Using WebSockets

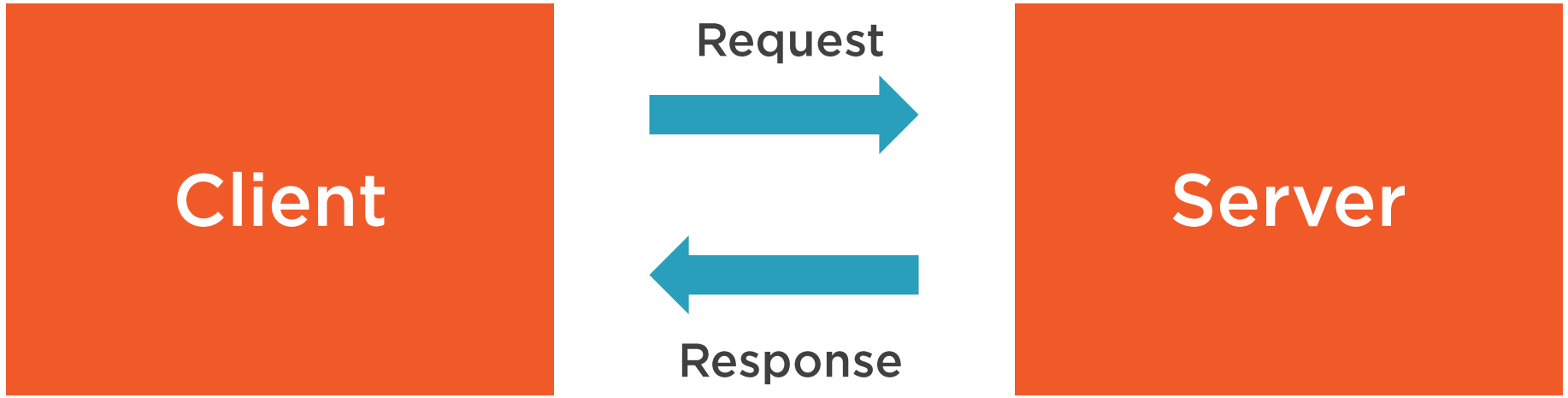
Alex Schultz
SOFTWARE ENGINEER | AWS ML HERO
@AlexCSchultz

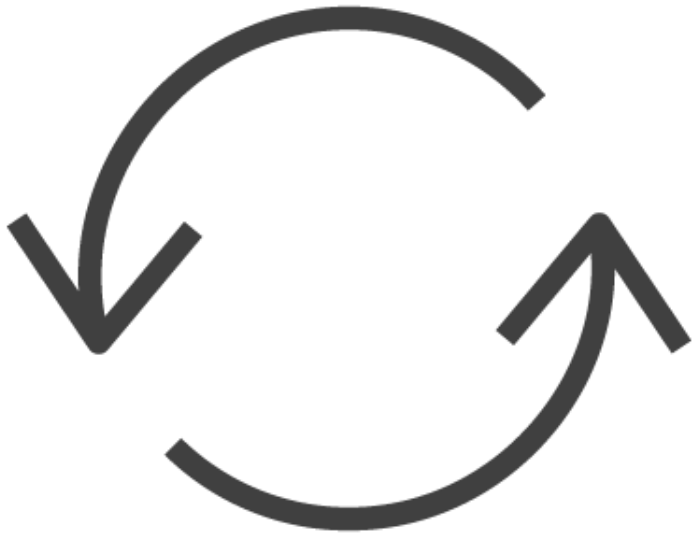
Overview

What are WebSockets

Creating WebSockets in Go

Standard Web Request Flow



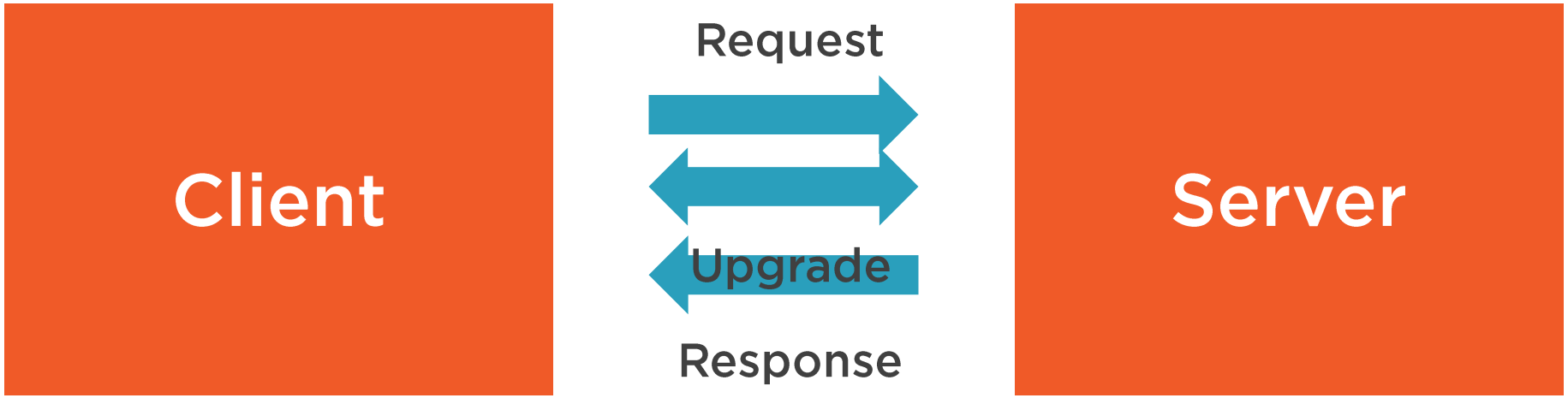


Updating data requires new HTTP requests

“Long Polling”

- Not responsive
- Not efficient

Web Socket Flow



WebSocket Flow

Client sends HTTP GET request

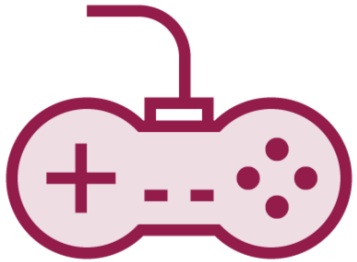
- Connection: Upgrade
- Upgrade: websocket
- Sec-WebSocket-Key: key

Server Responds with status code “101”

- Switching Protocols
- Upgrade: websocket
- Connection: Upgrade
- Sec-WebSocket-Accept: key

Uses for WebSockets

Chat Apps



Multiplayer Games

Stock Tickers

Dashboards

websocket.Conn

```
type Conn struct {
    PayloadType byte
    MaxPayloadBytes int
}
```

main.go

```
import (
  “net/http”
  “net/http”
)
func socket (ws *websocket.Conn) {
    // Handle receiving and sending data
}
func main() {
  http.Handle(“/websocket”, websocket.Handler(socket))
  log.Fatal(http.ListenAndServe(“:5000”, nil))
}
```

websocket.Codec

```
type Codec struct {
    Marshal func(v interface{}) (data []byte, payloadType byte, err error)
    Unmarshal func(data []byte, payloadType byte, v interface{}) (err error)
}
```

codec.Receive

```
func (cd Codec) Receive(ws *Conn, v interfaceP{}) (err error)
```

main.go

```
func socket (ws *websocket.Conn) {
  go func(c *websocket.Conn) {
    for {
      var msg message
      if err := websocket.JSON.Recieve(c, &msg); err != nil {
        break
      }
      fmt.Printf("received message %s\n", msg.Data)
    }
  }
  …
}
```

codec.Send

```
func (cd Codec) Send(ws *Conn, v interfaceP{}) (err error)
```

main.go

```
func socket (ws *websocket.Conn) {
  …
  products, _ := product.GetTopTenProducts()
  for {
      time.Sleep(10 * time.Second)
      if err := websocket.JSON.Send(ws, products); err != nil {
        break
      }
  }
}
```

Summary

What are WebSockets

Creating WebSockets in Go