**请求消息体**

# 请求头部分

# [请求方式] [资源路径] [版本]

GET /xxx HTTP/1.1

# 主机。

Host: server.example.com

# 协议升级。

Upgrade: websocket

# 连接状态。

Connection: Upgrade

# websocket客户端生成的随机字符。

Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==

# websocket协议的子协议，自定义字符，可以理解为频道。

Sec-WebSocket-Protocol: chat, superchat

# websocket协议的版本是13。

Sec-WebSocket-Version: 13

**响应结构体**

# 响应头部分

# [版本] [状态码]

HTTP/1.1 101 Switching Protocols

# 协议升级。

Upgrade: websocket

# 连接状态。

Connection: Upgrade

# WebSocket服务端根据Sec-WebSocket-Key生成的随机字符。

Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=

# WebSocket协议的子协议，自定义字符，可以理解为频道。

Sec-WebSocket-Protocol: chat

客户端发送的Sec-WebSocket-key实际上是随机字节的base64编码，这里随便发个xxx也是可以的，而服务端响应的Sec-WebSocket-Accept是该编码的hash，这个机制只是用来防止缓存代理发送同样的会话，并没有任何加密鉴权校验等功能。

**状态码**

连接成功状态码

101：HTTP协议切换为WebSocket协议。

连接关闭状态码

1000：正常断开连接。

1001：服务器断开连接。

1002：websocket协议错误。

1003：客户端接受了不支持数据格式（只允许接受文本消息，不允许接受二进制数据，是客户端限制不接受二进制数据，而不是websocket协议不支持二进制数据）。

1006：异常关闭。

1007：客户端接受了无效数据格式（文本消息编码不是utf-8）。

1009：传输数据量过大。

1010：客户端终止连接。

1011：服务器终止连接。

1012：服务端正在重新启动。

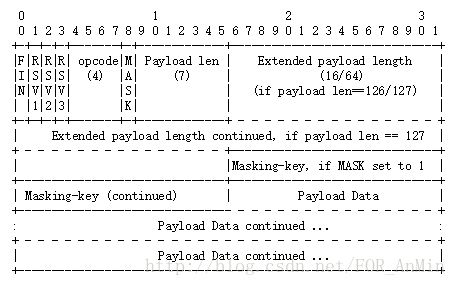
1013：服务端临时终止。

1014：通过网关或代理请求服务器，服务器无法及时响应。

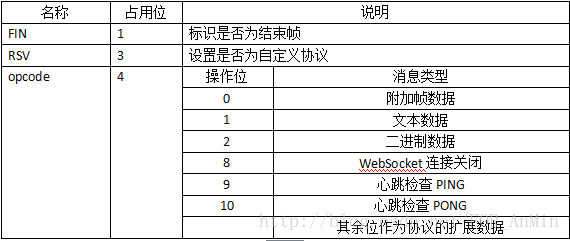
1015：TLS握手失败。

**WebSocket通信协议的数据传输**

WebSocket通讯协议的数据传输格式是以帧的形式传输的，其帧格式如图：



相关字段说明：



根据数据帧的设计可以看出，WebSocket通讯协议是通过心跳检查PING-PONG帧来实现WebSocket长连接。当WebSocket连接建立后，PING帧和PONG帧都会不携带数据地进行来回传输，当连接发生变化时，相应数据信息会被植入PING帧，PONG帧作为响应帧返回结果。