        b. Then select “iosApp” under Targets.
        c. Click on the “Signing & Capabilities” button to open the pane.
        d. Select your account under Team. In my case it says Pamela Hill (Personal Team).

- Make sure your physical device is unlocked, and listed as Execution Target in Android Studio.