# CONTRIBUTING

So you want to help out? Great! There's a number of ways you can get involved.

## Table of Contents

\* [Other repositories where issues could be filed](#other-repositories-where-issues-could-be-filed)

\* [Filing issues](#filing-issues)

\* [Reporting a Bug](#reporting-a-bug)

\* [Requesting a Feature](#requesting-a-feature)

\* [Labels](#labels)

\* [Contributing code](#contributing-code)

\* [Building video.js locally](#building-videojs-locally)

\* [Forking and cloning the repository](#forking-and-cloning-the-repository)

\* [Installing local dependencies](#installing-local-dependencies)

\* [Running tests](#running-tests)

\* [Building videojs](#building-videojs)

\* [Testing Locally](#testing-locally)

\* [Sandbox test directory](#sandbox-test-directory)

\* [Running a local web server](#running-a-local-web-server)

\* [Watching source and test changes](#watching-source-and-test-changes)

\* [Making Changes](#making-changes)

\* [Step 1: Verify](#step-1-verify)

\* [Step 2: Update remote](#step-2-update-remote)

\* [Step 3: Branch](#step-3-branch)

\* [Step 4: Commit](#step-4-commit)

\* [Step 5: Test](#step-5-test)

\* [Step 6: Push](#step-6-push)

\* [Code Style Guide](#code-style-guide)

\* [Developer's Certificate of Origin 1.1](#developers-certificate-of-origin-11)

\* [Doc Credit](#doc-credit)

## Other repositories where issues could be filed

There's also other Video.js projects where you can help. (check the [video.js org](https://github.com/videojs) for an up-to-date list of projects)

\* [Videojs.com](https://github.com/videojs/videojs.com)

\* [Video.js flash player](https://github.com/videojs/videojs-flash)

\* [HLS](https://github.com/videojs/videojs-contrib-hls)

\* [DASH](https://github.com/videojs/videojs-contrib-dash)

\* [Youtube Tech](https://github.com/videojs/videojs-youtube)

\* [Vimeo Tech](https://github.com/videojs/videojs-vimeo)

\* [Ads](https://github.com/videojs/videojs-contrib-ads)

\* [Plugin generator](https://github.com/videojs/generator-videojs-plugin)

\* [Linter][linter]

## Filing issues

[GitHub Issues](https://github.com/videojs/video.js/issues) are used for all discussions around the codebase, including \*\*bugs\*\*, \*\*features\*\*, and other \*\*enhancements\*\*.

When filling out an issue, make sure to fill out the questions in the

### Reporting a Bug

\*\*A bug is a demonstrable problem\*\* that is caused by the code in the repository. Good bug reports are extremely helpful. Thank You!

Guidelines for bug reports:

1. If your issue is with a particular video.js plugin or subproject, please open an issue against that project. See [list of some potential other projects above](#other-repositories-where-issues-could-be-filed)

1. Use the [GitHub issue search](https://github.com/videojs/video.js/issues) — check if the issue has already been reported.

1. Check if the issue has already been fixed — try to reproduce it using the latest `master` branch in the repository.

1. Isolate the problem — \*\*create a [reduced test case](https://css-tricks.com/reduced-test-cases/)\*\* with a live example. You can possibly use [this codepen template](https://codepen.io/gkatsev/pen/GwZegv?editors=1000#0) as a starting point -- don't forget to update it to the videojs version you use.

1. Answer all questions in the [issue template][]. The questions in the issue template are designed to try and provide the maintainers with as much information possible to minimize back-and-forth to get the issue resolved.

A good bug report should be as detailed as possible, so that others won't have to follow up for the essential details.

\*\*[File a bug report](https://github.com/videojs/video.js/issues/new)\*\*

### Requesting a Feature

1. [Check the plugin list](https://videojs.com/plugins/) for any plugins that may already support the feature.

1. [Search the issues](https://github.com/videojs/video.js/issues) for any previous requests for the same feature, and give a thumbs up or +1 on existing requests.

1. If no previous requests exist, create a new issue. Please be as clear as possible about why the feautre is needed and the intended use case.

1. Once again, be as details as possible and follow the [issue template][]

\*\*[Request a feature](https://github.com/videojs/video.js/issues/new)\*\*

### Labels

There are a few labels that might be added to your issue or PR by a maintainer. Here's a quick rundown of what they mean:

| Label | Issue or PR | Description |

| ------------------------ | ------------ | -------------------------------------------------------------------------------------------------------------------------------------------------------- |

| confirmed | Issue and PR | Issue: marks as reproducible by a maintainer. PR: marked by a maintainer as ready to be merged |

| 5.x | PR | Marks as a change to the 5.x branch only |

| bug | Issue | Marks as a confirmed bug by a maintainer |

| good first issue | Issue | Marked as a good bug or enhancement for first time contributors to Video.js |

| first-timers-only | Issue | Marked as a good bug or enhancement to be done by a newcomer to open source |

| minor, patch, major | PR | Marks PR with the expected [semver](https://semver.org/) classification of the change |

| needs: more info | Issue | Marked by a maintainer as needing more information from the issue reporter. Please update your issue with more information to help us reproduce the bug. |

| needs: reduced test case | Issue | Marked by a maintainer as needing a reduced test case from the issue reporter. Please create a test page that we can inspect to help us indentify a bug. |

## Contributing code

To contibute code you'll need to be able to build a copy of Video.js and run tests locally. There are a few requirements before getting started.

\* Node.js

Video.js uses Node for build and test automation. Node is available for Windows, Mac OS X, Linux, and SunOS, as well as source code if that doesn't scare you. [Download and install Node.js](http://nodejs.org/download/)

### Building video.js locally

#### Forking and cloning the repository

First, [fork](http://help.github.com/fork-a-repo/) the video.js git repository. At the top of every GitHub page, there is a Fork button. Click it, and the forking process will copy Video.js into your own GitHub account.

Clone your fork of the repo into your code directory

```sh

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

```

Navigate to the newly cloned directory

```sh

cd video.js

```

Assign the original repo to a remote called "upstream"

```sh

git remote add upstream https://github.com/videojs/video.js.git

```

> In the future, if you want to pull in updates to video.js that happened after you cloned the main repo, you can run:

>

> ```sh

> git remote update

> git checkout master

> git pull upstream master

> ```

#### Installing local dependencies

Install the required node.js modules using node package manager

```sh

npm install

```

> A note to Windows developers: If you run npm commands, and you find that your command prompt colors have suddenly reversed, you can configure npm to set color to false to prevent this from happening.

> `npm config set color false`

> Note that this change takes effect when a new command prompt window is opened; the current window will not be affected.

#### Running tests

Tests can be run either from the shell or from the browser.

To run the tests from the shell, just run

```sh

npm test

```

This will build video.js locally and run the test suite using [Karma](https://karma-runner.github.io/1.0/index.html), which runs our tests in actual browsers.

To run tests from the browser, first start a local server with `npm start` (this also watches for changes and rebuilds video.js and the test files as necessary). Then navigate to `http://localhost:9999/test`, and you'll see a page that displays the results of all the tests. To rerun the tests after making changes, just refresh the page. To run an individual test, click the "Rerun" link next to the test's title.

#### Building videojs

To build video.js, simply run

```sh

npm run build

```

This outputs an `es5/` and `dist/` folder. The `es5/` folder is used by bundling tools like browserify and webpack to package video.js into projects. The `dist/` folder has pre-compiled versions of video.js, including a minified version and the CSS file. This file can be included in page via a `<script></script>` tag.

#### Testing Locally

Besides running automated tests, you often want to run video.js manually and play around with things as you're developing. A few things are provided to make it easier.

#### Sandbox test directory

There's a sandbox directory where you can add any file and it won't get tracked in git. To start you can copy the example index file.

```sh

cp sandbox/index.html.example sandbox/index.html

```

See [the following section](#running-a-local-web-server) for how to open the page in a browser.

#### Running a local web server

This ties in nicely with the sandbox directory. You can always open the `sandbox/index.html` file directly but in some cases it may not work properly.

> Flash files (`.swf`) that are local and loaded into a locally accessed page (file:///) will NOT run.

> To get around this you must use a local web server.

To run the local webserver:

```sh

npm start

open http://localhost:9999/sandbox/index.html

```

The latter does some extra work which will be described in the next section.

#### Watching source and test changes

As you're developing, you want the build to re-run and update itself, and potentially re-run the tests. In addition, you want to launch a local web-server that you can open the `sandbox` directory in.

To do so, you just need to run

```sh

npm start

```

This sets up the local webserver using connect and then watches source files, test files, and CSS files for you and rebuilds things as they happen.

### Making Changes

#### Step 1: Verify

Whether you're adding something new, making something better, or fixing a bug, you'll first want to search the [GitHub issues](https://github.com/videojs/video.js/issues) and [plugins list](https://github.com/videojs/video.js/wiki/Plugins) to make sure you're aware of any previous discussion or work. If an unclaimed issue exists, claim it via a comment. If no issue exists for your change, submit one, follwing the [issue filing guidelines](#filing-issues).

#### Step 2: Update remote

Before starting work, you want to update your local repository to have all the latest changes.

```sh

git remote update

git checkout master

git rebase upstream/master

```

#### Step 3: Branch

You want to do your work in a separate branch.

```sh

git checkout -b my-branch

```

#### Step 4: Commit

Commit changes as you go. Write thorough descriptions of your changes in your commit messages.

For more information see our [conventional changelog guidelines for video.js](https://github.com/videojs/conventional-changelog-videojs/blob/master/convention.md)

Follow these guidelines:

1. The first line should be less than 50 characters and contain a short description of the commit.

1. The body should contain a more detailed description. It can contain things like reasoning for the change and specifics of what changed.

1. A footer can be added if this fixes a particular issue on GitHub.

```sh

git add src/js/player.js

git commit

```

An example of the first line of a commit message: `fix: changed the footer to correctly display foo`

In the body of the commit message, we can talk about why we made the change. What the change entails.

Any testing considerations or things to think about when looking at the commit. For Example:

```txt

fix: one line commit explanation

In the body of the commit message, we can talk about why we made the change. What the change entails.

Any testing considerations or things to think about when looking at the commit.

Fixes #123. The footer can contain Fixes messages.

```

> Make sure that git knows your name and email:

>

> ```sh

> git config --global user.name "Random User"

> git config --global user.email "random.user@example.com"

> ```

#### Step 5: Test

Any code change should come with corresponding test changes. Especially bug fixes.

Tests attached to bug fixes should fail before the change and succeed with it.

```sh

npm test

```

See [Running tests](#running-tests) for more information.

#### Step 6: Push

```sh

git push origin my-branch

```

Then go to the [repo page](https://github.com/videojs/video.js) and click the "Pull Request" button and fill out the [pull request template](/.github/PULL\_REQUEST\_TEMPLATE.md)

### Code Style Guide

Our javascript is linted using [videojs-standard][linter].

## [Developer's Certificate of Origin 1.1](https://github.com/nodejs/node/blob/master/CONTRIBUTING.md#developers-certificate-of-origin-11)

By making a contribution to this project, I certify that:

\* (a) The contribution was created in whole or in part by me and I

have the right to submit it under the open source license

indicated in the file; or

\* (b) The contribution is based upon previous work that, to the best

of my knowledge, is covered under an appropriate open source

license and I have the right under that license to submit that

work with modifications, whether created in whole or in part

by me, under the same open source license (unless I am

permitted to submit under a different license), as indicated

in the file; or

\* (c) The contribution was provided directly to me by some other

person who certified (a), (b) or (c) and I have not modified

it.

\* (d) I understand and agree that this project and the contribution

are public and that a record of the contribution (including all

personal information I submit with it, including my sign-off) is

maintained indefinitely and may be redistributed consistent with

this project or the open source license(s) involved.

## Doc Credit

This doc was inspired by some great contribution guide examples including [contribute.md template](https://github.com/contribute-md/contribute-md-template),

[grunt](https://github.com/gruntjs/grunt/wiki/Contributing),

[html5 boilerplate](https://github.com/h5bp/html5-boilerplate/blob/master/CONTRIBUTING.md),

[jquery](https://github.com/jquery/jquery/blob/master/CONTRIBUTING.md),

and [node.js](https://github.com/nodejs/node/blob/master/CONTRIBUTING.md).

[issue template]: /.github/ISSUE\_TEMPLATE.md

[linter]: https://github.com/videojs/standard