[PouchDB](http://pouchdb.com/) - The JavaScript Database that Syncs

==================================================

Welcome, so you are thinking about contributing to PouchDB? Awesome, this is a great place to start.

Get in Touch

------------

The following documentation should answer most of the common questions about how to get starting contributing, if you have any questions, please feel free to get in touch @ [Freenode IRC](https://www.irccloud.com/invite?channel=pouchdb&hostname=irc.freenode.net&port=6697&ssl=1), in [the Google Groups mailing list](https://groups.google.com/forum/#!forum/pouchdb), and [on StackOverflow](http://stackoverflow.com/questions/tagged/pouchdb). Or you can [tweet @pouchdb](http://twitter.com/pouchdb).

Most project discussions should happen on the Mailing list / Bug Tracker and IRC, however if you are a first time contributor and want some help getting started feel free to send a private email to any of the following maintainers:

\* Dale Harvey (dale@arandomurl.com, daleharvey on IRC)

Help Wanted

----------------

If you are looking for something to work on, we try to maintain a list of issues that should be suitable for first time contributions, they can be found tagged [good-first-bug](https://github.com/pouchdb/pouchdb/issues?labels=good-first-bug&state=open).

Guide to Contributions

--------------------------------------

\* Almost all Pull Requests for features or bug fixes will need tests

\* We follow [Felix's Node.js Style Guide](https://github.com/felixge/node-style-guide)

\* Almost all Pull Requests for features or bug fixes will need tests (seriously, it's really important)

\* Commit messages should follow the following style:

```

(#99) - A brief one line description < 50 chars

Followed by further explanation if needed, this should be wrapped at

around 72 characters. Most commits should reference an existing

issue

```

Dependencies

--------------------------------------

PouchDB needs the following to be able to build and test your build, if you haven't installed them then best to do so now, we will wait.

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

\*\*On Windows?\*\* PouchDB's build and tests work on Windows, but you will have to follow [Microsoft's guidelines for Windows](https://github.com/Microsoft/nodejs-guidelines/blob/master/windows-environment.md#environment-setup-and-configuration) to ensure you can install and compile native add-ons. Also, we recommend [Git Bash for Windows](https://git-scm.com/download/win) because our build relies on many Bash- and Unix-isms. Another option is [Windows Subsystem for Linux](https://en.wikipedia.org/wiki/Windows\_Subsystem\_for\_Linux).

Building PouchDB

--------------------------------------

All dependencies installed? Great, now building PouchDB itself is a breeze:

$ cd pouchdb

$ npm install

$ npm run build

You will now have various distributions of PouchDB in your `dist` folder, congratulations.

Note that the source code is in `src/`, which is built by [Rollup](http://rollupjs.org/) as a

Node module to `lib/`, which is then built by [Browserify](http://browserify.com/) as a browser-ready

UMD module to `dist/`. All of this logic is in `bin/build.sh`.

Testing PouchDB

--------------------------------------

The main PouchDB test suite can be run with:

$ npm test

If you would like to test against your a CouchDB instance you are currently running you can specify that with `COUCH\_HOST`:

$ COUCH\_HOST="http://127.0.0.1:5984" npm test

There is more information about the various test suites and testing options in [TESTING](./TESTING.md).

Git Essentials

--------------------------------------

Workflows can vary, but here is a very simple workflow for contributing a bug fix:

$ git clone https://github.com/pouchdb/pouchdb.git

$ git checkout -b 121-issue-keyword master

# Write tests + code

$ git add src/afile.js

$ git commit -m "(#121) - A brief description of what I changed"

Once you have some code to push, fork the [PouchDB repository](https://github.com/pouchdb/pouchdb) then push your changes to your fork:

$ git remote add myfork https://github.com/myfork/pouchdb.git

$ git push origin 121-issue-keyword

Now when you visit https://github.com/myfork/pouchdb there should be a button that will let you create a pull request.

Building PouchDB Documentation

--------------------------------------

The source for the website http://pouchdb.com is stored inside the `docs` directory of the PouchDB repository, you can make changes and submit pull requests as with any other patch. To build and view the website locally you will need to install [jekyll](http://jekyllrb.com/) and a few other gems. Jekyll is installed using [bundler](http://bundler.io/) so you need to install that first.

$ gem install bundler

$ npm run install-jekyll

If you haven't already done so, you'll also need to run `npm install` to pull in packages for the dev server:

$ npm install

Now you can build the site and start the dev server with:

$ npm run build-site

You should now find the documentation at http://127.0.0.1:4000

Writing a PouchDB Blog Post

--------------------------------------

Writing a blog post for PouchDB is exactly the same process as other contributions; all the blog posts are kept at https://github.com/pouchdb/pouchdb/tree/master/docs/\_posts. We always welcome blog posts from new contributors!

### Steps

1. Open up an issue proposing the blog post if you need help getting ideas or structuring it.

2. Add yourself as an author to https://github.com/pouchdb/pouchdb/blob/master/docs/\_data/authors.yml. (Make sure you have a [Gravatar](http://en.gravatar.com/) too.)

3. Add a new blog post with the date that you expect it will be published (we can always change it later).

4. Write something!

5. Run `npm run build-site` and you will always have a fresh version of the site at localhost:4000. You may need to Cmd-Shift-R or Ctrl-Shift-R (hard refresh) to see the latest version, since we use AppCache.

Committers!

--------------

With great power comes great responsibility yada yada yada:

\* Code is peer reviewed, you should (almost) never push your own code.

\* Please don't accidentally force push to master.

\* Cherry Pick / Rebase commits, \*\*don't use the big green button\*\*, see below for instructions on how to

merge a pull request.

\* Ensure reviewed code follows the above contribution guidelines, if it doesn't feel free to amend and make note.

\* Please try to watch when Pull Requests are made and review and / or commit them in a timely manner.

\* After you merge in a patch use tin to update the version accordingly. Run `tin -v x.x.x-prerelease` with x.x.x being the previous version upgraded appropriately via semver. When we are ready to publish to npm we can remove the `-prerelease`.

\* Thanks, you are all awesome human beings.

\*\*How to merge a pull request\*\*

\* Go to the pouchdb repository on your machine

\* Get the link to the patch of the pull request, which can be found under 'view command line instructions'

next to the green 'Merge pull request' button on the page on GitHub for the pull request

\* In your command line, run the following:

\* `curl https://patch-diff.githubusercontent.com/raw/pouchdb/pouchdb/pull/[PATCH NUMBER].patch | git am - && git push origin master`, replacing [PATCH NUMBER] with the number of the patch you want to merge.

\* Close the pull request once it has been merged, so no-one accidentally tries to merge it themselves

\* Make sure the issue associated with the pull request is closed, if the issue was resolved by that pull

request

Release Procedure

-----------------

\* Copy the last release post from ./docs/\_posts/date-pouchdb-version.md, amend date and version and fill in release notes

\* Push release post

\* `npm run set-version -- $VERSION`

\* `npm run release`

\* Copy the `dist/pouchdb\*` files from the $VERSION tag on github, paste the release notes and add the distribution files to Github Releases, rename `pouchdb.min.js` to `pouchdb-$VERSION.min.js` after you upload it.

\* Update docs/\_config.yml to the current version

\* Push updated versions to master

\* `npm run publish-site`

To do a dry run release, you can run:

DRY\_RUN=1 npm run release

To do a beta release to npm (using the dist-tag `beta`), do:

BETA=1 npm run release