# Contributing to ReactiveUI

We'd love for you to contribute to our source code and to make reactiveui even better than it is

today! Here are the guidelines we'd like you to follow:

- [Code of Conduct](https://reactiveui.net/code-of-conduct)

- [Question or Problem?](#question)

- [Issues and Bugs](#issue)

- [Feature Requests](#feature)

- [Submission Guidelines](#submit)

- [Coding Rules](#rules)

- [Commit Message Guidelines](https://reactiveui.net/contribute/software-style-guide/commit-message-convention)

## <a name="question"></a> Got a Question or Problem?

If you have questions about how to use reactiveui, please direct these to [StackOverflow](https://stackoverflow.com/questions/tagged/reactiveui). The project maintainers hang out in this [Slack](https://github.com/reactiveui/reactiveui#slack) channel.

## <a name="issue"></a> Found an Issue?

If you find a bug in the source code or a mistake in the documentation, you can help us by

submitting an issue to our [GitHub Repository](https://github.com/reactiveui/reactiveui). Even better you can submit a Pull Request

with a fix.

\*\*Please see the [Submission Guidelines](#submit) below.\*\*

## <a name="feature"></a> Want a Feature?

You can request a new feature by submitting an issue to our [GitHub Repository](https://github.com/reactiveui/reactiveui). If you

would like to implement a new feature then consider what kind of change it is:

\* \*\*Major Changes\*\* that you wish to contribute to the project should be discussed first in [Slack](https://github.com/reactiveui/reactiveui#slack) so that we can better coordinate our efforts,

prevent duplication of work, and help you to craft the change so that it is successfully accepted

into the project.

\* \*\*Small Changes\*\* can be crafted and submitted to the [GitHub Repository](https://github.com/reactiveui/reactiveui) as a Pull

Request.

## <a name="submit"></a> Submission Guidelines

### Submitting an Issue

If your issue appears to be a bug, and hasn't been reported, open a new issue. Help us to maximize

the effort we can spend fixing issues and adding new features, by not reporting duplicate issues.

Providing the following information will increase the chances of your issue being dealt with

quickly:

\* \*\*Overview of the Issue\*\* - if an error is being thrown a stack trace helps

\* \*\*Motivation for or Use Case\*\* - explain why this is a bug for you

\* \*\*reactiveui Version(s)\*\* - is it a regression?

\* \*\*Operating System\*\* - is this a problem with all browsers or only specific ones?

\* \*\*Reproduce the Error\*\* - provide a example or an unambiguous set of steps.

\* \*\*Related Issues\*\* - has a similar issue been reported before?

\* \*\*Suggest a Fix\*\* - if you can't fix the bug yourself, perhaps you can point to what might be

causing the problem (line of code or commit)

\*\*If you get help, help others. Good karma rulez!\*\*

### Submitting a Pull Request

Before you submit your pull request consider the following guidelines:

\* Search [GitHub](https://github.com/reactiveui/reactiveui/pulls) for an open or closed Pull Request

that relates to your submission. You don't want to duplicate effort.

\* Make your changes in a new git branch:

```shell

git checkout -b my-fix-branch master

```

\* Create your patch, \*\*including appropriate test cases\*\*.

\* Follow our [Coding Rules](#rules).

\* Run the test suite, as described below.

\* Commit your changes using a descriptive commit message that follows our

[commit message conventions](#commit).

```shell

git commit -a

```

Note: the optional commit `-a` command line option will automatically "add" and "rm" edited files.

\* Build your changes locally to ensure all the tests pass:

```shell

build.cmd

```

\* Push your branch to GitHub:

```shell

git push origin my-fix-branch

```

In GitHub, send a pull request to `reactiveui:master`.

If we suggest changes, then:

\* Make the required updates.

\* Re-run the test suite to ensure tests are still passing.

\* Commit your changes to your branch (e.g. `my-fix-branch`).

\* Push the changes to your GitHub repository (this will update your Pull Request).

If the PR gets too outdated we may ask you to rebase and force push to update the PR:

```shell

git rebase master -i

git push origin my-fix-branch -f

```

\_WARNING: Squashing or reverting commits and force-pushing thereafter may remove GitHub comments

on code that were previously made by you or others in your commits. Avoid any form of rebasing

unless necessary.\_

That's it! Thank you for your contribution!

#### After your pull request is merged

After your pull request is merged, you can safely delete your branch and pull the changes

from the main (upstream) repository:

\* Delete the remote branch on GitHub either through the GitHub web UI or your local shell as follows:

```shell

git push origin --delete my-fix-branch

```

\* Check out the master branch:

```shell

git checkout master -f

```

\* Delete the local branch:

```shell

git branch -D my-fix-branch

```

\* Update your master with the latest upstream version:

```shell

git pull --ff upstream master

```

## Coding

### Developer Environment

- Visual Studio 2019 (with latest patches/updates), with the following workloads/components

- Windows Universal Platform Development workload

- .NET desktop development workload

- Mobile development with .NET workload

- .NET Core cross-platform development

- Windows 10 SDK 10.0.17763.0 (individual component)

- PowerShell

- Android SDK Platform 26 (install through Visual Studio)

Once all the above are set up, running `.\build.cmd` in the repository root should get everything up and running.

### <a name="rules"></a> Coding Rules

To ensure consistency throughout the source code, keep these rules in mind as you are working:

\* All features or bug fixes \*\*must be tested\*\* by one or more unit tests.

\* All public API methods \*\*must be documented\*\* with XML documentation.