

SpdyStream

A multiplexed stream library using spdy

Usage

Client example (connecting to mirroring server without auth)

```
package main

import (
    "fmt"
    "github.com/moby/spdystream"
    "net"
    "net/http"
)

func main() {
    conn, err := net.Dial("tcp", "localhost:8080")
    if err != nil {
        panic(err)
    }
    spdyConn, err := spdystream.NewConnection(conn, false)
    if err != nil {
        panic(err)
    }
    go spdyConn.Serve(spdystream.NoOpStreamHandler)
    stream, err := spdyConn.CreateStream(http.Header{}, nil, false)
    if err != nil {
        panic(err)
    }

    stream.Wait()

    fmt.Fprint(stream, "Writing to stream")

    buf := make([]byte, 25)
    stream.Read(buf)
    fmt.Println(string(buf))

    stream.Close()
}
```

Server example (mirroring server without auth)

```
package main

import (
    "github.com/moby/spdystream"
    "net"
)
```

```
)

func main() {
    listener, err := net.Listen("tcp", "localhost:8080")
    if err != nil {
        panic(err)
    }
    for {
        conn, err := listener.Accept()
        if err != nil {
            panic(err)
        }
        spdyConn, err := spdystream.NewConnection(conn, true)
        if err != nil {
            panic(err)
        }
        go spdyConn.Serve(spdystream.MirrorStreamHandler)
    }
}
```

Copyright and license

Copyright 2013-2021 Docker, inc. Released under the [Apache 2.0 license](#).