

How to drive chat

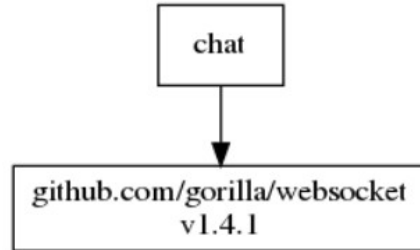


Figure 1 Dependency output from dep

After downloading chat from <https://github.com/rory911/misc/tree/master/chat>:

```
$ cd chat
$ go build chat
$ chat
```

You might have to give firewall permission in order to continue. Now, using a web browser, navigate to the home page at `localhost:8080/sales` and `localhost:8080/customer`.

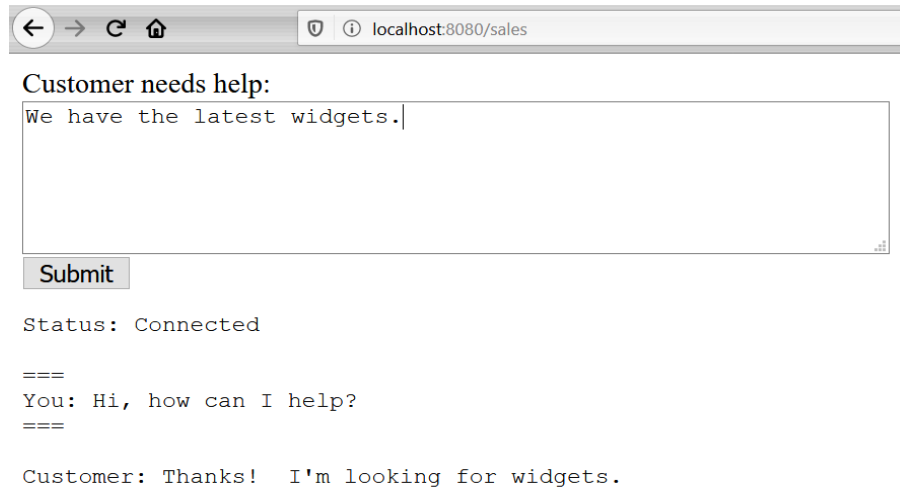
In this example, chat starts with the sales representative:

A screenshot of a web browser window. The address bar shows 'localhost:8080/sales'. The page content displays 'Customer needs help:' followed by a text input field containing 'Hi, how can I help?'. Below the input field is a 'Submit' button. At the bottom of the page, it says 'Status: Connected'.

The customer can then reply:

A screenshot of a web browser window. The address bar shows 'localhost:8080/customer'. The page content displays 'Our representatives are happy to help:' followed by a text input field containing 'Thanks! I'm looking for widgets.'. Below the input field is a 'Submit' button. At the bottom of the page, it says 'Status: Connected' and 'Sales: Hi, how can I help?'.

So here's what sales will see:



A screenshot of a web browser window. The address bar shows 'localhost:8080/sales'. The page content includes a heading 'Customer needs help:', a text input field containing 'We have the latest widgets.', a 'Submit' button, and a status message 'Status: Connected'. Below this, a chat log shows a user saying 'Hi, how can I help?' and a customer replying 'Thanks! I'm looking for widgets.'

Customer needs help:

We have the latest widgets.

Submit

Status: Connected

===

You: Hi, how can I help?

===

Customer: Thanks! I'm looking for widgets.

And the dialog continues. This code sample is a very simple example of Gorilla websockets running JavaScript and serving up a chat box dialog.