

# How to make a libFuzzer fuzzer in V8

This document describes how to make a new libFuzzer fuzzer for V8. A general introduction to libFuzzer can be found [here](#). In short, libFuzzer is an in-process coverage-driven evolutionary fuzzer. libFuzzer serves you with a sequence of byte arrays that you can use to test your code. libFuzzer tries to generate this sequence of byte arrays in a way that maximizes test coverage.

**Warning:** By itself libFuzzer typically does not generate valid JavaScript code.

## Changes to V8

**tldr:** Do the same as <https://codereview.chromium.org/2280623002> to introduce a new fuzzer to V8.

This is a step by step guide on how to make a new fuzzer in V8. In the example the fuzzer is called `foo`.

1. Copy one of the existing fuzzer implementations in [test/fuzzer/](#), e.g. `cp wasm.cc foo.cc`
  - Copying an existing fuzzer is a good idea to get all the required setup, e.g. setting up the isolate
2. Create a directory called `foo` in [test/fuzzer/](#) which contains at least one file
  - The file is used by the trybots to check whether the fuzzer actually compiles and runs
3. Copy the build rules of an existing fuzzer in [BUILD.gn](#), e.g. the build rules for the [wasm.cc](#) fuzzer are `v8_source_set("wasm_fuzzer")` and `v8_fuzzer("wasm_fuzzer")`. Note that the name has to be the name of the directory created in Step 2 + `_fuzzer` so that the scripts on the trybots work
4. Now you can already compile the fuzzer, e.g. with `ninja -j 1000 -C out/x64.debug/v8_simple_foo_fuzzer`
  - Use this binary to reproduce issues found by cluster fuzz, e.g. `out/x64.debug/v8_simple_foo_fuzzer testcase.foo`
5. Copy the binary name and the test directory name in [test/fuzzer/fuzzer.isolate](#)
6. Add the fuzzer to the FuzzerTestSuite in [test/fuzzer/testcfg.py](#)
  - This step is needed to run the fuzzer with the files created in Step 2 on the trybots
7. Commit the changes described above to the V8 repository

## Changes to Chromium

**tldr:** Do the same as <https://codereview.chromium.org/2344823002> to add the new fuzzer to cluster fuzz.

1. Copy the build rules of an existing fuzzer in [testing/libfuzzer/fuzzers/BUILD.gn](#), e.g. the build rule for the [wasm.cc](#) fuzzer is `v8_wasm_fuzzer`. There is no need to set a `dictionary`, or a `seed_corpus`. See [chromium-fuzzing-getting-started](#) for more information.
2. Compile the fuzzer in chromium (for different configurations see: <https://chromium.googlesource.com/chromium/src/+master/testing/libfuzzer/reference.md>):
  - `gn gen out/libfuzzer '--args=use_libfuzzer=true is_asan=true is_debug=false enable_nacl=false'`
  - `ninja -j 1000 -C out/libfuzzer/ v8_foo_fuzzer`

### 3. Run the fuzzer locally

- `mkdir /tmp/empty_corpus && out/libfuzzer/v8_foo_fuzzer /tmp/empty_corpus`