# Contributing to Manta

Thank you for considering a contribution to Manta! This guide explains how to:

\* Work on the Manta code base

\* Get help if you encounter trouble

\* Maximize the chance of your changes being accepted

## Get in touch

Before starting to work on a feature or a fix, please open an issue so we can discuss about it first. I believe this can save us a lot of time. For any non-trivial change, it would be best if you can create a short design/mockup/document explaining:

\* Why is this change necessary? What's the use case?

\* What will the UI look like? (If the changes will alter the UI in someway)

\* What could go wrong? What test cases should it have?

\* How will it roughly be implemented? (If you don't know where to begin and need some pointers, feel free to ask)

## Follow the Code of Conduct

In order to foster a more inclusive community, Manta has adopted the [Contributor Covenant](https://www.contributor-covenant.org/version/1/4/code-of-conduct/) and all contributors are expected to follow the Code of Conduct outlined in this document.

## Making Changes

### Development Setup

It's very easy to get started working on Manta, just follow these steps:

- Clone this project to your local machine.

- Open terminal and `cd` into the cloned folder, usually `cd Manta`.

- Run `yarn install` to install dependencies.

- Run `yarn dev` in one tab to start `webpack-dev-server`.

- Run `yarn start` in another tab to open the app.

> Note that on Linux you will need additional dependencies to provide the icns2png and gm commands. Ubuntu/Debian users can run: sudo apt install -y icnsutils graphicsmagick.

### Code Change General Guidelines

\* All code contribution need to contain at least some unit tests (using [Jest](https://github.com/facebook/jest)) for any logic, components, helpers ... that it introduces.

Also, please make sure your changes:

\* Do not break existing tests.

\* Do not include unnecessary code (code that's meant to solve other issues or not related to the current issue)

### Creating Commits And Writing Commit Messages

The commit messages that accompany your code changes are an important piece of documentation, please follow these guidelines when writing commit messages:

\* Keep commits discrete: avoid including multiple unrelated changes in a single commit

\* Keep commits self-contained: avoid spreading a single change across multiple commits. A single commit should make sense in isolation

\* If your commit pertains to a GitHub issue, include (`Issue: #123`) in the commit message on a separate line

### Adding/Refactoring Custom Fields

If you're planning to add a new custom field, or refactoring old ones, please read this guide: [Adding and Refactoring Custom Fields](https://github.com/hql287/Manta/wiki/Adding-and-Refactoring-Custom-Fields)

### Submitting Your Change

Your PR will be reviewed once you submitted it. It is normal that this takes several iterations so don't get discouraged by change requests. They ensure the high quality that we all enjoy.

## Thanks

Your effort toward improving Manta is deeply appreciated. For any contribution, large or small, you will be immortalized in the release notes for the version you've contributed to.