

Offline Installation Configuration Guide

1. Prerequisites

- **JDK 11:**
Ensure JDK 11 is installed on your offline machine. Set the `JAVA_HOME` environment variable to your JDK 11 installation path.
- **Android SDK (Compile SDK 34):**
Install the Android SDK offline with the required compile SDK version (34). Configure the `ANDROID_HOME` environment variable to point to your SDK installation.
- **Gradle 8.5:**
Download the Gradle 8.5-bin.zip on an internet-connected machine, transfer it to your offline environment, and extract it. Set `GRADLE_HOME` and add its `bin` directory to your system `PATH`.
- **Kotlin 1.0.7:**
Make sure you have access to the Kotlin compiler version 1.0.7. Download the necessary Kotlin plugin and related libraries and include them in your local repository as needed.

2. Setting Up a Local Maven Repository

Since your machine is offline, you need to cache all required dependencies locally:

- **Download Dependencies:**
While online, download all required dependencies (Android Gradle Plugin, Kotlin libraries, etc.) and store them in a local Maven repository or a directory.
- **Configure Local Repository in Gradle:**
Use the `flatDir` repository or a local Maven repository path to reference these dependencies in your Gradle build scripts.

3. Sample Gradle Configuration

Project-level `build.gradle`:

```
gradle
Copy
buildscript {
    repositories {
        // Point to your local Maven repository
        maven { url 'file:///path/to/local/maven/repo' }
        // Alternatively, use a flat directory for JARs
        flatDir {
            dirs 'libs'
        }
    }
}
```

```

    }
}
dependencies {
    // Use the appropriate Android Gradle Plugin version here
    classpath 'com.android.tools.build:gradle:8.5'
    classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:1.0.7"
}
}

allprojects {
    repositories {
        maven { url 'file:///path/to/local/maven/repo' }
        flatDir {
            dirs 'libs'
        }
    }
}
}

```

Module-level **build.gradle**:

```

gradle
Copy
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'

android {
    compileSdkVersion 34

    defaultConfig {
        applicationId "com.cv.status"
        minSdkVersion 21
        targetSdkVersion 34
        versionCode 1
        versionName "1.0"
    }

    buildTypes {
        release {

```

```

        minifyEnabled false
        proguardFiles
getDefaultProguardFile('proguard-android-optimize.txt'),
'proguard-rules.pro'
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation "org.jetbrains.kotlin:kotlin-stdlib:1.0.7"
    // Add other dependencies as required
}

```

4. Building the App Offline

Command Line:

Open your terminal in the project directory and run:

bash

Copy

```
gradle build --offline
```

- This command forces Gradle to use the local repository and build the app without needing an internet connection.
- **Using Android Studio:**
Import the project from your offline environment. In Android Studio, enable Gradle's offline mode (usually found in the Gradle settings) so that it doesn't try to fetch dependencies online.

5. Additional Tips

- **Gradle Wrapper:**
Ensure the Gradle wrapper (`gradlew` and `gradlew.bat`) is configured correctly to use your local Gradle distribution if needed.
 - **Dependency Verification:**
Before going fully offline, test the build process once while connected to confirm that all necessary dependencies are cached locally.
 - **Documentation:**
Keep a README file with these steps so that team members or future developers can reproduce the offline environment setup easily.
-

This configuration should help you build and install your Android app offline using the specified versions and dependencies.