Go培训第15天

tony

Outline

1. Websocket介绍

2. Tcp介绍

3. 一个简单的聊天室

websocket介绍

1. Websocket出现的背景

   a. http请求是典型的一问一答模式。

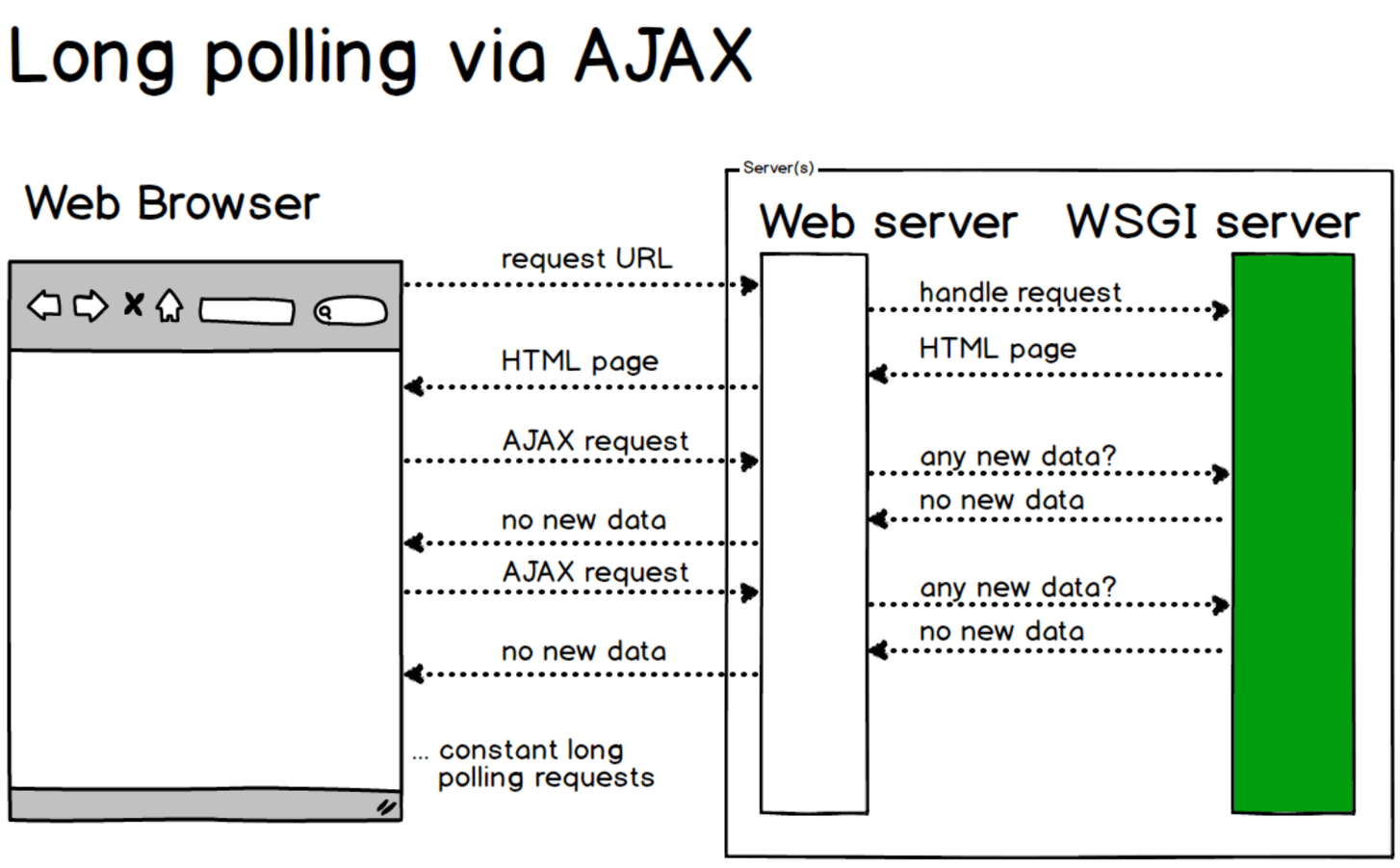
   b. 客户端与web服务器，无法建立长连接。

   c. 服务端缺乏主动推送数据到浏览器的能力。

websocket介绍

2. 基于http如何推送数据?

a. 通过长轮询的机制，以ajax的方式不断的发送请求给服务端来获取最新的数据



websocket介绍

3. TCP简介

a. Tcp是面向连接的，并且是全双工的

b. 当一个tcp连接建立之后，发送的数据就像水管里的水一样源源不断。

websocket介绍

4. TCP服务端的处理流程

a. 监听端口

b. 接收客户端的链接

c. 创建goroutine，处理该链接

```
package main

import (
	"fmt"
	"net"
)
func main() {
	fmt.Println("start server...")
	listen, err := net.Listen("tcp", "0.0.0.0:50000")
	if err != nil {
		fmt.Println("listen failed, err:", err)
		return
	}
	for {
		conn, err := listen.Accept()
		if err != nil {
			fmt.Println("accept failed, err:", err)
			continue
		}
		go process(conn)
	}
}
func process(conn net.Conn) {
	defer conn.Close()
	for {
		buf := make([]byte, 512)
		_, err := conn.Read(buf)
		if err != nil {
			fmt.Println("read err:", err)
			return
		}
		fmt.Println("read: ", string(buf))
	}
}
```

websocket介绍

5. 客户端的处理流程

a. 建立与服务端的链接

b. 进行数据收发

c. 关闭链接

```
package main

import (
	"bufio"
	"fmt"
	"net"
	"os"
	"strings"
)

func main() {

	conn, err := net.Dial("tcp", "localhost:50000")
	if err != nil {
		fmt.Println("Error dialing", err.Error())
		return
	}

	defer conn.Close()
	inputReader := bufio.NewReader(os.Stdin)
	for {
		input, _ := inputReader.ReadString('\n')
		trimmedInput := strings.Trim(input, "\r\n")
		if trimmedInput == "Q" {
			return
		}
		_, err = conn.Write([]byte(trimmedInput))
		if err != nil {
			return
		}
	}
}
```

websocket介绍

6. 如何基于web使用socket?

a. WebSocket 是 HTML5 开始提供的一种在单个 TCP 连接上进行全双工通讯的协议。

b. WebSocket 是一种网络通信协议。RFC6455 定义了它的通信标准。

c. WebSocket 是基于http协议上提供tcp socket通信的功能。

websocket介绍

7. Websocket握手过程

a. 浏览器发送websocket 握手包。

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
Origin: http://example.com
```

8. Websocket握手过程

a. 服务端响应握手包。

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: HSmrcOsM1YUkAGmm5OPpG2HaGWk=
Sec-WebSocket-Protocol: chat
```

Websocket示例

9. 基于websocket构建echo服务

a. 客户端发送字符串给服务端，服务端原样返回。

Websocket示例

10. 基于websocket构建简单聊天室服务

a. 简单的多人聊天室。

课后作业

1. 把今天的聊天室代码，自己实现一遍