# Contributing to Mailspring

Thanks for checking out Mailspring! We'd love for you to contribute. Whether you're a first-time open source contributor or an experienced developer, there are ways you can help make Mailspring great:

1. Grab an issue tagged with \*\*[Help Wanted](https://github.com/Foundry376/Mailspring/labels/help%20wanted)\*\* and dig in! We try to add context to these issues when adding the label so you know where to get started in the codebase. Be wary of working on issues without the \*\*Help Wanted\*\* label - just because someone has created an issue doesn't mean we'll accept a pull request for it. See [Where to Contribute](#where-to-contribute) below for more information.

2. Triage issues that haven't been addressed. With a large community of users on many platforms, we have trouble keeping up GitHub issues and moving the project forward at the same time. If you're good at testing and addressing issues, we'd love your help!

### Filing an Issue

If you have a feature request or bug to report, \_please\_ search for existing issues \*\*including closed ones!\*\*: https://github.com/Foundry376/Mailspring/issues?utf8=%E2%9C%93&q=is%3Aissue. If someone has already requested the feature you have in mind, upvote it using the "Add Reaction" feature - our team often sorts issues to find the most upvoted ones. For bugs, please verify that you're running the latest version of Mailspring. If you file an issue without providing detail, we may close it without comment.

### Pull requests

The first time you submit a pull request, a bot will ask you to sign a standard, bare-bones Contributor License Agreement. The CLA states that you waive any patent or copyright claims you might have to the code you're contributing. (e.g.: you can't submit a PR and then sue Mailspring for using your code.)

# Build and Run From Source

If you want to understand how Mailspring works or want to debug an issue, you'll want to get the source, build it, and run it locally.

### Installing Prerequisites

You'll need git and a recent version of Node.JS (any v7.2.1+ is recommended with npm v3.10.10+). [nvm](https://github.com/creationix/nvm) is also highly recommended. Based on your platform, you'll also need:

\*\*Windows:\*\*

- `npm install --global --production windows-build-tools`

\*\*OS X:\*\*

- Python

- Xcode and the Command Line Tools (Xcode -> Preferences -> Downloads), which will install `gcc` and the related toolchain containing `make`.

\*\*Linux:\*\*

- Python v2.7

- make

- A proper C/C++11 compiler tool chain, for example GCC

- Library dependencies:

- On Debian-based Linux: `sudo apt-get install libx11-dev libxkbfile-dev execstack libgconf-2-4 libsecret-1-dev`

- On Red Hat-based Linux: `sudo yum install libX11-devel.x86\_64 libxkbfile-devel.x86\_64 libsecret-1-dev libsecret-devel`.

After you have these tools installed, run the following commands to check out Mailspring,install dependencies, and launch the app:

```

git clone https://github.com/foundry376/mailspring

cd mailspring

npm install

npm start

```

# Development Workflow

#### App Data

When you're running Mailspring with `npm start`, it runs with the `--dev` flag and user data is located in a `Mailspring-dev` folder alongside the regular settings folder:

- Mac: `~/Library/Application Support/Mailspring-dev`

- Windows: `C:\Users\<you>\AppData\Roaming\Mailspring-dev`

- Linux: `~/.config/Mailspring-dev/`

#### Developer Tools

From Mailspring, you can open the Developer Tools from the menu: `Menu > Developer > Toggle Developer Tools`. Here are a few tips for getting started:

- Errors and warnings will show in the console.

- On the console, `$m` is a shorthand for `mailspring-exports`, and allows you to access global `Stores` and `Model` classes.

- You don't need to stop and restart the development version of Mailspring after each change. You can just reload the window via `CMD+R` (`CTRL+R` on Windows, Linux).

#### Linting

We use `prettier` and `eslint` for linting our sources. You can run both of these by running `npm run lint` on the command line. Always do this before submitting a pull request to ensure the CI servers accept your code.

#### Documentation

Back in 2015 and 2016, the Nylas Mail team made a huge effort to document the codebase. The docs are still [available on GitHub pages](https://foundry376.github.io/Mailspring/), and are still largely relevant to Mailspring development.

#### Discussion Etiquette

In order to keep the conversation clear and transparent, please limit discussion to English and keep things on topic with the issue. Be considerate to others and try to be courteous and professional at all times.

#### Testing Localization

You can run the app in different localizations by passing a langauge code to Electron via `--lang=de`. To use this in conjunction `npm start`, you'll need to use the `--` argument forwarding syntax: `npm start -- --lang=de`.

# Where to Contribute

Check out the full issues list for a list of all potential areas for contributions. Note that just because an issue exists in the repository does not mean we will accept a contribution to the core mail client for it. There are several reasons we may not accepts a pull requests, like:

- \*\*Maintainability\*\* - We're \_extremely\_ wary of adding options and preferences for niche behaviors. Email is a wild west, and we can't afford to support every possible configuration. Our general rule is that the code complexity of adding a preference isn't worth it unless the user base is fairly evenly divided about the desired behavior. [We don't want to end up with this!](https://cloud.githubusercontent.com/assets/1037212/14989123/2a74e810-110b-11e6-8b5d-6f343bca712f.png)

- \*\*User experience\*\* - We want to deliver a lightweight and smooth mail client, so UX and performance matter a lot. If you'd like to change or extend the UI, consider doing it in a plugin or theme.

- \*\*Architectural\*\* - The team and/or feature owner needs to agree with any architectural impact a change may make. Things like new extension APIs must be discussed with and agreed upon by the feature owner.

To improve the chances to get a pull request merged you should select an issue that is labelled with the help-wanted or bug labels. If the issue you want to work on is not labelled with help-wanted or bug, you can start a conversation with the issue owner asking whether an external contribution will be considered.