# Contributing Guide

## Table of Contents

- [How to Help?](#how-to-help)

- [Contribution Guidelines](#contribution-guidelines)

- [Getting Started](#getting-started)

- [Testing](#testing)

- [Before Sending a Pull Request](#before-sending-a-pull-request)

- [Code Style Guide](#code-style-guide)

- [Lint the Code](#lint-the-code)

- [Use EditorConfig](#use-editorconfig)

- [Documentation](#documentation)

- [JSDoc](#jsdoc)

## How to Help?

Help is always welcome. There are areas where you can help:

- The core functionality (performance improvements, bug fixes,

new features, etc.).

- Documentation ([markdown documents](https://github.com/date-fns/date-fns/search?l=markdown),

inline JSDoc comments).

- Test suite & development environment improvements.

- The [website](https://github.com/date-fns/date-fns.org).

If you see a gap, but don't have time, experience, or you just need help

with the library, don't hesitate to [shoot an issue](https://github.com/date-fns/date-fns/issues/new).

The date-fns functionality is comprehensive and covers most of the use cases,

however it doesn't have extended time zone support. Please leave a comment

to the [Extended time zones support issue](https://github.com/date-fns/date-fns/issues/180)

if you are interested in the functionality or want to help with development.

If you are interested in Elm/ClojureScript/etc. wrappers,

please [file an issue](https://github.com/date-fns/date-fns/issues/new).

## Contribution Guidelines

Due to the modular nature of date-fns, it's more than open to new features.

However, when a new function duplicates the existing functionality, native API

or causes significant build size increase, a PR might be rejected or

the author can be asked to move the code to a new or another package.

Please follow the main contributing rules, to maintain date-fns' top quality:

- Follow style guides:

- [Lint the code](#lint-the-code).

- [Use EditorConfig](#use-editorconfig).

- Write tests.

- [Write documentation](#documentation).

- Don't update the changelog.

- Don't change the library version.

## Getting Started

1. Install [Node.js](https://nodejs.org/en/download) and [Yarn](https://yarnpkg.com/en/docs/install).

2. Fork the project and clone the fork repo.

3. Run `yarn` to install the application dependencies.

## Testing

Run all tests:

```sh

yarn test

```

Run tests once:

```sh

yarn test --single-run

```

To test a function in REPL, use `babel-node` located in `./node\_modules/.bin`:

```sh

./node\_modules/.bin/babel-node

> const toDate = require('./src/toDate')

undefined

> toDate(1392098430000).toString()

'Tue Feb 11 2014 01:00:30 GMT-0500 (Eastern Standard Time)'

>

```

Build date-fns to test in in your project:

```sh

env PACKAGE\_OUTPUT\_PATH="$(pwd)/../PATH-TO-YOUR-MODULE/node\_modules/date-fns" ./scripts/build/package.sh

```

## Before Sending a Pull Request

Rebuild FP functions, typings and indices by using the following script. It could take around a minute:

```sh

./scripts/build/build.sh

```

## Code Style Guide

### Lint the Code

The project follows [Prettier] code style and uses [ESLint] as the linter.

To lint the code, run:

```bash

yarn lint

```

[prettier]: https://prettier.io/

[eslint]: https://eslint.org/

### Use EditorConfig

The project uses [EditorConfig] to define basic coding style guides.

Please install a plugin for your editor of choice or manually enforce

the rules listed in [.editorconfig].

[editorconfig]: http://editorconfig.org

[.editorconfig]: https://github.com/date-fns/date-fns.org/blob/master/.editorconfig

## Documentation

### JSDoc

[JSDoc](http://usejsdoc.org) is used for the code documentation. Along with

standard JSDoc tags, date-fns uses `@category` tag that allows

to group functions.

[jsdoc-parse](https://github.com/jsdoc2md/jsdoc-parse) is used to parse

JSDoc annotations.