**GO-CONVEY TESTING**

Go Convey supports Go's native [**testing**](http://golang.org/pkg/testing/) package. Neither the web UI nor the DSL are required; you can use either one independently.

**Features:**

* Directly integrates with go test
* Huge suite of regression tests
* Readable, colorized console output (understandable by any manager, IT or not)
* Fully-automatic web UI (works with native Go tests, too)
* Test code generator
* Idler script automatically runs tests in the terminal
* Go convey port- http://127.0.0.1:8080/

**JUMP TO:**

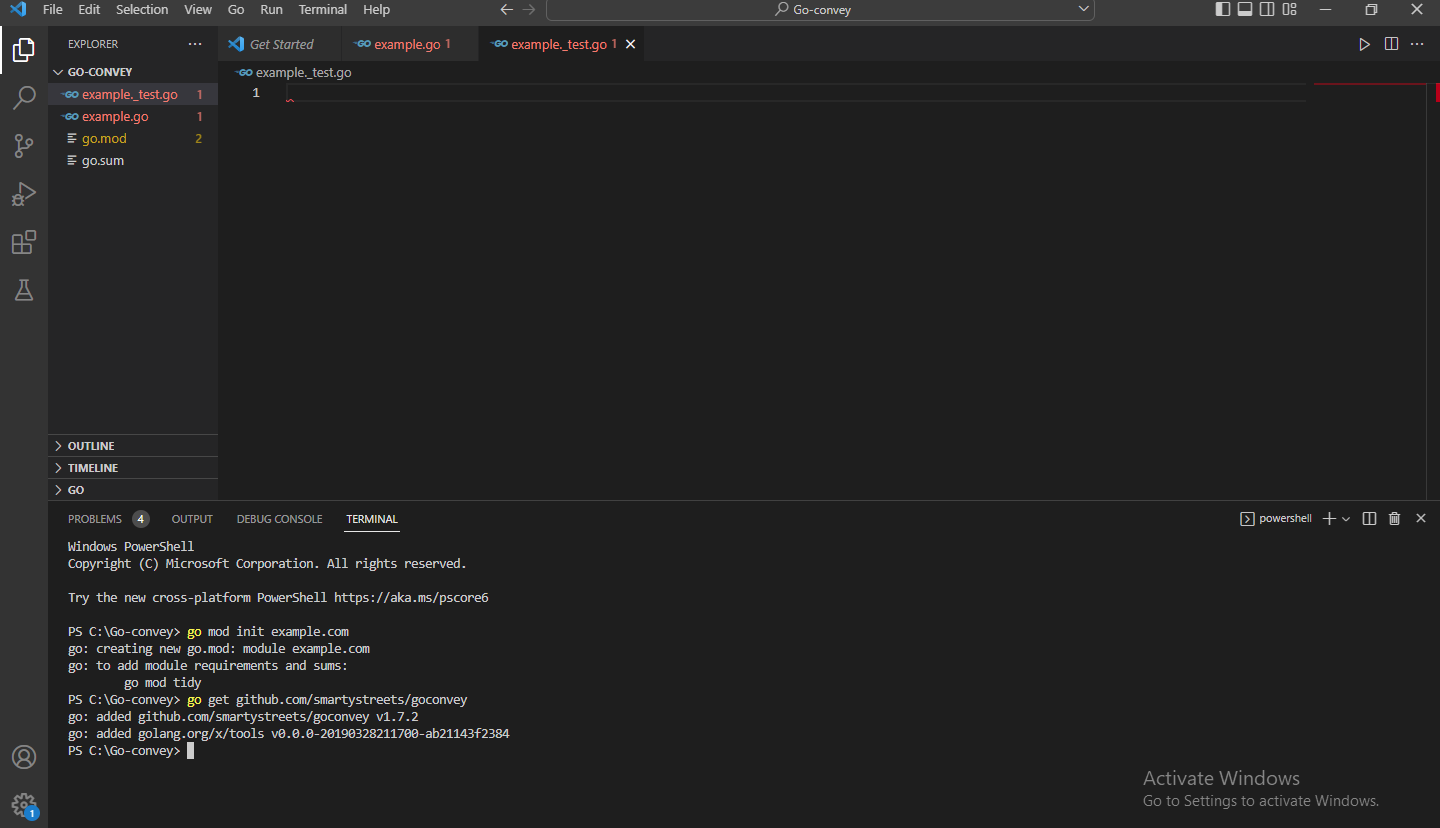
* [Installation](https://chromium.googlesource.com/external/github.com/smartystreets/goconvey/+/refs/tags/1.3.1/README.md#installation)
* Quick start
* Web UI
* [Terminal output](https://chromium.googlesource.com/external/github.com/smartystreets/goconvey/+/refs/tags/1.3.1/README.md#terminal-output-with-go-test--v)
* [Writing tests](https://chromium.googlesource.com/external/github.com/smartystreets/goconvey/+/refs/tags/1.3.1/README.md#writing-tests)
* Execution
* [Assertions](https://chromium.googlesource.com/external/github.com/smartystreets/goconvey/+/refs/tags/1.3.1/README.md#assertions)

[**Installation**](https://chromium.googlesource.com/external/github.com/smartystreets/goconvey/+/refs/tags/1.3.1/README.md#installation)

1. Open Visual studio code and Create New folder.
2. Create New mod – go mod init <example.com>.
3. Get and Install below commands.

go get github.com/smartystreets/goconvey

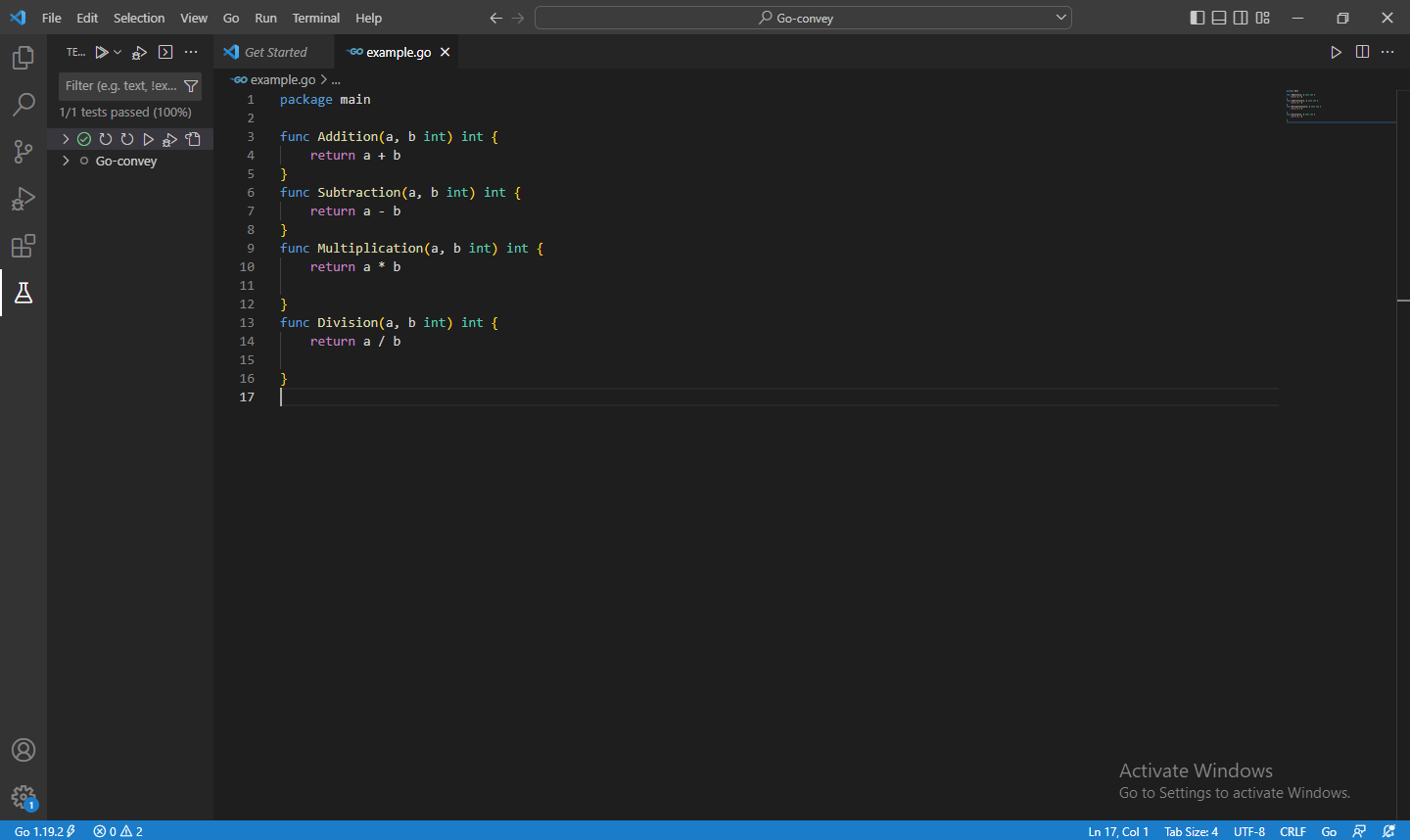
go install github.com/smartystreets/goconvey/web/server



**Quick start**

Make a test, for example:

1. Make a code in Example.go, having simple math functions.

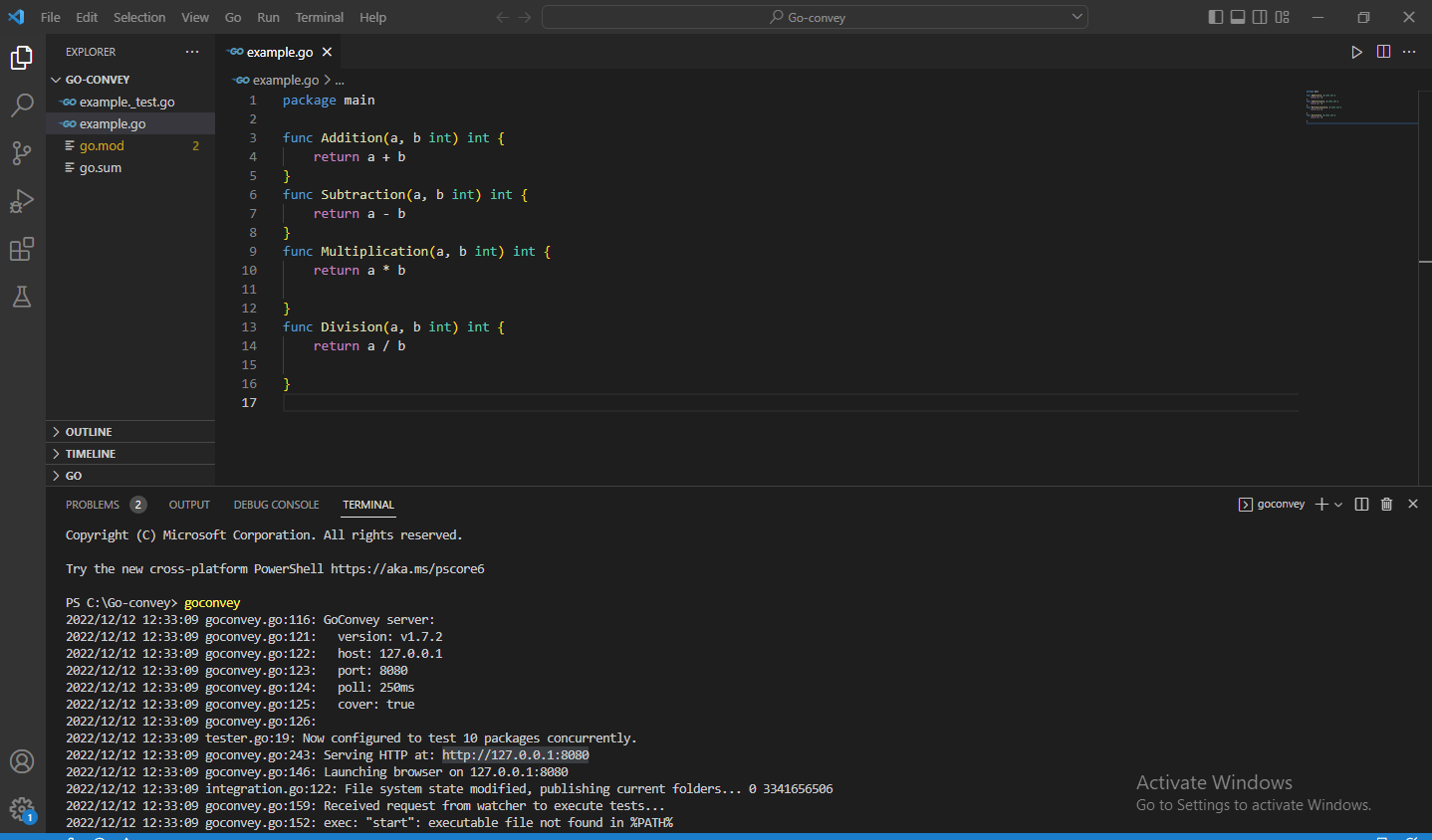
****

1. Open new terminal – go convey.

goconvey

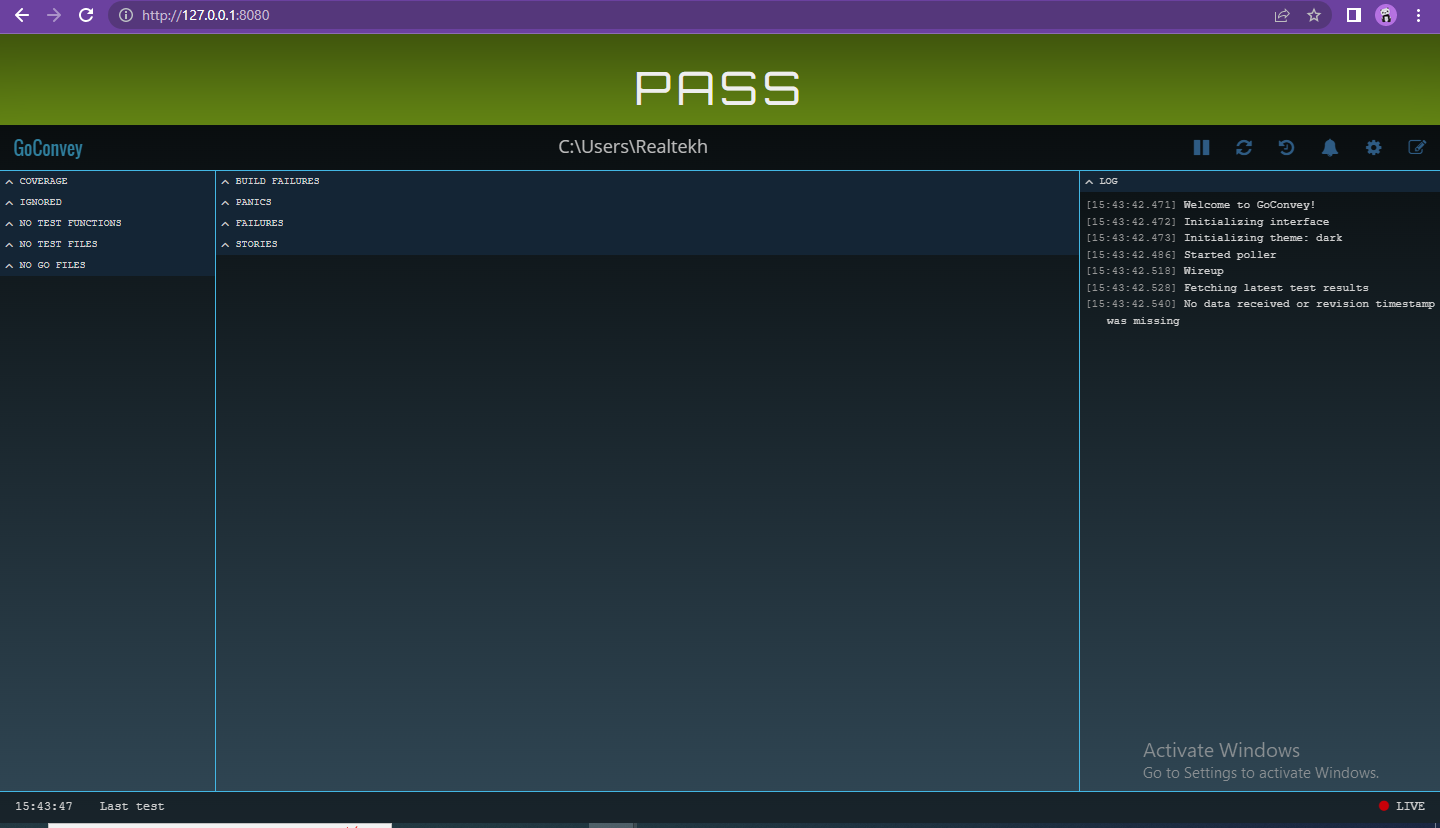
Copy the Launching browser on 127.0.0.1:8080

Package of convey contains all of the public-facing entry points to this project. This means that it should never be required of the user to import any other packages from this project as they serve internal purposes.

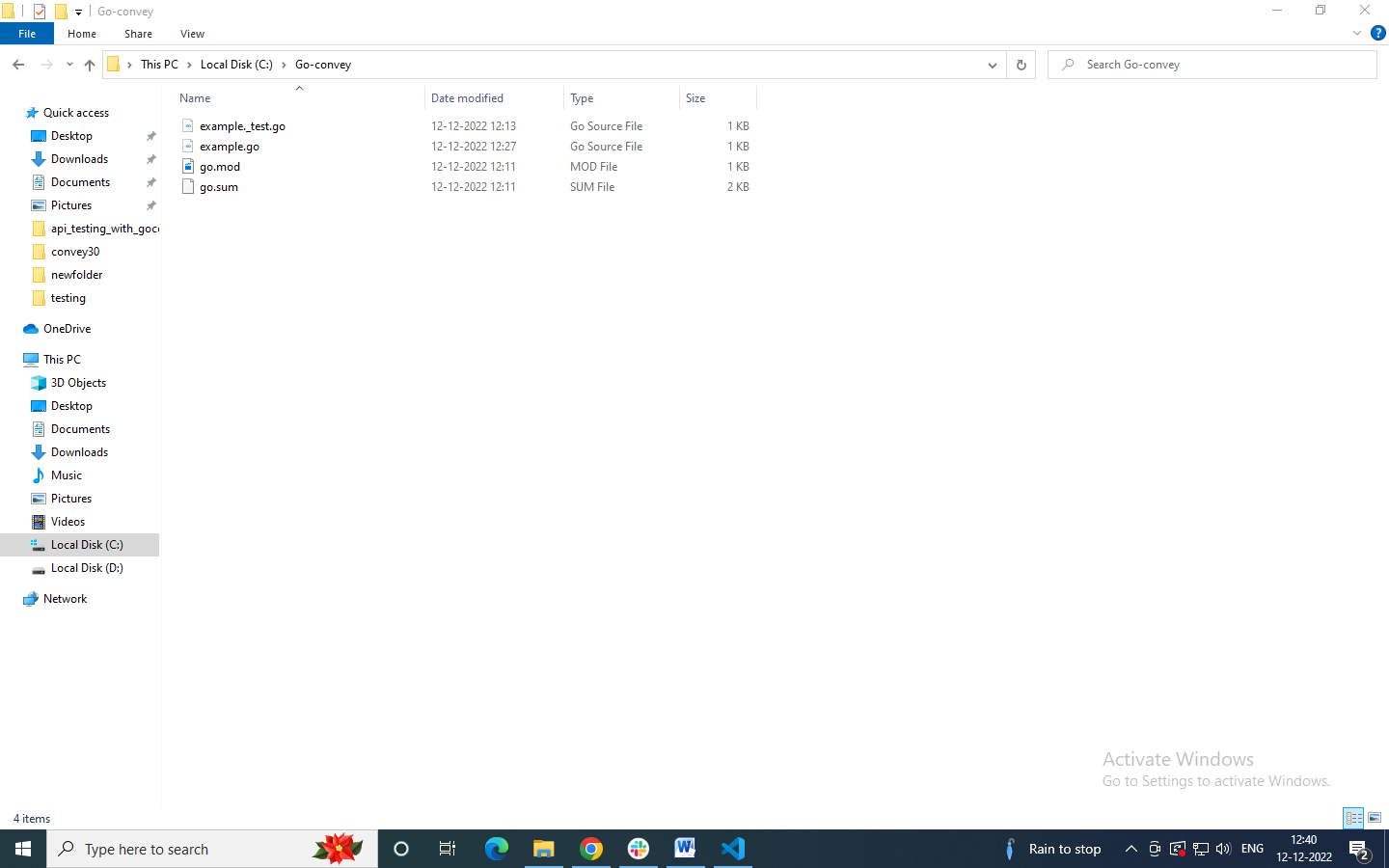


Then open your browser to:

http://localhost:8080



As long as Go-Convey are running, test results will automatically update in your browser window. The design is responsive, so you can squish the browser real tight if you need to put it beside your code.

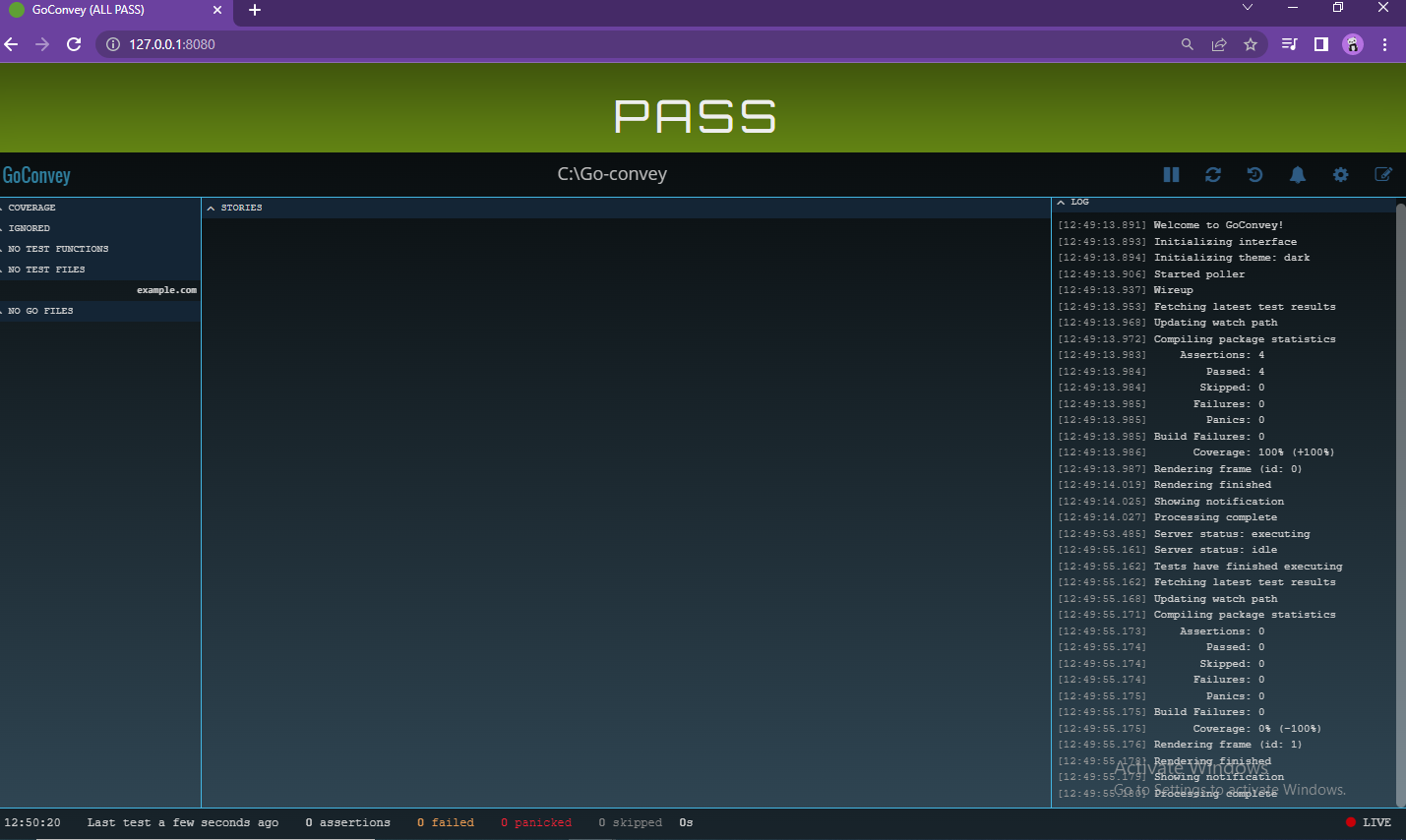
1. Copy the example.go Directory path – enter in patched directory. 

**Web UI**

A Web user interface or Web app allows the user to interact with content or software running on a remote server through a Web browser

Go-Convey port

Before using go composer the following go convey port is seen below.



#### Tests run automatically: Simply save the example.go file or click the <run tests > icon to execute tests. Your browser page will update automatically.

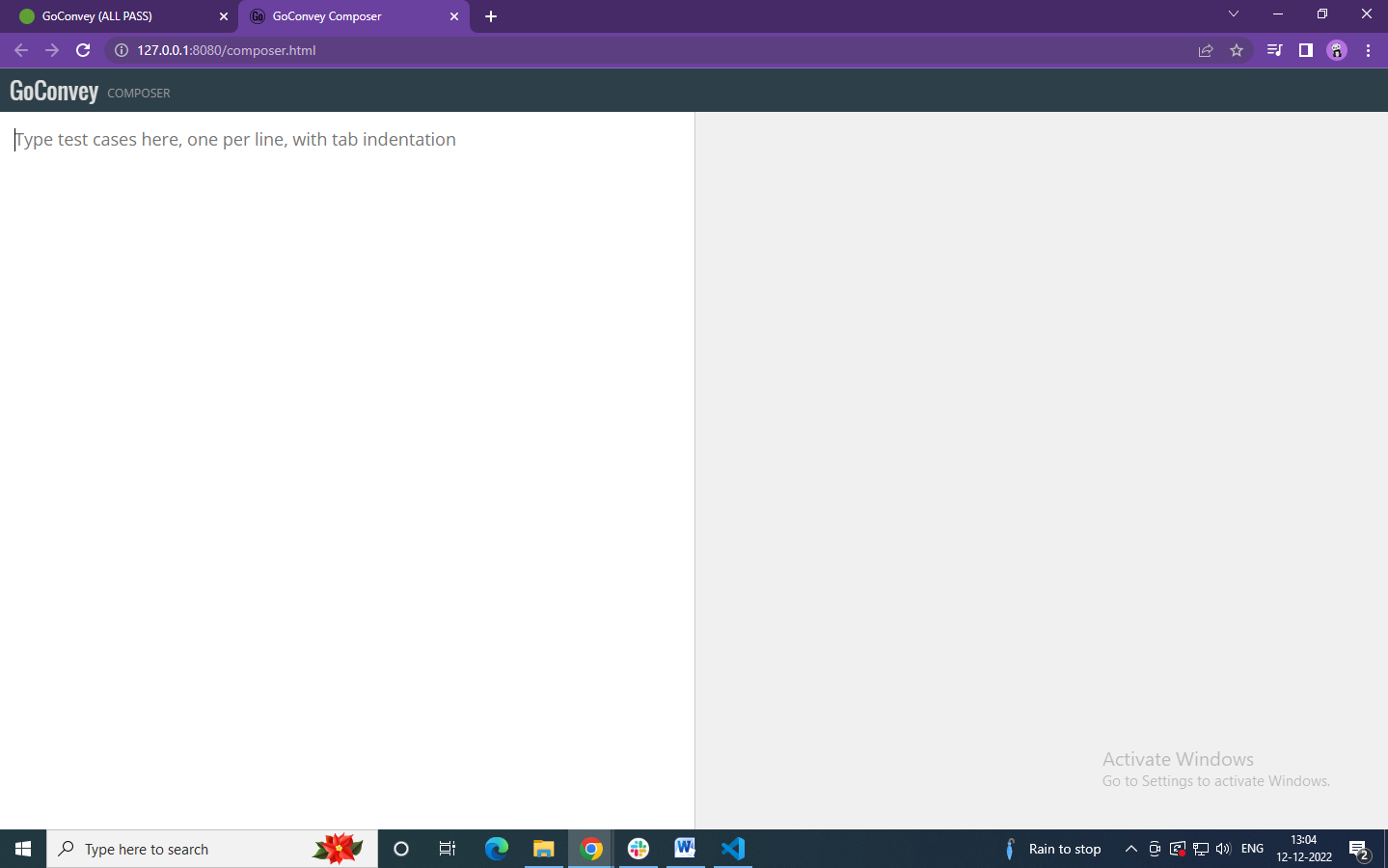
#### Dark, light, and custom theme: Go-Convey come with two themes built-in. ([Toggle the theme](about:blank) on this page to try it!) You can also use third-party themes or roll your own.

#### Pause, resume, and review: You can Pause auto-test execution and go through recent test history to see where, when, and why your code is breaking.

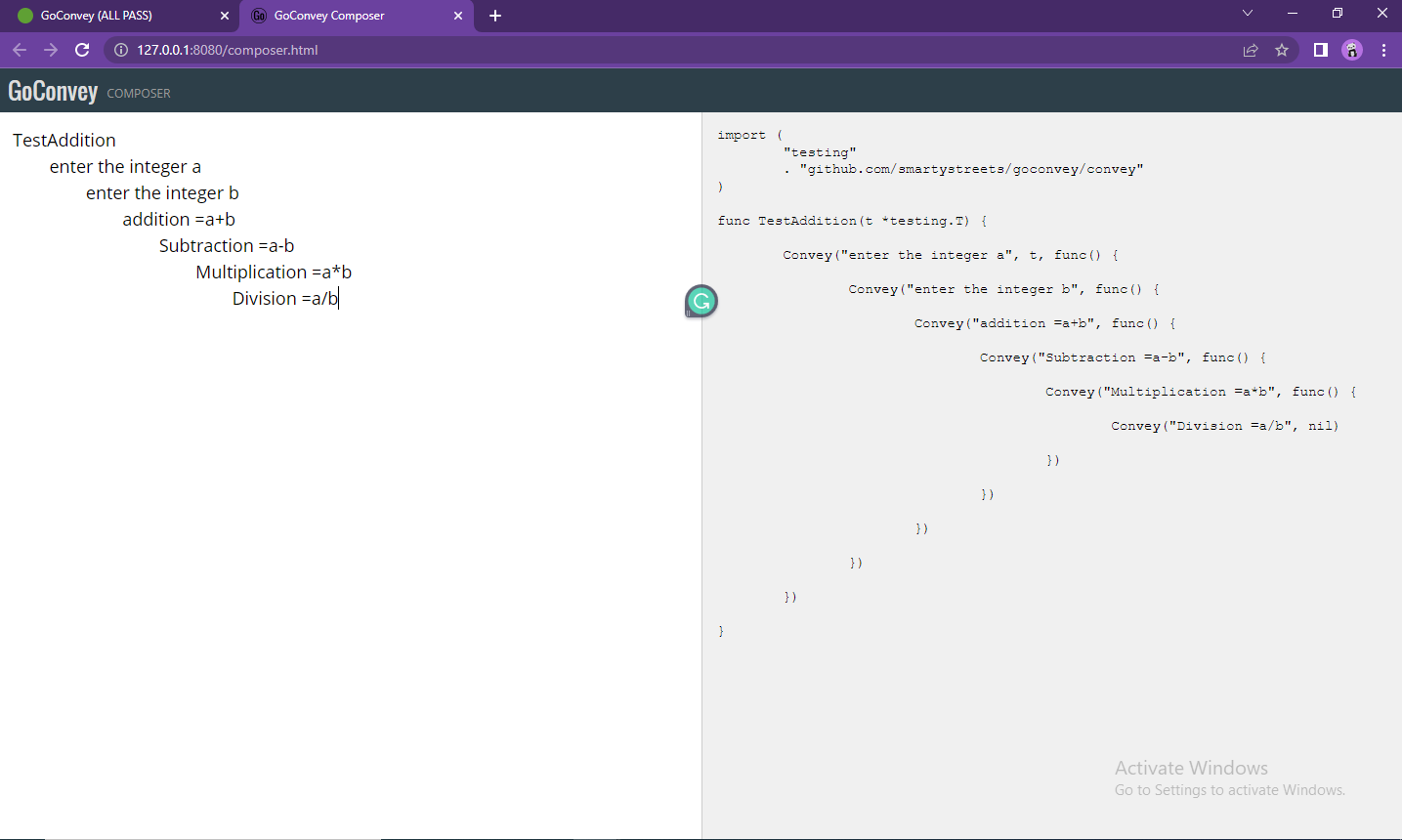
[**Writing tests**](https://chromium.googlesource.com/external/github.com/smartystreets/goconvey/+/refs/tags/1.3.1/README.md#writing-tests)

Code composer of example.go

Go Convey Generator/ Composer: Tell your program story; write one line behavioral comments to create BDD (Behavioral -Driven Development) test structure.

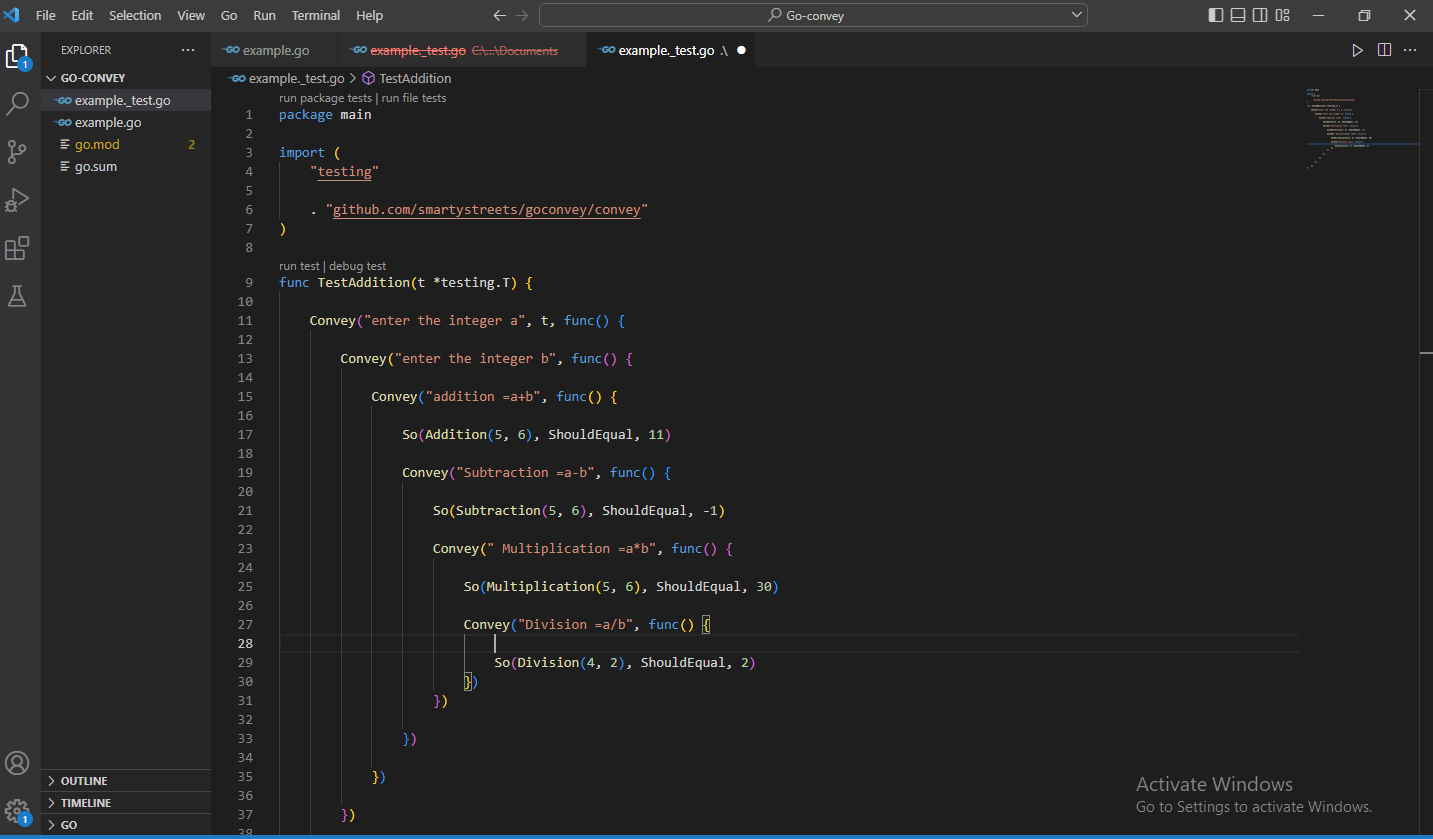


Write BDD structure in following example code



Go testing code of example.go

Example\_test.go



#### 1. Pass:

#### Test all Go Packages nested within the Specified Directory.

#### Supports all traditional test cases.

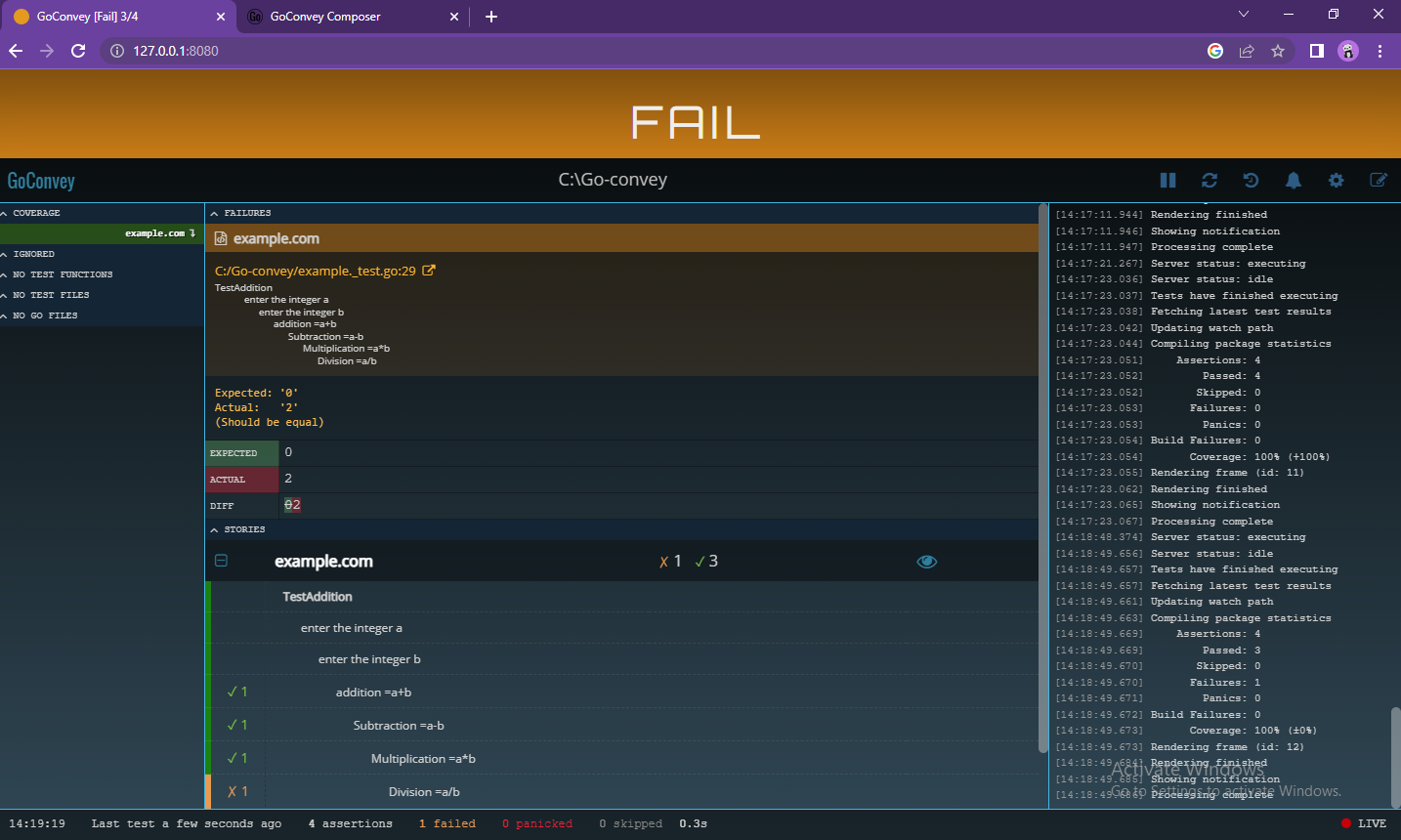
#### When go files are saved, tests are run and the page update automatically,

#### 

**2. Failure:**

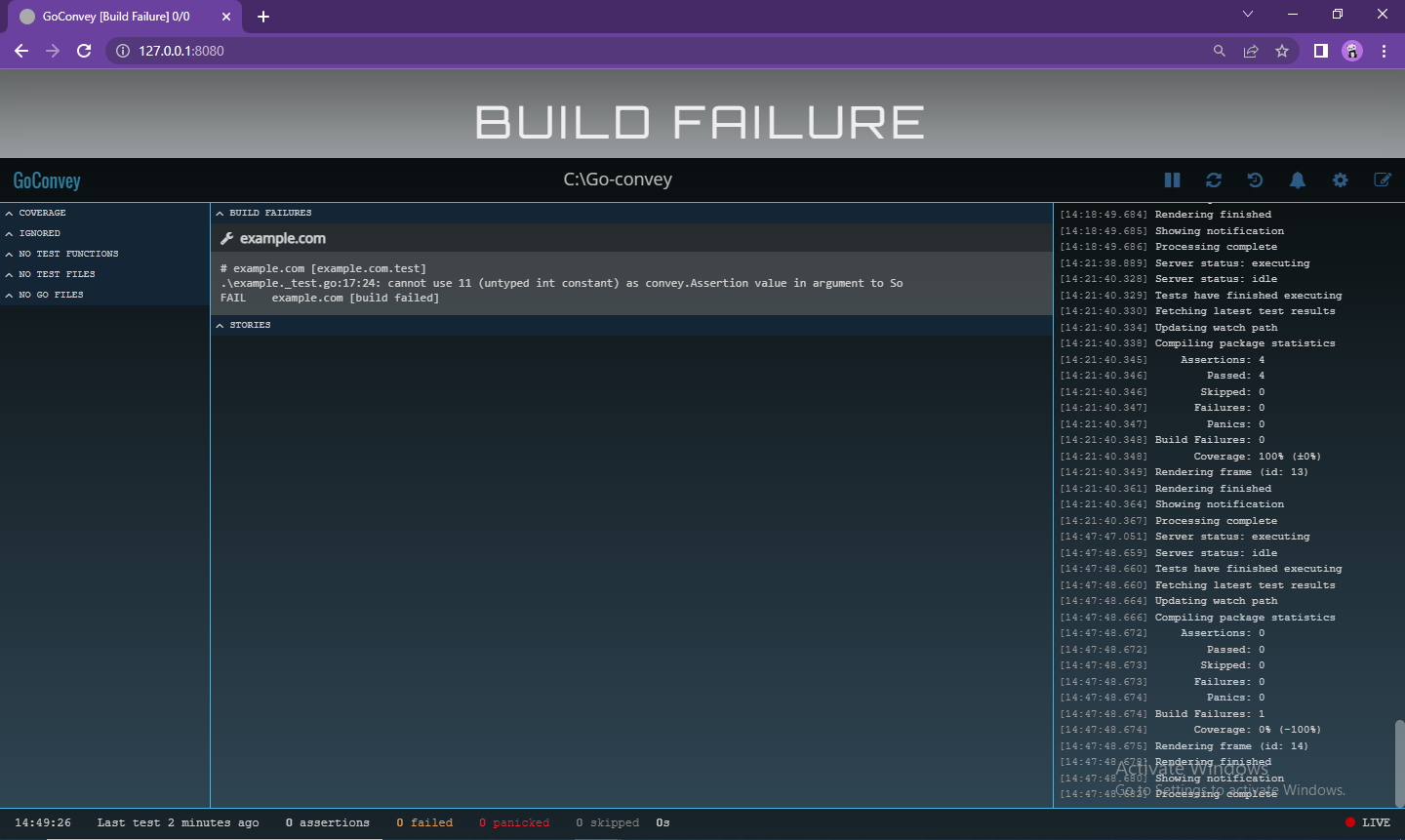
Responsive design so you can squish the tests beside your code

Each test in context with assertion counts, results of fail and pass counts.



**3.Build Failure:**

A failed build means **the code cannot be compiled** => there is nothing to run. Or maybe there is a previous successful build and there is something to run and debug.



[**Terminal output**](https://chromium.googlesource.com/external/github.com/smartystreets/goconvey/+/refs/tags/1.3.1/README.md#terminal-output-with-go-test--v)

Just do what you do best:

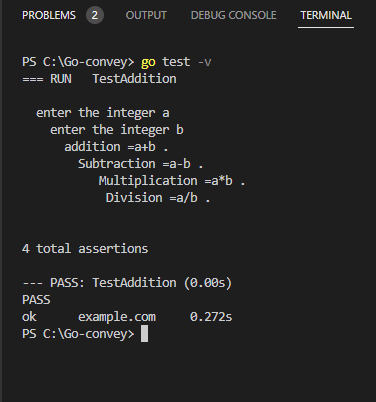
go test

Or if you want the output to include the story:

go test -v

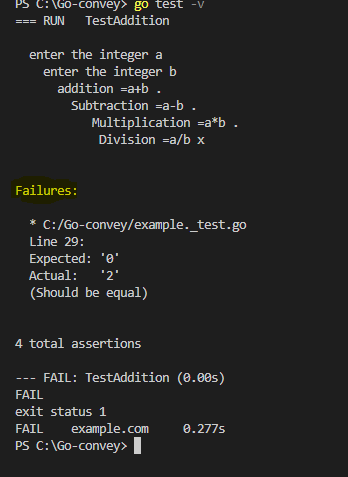
**Test Pass:**

When the all test case and assertion is pass.



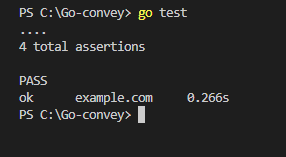
**Test Fail:**

When the test case is fail it shows the failure of the particular case remaining was passed.



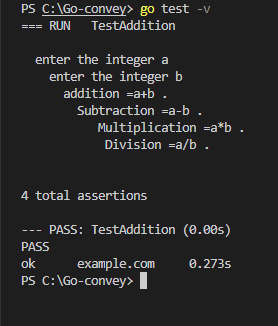
**Execution**

Concise mode (default):



Verbose mode:

go test -v



[**Assertions**](https://chromium.googlesource.com/external/github.com/smartystreets/goconvey/+/refs/tags/1.3.1/README.md#assertions)

A test assertion is an expression which encapsulates some testable logic specified about a target under test.

Here's the listing of assertions that this project aims to implement.

General Equality:

So(thing, ShouldEqual, thing2)

