User manual

（Signal Server）

Declaration

Since the product version upgrade or other reasons, this manual will subsequently be updated. Unless otherwise agreed, this manual only as a guide, this manual all statements, information, recommendations do not constitute any express or implied warranties。

www.happytimesoft.com

Table of Contents

Chapter 1 Introduction

The role of the Happytime Signal Server is to coordinate the exchange of information between communicating parties, mainly for exchanging network information, ICE candidates, and media negotiation signaling information. It plays a key role in WebRTC communication processes, and its role is mainly reflected in the following aspects:

1. Network address exchange: Because WebRTC is based on peer-to-peer connections, both parties need to know each other's network addresses to establish a connection. The signal server is responsible for collecting and exchanging the network addresses of both parties, enabling them to discover each other and establish a connection.

2. Media negotiation: Before establishing a connection, both parties need to negotiate parameters such as media codec, resolution, and frame rate. The signal server can assist both parties in media negotiation to ensure that they can communicate in a compatible manner.

3. Connection establishment and maintenance: The signal server is responsible for assisting communication parties in establishing connections and maintaining and managing connections during the communication process. It can handle operations such as establishing, reconnecting, and disconnecting connections, ensuring the stability and reliability of communication.

Chapter 2 Configuration

If the configuration file path is not specified during startup, Signal Server uses the default configuration file signalserver.cfg.

2.1 Configuration Templates

```xml
<?xml version="1.0" encoding="utf-8"?>
<config>
    <http_enable>1</http_enable>
    <http_server_ip></http_server_ip>
    <http_port>8080</http_port>
    <https_enable>1</https_enable>
    <https_server_ip></https_server_ip>
    <https_port>5443</https_port>
    <cert_file>ssl.ca</cert_file>
    <key_file>ssl.key</key_file>
    <log_enable>1</log_enable>
    <log_level>1</log_level>

    <ice_servers>
        <url>stun:192.168.3.168:3478</url>
        <url>turn:test:123456@192.168.3.168:3478</url>
    </ice_servers>
</config>
```

2.2 Configuring Node Description

2.2.1 System parameters

<http_enable>

Whether to enable http server, 0-disable,1-enable.

<http_server_ip>

Specify the http server ip, if not specified, it will listen on all interfaces.

<http_port>

Specify the http server port, default is 80.

Note: On Linux systems, ports below 1024 are reserved by the system and require root privileges to be used.

<https_enable>
Whether to enable https server, 0-disable,1-enable.

<https_server_ip>
Specify the https server ip, if not specified, it will listen on all interfaces.

<https_port>
Specify the https server port, default is 443.

Note: On Linux systems, ports below 1024 are reserved by the system and require root privileges to be used.

<cert_file>
Specify the https server certificate file.

<key_file>
Specify the https server key file.

Note: The certificate file ssl.ca and key file ssl.key provided by default are self signed local hosts certificates, only for testing purposes (browsers may pop up untrusted certificate warnings), and cannot be used in formal deployment environments.

<log_enable>
Whether enable the log function, 0-disable, 1-enable.

<log_level>
The log level:

TRACE	0
DEBUG	1
INFO	2
WARN	3
ERROR	4
FATAL	5

2.2.2 **ice_servers**

<ice_servers> **:** Specify ICE server configuration

<url>

Specify the ICE server URL address,up to 8 ICE service URLs can be configured.

The URL format of the STUN server is as follows:

stun:ip or **stun:ip:port** or **stun:username:password@ip:port**

If no port is specified, the default port 3478 is used, and username and password are the authentication credentials required by the STUN server.

The URL format of the TURN server is as follows:

turn:ip or **turn:ip:port** or **turn:username:password@ip:port**

If no port is specified, the default port 3478 is used, and username and password are the authentication credentials required by the TURN server.

Some browsers require the TURN service to be configured with username and password authentication credentials, if there is no username and password authentication credentials, creating a PeerConnection fails,therefore, it is recommended to configure the TURN server with username and password authentication credentials.

We have tested the stunman and coturn servers.

You can download the stunman or coturn server and run it.

Stunman server download address:

https://www.stunprotocol.org/

Coturn server download address:

https://github.com/coturn/coturn

Chapter 3 Run Signal Server

The signal server is a console application.

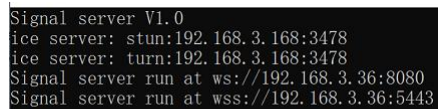Windows: to run the server, simply type "signalserver".

Linux: to run the server, type "./start.sh", on linux platform, signal server run as deamon by default.

Signal server supports the following command line options:

-c config   specify the configuration file

-c option specifies the configuration file,if not specified, the default configuration signalserver.cfg is used.

After the signaling service starts normally, it will output the following information, as shown in the following figure:

```
Signal server V1.0
ice server: stun:192.168.3.168:3478
ice server: turn:192.168.3.168:3478
Signal server run at ws://192.168.3.36:8080
Signal server run at wss://192.168.3.36:5443
```

Output version number, output configured ICE service address.

Output the signaling server address.

Portablertc requires the configuration of signaling server. The output signaling server address can be set to the configuration file of Portablertc, where ws_signal_server corresponds to the address starting with ws://, and wss_signal_server corresponds to the address starting with wss://.