

How to add local libraries to Gradle

There is more than one way you can add AAR files to your project. For example, you can add libraries to all modules of your project or only for a single one. Or you can store all your libraries in single place and even create local maven repository and only add references to your libraries in your project. But this article is not going to describe all possible solution or explain how they work, here you can find just a few straightforward ways how to do it. You can find more detailed information on dependencies and libraries in [extras](#).

How to add AAR files to your Gradle project

1. Copy your AAR file to your module 'libs' folder. If you don't have 'libs' folder then create one.

For example, new projects module 'libs' folder location should be 'My Application/app/libs'

2. Now open your module level Gradle settings file

For example, new projects module Gradle settings location should be 'My Application/app/build.gradle'

3. Now you need to make 'libs' folder easily accessible. To do that add this lines to your 'module' level Gradle file:

```
repositories {  
    flatDir {  
        dirs 'libs'  
    }  
}
```

By default, you might not have any repositories settings in your 'module' level 'build.gradle' in that case you just can add it to the end of the file outside any other objects.

Also, it is possible that you already have `flatDir { dirs 'libs' }` in your repositories, in this case, you don't need to change anything

4. And finally, add the library to your 'module' Gradle dependencies

```
dependencies {  
    // Your might already have other librarieres/dependencies here you should leave them  
    // as they are and just add your on new line
```

```
compile (name:'your-library-file-name', ext:'aar')
}
```

Replace your-library-file-name with the name of your library without the file extension

How to add JAR files to your Gradle project

NB: You will have to select a single approach for your module because if the same library will be attached to the project twice it will give compilation errors. So please either Option 1 or Option 2 and not both together.

Option 1.

This is the default approach, but if you use it all JAR libraries that are in your 'libs' will be imported to your project, which can be both good and bad depending on your case.

1. Copy your JAR file to your module 'libs' folder. If you don't have 'libs' folder then create one.

For example, new projects module 'libs' folder location should be 'My Application/app/libs'

2. Add whole 'libs' folder to your 'module' dependencies

```
dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    // Your might already have other librarires/dependencies here you should leave them
    // as they are and just add your on new line
}
```

Note, that fileTree(include: ['.jar'], dir: 'libs') should be added only once to your module dependencies.*

Option 2.

1. Copy your JAR file to your module 'libs' folder. If you don't have 'libs' folder then create one.

For example, new projects module 'libs' folder location should be 'My Application/app/libs'

2. Add the library to your 'module' Gradle dependencies

```
dependencies {
    // Your might already have other librarires/dependencies here you should leave them
    // as they are and just add your on new line
    compile files('libs/your-library-file-name.jar')
}
```

Replace your-library-file-name with the full name of your library as well as the file extension

Extras

- Android Studio official explanation on how to add dependencies –
<https://developer.android.com/studio/build/dependencies.html>
- Gradle official explanation on dependencies –
https://docs.gradle.org/current/userguide/declaring_dependencies.html

<input type="text"/>	Search
----------------------	--------

Recent Posts

The happiest hours at MWC Barcelona 2019

Update: SDK 2.0 is now available

Recap of October's mobile industry events

The bigger and better Pocket Gamer Connects Helsinki 2018

Top Tips for Successful App Monetization

Archives

March 2019

January 2019

November 2018

September 2018

July 2018

June 2018

May 2018

April 2018

March 2018





January 2018

December 2017

Legal

- Legal
- Privacy
- COPPA
- Terms and Conditions

Follow us

-  Facebook
-  Twitter
-  LinkedIn
-  Instagram

Reach us

email:
contact@appmediation.com

Copyright © 2020 **Appmediation**