

Gorilla WebSocket



Gorilla WebSocket is a [Go](#) implementation of the [WebSocket](#) protocol.

Documentation

- [API Reference](#)
- [Chat example](#)
- [Command example](#)
- [Client and server example](#)
- [File watch example](#)

Status

The Gorilla WebSocket package provides a complete and tested implementation of the [WebSocket](#) protocol. The package API is stable.

Installation

```
go get github.com/gorilla/websocket
```

Protocol Compliance

The Gorilla WebSocket package passes the server tests in the [Autobahn Test Suite](#) using the application in the [examples/autobahn](#) subdirectory.

Gorilla WebSocket compared with other packages

	github.com/gorilla	golang.org/x/net
RFC 6455 Features		
Passes Autobahn Test Suite	Yes	No
Receive fragmented message	Yes	No , see note 1
Send close message	Yes	No
Send pings and receive pongs	Yes	No
Get the type of a received data message	Yes	Yes, see note 2
Other Features		
Compression Extensions	Experimental	No
Read message using io.Reader	Yes	No, see note 3
Write message using io.WriteCloser	Yes	No, see note 3

Notes:

1. Large messages are fragmented in [Chrome's new WebSocket implementation](#).
2. The application can get the type of a received data message by implementing a [Codec marshal](#) function.
3. The go.net io.Reader and io.Writer operate across WebSocket frame boundaries. Read returns when the input buffer is full or a frame boundary is encountered. Each call to Write sends a single frame message. The Gorilla io.Reader and io.WriteCloser operate on a single WebSocket message.