[![Build Status](https://img.shields.io/badge/build-passing-brightgreen.svg?style=flat-square)](https://github.com/johnbillion/query-monitor/actions)

# Contributing to Query Monitor

Code contributions, bug reports, and feedback are very welcome. These should be submitted through [the GitHub repository](https://github.com/johnbillion/query-monitor). Development happens in the `develop` branch, and any pull requests should be made against that branch please.

\* [Reporting Security Issues](#reporting-security-issues)

\* [Setting up Locally](#setting-up-locally)

\* [Building the Assets](#building-the-assets)

\* [Running the Tests](#running-the-tests)

\* [Releasing a New Version](#releasing-a-new-version)

## Reporting Security Issues

If you discover a security issue in Query Monitor, please report it to [the security program on HackerOne](https://hackerone.com/johnblackbourn). Do not report security issues on GitHub or the WordPress.org support forums. Thank you.

## Setting up Locally

You can clone this repo and activate it like a normal WordPress plugin, but you'll need to install the developer dependencies in order to build the assets and to run the tests.

### Prerequisites

\* [Composer](https://getcomposer.org/)

\* [Node](https://nodejs.org/)

### Setup

1. Install the PHP dependencies:

composer install

2. Install the Node dependencies:

npm install

3. Check the MySQL database credentials in the `tests/.env` file and amend them as necessary.

## Building the Assets

To compile the Sass files into CSS:

npm run build

To start the file watcher which will watch for changes and automatically compile the Sass:

npm run watch

## Running the Tests

To run the whole test suite which includes unit tests and linting:

composer test

To run just the PHPUnit tests:

composer test:ut

To run just the code sniffer:

composer test:cs

## Releasing a New Version

These are the steps to take to release a new version of Query Monitor (for contributors who have push access to the GitHub repo).

### Prior to Release

1. Check [the milestone on GitHub](https://github.com/johnbillion/query-monitor/milestones) for open issues or PRs. Fix or reassign as necessary.

1. If this is a non-patch release, check issues and PRs assigned to the patch or minor milestones that will get skipped. Reassign as necessary.

1. Ensure you're on the `develop` branch and all the changes for this release have been merged in.

1. Ensure both `README.md` and `readme.txt` contain up to date descriptions, "Tested up to" versions, FAQs, screenshots, etc.

- This is currently a manual process while I decide whether I want to sync parts of these files.

1. Ensure `.distignore` is up to date with all files that shouldn't be part of the build.

- To do this, run `wp dist-archive . archive.zip` then check the contents for files that shouldn't be part of the package.

1. Run `composer test` and ensure everything passes.

1. Prepare a changelog for [the Releases page on GitHub](https://github.com/johnbillion/query-monitor/releases).

- The `git changelog -x` command from [Git Extras](https://github.com/tj/git-extras) is handy for this.

### For Release

1. Bump the plugin version number:

- `npm run bump:patch` for a patch release (1.2.3 => 1.2.4)

- `npm run bump:minor` for a minor release (1.2.3 => 1.3.0)

- `npm run bump:major` for a major release (1.2.3 => 2.0.0)

1. Commit the version number changes

1. `git push origin develop`

1. Wait until (and ensure that) [the build passes](https://github.com/johnbillion/query-monitor/actions)

1. `git checkout master`

1. `git merge develop`

1. `git push origin master`

1. `git tag <version>` where `<version>` is the new version number

1. `git push origin --tags`

1. Enter the changelog into [the release on GitHub](https://github.com/johnbillion/query-monitor/releases) and publish it.

### Post Release

Publishing a release on GitHub triggers an action which deploys the release to the WordPress.org Plugin Directory. No need to touch Subversion.

The corresponding milestone on GitHub gets automatically closed via [ProBot semver](https://github.com/apps/probot-semver). New milestones are automatically created for the next major, minor, and patch releases where appropriate.

1. If this is a non-patch release, manually delete any [unused patch and minor milestones on GitHub](https://github.com/johnbillion/query-monitor/milestones) as ProBot semver doesn't handle this.

1. Check the new version has appeared [on the WordPress.org plugin page](https://wordpress.org/plugins/query-monitor/) (it'll take a few minutes).

1. Resolve relevant threads on [the plugin's support forums](https://wordpress.org/support/plugin/query-monitor/).

1. Consume tea and cake as necessary.

### Asset Updates

Assets such as screenshots and banners are stored in the `.wordpress-org` directory. These get deployed as part of the automated release process too, but will be deployed separately if they're the only changes in the master branch since the last release. This allows for the "Tested up to" value to be bumped and assets to be updated in between releases.