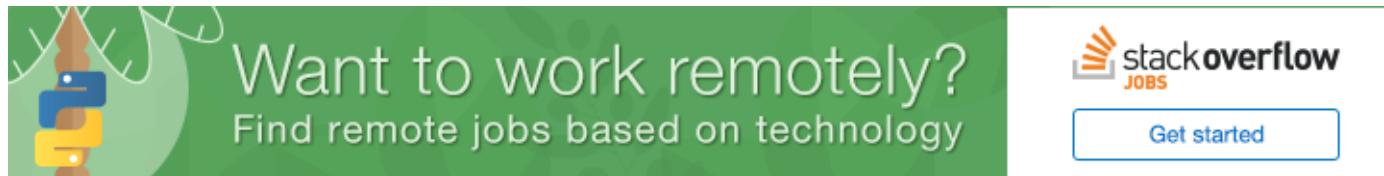Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

The results are in! **See the 2018 Developer Survey results. »**   ✕

C++ project with Bazel and GTest

I want to create a Bazel C++ project with gtest for unit tests.

What is the minimal setup?

(I only have Bazel installed on my computer and I am running under Linux)

c++   googletest   bazel

asked Aug 22 '17 at 10:05

Picaud Vincent
2,603   1   7   26

1 Answer

The project structure is:

```
.
├── bin
│   ├── BUILD
```

```
├── test
│   ├── BUILD
│   ├── message_test.cpp
│   ├── ...
├── gmock.BUILD
└── WORKSPACE
```

Files related to Bazel+GTest

- **WORKSPACE**

There you download gtest from github:

```
new_git_repository(
    name = "googletest",
    build_file = "gmock.BUILD",
    remote = "https://github.com/google/googletest",
    tag = "release-1.8.0",
)
```

You define a gmock BUILD file defined below:

- **gmock.BUILD**

This BUILD file is in charge of compiling gtest/gmock:

```
cc_library(
    name = "gtest",
    srcs = [
        "googletest/src/gtest-all.cc",
        "googlemock/src/gmock-all.cc",
    ],
    hdrs = glob([
        "**/*.h",
        "googletest/src/*.cc",
        "googlemock/src/*.cc",
    ]),
    includes = [
        "googlemock",
        "googletest",
        "googletest/include",
        "googlemock/include",
    ],
    linkopts = ["-pthread"],
    visibility = ["//visibility:public"],
```

```
cc_library(
    name = "gtest_main",
    srcs = ["googlemock/src/gmock_main.cc"],
    linkopts = ["-pthread"],
    visibility = ["//visibility:public"],
    deps = [":gtest"],
)
```

- **test/BUILD**

This build file generate the tests:

```
cc_test(
    name = "MyTest",
    srcs = glob(["**/*.cpp"]),
    deps = ["//MyLib:MyLib",
            "@googletest//:gtest_main"],
)
```

The **test/message_test.cpp** file is defined by:

```
#include "gtest/gtest.h"

#include "MyLib/message.hpp"

TEST(message_test,content)
{
  EXPECT_EQ(get_message(),"Hello World!");
}
```

And that is all! The other files are defined as usual:

Files for the supporting example

- **MyLib/BUILD**

Creates the **libMyLib.so** and **libMyLib.a** libraries.

```
cc_library(
    name="MyLib",
    hdrs=glob(["**/*.hpp"]),
    srcs=glob(["**/*.cpp"]),
    visibility = ["//visibility:public"],
)
```

```
#include <string>

std::string get_message();
```

and **message.cpp**

```
#include "MyLib/message.hpp"

std::string get_message()
{
    return "Hello World!";
}
```

example.

- **bin/BUILD**

Creates the *hello* executable.

```
cc_binary(
    name = "hello",
    srcs = ["hello.cpp"],
    deps = ["//MyLib:MyLib"],
)
```

which is:

```
#include "MyLib/message.hpp"

#include <iostream>

int main()
{
  std::cout << "\n" << get_message() << std::endl;

  return EXIT_SUCCESS;
}
```

Usage:

- Compiles all targets:

This will also download gtest from its github repo and compile it

- Checks the hello target:

You can run it with:

```
bazel run bin:hello
```

- Running your tests using GTest

That was the main point of this note:

```
bazel test ... --test_output=errors
```

You should get something like:

```
INFO: Analysed 3 targets (0 packages loaded).
INFO: Found 2 targets and 1 test target...
INFO: Elapsed time: 0.205s, Critical Path: 0.05s
INFO: Build completed successfully, 2 total actions
//test:MyTest
PASSED in 0.0s
Executed 1 out of 1 test: 1 test passes.
```

Reproduce the results

For your ease I have created a github repo containing this example. I hope it works out of the box.

edited Aug 22 '17 at 19:38   answered Aug 22 '17 at 10:05

Picaud Vincent
2,603   1   7   26

If you also need some data files for your tests, stackoverflow.com/questions/46100906/… describes how to manage that. – Picaud Vincent   Sep 9 '17 at 14:38