hideX - Complete Setup Guide

This guide will walk you through setting up and running the hideX app in Android Studio.

Prerequisites

Before you begin, ensure you have:

1. **Android Studio** - Download from https://developer.android.com/studio
   - Recommended version: Android Studio Hedgehog (2023.1.1) or later

2. **Java Development Kit (JDK)**
   - JDK 17 is required
   - Usually comes bundled with Android Studio

3. **Android SDK**
   - SDK 34 (Android 14) must be installed
   - Android Studio will prompt you to install it if missing

4. **Android Device or Emulator**
   - Physical device: Android 8.0 (API 26) or higher
   - Emulator: Create one with API 26-34

Step 1: Open the Project

Option A: Open in Android Studio

1. Launch Android Studio
2. Click **“Open”** on the welcome screen
3. Navigate to `/home/ubuntu/hideX_app`
4. Click **“OK”**

Option B: Import from Command Line

```
cd /home/ubuntu/hideX_app
studio . # If Android Studio is in PATH
```

Step 2: Wait for Gradle Sync

When you open the project:

1. Android Studio will automatically start syncing Gradle
2. You’ll see progress in the bottom status bar: “Gradle sync in progress…”
3. This may take 2-10 minutes depending on your internet speed
4. If prompted to update Gradle or plugins, you can choose to update or keep current versions

Troubleshooting Gradle Sync

If sync fails:

Problem: “SDK location not found”

```
# Create local.properties file
echo "sdk.dir=/path/to/Android/Sdk" > local.properties

# Common SDK locations:
# Linux: /home/username/Android/Sdk
# macOS: /Users/username/Library/Android/sdk
# Windows: C:\\Users\\username\\AppData\\Local\\Android\\Sdk
```

Problem: “Failed to download Gradle”
- Check your internet connection
- Try using a VPN if behind a firewall
- Manually download Gradle 8.2 and place in `~/.gradle/wrapper/dists/`

Problem: “Compilation error with Kotlin”
- Invalidate caches: **File → Invalidate Caches → Invalidate and Restart**

Step 3: Install Required SDK Components

If Android Studio shows a warning about missing SDK components:

1. Click the **“Install missing components”** link
2. Or go to: **Tools → SDK Manager**
3. Ensure these are installed:
   - ✅ Android SDK Platform 34
   - ✅ Android SDK Build-Tools 34.0.0
   - ✅ Android Emulator
   - ✅ Android SDK Platform-Tools

Step 4: Create App Icons (Optional but Recommended)

The app currently has placeholder icons. To create proper icons:

Using Android Studio Image Asset Studio

1. Right-click on `res` folder
2. Select **New → Image Asset**
3. Choose **Launcher Icons (Adaptive and Legacy)**
4. Upload your icon or use the built-in icon library
5. Suggested: Use a lock icon or stylized “X”
6. Click **Next → Finish**

Using Online Tools

1. Visit: https://romannurik.github.io/AndroidAssetStudio/icons-launcher.html
2. Design your icon with black/white theme
3. Download the generated assets
4. Replace the contents of `app/src/main/res/mipmap-*` folders

Step 5: Build the Project

Before running, build the project to ensure everything compiles:

1. Click **Build → Make Project** (or press `Ctrl+F9` / `Cmd+F9`)
2. Wait for the build to complete
3. Check the **Build** tab at the bottom for any errors

Common Build Errors

Error: “Cannot resolve symbol ‘R’“
- Solution: **Build → Clean Project**, then **Build → Rebuild Project**

Error: “Duplicate class found”
- Solution: **File → Invalidate Caches → Invalidate and Restart**

Error: “OutOfMemoryError”
- Solution: Increase heap size in `gradle.properties` :
`org.gradle.jvmargs=-Xmx4096m`

Step 6: Set Up a Device

Option A: Use a Physical Device (Recommended)

1. **Enable Developer Options** on your Android device:
   - Go to **Settings → About Phone**
   - Tap **Build Number** 7 times
   - You’ll see “You are now a developer!”

2. **Enable USB Debugging**:
   - Go to **Settings → System → Developer Options**
   - Toggle **USB Debugging** ON

3. **Connect Device**:
   - Connect your device via USB cable
   - Allow USB debugging when prompted on device
   - Device should appear in Android Studio’s device dropdown

Option B: Use an Emulator

1. Click **Device Manager** in Android Studio (phone icon in toolbar)
2. Click **Create Device**
3. Choose a device (e.g., Pixel 6)
4. Select a system image (API 30-34 recommended)
5. Download the system image if needed
6. Click **Finish**
7. Start the emulator by clicking the play button

Emulator Tips:
- Use x86_64 images for better performance
- Allocate at least 2GB RAM to emulator
- Enable hardware acceleration in BIOS if slow

Step 7: Run the App

1. Select your device/emulator from the device dropdown in the toolbar
2. Click the **Run** button (green play icon) or press `Shift+F10`
3. Wait for the app to install and launch

First Launch

When the app launches for the first time:

1. You'll see the **"Create Password"** screen
2. Enter a password (minimum 4 characters)
3. Confirm the password
4. Click **"Create Password"**

Step 8: Enable Accessibility Service

This is CRITICAL for the app to work!

After creating your password:

1. You'll see a red warning banner: **"Service Disabled"**
2. Tap on the warning banner
3. You'll be taken to **Accessibility Settings**
4. Find **"hideX App Protection"** in the list
5. Tap on it
6. Toggle the switch to **ON**
7. Accept the permission warning
8. Go back to the hideX app

The red warning should now disappear.

If Accessibility Settings Don't Open

If the settings don't open automatically:

1. Open your device **Settings**
2. Navigate to: **Settings → Accessibility**
3. Scroll down to **Downloaded apps** or **Services** section
4. Find **"hideX App Protection"**
5. Enable it

Step 9: Add Apps to Protect

Now you can start protecting apps:

1. Tap the **"+"** button (Floating Action Button)
2. You'll see a list of all installed apps
3. **Select apps** you want to protect (e.g., WhatsApp, Gallery, etc.)
4. Tap the **checkmark** icon at the top right
5. Selected apps will appear on the main screen

Testing App Protection

1. Exit the hideX app (press Home button)
2. Try to open a protected app
3. The password overlay should appear immediately
4. Enter your password to unlock the app

Step 10: Configure Settings (Optional)

Tap the **Settings** icon in the top right to:

Enable Biometric Authentication

If your device has fingerprint/face unlock:

1. Go to **Settings** in hideX
2. Toggle **“Biometric Authentication”** ON
3. Now you can use fingerprint instead of password

Change Password

1. Go to **Settings → Change Password**
2. Enter current password
3. Enter new password
4. Confirm new password
5. Tap **“Change”**

Troubleshooting

Password Overlay Not Appearing

Symptoms: Protected apps open without asking for password

Solutions:
1. ✅ Verify Accessibility Service is enabled
2. ✅ Try disabling and re-enabling the service
3. ✅ Reboot the device
4. ✅ Clear app data and set up again: **Settings → Apps → hideX → Storage → Clear Data**

App Crashes on Launch

Check Logcat for error details:

1. In Android Studio, click **Logcat** tab at the bottom
2. Select your device
3. Filter by package: `com.hidex.app`
4. Look for red error lines

Common solutions:
- Clear app data
- Reinstall the app
- Check if Room database migration is needed

Biometric Not Working

Symptoms: Fingerprint option doesn’t appear

Requirements:
- Device must have biometric hardware
- Biometric must be set up in device settings
- Enable biometric in hideX settings

Build Errors

“Cannot find symbol” errors:

```
# In terminal:
./gradlew clean build --refresh-dependencies
```

Gradle version issues:
- Update Gradle wrapper: `./gradlew wrapper --gradle-version=8.2`

Performance Issues

App is slow or laggy:
- Use a physical device instead of emulator
- Close other apps to free RAM
- Reduce app animations in device Developer Options

Development Tips

Enable Debug Logging

To see more detailed logs, add this to `MainActivity.kt`:

```
Log.d("hideX", "Debug message here")
```

Hot Reload / Live Edit

Jetpack Compose supports Live Edit:
1. Make UI changes in Compose code
2. Save the file (`Ctrl+S`)
3. Changes appear instantly without rebuilding

Database Inspection

To inspect the Room database:

1. **Run the app**
2. In Android Studio: **View → Tool Windows → App Inspection**
3. Select **Database Inspector**
4. Browse the `hidex_database` and `protected_apps` table

Debugging

Set breakpoints in code:
1. Click in the left gutter next to line numbers

2. Run app in **Debug** mode (bug icon)
3. App will pause at breakpoints

Building APK for Distribution

Debug APK (for testing)

```
./gradlew assembleDebug
```

APK location: `app/build/outputs/apk/debug/app-debug.apk`

Release APK (for production)

1. Create a keystore (one-time setup):

```
keytool -genkey -v -keystore release.keystore -alias hidex -keyalg RSA -keysize 2048 -validity 10000
```

1. Add signing config to `app/build.gradle.kts`:

```
android {
    signingConfigs {
        create("release") {
            storeFile = file("../release.keystore")
            storePassword = "your_password"
            keyAlias = "hidex"
            keyPassword = "your_password"
        }
    }

    buildTypes {
        getByName("release") {
            signingConfig = signingConfigs.getByName("release")
            isMinifyEnabled = true
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }
}
```

1. Build release APK:

```
./gradlew assembleRelease
```

APK location: `app/build/outputs/apk/release/app-release.apk`

Useful Commands

```
# Clean build
./gradlew clean

# Build debug
./gradlew assembleDebug

# Install on connected device
./gradlew installDebug

# Run tests
./gradlew test

# Check dependencies
./gradlew dependencies

# Check for dependency updates
./gradlew dependencyUpdates
```

Next Steps

Once the app is running successfully:

1. ✅ Test with different apps
2. ✅ Test biometric authentication
3. ✅ Test password change functionality
4. ✅ Test app protection with back button, home button, etc.
5. ✅ Customize the app icon and theme if desired
6. ✅ Add more features or customize the code

Getting Help

If you encounter issues:

1. **Check Logcat** in Android Studio for error messages
2. **Review README.md** for common issues
3. **Search StackOverflow** for specific error messages
4. **Check Android Studio documentation**: https://developer.android.com/studio

Resources

- **Android Developers**: https://developer.android.com/
- **Jetpack Compose**: https://developer.android.com/jetpack/compose
- **Material Design 3**: https://m3.material.io/
- **Room Database**: https://developer.android.com/training/data-storage/room
- **Accessibility Services**: https://developer.android.com/guide/topics/ui/accessibility/service

---

Happy Coding! 🚀