مدينة زويل للعلوم والتكنولوجيا
Zewail City of Science and Technology

Pre-Session Manual for Mobile App Development Session - CSAI Workshop 2023

Sara: I created this manual to help you prepare for the Mobile App development session. You'll find below an index of the contents of this file. You'll need to install Flutter and all its requirements first. Try running the hello world app that comes pre-built with it to make sure your environment is all set. Or otherwise make sure that the output to the command `flutter doctor` does not give you any errors or warnings.

Flutter's documentation is pretty awesome, so hopefully you won't run into any problems following it. There's an extra section that I added for installing Flutter *without Android Studio*, if you're a bit short on disk storage, which is not included in the official documentation so I prepared it myself. If you run into any problems following this section, please do get in touch with me before the session at t-sarahahmed@zewailcity.edu.eg. I'll be happy to help.

Index:

Setting Up Flutter

- For installing Flutter on any operating system, go to the official Flutter documentation, choose your operating system and follow the instructions.

Setting Up Flutter on VSCode (without installing Android Studio)

1. Install Flutter SDK

- Download the Flutter SDK from the official Flutter website.
- Extract the downloaded SDK archive to a desired location on your computer.
- Add the Flutter SDK to your system's PATH variable. This allows you to run Flutter commands from anywhere in the terminal/command prompt.
- Verify the installation by opening a new terminal/command prompt window and running the command: `flutter doctor`. It should display a checklist with any missing dependencies or configurations.

2. Install Visual Studio Code (VSCode)

- Download and install Visual Studio Code from the official VSCode website.
- Open VSCode and install the Flutter and Dart extensions. Go to the Extensions view (Ctrl+Shift+X), search for "Flutter" and "Dart", and install the respective extensions provided by the Dart Code team.

3. Set up Flutter in VSCode

- Open VSCode and navigate to the settings by going to File > Preferences > Settings or using the shortcut Ctrl+, (comma).
- In the settings, search for "flutter.sdkPath".
- Click on "Edit in settings.json" to modify the user settings.
- Add the following line to the settings.json file, replacing `<flutter-sdk-path>` with the path to your Flutter SDK:

  ```
  "dart.flutterSdkPath": "<flutter-sdk-path>"
  ```

- Save the file and close the settings.

4. Create a new Flutter project

- Open a terminal or command prompt window.
- Navigate to the directory where you want to create your Flutter project.
- Run the following command to create a new Flutter project:

  ```
  flutter create my_app
  ```

  (Replace "my_app" with the desired name for your project.)

5. Open the Flutter project in VSCode

- Open VSCode.

- Go to File > Open Folder (Ctrl+K Ctrl+O) and select the folder of your Flutter project.
- VSCode will analyze the project and install any missing dependencies automatically.

6. Configure and run the Flutter project

- Connect a physical device or set up an emulator for running the Flutter project.
- Open the terminal in VSCode by going to View > Terminal (Ctrl+`).
- Run the following command to check if the Flutter project can be run:

```
flutter run
```

- VSCode will detect the connected device/emulator and build the project. The app should launch on the device/emulator if everything is set up correctly.

7. Enable Developer Options on your Android device

- Go to the "Settings" app on your Android device.
- Scroll down and tap on "About phone" or "About device."
- Look for the "Build number" or "Build version" entry.
- Tap on the "Build number" entry seven times consecutively. You should see a message indicating that Developer Options are enabled.

8. Enable USB Debugging on your Android device

- Go back to the main "Settings" screen.
- Scroll down and tap on "Developer options" or "Developer settings."
- Toggle the "Developer options" switch at the top of the screen to enable it.
- Scroll down and locate the "USB debugging" option.
- Enable the "USB debugging" option by toggling the switch next to it.

9. Connect your Android device to your computer

- Use a USB cable to connect your Android device to your computer.
- Ensure that your device is unlocked and the screen is active.

10. Verify device connection

- Open a terminal or command prompt window on your computer.
- Run the following command to verify that your device is detected by Flutter:

```
flutter devices
```

- You should see your connected device listed in the output.

11. Run the Flutter project on your device

- In VSCode, make sure you have your Flutter project open.
- Open the terminal in VSCode by going to View > Terminal (Ctrl+`).
- Run the following command to build and run your Flutter project on your connected device:

```
flutter run
```

- Flutter will build the project and install it on your device.
- If prompted, grant any necessary permissions on your device to run the app.

12. Test and debug your Flutter app

- Once the app is installed on your device, you can interact with it as you would with any other app.
- If you encounter any issues or want to debug your app, you can use the debugging tools provided by VSCode and Flutter. Set breakpoints, inspect variables, and use the debugging features to analyze and fix any problems.

By following these steps, you should be able to set up Flutter, run it on VSCode, and utilize USB debugging on your mobile devices. Enjoy developing and testing your Flutter applications using this setup!

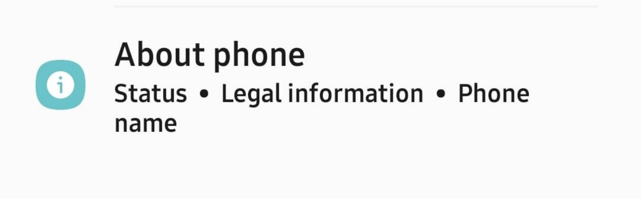
Setting Up your Phone for USB Debugging

To be able to run apps on your Android phone, you'll need to set up USB debugging, which basically means giving permission to your laptop to build and install apps on your phone through a USB cable. The following instructions would only work for an Android phone.

To set up USB debugging on your Android phone, you'll need to enable Developer Options and then enable USB debugging within those options. Here's a step-by-step guide:

1. **Enable Developer Options:**

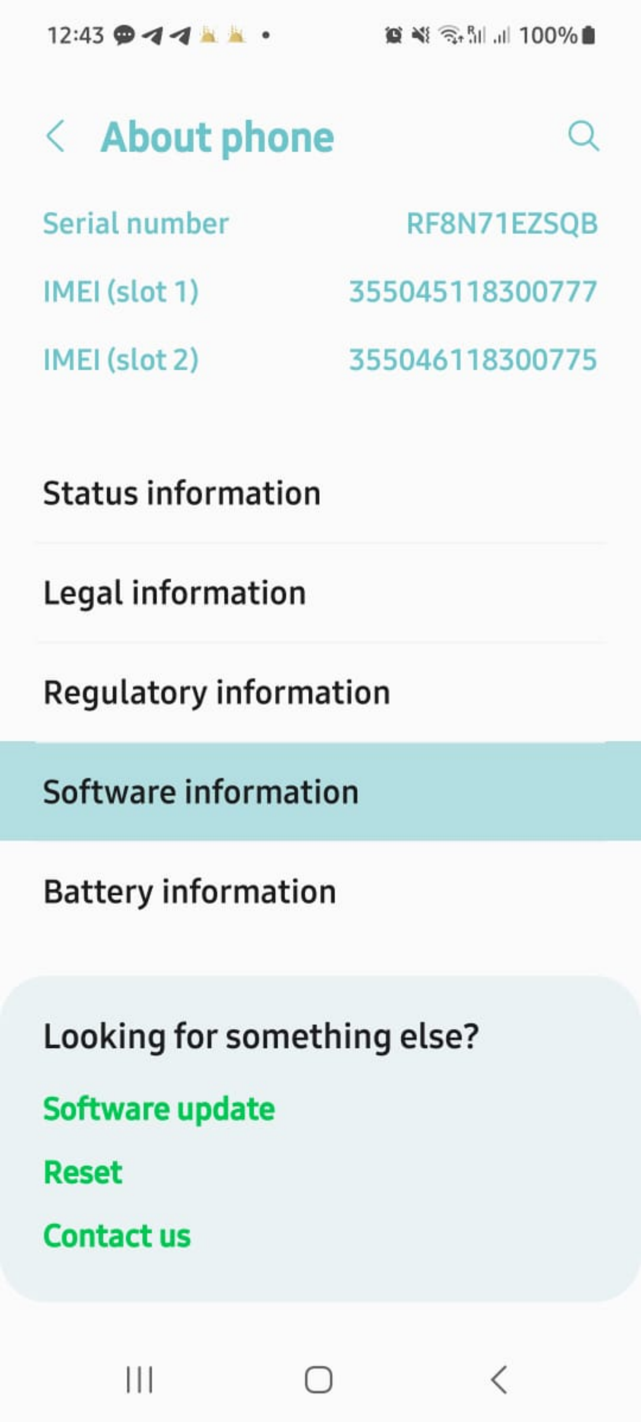
    - Go to the "Settings" app on your Android phone.
    - Scroll down and look for "About phone" or "About device" (the exact name may vary depending on your device).



About phone

Status • Legal information • Phone name

- Tap on "About phone" and locate the "Build number" or "Build version" entry.



12:43 100%

About phone

Serial number RF8N71EZSQB

IMEI (slot 1) 355045118300777

IMEI (slot 2) 355046118300775

Status information

Legal information

Regulatory information

Software information

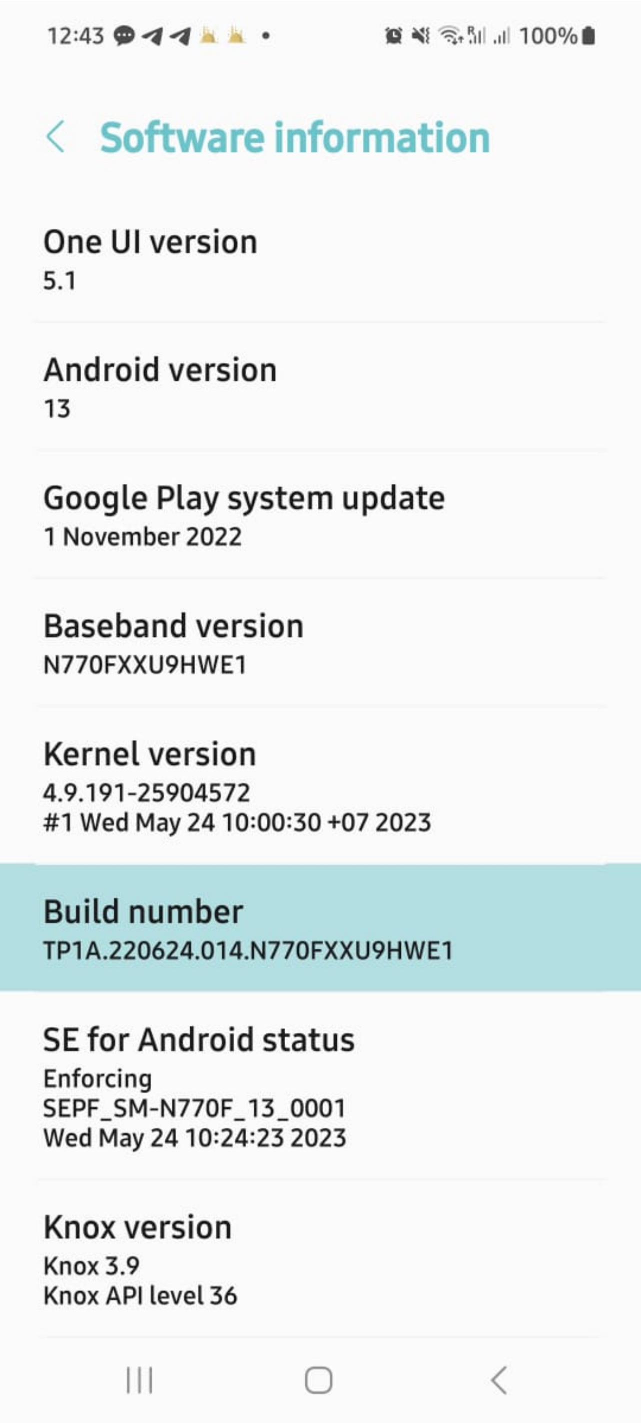
Battery information

Looking for something else?

Software update

Reset

Contact us



12:43 100%

Software information

One UI version
5.1

Android version
13

Google Play system update
1 November 2022

Baseband version
N770FXXU9HWE1

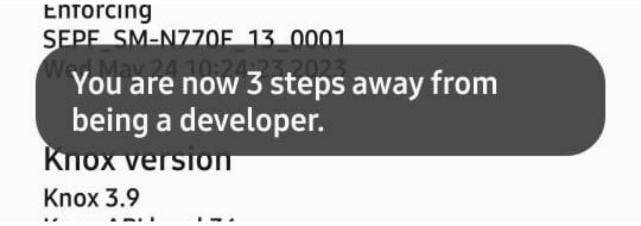
Kernel version
4.9.191-25904572
#1 Wed May 24 10:00:30 +07 2023

Build number
TP1A.220624.014.N770FXXU9HWE1

SE for Android status
Enforcing
SEPF_SM-N770F_13_0001
Wed May 24 10:24:23 2023

Knox version
Knox 3.9
Knox API level 36

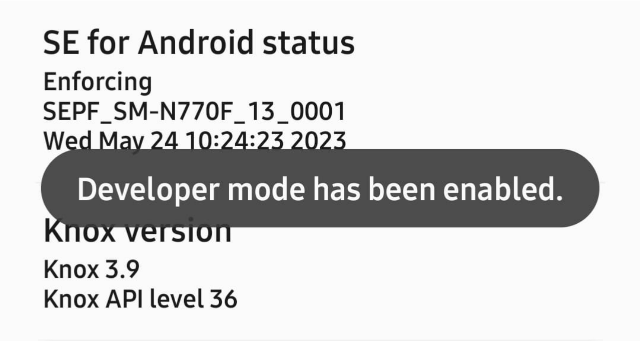
- Tap on the "Build number" entry multiple times (typically seven times) until you see a message indicating that you are now a developer.



Enforcing
SEPF_SM-N770F_13_0001

You are now 3 steps away from being a developer.

Knox version
Knox 3.9



SE for Android status

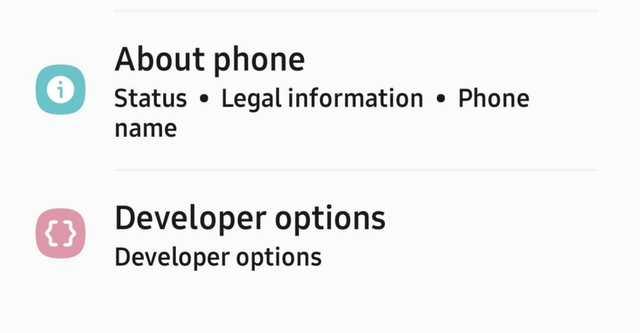
Enforcing
SEPF_SM-N770F_13_0001
Wed May 24 10:24:23 2023

Developer mode has been enabled.

Knox version

Knox 3.9
Knox API level 36

2. **Enable USB Debugging:**
   - Go back to the main settings menu.
   - You should now see a new option called "Developer Options" or "Developer settings" listed. Tap on it.



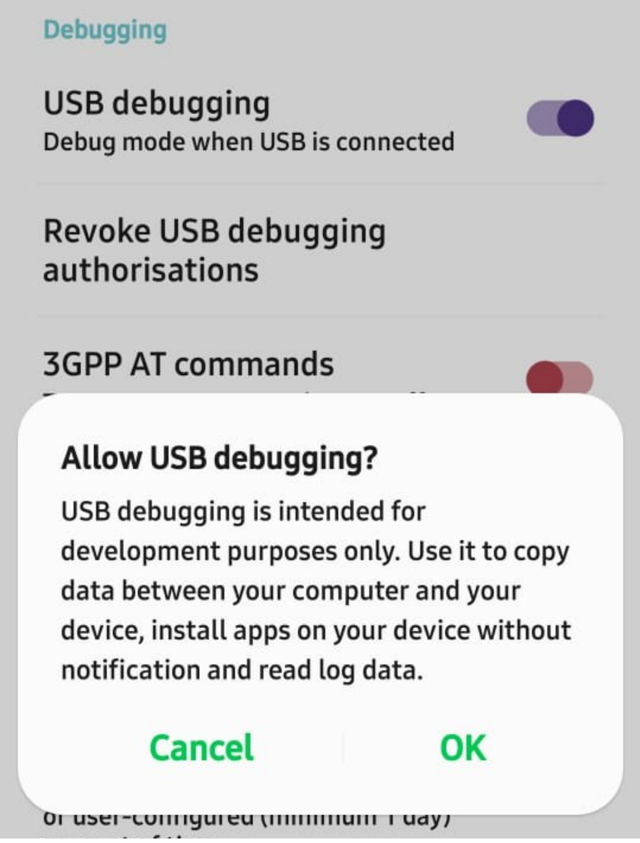
About phone

Status • Legal information • Phone name

Developer options

Developer options

   - Look for the "USB debugging" option and toggle it on.



Debugging

USB debugging

Debug mode when USB is connected

Revoke USB debugging authorisations

3GPP AT commands

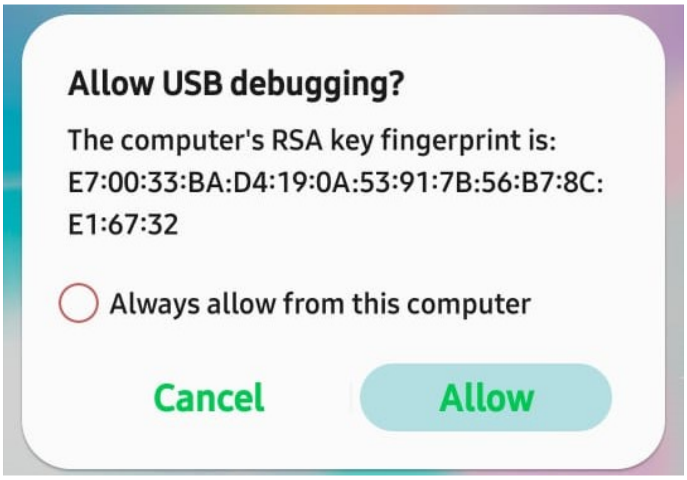
Allow USB debugging?

USB debugging is intended for development purposes only. Use it to copy data between your computer and your device, install apps on your device without notification and read log data.

or user-configured (minimum 1 day)

   - You may see a warning message explaining the risks of enabling USB debugging. Read it carefully, and if you agree, select "OK" or "Allow" to proceed.

3. **Connect your Phone to your Computer:**
   - Use a USB cable to connect your Android phone to your computer.
   - If prompted on your phone, choose the "File Transfer" or "Transfer files" option.
   - On your computer, make sure you have the appropriate USB drivers installed for your Android phone. Most devices will automatically install the necessary drivers, but if not, you can download them from your phone manufacturer's website.



Allow USB debugging?

The computer's RSA key fingerprint is:
E7:00:33:BA:D4:19:0A:53:91:7B:56:B7:8C:
E1:67:32

○ Always allow from this computer

4. **Verify USB Debugging:**
   - Open a command prompt or terminal window and navigate to the location where you extracted the platform tools.
   - Type the command `adb devices` or `flutter devices` and press Enter.
   - If everything is set up correctly, you should see your device listed as a connected device with a device ID.

That's it! USB debugging is now enabled on your Android phone, and you can use it for various development tasks, such as deploying and debugging apps from your computer.

Session Project

To prepare for the hands-on exercise before the session:

1. 💻 Clone this github repo.

2. Explore the code inside the `app` folder!