# Contributing to OWASP ZAP

ZAP is a community project, and we are always delighted to welcome new contributors!

There are lots of ways you can contribute:

## Got a Question or Problem?

If you have a question or problem relating to using ZAP then the first thing to do is to check the [Frequently Asked Questions](https://www.zaproxy.org/faq/).

We also include a comprehensive User Guide with ZAP which is also available online: https://github.com/zaproxy/zap-core-help/wiki

If they don't help then please ask on the [User Group](https://groups.google.com/group/zaproxy-users)

## Found an Issue?

If you have found a bug then raise an issue on the zaproxy repo: https://github.com/zaproxy/zaproxy/issues

Its worth checking to see if its already been reported, and including as much information as you can to help us diagnose your problem.

This FAQ explains some useful steps you can follow: https://www.zaproxy.org/faq/somethings-not-working-what-should-i-do/

## Found a Vulnerability?

If you think you have found a vulnerability in ZAP then please report it via our [bug bounty program](https://bugcrowd.com/owaspzap).

We are always very grateful to researchers who report vulnerabilities responsibly and will be very happy to give credit for the valuable assistance they provide.

## Have a Feature Request?

If you have a suggestion for new functionality then you can raise an issue on the zaproxy repo: https://github.com/zaproxy/zaproxy/issues

Its worth checking to see if its already been requested, and including as much information as you can so that we can fully understand your requirements.

## Translate ZAP to Other Languages

You can help translate the ZAP UI via the [Crowdin owasp-zap](https://crowdin.com/project/owasp-zap) project.

You can help translate the ZAP User Guide via the [Crowdin owasp-zap-help](https://crowdin.com/project/owasp-zap-help) project.

## Become a ZAP Evangelist

For information about the ZAP Evangelists and how to join up see the [ZAP Evangelists page](https://www.zaproxy.org/evangelists/)

## Help Improve the Documentation

The source for the ZAP [User Guide](https://github.com/zaproxy/zap-core-help/wiki) is underneath the zap-core-tree repo [addOns/help/src/main/javahelp/contents](https://github.com/zaproxy/zap-core-help/tree/master/addOns/help/src/main/javahelp/contents) directory.

The Java Help included with ZAP and the online version are both generated from these HTML pages. Send Pull Requests to help us improve it.

If you have a GitHub account you can contribute to the ZAP wikis.

The following resources may assist you to that end:

\* [Editing wiki pages via the online interface](https://help.github.com/articles/editing-wiki-pages-via-the-online-interface/)

\* [Adding and editing wiki pages locally (via Git)](https://help.github.com/articles/adding-and-editing-wiki-pages-locally/)

## Coding

There's always lots of coding to be done! So much so that we've split it into different categories.

Refer to [BUILDING] on how to build and run ZAP.

All code should follow the [Development Rules and Guidelines](https://github.com/zaproxy/zaproxy/wiki/DevGuidelines).

Other resources for ZAP Developers include:

\* The [Hacking ZAP blog posts](https://github.com/zaproxy/zaproxy/wiki/Development#Hacking\_ZAP)

\* The [Contributing Changes](https://github.com/zaproxy/zaproxy/wiki/Contributing-Changes) wiki page

\* The [Internal Details](https://github.com/zaproxy/zaproxy/wiki/InternalDetails) wiki pages

If you are interested in working on any of the code then the [Developer Group](https://groups.google.com/group/zaproxy-develop) is the best place to ask questions.

### Improve Existing Scan Rules or Write New Ones

The scan rules define how ZAP can automatically detect vulnerabilities.

We are always looking to improve existing ones and add new ones, so this is a great place to start helping with the ZAP code base.

### Improve Existing Add-Ons or Write New Ones

Much of the ZAP functionality is implemented as add-ons, even features that are included 'as standard' in ZAP releases.

Add-ons are a great way to extend ZAP and can be ideal for student projects - many of the existing add-ons have been implemented by students, either through programs like Google Summer of Code and the Mozilla Winter of Security or directly as part of course work.

### Improve the ZAP Core

The ZAP 'core' underpins all of the other ZAP features, and so ensuring it is as robust as possible is very important.

Fixing [issues](https://github.com/zaproxy/zaproxy/issues) is very valuable (ones flagged as [IdealFirstBug](https://github.com/zaproxy/zaproxy/issues?q=is%3Aopen+is%3Aissue+label%3AIdealFirstBug) are good ones to start on) and there are always many core improvements we want to make.

#### Guidelines for Pull Request (PR) submission and processing:

##### Developer Certificate of Origin

The Git commits must be [signed off] to indicate your agreement with [Developer Certificate of Origin].

##### Format/Style Java Code

The Java code is formatted according to Google Java Style (AOSP variant). The build automatically checks

that the code conforms to the style (using [Spotless], which delegates to [google-java-format]), it can

also be used to format the code (with the Gradle task `spotlessApply`) if the IDE/editor in use

does not support it.

##### What should you, the author of a pull request, expect from us (ZAP Team)?

\* How much time (maximum) until the first feedback? 1 week.

\* And following iterations? 1 week.

\* This is a guideline we should normally be able to hit. If it's been more than a week and you haven't heard then please feel free to add a comment to your PR and @ mention the team (@zaproxy/team-zaproxy).

##### What we (ZAP Team), expect from you?

\* "Atomic commits" (logical changes to be in a single commit). Please don't group disjointed changes into a single commit/PR.

\* Descriptive commits (subject and message):

\* For example: https://github.com/spring-projects/spring-framework/blob/master/CONTRIBUTING.md#format-commit-messages

\* Please, leave a comment after force pushing changes. It allows everyone to know that new changes are available.

\* Discussion about the changes:

\* Should be done in/on the PR or via the Dev Group and a link to that Dev Group thread added to the PR comments. (i.e.: Shared information is important, if something happens via IRC or private email please ensure a summary makes it to the PR.)

\* Discussion will be kept in the pull request unless off topic.

\* No merge commits. Please, rebase.

\* Rebase if the branch has conflicts.

\* How much time will a pull request be left open?

\* This isn't static, one or more members of the ZAP Team will reach out (using @ mentions in PR comments) once or twice in order to get things back on track. If no input is received after a month or two then the PR will be closed. Total stale time will likely be 2 to 3 months.

\* Close with a message such as: "The pull request was closed because of lack of activity (as per CONTRIBUTING guidelines)". Labeled as "Stale".

\* If the contribution is deemed important or still valuable the code may be:

\* Manually merged (if possible).

\* Retrieved by another member of the team, fixed up and resubmitted. In which case the commit message (PR message) should contain a reference to the original submission.

##### Approval process:

\* All PRs must be approved by a minimum of 2 members of the ZAP Core Team (other than the author) and anyone who is flagged as a reviewer on the PR.

\* The PR author can optionally specify any reviewer they would like to review their PR and any member of the Core Team can add themselves as a reviewer. This will effectively prevent the PR from being merged until they approve it.

\* Any member of the Core Team can merge a PR as long as the above conditions are met.

\* Reviews by people outside of the Core Team are still appreciated :)

[BUILDING]: BUILDING.md

[signed off]: https://git-scm.com/docs/git-commit#Documentation/git-commit.txt--s

[Developer Certificate of Origin]: https://developercertificate.org/

[Spotless]: https://github.com/diffplug/spotless

[google-java-format]: https://github.com/google/google-java-format