#Contributing to textAngular

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

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

- [Issues and Bugs](#issue)

- [Feature Requests](#feature)

- [Submission Guidelines](#submit)

- [Coding Rules](#rules)

- [Commit Message Guidelines](#commit)

## <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][github]. Even better you can submit a Pull Request

with a fix.

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

You can request a new feature by submitting an issue to our [GitHub Repository][github]. 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 as an issue in the repository 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][github] as a Pull Request.

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

### Submitting an Issue

Before you submit your issue search the archive, maybe your question was already answered.

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 non-minified stack trace helps

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

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

\* \*\*Browsers and Operating System\*\* - is this a problem with all browsers or only IE8?

\* \*\*Reproduce the error\*\* - provide a live example ([Plunkr](http://plnkr.co/edit/tpl:RaBEszV5xrAZ0VRoZ7PM?p=preview)) or a 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)

Here is a great example of a well defined issue: https://github.com/angular/angular.js/issues/5069

\*\*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/fraywing/textAngular/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 full Angular test suite, as described in the [developer documentation][dev-doc],

and ensure that all tests pass.

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

[commit message conventions](#commit-message-format) and passes our commit message presubmit hook

`validate-commit-msg.js`. Adherence to the [commit message conventions](#commit-message-format)

is required because release notes are automatically generated from these messages.

```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

grunt test

```

\* Push your branch to GitHub:

```shell

git push origin my-fix-branch

```

\* In GitHub, send a pull request to `angular:master`.

\* If we suggest changes then

\* Make the required updates.

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

\* Rebase your branch and force push to your GitHub repository (this will update your Pull Request):

```shell

git rebase master -i

git push -f

```

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

```

## <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 [specs][unit-testing].

\* We \*\*love functions and closures\*\* and, whenever possible, prefer them over objects.

\* Where possible follow the coding style that is already in place.

\* Use \*\*tabs\*\* not spaces.

## <a name="commit"></a> Git Commit Guidelines

We have very precise rules over how our git commit messages can be formatted. This leads to \*\*more

readable messages\*\* that are easy to follow when looking through the \*\*project history\*\*.

### Commit Message Format

Each commit message consists of a \*\*header\*\*, a \*\*body\*\* and a \*\*footer\*\*. The header has a special

format that includes a \*\*type\*\*, a \*\*scope\*\* and a \*\*subject\*\*:

```

<type>(<scope>): <subject>

<BLANK LINE>

<body>

<BLANK LINE>

<footer>

```

Any line of the commit message cannot be longer 100 characters! This allows the message to be easier

to read on github as well as in various git tools.

### Type

Must be one of the following:

\* \*\*feat\*\*: A new feature

\* \*\*fix\*\*: A bug fix

\* \*\*style\*\*: Changes that do not affect the meaning of the code (white-space, formatting, missing

semi-colons, etc)

\* \*\*refactor\*\*: A code change that neither fixes a bug or adds a feature

\* \*\*perf\*\*: A code change that improves performance

\* \*\*test\*\*: Adding missing tests

\* \*\*chore\*\*: Changes to the build process or auxiliary tools and libraries such as documentation

generation

### Scope

The scope could be anything specifying place of the commit change. For example `$location`,

`$browser`, `$compile`, `$rootScope`, `ngHref`, `ngClick`, `ngView`, etc...

### Subject

The subject contains succinct description of the change:

\* use the imperative, present tense: "change" not "changed" nor "changes"

\* don't capitalize first letter

\* no dot (.) at the end

###Body

Just as in the \*\*subject\*\*, use the imperative, present tense: "change" not "changed" nor "changes"

The body should include the motivation for the change and contrast this with previous behavior.

###Footer

The footer should contain any information about \*\*Breaking Changes\*\* and is also the place to

reference GitHub issues that this commit \*\*Closes\*\*.

A detailed explanation can be found in this [document][commit-message-format].