If you have a bugfix or new feature that you would like to contribute to elasticsearch-php, please find or open an issue about it first. Talk about what you would like to do. It may be that somebody is already working on it, or that there are particular issues that you should know about before implementing the change.

We enjoy working with contributors to get their code accepted. There are many approaches to fixing a problem and it is important to find the best approach before writing too much code.

The process for contributing to any of the Elasticsearch repositories is similar.

1. Sign the contributor license agreement

Please make sure you have signed the [Contributor License Agreement](https://www.elastic.co/contributor-agreement/). We are not asking you to assign copyright to us, but to give us the right to distribute your code without restriction. We ask this of all contributors in order to assure our users of the origin and continuing existence of the code. You only need to sign the CLA once.

2. Set up your fork for development

$> git clone https://github.com/elasticsearch/elasticsearch-php.git

$> cd elasticsearch-php

$> git submodule update --init --recursive

$> curl -s http://getcomposer.org/installer | php

$> php composer.phar install --dev

3. Ensure a version of Elasticsearch is running on your machine. Recommended "test" configuration is:

$> bin/elasticsearch -Des.gateway.type=none -Des.http.port=9200 \

-Des.index.store.type=memory -Des.discovery.zen.ping.multicast.enabled=false \

-Des.node.bench=true -Des.script.disable\_dynamic=false

4. Run the unit and yaml integration tests to ensure your changes do not break existing code. The `util/RestSpecRunner.php` script will automatically checkout the tests for the exact same build as is the Elasticsearch you are running on your machine (since tests are specific to the server version):

$> export ES\_TEST\_HOST='http://localhost:9200'

Then proceed to initialize the REST yaml tests and run the package. \*\*WARNING: the unit tests will clear your cluster

and data...\_do not\_ run the tests on a production cluster!\*\*

$> php util/RestSpecRunner.php

$> php vendor/bin/phpunit

5. Ensure your changes follow the [PSR-2 Coding Style Guide](http://www.php-fig.org/psr/psr-2/). You can run tools such as [PHP-CS-Fixer](http://cs.sensiolabs.org/) or [PHP\_CodeSniffer](http://pear.php.net/package/PHP\_CodeSniffer) to enforce PSR-2 automatically.

6. Rebase your changes

Update your local repository with the most recent code from the main elasticsearch-php repository, and rebase your branch on top of the latest master branch. We prefer your changes to be squashed into a single commit.

7. Submit a pull request

Push your local changes to your forked copy of the repository and submit a pull request. In the pull request, describe what your changes do and mention the number of the issue where discussion has taken place, eg “Closes #123?. Please consider adding or modifying tests related to your changes.

Then sit back and wait. There will probably be discussion about the pull request and, if any changes are needed, we would love to work with you to get your pull request merged into elasticsearch-php.