

# How To Add Flutter Packages To Jfrog Artifactory

Recent Dart 2.15 release added support for private package repositories. A package repository is a server that hosts Dart packages for consumption by the `dart pub` client. This post discusses how to use private package repositories alongside [pub.dev](https://pub.dev), leveraging the new token authentication mechanism introduced in Dart 2.15.

This feature has already been adopted by [Cloudsmith](#) and [JFrog Artifactory](#) who offer custom package repositories as a service.

## Why use a custom package repository?

The default package repository, [pub.dev](https://pub.dev), is operated by the Dart team. It facilitates publication of Dart packages for public use, and allows any Dart developer to consume those packages simply by adding the name of the package to their pubspec file. This enables a rich collection of Dart packages to be created by an active ecosystem. However, a custom package repository might be useful in the following scenarios:

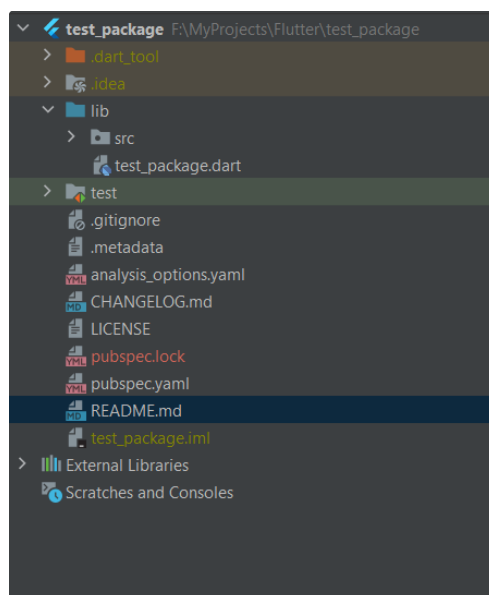
1. Sharing private packages within an organization. This could be an organization looking to share internal proprietary packages among the members of the organization, and not with the general public.
2. Tight control of dependencies in enterprise environments. Some corporations want to explicitly create allow-lists for specific *package versions* approved for usage. This might be done to enforce code review policies or license compliance.
3. Secure environments without public internet access. Some organizations operate in a regulatory environment where connections to the public internet, and by extension [pub.dev](https://pub.dev), are not allowed. This typically applies to government agencies/contractors and some banking service providers.

## Create Package

Run Following command to create sample test package in your local machine.

```
1 flutter create --template=package <your package name>
```

Open the package in your favorite editor and create src package.



After that implement the your package in side the `src` folder and import it your main dart file and export the package.

```
test_package.dart × pubspec.yaml × README.md ×
Android SDK "Android API 29 Platform" is not found on the disk or corrupted Configure...

1 import 'package:flutter/material.dart';
2 import 'package:test_package/src/utls/test_package_typedef.dart';
3
4 class TestPackage extends StatefulWidget {...}
37
38 class _TestPackageState extends State<TestPackage> {
39   @override
40   Widget build(BuildContext context) {
41     return GestureDetector(
42       onTap: widget.onTap ?? () {},
43       child: Container(
44         width: widget.width ?? MediaQuery.of(context).size.width * 0.90,
45         height: widget.height,
46         decoration: BoxDecoration(
47           borderRadius: const BorderRadius.all(Radius.circular(20.0)),
48           gradient: LinearGradient(colors: [
49             widget.color1 ?? const Color(0xFFCB1841),
50             widget.color2 ?? const Color(0xFF2604DE)
51           ], begin: Alignment.topLeft, end: Alignment.bottomRight), // LinearGradient
52           boxShadow: const [
53             BoxShadow(
54               color: Colors.grey,
55               blurRadius: 12,
56               offset: Offset(0, 6),
57             ), // BoxShadow
58           ],
59         ), // BoxDecoration
60         child: Column(
61           mainAxisAlignment: MainAxisAlignment.center,
62           crossAxisAlignment: CrossAxisAlignment.center,
```

```
src/test_package.dart × lib/test_package.dart × pubspec.yaml × README.md ×
Android SDK "Android API 29 Platform" is not found on the disk or corrupted Configure...

1 library test_package;
2
3 export 'src/test_package.dart';
```

Sample Code you can find this git hub repository: [GitHub - contactpkumara/test\\_package](https://github.com/contactpkumara/test_package): Fancy container package lets you add a beautiful gradient container to your Flutter app.

Then you need to create read me file and need to add package installation details as well as example for the usage of package.

# TestPackage Containers

Fancy container package lets you add a beautiful gradient container to your Flutter app.

## Installation

1. Authenticate the Jfrog Artifactory

Run the Following Command

```
dart pub token add "https://premiumsar1.jfrog.io/artifactory/api/pub/math-pub"
```



Then It will asked authentication token. Which will be provided by personally.

2. Add the latest version of package to your pubspec.yaml (and run `dart pub get`):

```
dependencies:
  test_package:
    hosted: https://premiumsar1.jfrog.io/artifactory/api/pub/math-pub
    version: ^0.0.1
```



3. Import the package and use it in your Flutter App.

```
import 'package:test_package/test_package.dart';
```



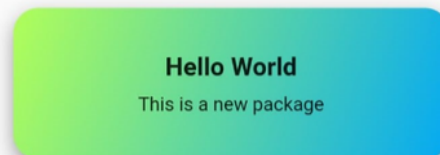
## Example

There are a number of properties that you can modify:

- height
- width
- title
- subtitle
- gradient (color1 and color2)

```
class TestPackage extends StatelessWidget {
  const TestPackage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: const TestPackage(
          title: 'Hello World',
          color1: Colors.lightGreenAccent,
          color2: Colors.lightBlue,
          subtitle: 'This is a new package',
          subtitleColor: Colors.black,
        ),
      ),
    );
  }
}
```



## Next Goals

- ☒ Add onTap for functions. Now, you can specify the onTap and specify a function.
- ☒ Change font and color style for text. Change color by specifying `textcolor` and `subtitlecolor` properties.
- ☐ Add more containers to the package.

Then we need to configure our private repository URL in the `pubspec.yaml` file using `publish_to` key. After that add `homepage` key value as our git hub repository URL.

```
name: test_package
description: A new Flutter package project.
version: 0.0.1
publish_to: https://tes.repository.url
homepage: https://github.com/adityathakurxd/fancy_containers

environment:
  sdk: '>=3.0.6 <4.0.0'
  flutter: ">=1.17.0"

dependencies:
  flutter:
    sdk: flutter

dev_dependencies:
  flutter_test:
    sdk: flutter
  flutter_lints: ^2.0.0

flutter:
```

Then Login to your private repository using following command and it will asked to provide the authentication token and you need to provide it.

```
1 dart pub token add "JFORG Repository URL"
```

After all the things are done we can run dry run command to check is their any thing missing or any warning we have in the created package.

```
1 dart pub publish --dry-run
```

Finally you can publish the your flutter packages to your private repository using following command.

```
1 dart pub publish
```