# Contributing

:mega: \*\*Support/Questions?\*\*: Please see our [Support

Page](https://ionicframework.com/support) for general support questions. The

issues on GitHub should be reserved for bug reports and feature requests.

### Branches

\* [`develop`](https://github.com/ionic-team/ionic-cli/tree/develop): \*\*development\*\* branch

\* [`stable`](https://github.com/ionic-team/ionic-cli/tree/stable): \*\*stable\*\* version

##### Version Branches

These are mostly for reference--older major versions are typically not maintained.

\* [`5.x`](https://github.com/ionic-team/ionic-cli/tree/5.x)

\* [`4.x`](https://github.com/ionic-team/ionic-cli/tree/4.x)

\* [`3.x`](https://github.com/ionic-team/ionic-cli/tree/3.x)

\* [`2.x`](https://github.com/ionic-team/ionic-cli/tree/2.x) (\*legacy version\*)

### Bug Reports

Run the command(s) with `--verbose` to produce debugging output. We may ask for

the full command output, including debug statements.

Please also copy/paste the output of the `ionic info` command into your issue

and be as descriptive as possible. Include any steps that might help us

reproduce your issue.

### Feature Requests

Post an issue describing your feature to open a dialogue with us. We're happy

to hear from you!

### Pull Requests

Pull requests are most welcome! But, if you plan to add features or do large

refactors, please \*\*open a dialogue\*\* with us first by creating an issue. Small

bug fixes are welcome any time.

#### Help Wanted

Looking for small issues to help with? You can browse the [`help

wanted`](https://github.com/ionic-team/ionic-cli/labels/help%20wanted) label.

These are issues that have been marked as great opportunities for someone's

first PR to the Ionic CLI. :heart\_eyes:

### Local Setup

#### Structure

The Ionic CLI is organized into a monorepo. Here are the packages:

##### CLIs

\* [`packages/@ionic/cli`](https://github.com/ionic-team/ionic-cli/tree/develop/packages/%40ionic/cli): Ionic CLI executable and library.

\* [`packages/@ionic/lab`](https://github.com/ionic-team/ionic-cli/tree/develop/packages/%40ionic/lab): Utility CLI for Ionic Lab, used by `ionic serve`.

\* [`packages/@ionic/v1-toolkit`](https://github.com/ionic-team/ionic-cli/tree/develop/packages/%40ionic/v1-toolkit): Utility CLI for Ionic v1 Apps, used by `ionic serve`.

##### Libraries

\* [`packages/@ionic/cli-framework`](https://github.com/ionic-team/ionic-cli/tree/develop/packages/%40ionic/cli-framework): Framework for command-line programs.

\* [`packages/@ionic/cli-framework-prompts`](https://github.com/ionic-team/ionic-cli/tree/develop/packages/%40ionic/cli-framework-prompts): Command-line prompting framework that wraps [Inquirer.js](https://github.com/SBoudrias/Inquirer.js).

\* [`packages/@ionic/discover`](https://github.com/ionic-team/ionic-cli/tree/develop/packages/%40ionic/discover): Service discovery library used for `ionic serve` with the [Ionic DevApp](https://ionicframework.com/docs/appflow/devapp/) (now retired).

\* [`packages/@ionic/utils-array`](https://github.com/ionic-team/ionic-cli/tree/develop/packages/%40ionic/utils-array): General purpose array library with asynchronous map/filter/reduce.

\* [`packages/@ionic/utils-fs`](https://github.com/ionic-team/ionic-cli/tree/develop/packages/%40ionic/utils-fs): Filesystem library that wraps [fs-extra](https://github.com/jprichardson/node-fs-extra) for Node.js.

\* [`packages/@ionic/utils-network`](https://github.com/ionic-team/ionic-cli/tree/develop/packages/%40ionic/utils-network): Network library for Node.js.

\* [`packages/@ionic/utils-object`](https://github.com/ionic-team/ionic-cli/tree/develop/packages/%40ionic/utils-object): General purpose object library.

\* [`packages/@ionic/utils-process`](https://github.com/ionic-team/ionic-cli/tree/develop/packages/%40ionic/utils-process): OS process library for Node.js.

\* [`packages/@ionic/utils-stream`](https://github.com/ionic-team/ionic-cli/tree/develop/packages/%40ionic/utils-stream): Stream library for Node.js.

\* [`packages/@ionic/utils-subprocess`](https://github.com/ionic-team/ionic-cli/tree/develop/packages/%40ionic/utils-subprocess): Subprocess library that uses [cross-spawn](https://github.com/moxystudio/node-cross-spawn) for Node.js.

\* [`packages/@ionic/utils-terminal`](https://github.com/ionic-team/ionic-cli/tree/develop/packages/%40ionic/utils-terminal): Terminal and command-line environment library for Node.js.

#### Toolset

\* npm 6+ is required.

\* Node 12+ is required.

\* The codebase is written in [TypeScript](https://www.typescriptlang.org/). If

you're unfamiliar with TypeScript, we recommend using [VS

Code](https://code.visualstudio.com/) and finding a tutorial to familiarize

yourself with basic concepts.

\* The test suite uses [Jest](https://facebook.github.io/jest/).

#### Setup

1. Fork the repo & clone it locally.

1. `npm install` to install the dev tools.

1. `npm run bootstrap` (will install package dependencies and link packages

together)

1. Optionally `npm run link` to make `ionic` and other bin files point to your

dev CLI.

1. `npm run watch` will spin up TypeScript watch scripts for all packages.

1. TypeScript source files are in `packages/\*\*/src`.

1. Good luck! :muscle: Please open an issue if you have questions or something

is unclear.

#### Running Dev CLI

To switch between dev CLI and stable CLI, you can use a Node version manager

such as [nvm](https://github.com/creationix/nvm) and switch between

environments, e.g.

```bash

nvm install v12

nvm alias cli-local v12

```

You can even set up an alias in your terminal that sets `IONIC\_CONFIG\_DIRECTORY`

to have seperate configuration environments.

```bash

alias cli-local="nvm use cli-local && export IONIC\_CONFIG\_DIRECTORY=$HOME/.ionic/cli-local"

```

When the Node environment is created, create a symlink to `packages/@ionic/cli/bin/ionic` within `$NVM\_BIN`:

```bash

ln -s $(pwd)/packages/@ionic/cli/bin/ionic $NVM\_BIN

```

##### Debugging

The following workflow is recommended for debugging the Ionic CLI:

1. Place

[`debugger;`](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/debugger)

statements where desired.

1. Run the CLI via `node` to use Node's `--inspect-brk` flag:

\* Instead of `~/path/to/ionic <command>` use `node --inspect-brk

~/path/to/ionic <command>`.

\* Instead of `ionic <command>`, try `node --inspect-brk $(which ionic)

<command>` (works on Mac and Linux).

1. Open `chrome://inspect` in Chrome and select the remote target to use

DevTools for debugging.

Read more about Node debugging in the [Debugging

Guide](https://nodejs.org/en/docs/guides/debugging-getting-started/).

##### Publishing

CI automatically publishes the next version semantically from analyzing commits in `stable`. To maintain a shared history between `develop` and `stable`, the branches must be rebased with each other locally.

\* When it's time to cut a release from `develop`:

```

git checkout stable

git rebase develop

git push origin stable

```

\* Await successful publish in CI. Ionitron will push the updated versions and tags to `stable`.

\* Sync `develop` with `stable`.

```

git pull origin stable

git checkout develop

git rebase stable

git push origin develop

```

To publish \*\*testing\*\* versions, follow these steps:

1. Cancel any watch scripts.

1. Run `npm run publish:testing`