# Contributing to Web Essentials

Looking to contribute something to Web Essentials? \*\*Here's how you can help.\*\*

Please take a moment to review this document in order to make the contribution

process easy and effective for everyone involved.

Following these guidelines helps to communicate that you respect the time of

the developers managing and developing this open source project. In return,

they should reciprocate that respect in addressing your issue or assessing

patches and features.

## Using the issue tracker

The [issue tracker](https://github.com/madskristensen/WebEssentials2013/issues) is

the preferred channel for [bug reports](#bug-reports), [features requests](#feature-requests)

and [submitting pull requests](#pull-requests), but please respect the following

restrictions:

\* Please \*\*do not\*\* use the issue tracker for personal support requests. Stack

Overflow ([`web-essentials`](http://stackoverflow.com/questions/tagged/web-essentials) tag)

is a better place to get help.

\* Please \*\*do not\*\* derail or troll issues. Keep the discussion on topic and

respect the opinions of others.

\* Please \*\*do not\*\* open issues or pull requests which \*belongs to\* the third party

components such as SignalR, LESS, SASS, CoffeeScript, LiveScript, TypeScript, Node.js,

etc.

## Bug reports

A bug is a \_demonstrable problem\_ that is caused by the code in the repository.

Good bug reports are extremely helpful, so thanks!

Guidelines for bug reports:

1. \*\*Use the GitHub issue search\*\* &mdash; check if the issue has already been

reported.

2. \*\*Check if the issue has been fixed\*\* &mdash; try to reproduce it using the

latest `master` or development branch in the repository.

3. \*\*Isolate the problem\*\* &mdash; ideally create an

[SSCCE](http://www.sscce.org/) and a live example.

Uploading the project on cloud storage (OneDrive, DropBox, et el.)

or creating a sample GitHub repository is also helpful.

A good bug report shouldn't leave others needing to chase you up for more

information. Please try to be as detailed as possible in your report. What is

your environment? What steps will reproduce the issue? What browser(s) and OS

experience the problem? Do other browsers show the bug differently? What

would you expect to be the outcome? All these details will help people to fix

any potential bugs.

Example:

> Short and descriptive example bug report title

>

> A summary of the issue and the Web Essentials, Visual Studio, browser, OS environments

> in which it occurs. If suitable, include the steps required to reproduce the bug.

>

> 1. This is the first step

> 2. This is the second step

> 3. Further steps, etc.

>

> `<url>` - a link to the project/file uploaded on cloud storage or other publicly accessible medium.

>

> Any other information you want to share that is relevant to the issue being

> reported. This might include the lines of code that you have identified as

> causing the bug, and potential solutions (and your opinions on their

> merits).

## Feature requests

Feature requests are welcome. But take a moment to find out whether your idea

fits with the scope and aims of the project. It's up to \*you\* to make a strong

case to convince the project's developers of the merits of this feature. Please

provide as much detail and context as possible.

## Pull requests

Good pull requests?patches, improvements, new features?are a fantastic

help. They should remain focused in scope and avoid containing unrelated

commits.

\*\*Please ask first\*\* before embarking on any significant pull request (e.g.

implementing features, refactoring code, porting to a different language),

otherwise you risk spending a lot of time working on something that the

project's developers might not want to merge into the project.

Please adhere to the [coding guidelines](#code-guidelines) used throughout the

project (indentation, accurate comments, etc.) and any other requirements

(such as test coverage).

Adhering to the following process is the best way to get your work

included in the project:

1. [Fork](http://help.github.com/fork-a-repo/) the project, clone your fork,

and configure the remotes:

```bash

# Clone your fork of the repo into the current directory

git clone https://github.com/<your-username>/WebEssentials2013.git

# Navigate to the newly cloned directory

cd WebEssentials2013

# Assign the original repo to a remote called "we2013" (or "upstream")

git remote add we2013 https://github.com/madskristensen/WebEssentials2013.git

```

2. If you cloned a while ago, get the latest changes from upstream:

```bash

git checkout master

git pull we2013 master

```

3. Create a new topic branch (off the main project development branch) to

contain your feature, change, or fix:

```bash

git checkout -b <topic-branch-name>

```

4. Commit your changes in logical chunks. Please adhere to these [git commit

message guidelines](http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html)

or your code is unlikely be merged into the main project. Use Git's

[interactive rebase](https://help.github.com/articles/interactive-rebase)

feature to tidy up your commits before making them public. Also, prepend name of the feature

to the commit message. For instance: "SCSS: Fixes compiler results for IFileListener.\nFixes `#123`"

5. Locally merge (or rebase) the upstream development branch into your topic branch:

```bash

git pull [--rebase] we2013 master

```

6. Push your topic branch up to your fork:

```bash

git push origin <topic-branch-name>

```

7. [Open a Pull Request](https://help.github.com/articles/using-pull-requests/)

with a clear title and description against the `master` branch.

## Code guidelines

- Always use proper indentation.

- In Visual Studio under `Tools > Options > Text Editor > C# > Advanced`, make sure

`Place 'System' directives first when sorting usings` option is enabled (checked).

- Before committing, organize usings for each updated C# source file. Either you can

right-click editor and select `Organize Usings > Remove and sort` OR use extension

like [BatchFormat](http://visualstudiogallery.msdn.microsoft.com/a7f75c34-82b4-4357-9c66-c18e32b9393e).

- Before committing, run Code Analysis in `Debug` configuration and follow the guidelines

to fix CA issues. Code Analysis commits can be made separately.