# Contributions

We welcome contributions in the form of issues and pull requests. We view the contributions and process as the same for internal and external contributors.

## Issues

Log issues for both bugs and enhancement requests. Logging issues are important for the open community.

Issues in this repository should be for the toolkit packages. General feedback for GitHub Actions should be filed in the [community forums.](https://github.community/t5/GitHub-Actions/bd-p/actions) Runner specific issues can be filed [in the runner repository](https://github.com/actions/runner).

## Enhancements and Feature Requests

We ask that before significant effort is put into code changes, that we have agreement on taking the change before time is invested in code changes.

1. Create a feature request.

2. When we agree to take the enhancement, create an ADR to agree on the details of the change.

An ADR is an Architectural Decision Record. This allows consensus on the direction forward and also serves as a record of the change and motivation. [Read more here](../docs/adrs/README.md).

## Development Life Cycle

This repository uses [Lerna](https://github.com/lerna/lerna#readme) to manage multiple packages. Read the documentation there to begin contributing.

Note that before a PR will be accepted, you must ensure:

- all tests are passing

- `npm run format` reports no issues

- `npm run lint` reports no issues

### Useful Scripts

- `npm run bootstrap` This runs `lerna bootstrap` which will install dependencies in this repository's packages and cross-link packages where necessary.

- `npm run build` This compiles TypeScript code in each package (this is especially important if one package relies on changes in another when you're running tests). This is just an alias for `lerna run tsc`.

- `npm run format` This checks that formatting has been applied with Prettier.

- `npm test` This runs all Jest tests in all packages in this repository.

- If you need to run tests for only one package, you can pass normal Jest CLI options:

```console

$ npm test -- packages/toolkit

```

- `npm run create-package [name]` This runs a script that automates a couple of parts of creating a new package.

### Creating a Package

1. In a new branch, create a new Lerna package:

```console

$ npm run create-package new-package

```

This will ask you some questions about the new package. Start with `0.0.0` as the first version (look generally at some of the other packages for how the package.json is structured).

2. Add `tsc` script to the new package's package.json file:

```json

"scripts": {

"tsc": "tsc"

}

```

3. Start developing ?.