Thank you for your contribution. OpenEMR (and global healthcare) continues to get better because of people like you!

The maintainers of OpenEMR want to get your pull request in as seamlessly as possible, so please ensure your code is consistent with our [development policies](https://open-emr.org/wiki/index.php/Development\_Policies).

## Code Contributions (local development)

You will need a "local" version of OpenEMR to make changes to the source code. The easiest way to do this is with [Docker](https://hub.docker.com/r/openemr/openemr/):

1. [Create your own fork of OpenEMR](https://github.com/openemr/openemr/fork) (you will need a GitHub account) and `git clone` it to your local machine.

- It's best to also add an `upstream` origin to keep your local fork up to date. [Check out this guide](https://oneemptymind.wordpress.com/2018/07/11/keeping-a-fork-up-to-date/) for more info.

- If you haven't already, [install git](https://git-scm.com/downloads) for your system

2. `cd openemr` (the directory you cloned the code into)

- If you haven't already, [install Docker](https://docs.docker.com/install/) and [install compose](https://docs.docker.com/compose/install/) for your system

- If you want to troubleshoot with the below steps easier, please also [install openemr-cmd](https://github.com/openemr/openemr-devops/tree/master/utilities/openemr-cmd) for your system

3. Run `docker-compose up` from your command line

- When the build is done, you'll see the following message:

```sh

openemr\_1 | Love OpenEMR? You can now support the project via the open collective:

openemr\_1 | > https://opencollective.com/openemr/donate

openemr\_1 |

openemr\_1 | Starting cron daemon!

openemr\_1 | Starting apache!

```

4. Navigate to `http://localhost:8300/` to login as `admin`. Password is `pass`.

5. Make changes to any files on your local file system. Most changes will appear after a refresh of the page or iFrame you're working on.

- An exception to this is if making changes to styling scripts in interface/themes/. In that case will need to clear web browser cache and run the following command to rebuild the theme files:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools build-themes'

```

6. If you wish to connect to the sql database, this docker environment provides the following 2 options:

- Navigate to `http://localhost:8310/` where you can login into phpMyAdmin.

- Or you can directly connect to port 8320 via your favorite sql tool (Mysql Workbench etc.).

- Use `username/user`: openemr, `password`: openemr .

7. Developer tools for php syntax checking, psr12 checking, and automated testing.

- To check PHP error logs:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools php-log'

```

- To create a report of PSR12 code styling issues (this takes several minutes):

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools psr12-report'

```

- To fix PSR12 code styling issues (this takes several minutes):

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools psr12-fix'

```

- To create a report of theme styling issues:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools lint-themes-report'

```

- To fix theme styling issues:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools lint-themes-fix'

```

- To check PHP parsing errors (this takes several minutes):

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools php-parserror'

```

- To run unit testing:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools unit-test'

```

- To run api testing:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools api-test'

```

- To run e2e testing:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools e2e-test'

```

- To run services testing:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools services-test'

```

- To run fixtures testing:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools fixtures-test'

```

- To run validators testing:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools validators-test'

```

- To run controllers testing:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools controllers-test'

```

- To run common testing:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools common-test'

```

8. To run the entire dev tool suite (PSR12 fix, lint themes fix, PHP parse error, unit/API/e2e/services/fixtures/validators/controllers/common tests) in one command, run

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools clean-sweep'

```

9. To run only all the automated tests (unit/API/e2e/services/fixtures/validators/controllers/common tests) in one command, run

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools clean-sweep-tests'

```

10. Developer tools to reset openemr and to load demo data.

- To reset OpenEMR only (then can reinstall manually via setup.php in web browser):

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools dev-reset'

```

- When running setup.php, need to use `mysql` for 'Server Host', `root` for 'Root Password', and `%` for 'User Hostname'.

- To reset and reinstall OpenEMR:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools dev-reset-install'

```

- To reset and reinstall OpenEMR with demo data (this includes several users with access controls setup in addition to patient portal logins. [See HERE for those credentials](https://www.open-emr.org/wiki/index.php/Development\_Demo#Demo\_Credentials).):

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools dev-reset-install-demodata'

```

- hint: this is also a great way to test any changes a developer has made to the sql upgrade stuff (ie. such as sql/5\_0\_2-to-6\_0\_0\_upgrade.sql)

11. Developer tools to backup and restore OpenEMR data (database and data on drive) via snapshots.

- Create a backup snapshot (using `example` below, but can use any alphanumeric identifier):

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools backup example'

```

- Restore from a snapshot (using `example` below, but can use any alphanumeric identifier)

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools restore example'

```

- To list the snapshots

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools list-snapshots'

```

12. Developer tools to send/receive snapshots (via capsules) that are created above in item 11.

- Here is how to grab a capsule from the docker, which can then store or share with friends.

- List the capsules:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools list-capsules'

```

- Copy the capsule from the docker to your current directory (using `example.tgz` below):

```sh

docker cp $(docker ps | grep \_openemr | cut -f 1 -d " "):/snapshots/example.tgz .

```

- Here is how to send a capsule into the docker.

- Copy the capsule from current directory into the docker (using `example.tgz` below):

```sh

docker cp example.tgz $(docker ps | grep \_openemr | cut -f 1 -d " "):/snapshots/

```

- Restore from the new shiny snapshot (using `example` below):

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools restore example'

```

- Ensure run upgrade to ensure will work with current version OpenEMR:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools upgrade 5.0.2'

```

13. Developer tools to turn on and turn off support for multisite feature.

- Turn on support for multisite:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools enable-multisite'

```

- Turn off support for multisite:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools disable-multisite'

```

14. Developer tool to change the database character set and collation (character set is the encoding that is used to store data in the database; collation are a set of rules that the database uses to sort the stored data).

- Best to demonstrate this devtool with examples.

- Set character set to utf8mb4 and collation to utf8mb4\_general\_ci (this is default for OpenEMR 6 and higher):

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools change-encoding-collation utf8mb4 utf8mb4\_general\_ci'

```

- Set character set to utf8mb4 and collation to utf8mb4\_unicode\_ci:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools change-encoding-collation utf8mb4 utf8mb4\_unicode\_ci'

```

- Set character set to utf8mb4 and collation to utf8mb4\_vietnamese\_ci:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools change-encoding-collation utf8mb4 utf8mb4\_vietnamese\_ci'

```

- Set character set to utf8 and collation to utf8\_general\_ci (this is default for OpenEMR 5 and lower):

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools change-encoding-collation utf8 utf8\_general\_ci'

```

15. Developer tools to test ssl certificate (to test client based certificates and revert back to default self signed certificate) and force/unforce https.

- To test client based certificates, create a zip package of the certificate in OpenEMR at Administration->System->Certificates. Then can import this zip package (example `ssl.zip`) into the docker via:

```sh

docker cp ssl.zip $(docker ps | grep \_openemr | cut -f 1 -d " "):/certs/

```

- To list the available certificate packages on docker:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools list-client-certs'

```

- To install and configure a certificate package (example `ssl`):

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools setup-client-cert ssl'

```

- To revert back to selfsigned certicates (ie. revert the changes required for client based certificates):

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools on-self-signed-cert'

```

- To force https in apache script via redirect:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools force-https'

```

- To revert the changes that forced https in apache script:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools un-force-https'

```

16. Developer tools to place/remove testing sql ssl certificate and testing sql ssl client key/cert.

- Place the testing sql ssl CA cert:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools sql-ssl'

```

- Remove the testing sql ssl CA cert:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools sql-ssl-off'

```

- Place the testing sql ssl CA cert and testing sql ssl client key/cert:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools sql-ssl-client'

```

- Remove the testing sql ssl CA cert and testing sql ssl client key/cert:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools sql-ssl-client-off'

```

17. CouchDB integration.

- In OpenEMR, CouchDB is an option for the patients document storage. For this reason, a CouchDB docker is included in this OpenEMR docker development environment. You can visit the CouchDB GUI directly via http://localhost:5984/\_utils/ or https://localhost:6984/\_utils/ with username `admin` and password `password`. You can configure OpenEMR to use this CouchDB docker for patient document storage in OpenEMR at Administration->Globals->Documents:

- Document Storage Method->CouchDB

- After running the following devtools, 'dev-reset', 'dev-install', 'dev-reset-install', 'dev-reset-install-demodata', 'restore-snapshot', then need to restart the couchdb docker via the following command:

```sh

docker restart $(docker ps | grep \_couchdb\_1 | cut -f 1 -d " ")

```

- Developer tools to place/remove testing couchdb ssl certificate and testing couchdb ssl client key/cert.

- Place the testing couchdb ssl CA cert:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools couchdb-ssl'

```

- Remove the testing couchdb ssl CA cert:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools couchdb-ssl-off'

```

- Place the testing couchdb ssl CA cert and testing couchdb ssl client key/cert:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools couchdb-ssl-client'

```

- Remove the testing couchdb ssl CA cert and testing couchdb ssl client key/cert:

```sh

docker exec -i $(docker ps | grep \_openemr | cut -f 1 -d " ") sh -c '/root/devtools couchdb-ssl-client-off'

```

18. Xdebug and profiling is also supported for PHPStorm.

- Firefox install xdebug helper add on (configure for PHPSTORM)

- PHPStorm Settings->Language & Frameworks->PHP->Debug

- Start listening

- Untoggle "Break at first line in PHP scripts"

- Untoggle both settings that start with "Force Break at first line..."

- Make sure port 9000 is open on your host operating system

- Profiling output can be found in /tmp directory in the docker

19. When you're done, it's best to clean up after yourself with `docker-compose down -v`

- If you don't want to build from scratch every time, just use `docker-compose down` so your next `docker-compose up` will use the cached volumes.

20. [Submit a PR](https://github.com/openemr/openemr/compare) from your fork into `openemr/openemr#master`!

We look forward to your contribution...

If you do not want to use Docker, you can always install OpenEMR directly on your local environment. This will require installing additional dependencies for your operating system. For more info see [OpenEMR Development Versions](https://open-emr.org/wiki/index.php/OpenEMR\_Installation\_Guides#OpenEMR\_Development\_Versions) on the wiki.

## Financial contributions

We also welcome financial contributions in full transparency on our [open collective](https://opencollective.com/openemr).

Anyone can file an expense. If the expense makes sense for the development of the community, it will be "merged" in the ledger of our open collective by the core contributors and the person who filed the expense will be reimbursed.

## Credits

### Contributors

Thank you to all the people who have already contributed to openemr!

<a href="https://github.com/openemr/openemr/graphs/contributors"><img src="https://opencollective.com/openemr/contributors.svg?width=890" /></a>

### Backers

Thank you to all our backers! [[Become a backer](https://opencollective.com/openemr#backer)]

<a href="https://opencollective.com/openemr#backers" target="\_blank"><img src="https://opencollective.com/openemr/backers.svg?width=890"></a>

### Sponsors

Thank you to all our sponsors! (please ask your company to also support this open source project by [becoming a sponsor](https://opencollective.com/openemr#sponsor))

<a href="https://opencollective.com/openemr/sponsor/0/website" target="\_blank"><img src="https://opencollective.com/openemr/sponsor/0/avatar.svg"></a>

<a href="https://opencollective.com/openemr/sponsor/1/website" target="\_blank"><img src="https://opencollective.com/openemr/sponsor/1/avatar.svg"></a>

<a href="https://opencollective.com/openemr/sponsor/2/website" target="\_blank"><img src="https://opencollective.com/openemr/sponsor/2/avatar.svg"></a>

<a href="https://opencollective.com/openemr/sponsor/3/website" target="\_blank"><img src="https://opencollective.com/openemr/sponsor/3/avatar.svg"></a>

<a href="https://opencollective.com/openemr/sponsor/4/website" target="\_blank"><img src="https://opencollective.com/openemr/sponsor/4/avatar.svg"></a>

<a href="https://opencollective.com/openemr/sponsor/5/website" target="\_blank"><img src="https://opencollective.com/openemr/sponsor/5/avatar.svg"></a>

<a href="https://opencollective.com/openemr/sponsor/6/website" target="\_blank"><img src="https://opencollective.com/openemr/sponsor/6/avatar.svg"></a>

<a href="https://opencollective.com/openemr/sponsor/7/website" target="\_blank"><img src="https://opencollective.com/openemr/sponsor/7/avatar.svg"></a>

<a href="https://opencollective.com/openemr/sponsor/8/website" target="\_blank"><img src="https://opencollective.com/openemr/sponsor/8/avatar.svg"></a>

<a href="https://opencollective.com/openemr/sponsor/9/website" target="\_blank"><img src="https://opencollective.com/openemr/sponsor/9/avatar.svg"></a>