# Contributing Guide

- For minor fixes such as a typo in existing cheat sheet, a simple pull request is all that's needed. For more involved changes, please follow the process laid out below.

- :heavy\_exclamation\_mark: Focus on updating a single file in a Pull Request to make the review processes simpler for the core team.

- :warning: Pull Requests marked as \*\*WAITING\_UPDATE\*\* (indicating that the core team are waiting for an update from the author of the Pull Request) that do not receive any updates from the author in one month will be closed.

- :warning: If the assignees of an issue do not provide a Pull Request within one month then the issue will go back to the \*\*HELP\_WANTED\*\* state and assignees will be removed.

To propose changes to the existing cheat sheets or the creation of a new one, the process is as follows:

1. Create an new [issue](https://github.com/OWASP/CheatSheetSeries/issues/new/choose) using either:

- The `new\_cheatsheet\_proposal` template if you want to propose a new cheat sheet.

- The `update\_cheatsheet\_proposal` template if you want to modify a existing cheat sheet.

2. Once the issue has been discussed and approved:

1. Fork and clone this repository.

2. Either:

- Create the cheat sheet using the [new cheat sheet template](templates/New\_CheatSheet.md).

- Modify the target cheat sheet in case of an update or refactor.

3. Submit your [Pull Request](https://help.github.com/articles/creating-a-pull-request/).

4. Verify that the CI job [applied on your Pull Request](https://travis-ci.org/OWASP/CheatSheetSeries/pull\_requests) does not fail!

- If you believe they're failing due to something that's not your fault (such as another untouched file), add a comment in the Pull Request.

## Style Guide

### Markdown

- Use the markdown syntax described in this [guide](https://guides.github.com/features/mastering-markdown/).

- Use `\*\*bold\*\*` syntax for \*\*bold\*\* text.

- Lists and nested lists should use `-` strictly.

- Avoid the use of HTML in the cheat sheets (stick to pure Markdown).

- Quotes from other articles should use quote syntax: `> Quote here`

- If you use `{{` or `}}` pattern in code fencing then add a space between both curly braces (ex: `{ {`).

- Cheat Sheet filenames should only contain letters, numbers, hyphens and underscores.

- Store all assets in the \*\*assets\*\* folder and use the following syntax:

- `![ALTERNATE\_NAME](../assets/ASSET\_NAME.png)` for images (which should be in the PNG format).

- `[ALTERNATE\_NAME](../assets/ASSET\_NAME.EXT)` for other types of files.

- Use this [site](https://www.tablesgenerator.com/markdown\_tables) for generation of tables.

- Links should be inline with a useful description, such as `[Description](https://example.org)`.

- Always use HTTPS links where possible

- Code snippets should be short and should be appropriately marked to provide syntax highlighting:

```md

```php

<?php

echo "Example code";

```

```

### Content

The intended audience of the cheat sheets is developers, \_not\_ security experts. As such, do not assume that the person reading the cheat sheet has a strong understanding of security topics. In depth or academic discussions are generally not appropriate in cheat sheets, and should be linked to as external references where appropriate.

The purpose of the cheat sheets is to provide \*\*useful, practical advice\*\* that can be followed by developers. It is much better to give \_good\_ practices that can actually be followed than \_best\_ practices that are completely impractical

When submitting changes in a PR, consider the following areas:

- The content should be useful to developers.

- The content should be factual and correct.

- Statements should be supported by authoritative references where possible.

- Recommendations should be feasible for the majority of developers to implement.

### Structure

- Start with a H1 of the cheat sheet name

- The first section of the cheat sheet should be an introduction which briefly sums up the contents, and provides a short list of key bullet points.

- The table of contents will be automatically generated on the site, so does not need to be added as a section.

- Headings should have a blank line after them.

### Language

- Use US English.

- Spell check before submitted a PR.

- Try and keep the language relatively simple to make it easier for non-native speakers

- Define any non-ubiquitous acronyms when they are first used.

- This is not necessary for extremely common acronyms such as "HTTP" or "URL".

## How to setup my contributor environment

Follow these steps:

1. Install [Visual Studio Code (VSCode)](https://code.visualstudio.com/).

2. Install the [vscode-markdownlint plugin](https://github.com/DavidAnson/vscode-markdownlint#install).

3. Open the file [Project.code-workspace](Project.code-workspace) from VSCode via the menu `File > Open Workspace...`.

4. You are ready to contribute :+1:

:alarm\_clock: What to verify before pushing the updates?

1. Ensure that the markdown files you have created or modified do not have any warnings/errors raised by the linter. You can see it in this bottom bar when the markdown file is opened in VSCode:

![PluginWarningUI](assets/README\_PluginWarningUI.png)

2. Ensure that the markdown file you have created/modified do not have any dead links. You can verify that by using this [plugin](https://www.npmjs.com/package/markdown-link-check). If you cannot use this plugin then, verify that all the links you have changed or added are valid before pushing.

1. Install [NodeJS](https://nodejs.org/en/download/) to install NPM.

2. Install the validation plugin via the command `npm install -g markdown-link-check`

3. Use this command (from the repository root folder) on your markdown file to verify the presence of any dead links:

```bash

markdown-link-check -c .markdownlinkcheck.json [MD\_FILE]

```

The should produce output similar to the below. Any identified dead links are shown using a red cross instead of a green tick before the link.

```bash

$ markdown-link-check -c .markdownlinkcheck.json cheatsheets/Transaction\_Authorization\_Cheat\_Sheet.md

FILE: cheatsheets/Transaction\_Authorization\_Cheat\_Sheet.md

[?] https://en.wikipedia.org/wiki/Time-based\_One-time\_Password\_Algorithm

[?] https://en.wikipedia.org/wiki/Chip\_Authentication\_Program

[?] http://www.cl.cam.ac.uk/~sjm217/papers/fc09optimised.pdf

...

```