

Every developer at some point in time has thought about contributing to open source, but for one reason or another, they don't. This blog will cover everything from myth-busting to how you can make your first contribution to Flutter. Even if you are not a Flutter Developer, the general ideas will apply to contributing to any open source framework, language, or project of your choice.

Why should you care about Open Source contributions?

You might be thinking — why should you contribute to open source? You are a 10x Developer, what does open source have to offer you. You don't have to contribute to open source for benefits, but if you do, open source offers a lot to drive you through.

Mentoring

When you contribute to a project, you get mentored from the best folks in the form of PR reviews. When your PR gets reviewed, you will get comments which can help you improve your code contributions and learn the best practices.

animations/test/misc/expand_card_test.dart

Outdated

```
11 + );  
12 +  
13 + void main() {  
14 +   testWidgets('size of ExpandCard changes on tap', (tester) async {
```



RedBrogdon on Apr 12

Contributor



This is a good idea for a test, but it's reaching farther into the details of `ExpandCard` than it needs to. For example, it's coupled to the fact that `ExpandCard` is using an `AnimatedContainer` widget internally and the exact sizes of that widget (128, 256).

Consider instead merely checking that the `ExpandCard` itself gets bigger. You could:

- Create the screen
- Find the `ExpandCard` widget and record its size
- Tap directly on it
- `pumpAndSettle` (no duration necessary, I believe)
- Check the size again and verify just that it's bigger than the first value.

This would decouple your test from the specific size and the exact widgets being used *inside* `ExpandCard`. If you simulate the tap on `ExpandCard` itself, you'd also be able to avoid using hardcoded screen coordinates. That'd be a good thing, since they would change if the test were run on different devices.



AyushBherwani1998 on Apr 13

Author

Contributor



Makes sense, I was focusing on the implementation rather than `ExpandCard` itself. I'll do the required changes.

Andrew Brogdon, the maintainer of `flutter/samples` repository guiding me to write more efficient tests.

While contributing to [flutter/flutter](https://github.com/flutter/flutter) and [flutter/samples](https://github.com/flutter/samples) repository, I learned the best practices for writing test coverage in Flutter.

Learn Something New

Contributing to your favorite project is a good way to understand the project better. You will learn how to break down the project and understand the codebase through fixing something.

When you contribute to Flutter, you will learn how your favorite widget works, as well as how to write test coverage for every change you make in the codebase. It will also give you a deeper understanding of the framework.

Impacting Community

When you contribute to an open source project, you not only help yourself, but you also help the community around the project. The changes you make in the project will probably be used by thousands out there. You get a chance to connect with a beautiful community and give back to it.

But, I can't!

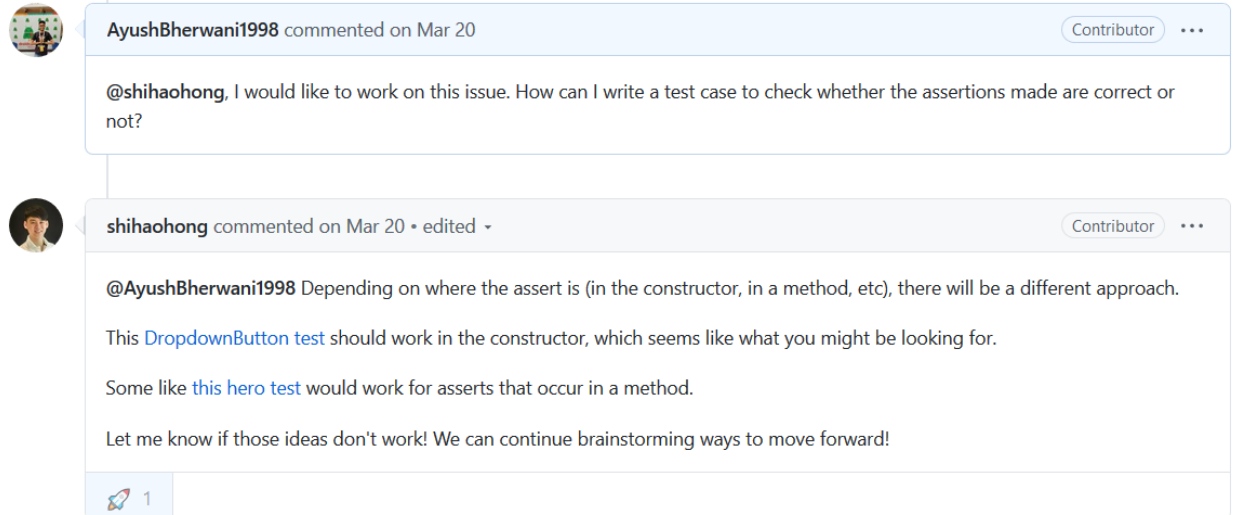
Many of us think that open source contribution is not for me. Only good developers can contribute to open source. You also might have come across one of these thoughts.

I should be a PRO to land a PR.

You don't have to be a PRO or a 10x developer to land a PR. If you can't make a code contribution because you have just started, Flutter has a lot of easy fix issues which includes document changes or adding new samples to the documentation.

I'm worried about asking help from the maintainers.

The thing I like most about the Flutter team is that they are very generous, and they are always there to help you land a PR. Whenever I don't know how to fix the issue or am not sure about something, I ask the maintainers, and they always help me, no matter how silly my question may seem.



AyushBherwani1998 commented on Mar 20 Contributor ...

@shihaohong, I would like to work on this issue. How can I write a test case to check whether the assertions made are correct or not?

shihaohong commented on Mar 20 • edited Contributor ...

@AyushBherwani1998 Depending on where the assert is (in the constructor, in a method, etc), there will be a different approach.

This [DropdownButton test](#) should work in the constructor, which seems like what you might be looking for.


Some like [this hero test](#) would work for asserts that occur in a method.

Let me know if those ideas don't work! We can continue brainstorming ways to move forward!

1

Shi-Hao Hong who is a member of the Flutter team helping me with writing a test.

If you have submitted a PR and you are not confident about the approach for the solution, you can always leave comments on the code and ask maintainers to help you select a better approach.



```
platform_channels/lib/src/basic_message_channel_demo.dart
84 +           onPressed: () {
85 +               setState(() {
86 +                   platformReply = PlatformMessages.sendPlatformMessage(
87 +                       controller.value.text,
```

AyushBherwani1998 8 days ago Author Contributor 😊 ...

@RedBrogdon Right now, I'm not checking whether the String is empty or not, and sending it anyways to the platform. Let me know if I should add a validation for empty String or not.

Reply...

Resolve conversation

Me asking question on whether my approach is correct or not in my one of recent PR.

Don't believe them, it's myth-busting time

When it comes to Open Source contribution, you might have heard a lot of things and myths about it. Don't worry, today we will bust few of the them.

Myth 1 : Open Source contribution is hard.

A lot of times, I have seen people say to me contributing to open source is hard. Only good developers can contribute to open source. It is a myth, and the one thing which I like about open source is that there is something to work on for everyone. If you are a developer, you can contribute to the codebase. If you are a technical writer, you can improve the existing documentation.

You need to take a leap of faith and comment on the issue that you would like to work on it. If you don't know how to solve it, don't worry, take some leads or help from the maintainers. They are always there to mentor you.

Myth 2: Writing documentation & creating issues are not open source contributions.

Documentation plays an important role in any project. No matter how awesome it is, if it doesn't have good documentation, people will have a hard time using it. So, even if you are contributing to improving the documentation, it's a valuable contribution to any open source project.

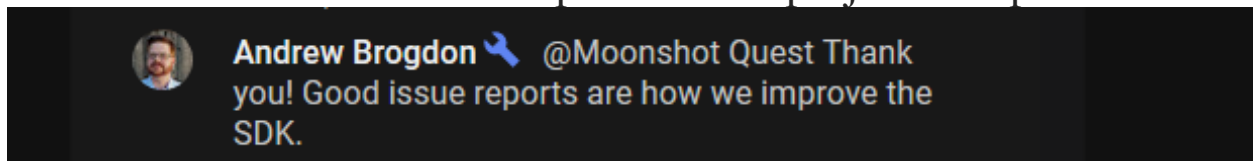
Improving documentation also gives you an opportunity to learn more about the API that you are documenting.

[Api Docs] Improves contentPadding documentation in InputDecoration #52807

Merged fluttergithubbot merged 4 commits into flutter:master from AyushBherwani1998:inputDecoration_docs on Mar 19

While working on the PR to improve the `InputDecoration.contentPadding` documentation, I learned how different properties are affecting the `contentPadding` and got a better understanding of the widget.

Many open source projects thrive on good and valuable issues, it's as important as sending a PR to land a fix. Finding bugs, giving feedback, and constructive criticism is important for a project to improve.



Comment from Andrew during Flutter Day Live stream.

How can you contribute to Flutter?

There are different ways you can contribute to Flutter, you can create issues, or you can make code contributions to the Flutter framework, Flutter engine, improve the existing samples, or land the requested features for the plugins.

Step 1

Select a repository from the Flutter organization, which has 33 repositories as of now including the SDK, engine, samples maintained by Flutter team, a Flutter gallery, the Flutter website, as well as your favorite Flutter plugins maintained by the team. For the whole list of repositories, take a look [here](#).

Step 2

Find issues labeled as ‘easy fix’, ‘good first contribution’, ‘entertaining new contributor project’, ‘documentation’, ‘d: api docs’, or take any issue you are comfortable fixing.

Step 3

Talk to the maintainers if you are not sure about how to solve the issue. You can comment on the issue and take some leads. If you are not sure who is the right person to reach out, you can also join the Flutter project’s [discord server](#) and take some leads. Check out the different channels and their purpose [here](#). As I mentioned earlier, the Flutter team is really generous and will be there to help you out. So don’t be hesitant to ask for help.

Step 4

Fork the repository, clone it, and do the changes to fix the issue. Once you have fixed the issue, you might also have to write a test to ensure your changes are solid and not breaking anything.

If you are working on the flutter repository and if your change is a breaking change, you might have to write a design document for proposing the breaking change before creating a PR. How can I define my change as a breaking change? Check out flutter breaking policy [here](#).

For more detailed information on submitting a PR to flutter, you can check out the Flutter tree hygiene wiki page [here](#).

Step 5

Wait for reviews and work on it. Once you receive an LGTM (looks good to me) comment on your PR, it will be ready to be merged 🎉.

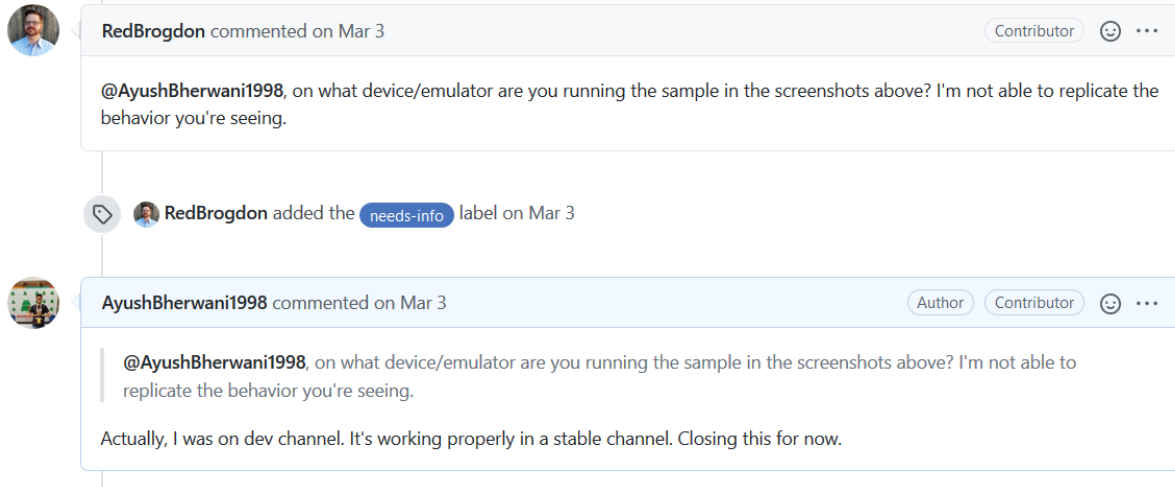
Don't get demotivated if your PR gets closed. It's okay and has happened to me. My PR was closed because I didn't create an issue before sending the PR. Later, I discovered that I was working on the beta channel instead of stable, and the samples target the stable branch. Every closed PR will also help you to learn something from it. I learned the importance of creating an issue and discussing the issue with other maintainers for validation and discussing potential solutions before sending a PR.

It's time for some tips

If you are willing to contribute to Flutter these tips will come handy.

Don't send a PR without creating an issue.

It's always a good habit to create an issue so that you can discuss the issue with other maintainers before sending a PR. You might be able to come up with a better solution together, or realize that the behavior is working as intended.



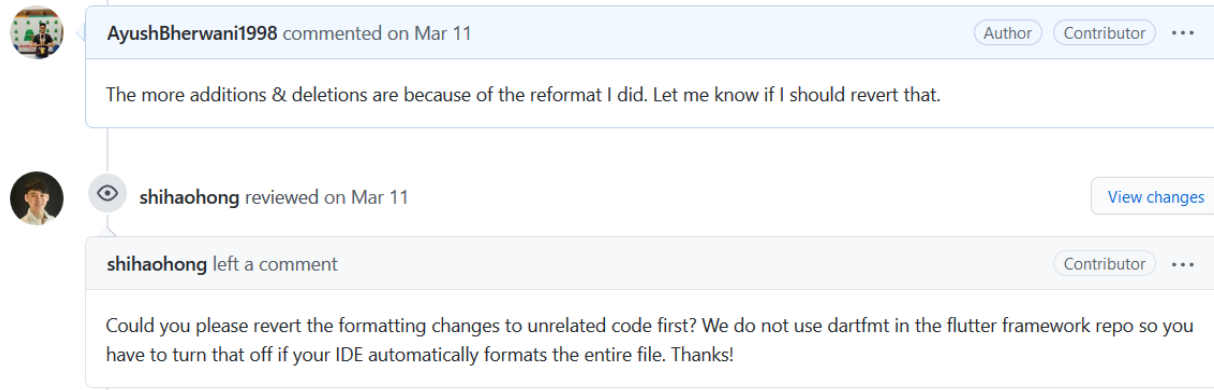
I did this mistake when I started contributing to the samples repository which I don't want you to do :p

Read the CONTRIBUTING guide or ask for TL,DR from someone.

The mistake I made above was because I didn't read the CONTRIBUTING guide, which clearly said that all the samples in the repository target the stable channel.

Don't use dartfmt in the Flutter repository.

Never never never use dartfmt in the Flutter repository cause they don't use it, and you will end up with a million additions and deletions. I made the same mistake in my first PR :(



P.S. I wish I had read the CONTRIBUTING guide properly. For more information on the style guide used in the Flutter repository, you can check [here](#).

Tips for sending a PR

1. One PR for one fix, no matter how small the fix is. Consider if you land three changes in a single PR, and due to some reasons maintainers need to revert one change, your other two changes will also get reverted along with it. It also helps the team to keep track of changes and review them easily.
2. Be open to changes, and healthy conversation. It's okay to get a lot of comments on the first PR, every project follows certain guidelines that you will understand in your first PR.
3. Don't forget to follow up on the reviews. Sometimes it may take weeks to land a PR, have patience, and trust the process.

That's it for this article, I hope this article was informative & will help you make your first contribution to Flutter 🎉

If need any help, don't forget that Flutter team is always there to help you land a PR.