
Subject: Re: How to close the websocket connection

Posted by [uppjj](#) on Sat, 24 Feb 2018 14:59:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

thanks for your reply Mirek

In my case the server no longer called do () after launching close (), I should have better read the documentation!

The problem is rather client side, see below.

I encountered several difficulties to write a client application (IoT) with Websocket class for the real world of the web, but it does not miss much:

1) the connection header is currently fixed, it is essential to be able to modify it

=> it could be something like a Vector <String> to adapt before connecting.

2) Need a public function for really close the socket. Websoket :: Close does not close the socket but sends a request to the server.

If it does not respond for some reason, the client loops indefinitely (my first problem !)

I replaced the Close by Disconnect (), and added a new Close ():

```
void Disconnect (const String & msg = Null);    // old Close (), just a message to the server
```

```
void Close () {socket-> Close ();}              // real TCP close, even is server is sleeping...
```

3) Sending Masked requests is not available. This works:

add this function in class WebSocket (Inet.h) :

```
void SendTextMasked(const String& data)          { SendRaw(MASK|FIN|TEXT, data); }
```

Change SendRaw() in WebSocket.cpp :

```
void WebSocket::SendRaw(int hdr, const String& data)
```

```
{  
    if(IsError())  
        return;
```

```
    ASSERT(!close_sent);
```

```
    LLOG("Send " << data.GetCount() << " bytes, hdr: " << Format("%04X",hdr));
```

```
    // mask detect
```

```
    int LocMask = (hdr & MASK)?0x80:0;
```

```
    hdr &= 0xFF;
```

```
    //---- header construct
```

```
    // opcode
```

```
    String header;
```

```
    header.Cat(hdr);
```

```
    // Length
```

```
    int len = data.GetCount();
```

```
    if(len > 65535) {
```

```
        header.Cat(LocMask | 127);
```

```

header.Cat(0);
header.Cat(0);
header.Cat(0);
header.Cat(0);
header.Cat(byte(len >> 24));
header.Cat(byte(len >> 16));
header.Cat(byte(len >> ));
header.Cat(byte(len));
}
else
if(len > 125) {
    header.Cat(LocMask | 126);
    header.Cat(byte(len >> ));
    header.Cat(byte(len));
}
else
    header.Cat(LocMask |(int)len);

if (LocMask)
{
    //add masking-key
    byte Cle[4];
    Cle[0] = Random();
    Cle[1] = Random();
    Cle[2] = Random();
    Cle[3] = Random();
    for(int i = 0; i < 4; i++) header.Cat(Cle[i]);

    //---- send header with mask
    Out(header);

    //---- send masked data
    if(data.GetCount() != 0)
    {
        char buf[32768];
        int n = data.GetCount();
        for(int i = 0; i < n; i++)
            buf[i] = data[i] ^ (byte) Cle[i & 3];
        Out(String(buf,n));
    }
}

else
{
    //---- send header (not masked)
    Out(header);

    //--- send data (not masked)

```

```
if(data.GetCount() != 0)
    Out(data);
}
}
hope this can help
```
