# Contributing to the PHP Library for MongoDB

## Initializing the Repository

Developers who would like to contribute to the library will need to clone it and

initialize the project dependencies with [Composer](https://getcomposer.org/):

```

$ git clone https://github.com/mongodb/mongo-php-library.git

$ cd mongo-php-library

$ composer update

```

In addition to installing project dependencies, Composer will check that the

required extension version is installed. Directions for installing the extension

may be found [here](http://php.net/manual/en/mongodb.installation.php).

Installation directions for Composer may be found in its

[Getting Started](https://getcomposer.org/doc/00-intro.md) guide.

## Testing

The library's test suite uses [PHPUnit](https://phpunit.de/), which should be

installed as a development dependency by Composer.

The test suite may be executed with:

```

$ vendor/bin/phpunit

```

The `phpunit.xml.dist` file is used as the default configuration file for the

test suite. In addition to various PHPUnit options, it defines required

`MONGODB\_URI` and `MONGODB\_DATABASE` environment variables. You may customize

this configuration by creating your own `phpunit.xml` file based on the

`phpunit.xml.dist` file we provide.

## Checking coding standards

The library's code is checked using [PHP\_CodeSniffer](https://github.com/squizlabs/PHP\_CodeSniffer),

which is installed as a development dependency by Composer. Due to the PHP

requirement, the base version of the coding standard is not installed and needs

to be added manually if you plan to contributing code:

```

$ composer require --dev doctrine/coding-standard=^6.0

```

Once the coding standard has been installed, you can check the code for style

errors:

```

$ vendor/bin/phpcs

```

To automatically fix all fixable errors, use the `phpcbf` binary:

```

$ vendor/bin/phpcbf

```

## Documentation

Documentation for the library lives in the `docs/` directory and is built with

tools in the related

[mongodb/docs-php-library](https://github.com/mongodb/docs-php-library)

repository. The tools repository is already configured to reference our sources.

That said, any changes to the documentation should be tested locally before

committing. Follow the following steps to build the docs locally with the tools

repository:

\* Clone the

[mongodb/docs-php-library](https://github.com/mongodb/docs-php-library) tools

repository.

\* Install [giza](https://pypi.python.org/pypi/giza/), as noted in the tools

README.

\* Sync your working copy of the documentation to the `source/` directory with

`rsync -a --delete /path/to/mongo-php-library/docs/ source/`.

\* Build the documentation with `giza make publish`. You can suppress

informational log messages with the `--level warning` option.

\* Generated documentation may be found in the `build/master/html` directory.

## Releasing

The follow steps outline the release process for a maintenance branch (e.g.

releasing the `vX.Y` branch as X.Y.Z).

### Ensure PHP version compatibility

Ensure that the library test suite completes on supported versions of PHP.

### Transition JIRA issues and version

All issues associated with the release version should be in the "Closed" state

and have a resolution of "Fixed". Issues with other resolutions (e.g.

"Duplicate", "Works as Designed") should be removed from the release version so

that they do not appear in the release notes.

Check the corresponding ".x" fix version to see if it contains any issues that

are resolved as "Fixed" and should be included in this release version.

Update the version's release date and status from the

[Manage Versions](https://jira.mongodb.org/plugins/servlet/project-config/PHPLIB/versions)

page.

### Update version info

The PHP library uses [semantic versioning](http://semver.org/). Do not break

backwards compatibility in a non-major release or your users will kill you.

Before proceeding, ensure that the `master` branch is up-to-date with all code

changes in this maintenance branch. This is important because we will later

merge the ensuing release commits up to master with `--strategy=ours`, which

will ignore changes from the merged commits.

### Tag release

The maintenance branch's HEAD will be the target for our release tag:

```

$ git tag -a -m "Release X.Y.Z" X.Y.Z

```

### Push tags

```

$ git push --tags

```

### Merge the maintenance branch up to master

```

$ git checkout master

$ git merge vX.Y --strategy=ours

$ git push

```

The `--strategy=ours` option ensures that all changes from the merged commits

will be ignored.

### Publish release notes

The following template should be used for creating GitHub release notes via

[this form](https://github.com/mongodb/mongo-php-library/releases/new).

```

The PHP team is happy to announce that version X.Y.Z of the MongoDB PHP library is now available. This library is a high-level abstraction for the [`mongodb`](http://php.net/mongodb) extension.

\*\*Release Highlights\*\*

<one or more paragraphs describing important changes in this release>

A complete list of resolved issues in this release may be found at:

$JIRA\_URL

\*\*Documentation\*\*

Documentation for this library may be found at:

https://docs.mongodb.com/php-library/

\*\*Feedback\*\*

If you encounter any bugs or issues with this library, please report them via this form:

https://jira.mongodb.org/secure/CreateIssue.jspa?pid=12483&issuetype=1

\*\*Installation\*\*

This library may be installed or upgraded with:

composer require mongodb/mongodb

Installation instructions for the `mongodb` extension may be found in the [PHP.net documentation](http://php.net/manual/en/mongodb.installation.php).

```

The URL for the list of resolved JIRA issues will need to be updated with each

release. You may obtain the list from

[this form](https://jira.mongodb.org/secure/ReleaseNote.jspa?projectId=12483).

If commits from community contributors were included in this release, append the

following section:

```

\*\*Thanks\*\*

Thanks for our community contributors for this release:

\* [$CONTRIBUTOR\_NAME](https://github.com/$GITHUB\_USERNAME)

```

Release announcements should also be sent to the [MongoDB Product & Driver Announcements](https://community.mongodb.com/tags/c/community/release-notes/35/php-driver).

Consider announcing each release on Twitter. Significant releases should also be

announced via [@MongoDB](http://twitter.com/mongodb) as well.

### Update compatibility tables in MongoDB docs

The [compatibility tables](https://docs.mongodb.com/drivers/driver-compatibility-reference#php-driver-compatibility) in

the MongoDB documentation must be updated to account for new releases. Make sure to update both MongoDB and Language

compatibility tables, as shown in [this pull request](https://github.com/mongodb/docs-ecosystem/pull/642).