# Python Elasticsearch Client

If you have a bugfix or new feature that you would like to contribute to

elasticsearch-py, 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.

## API Code Generation

All the API methods (any method in `elasticsearch.client` classes decorated

with `@query\_params`) are actually auto-generated from the

[rest-api-spec](https://github.com/elastic/elasticsearch/tree/master/rest-api-spec/src/main/resources/rest-api-spec/api)

found in the `Elasticsearch` repository. Any changes to those methods should be

done either by submitting a PR to Elasticsearch itself (in case of adding or

modifying any of the API methods) or to the [Generate

Script](https://github.com/elastic/elasticsearch-py/blob/master/utils/generate\_api.py).

To run the code generation make sure you have pre-requisites installed:

\* by running `pip install -e '.[develop]'`

\* having the [elasticsearch](https://github.com/elastic/elasticsearch) repo

cloned on the same level as `elasticsearch-py` and switched to appropriate

version

Then you should be able to run the code generation by invoking:

```

$ python utils/generate\_api.py

```

## Contributing Code Changes

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

1. Please make sure you have signed the [Contributor License

Agreement](http://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. Run the linter and test suite to ensure your changes do not break existing code:

````

# Install Nox for task management

$ python -m pip install nox

# Auto-format and lint your changes

$ nox -s blacken

# Run the test suite

$ python setup.py test

````

See the README file in `test\_elasticsearch` directory for more information on

running the test suite.

3. Rebase your changes.

Update your local repository with the most recent code from the main

elasticsearch-py repository, and rebase your branch on top of the latest master

branch. We prefer your changes to be squashed into a single commit.

4. 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 a 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-py.