Role CHAT (Debian/Win2019)

The CHAT role is used for conducting chat sessions with agents from external websites. The CHAT role is also required for WhatsApp integrations.

The chat server is a Java application, which can be installed in parallel on the web servers with corresponding additional RAM of about 4 GB per server.

Alternatively, the chat-server can be installed on a separate server, or on the load balancer.

The following recommendations are made:

Where	Requirements
On an existing webserver	Additional 4 GB of RAM
On the load balancer	Additional 4 GB of RAM and 2 CPU cores, expand system drive (home directories) to 32 GB
On a separate server	8 GB RAM, 32 GB HDD, 2 CPU Cores (can be extended later to 4 CPU cores, depending on system load)

Installation

Java Installation

```
apt-get -y install openjdk-17-jdk
```

OLD - Java 8 Installation

```
apt-get update
apt-get install software-properties-common
apt-add-repository 'deb http://security.debian.org/debian-security stretch/updates main'
apt-get update
apt-get -y install default-jdk
```

Installing and Configuring ClientMessenger

ClientMessenger Installation

First of all, the ClientMessenger directory is copied to /home/jtel

```
cp -R /home/jtel/shared/JTELCarrierPortal/Utils/Install/ClientMessenger/ /home/jtel/ClientMessenger
chown -R jtel:jtel /home/jtel/ClientMessenger
```

Hazelcast Cluster Configuration File

The hazelcast.xml file is copied to the configuration directory of the chat server.

```
rm -f /home/jtel/ClientMessenger/conf/hazelcast.xml
cp /home/jtel/shared/hazelcast.xml /home/jtel/ClientMessenger/conf/hazelcast.xml
chown jtel:jtel /home/jtel/ClientMessenger/conf/hazelcast.xml
```

Client Messenger Configuration

The configuration or the start file jtel-clientmessenger may have to be adapted so that the database connection can be established. Here **<password>** must be replaced with the password for the root user of the database:

CAUTION PASSWORD

```
# Edit this file:
vi /home/jtel/ClientMessenger/systemd/jtel-clientmessenger.service

# Check this setting
...
-Dde.jtel.platform.clientmessenger.connection=jdbc:mysql://acd-dbm/JTELWeb?user=root&password=<password>&characterEncoding=utf8&serverTimezone=Europe/Berlin \
...
```

Install ClientMessenger with systemd

```
cp /home/jtel/ClientMessenger/systemd/jtel-clientmessenger.service /etc/systemd/system/jtel-clientmessenger.service
systemctl daemon-reload
systemctl enable jtel-clientmessenger.service
systemctl start jtel-clientmessenger.service
```

Create the Update Script

An update script is created for updating the chat server:

```
cat <<EOFF>/usr/local/bin/updatesc.sh
#!/bin/bash
systemctl stop jtel-clientmessenger.service
cp /home/jtel/shared/JTELCarrierPortal/Utils/Install/ClientMessenger/bin/* /home/jtel/ClientMessenger/bin
chown -R jtel:jtel /home/jtel/ClientMessenger
systemctl start jtel-clientmessenger.service
EOFF
chmod +x /usr/local/bin/updatesc.sh
```

Configure the Firewall

Next, the *open* ports for the chat service must be entered in the firewall and saved persistently. Here you can distinguish whether http (speak unsecure websocket) or https (secure websocket) is used.

Since the load balancer usually decrypts the https, for https only port 3003 needs to be opened to the load balancer.

Firewall configuration http

Execute the following commands, on the following machines:

- On CHAT server (here the chat port and the Hazelcast Cluster ports)

```
ufw allow 5701:5801/tcp
ufw allow 3000/tcp
```

  - For a specific interface

```
ufw allow in on eth0 to any port 5701:5801 proto tcp
ufw allow in on eth0 to any port 3000 proto tcp
```

- On all LB - with http connection to the outside

```
ufw allow 3000/tcp
```

Firewall configuration https

- On all LB - with https connection to the outside

```
ufw allow 3003/tcp
```

Load Balancer Configuration

haproxy.cfg Adjustments

For http

The following adjustments are made on the haproxy, in case of a http connection for the chat to the outside:

```
frontend acdportal_chat_ws
        mode            http
        bind            :3000
        timeout         client 1d
        default_backend backend_chat_ws

backend backend_chat_ws
        mode            http
        balance         leastconn # roundrobin
        server          chatserver1 <ip_addresse_oder_name>:3000 weight 1 check inter 1m
        server          chatserver2 <ip_addresse_oder_name>:3000 weight 1 check inter 1m
```

For https

```
frontend acdportal_chat_wss
        mode            http
        bind            :3003 ssl crt /etc/haproxy/haproxy.pem
        timeout         client 1d
        default_backend backend_chat_ws

backend backend_chat_ws
        mode            http
        balance         leastconn # roundrobin
        server          chatserver1 <ip_addresse_oder_name>:3000 weight 1 check inter 1m
        server          chatserver2 <ip_addresse_oder_name>:3000 weight 1 check inter 1m
```

Reload Haproxy

Then, reload the haproxy with the following command:

```
service haproxy reload
```

Chat Configuration

Parameters (as sysadmin)

For https / wss

ACD.Chat.Script.Library.URL	https://<load-balancer>:3003/gui/
ACD.Chat.Server.Base.URL	wss://<load-balancer>:3003

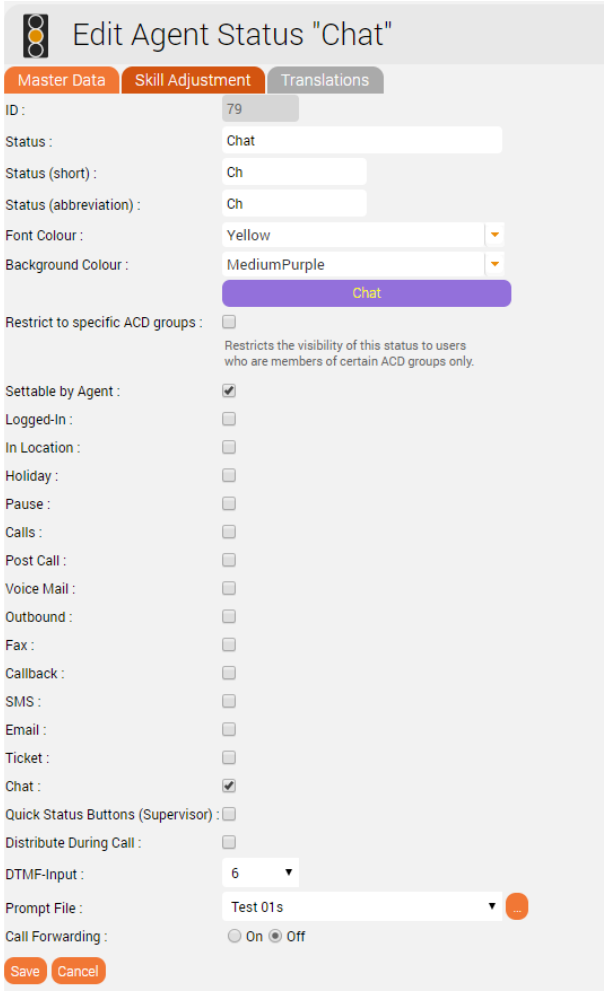
For http / ws

ACD.Chat.Script.Library.URL	http://<load-balancer>:3000/gui/
ACD.Chat.Server.Base.URL	ws://<load-balancer>:3000

P.S: To enable the chat attachement, please configure the following parameters: **ACD.Whatsapp.Server.Base.URL** and **ACD.Whatsapp.Whatsapp.Url.** See Add Additional parameters below for more details

Status

An appropriate agent status for "Chat" - or an existing status - must be configured so that chat is distributed to the agent:



Max chats per agent

The maximum number of parallel chats per agent must be configured. This value is set to 0 by default for all agents.

Edit User "admin"

Maximum call recordings each day :

This counter applies only to call recording every x calls

Web Logout Options

Agent status change after "Logout" in web :

Agent status after "Logout" in web :

Auto-Logoff Options

Maximum Missed Calls - Busy : -1

Agent Status after Logout - Busy :

Maximum Missed Calls - No Answer : -1

Agent Status after Logout - No Answer : Logged Off

Maximum Missed Calls - Rejected : -1

Agent Status after Logout - Rejected :

Maximum Missed Calls - Total : -1

Agent Status after Logout - All :

Notify Auto-Logout by Email :

Daily Autologout

Daily Autologout Active :

Execution Time : 00:00

Agent Status after Logout :

Deactivate Daily Autologout ACD Groups :

If this option is selected, autologout is not performed for individual ACD groups.

Media Delivery Options

Voicemail via Web Interface :

Fax via Web Interface :

SMS via Web Interface :

Callback via Web Interface :

Email via Web Interface :

Ticket via Web Interface :

Max Chats Per Agent : 8

Save Cancel

Additional configurations for WhatsApp

In order for What's App to work, further configurations still need to be made.

Note: setting up the WhatsApp business account or connecting to the provider (currently TynTec) is done directly with the provider.

https

The load balancer must be operated with https and a valid certificate.

Additional configuration in jtel-clientmessenger

The parameter DATADIR must be checked. This must point to either the **/srv/jtel/shared/Data/Clients/** directory if it is hosted directly on the chat server, or **/home/jtel/shared/Data/clients/** if the server has mouted the directory.

```
Environment=DATADIR=/home/jtel/shared/Data/clients/
```

Additional configuration on the load balancer

Additional entries are required in the frontend area for the portal:

```
frontend acdportal_https
...
        acl whatsapp_req    path /incoming
        acl whatsapp_req    path /delivery
...
        use_backend     backend_chat_ws if whatsapp_req
```

Additional parameters

The following system parameters must be checked in the portal:

Parameter	Value	Comment
ACD.Whatsapp.Server.Base.URL	https://jtel-portal:3003	The URL used by the agent client to allow the agent to reply to messages. This must be resolved from all agent workstations to a valid URL via DNS, and is directed to the chat server via the haproxy.
ACD.Whatsapp.Whatsapp.Url	https://jtel-portal:3003	The URL used by the outside world (WhatsApp users) to download the attachments of the messages. This must be resolved from the Internet to a valid URL via DNS, and is directed to the chat server via the haproxy.
ACD.Whatsapp.Tyntec.Url	https://api.tyntec.com/chat-api/v2/messages	The URL of the TynTec API for WhatsApp.

Customize TynTec API endpoints

It may be necessary to adjust the API endpoints to which the messages are sent. Currently TynTec does not offer the possibility to do this via a web configuration, this must be done via a REST call. The easiest way to do this is to use Postman.

The following parameters are required, both callback URLs are adjusted according to the end point of the system.

Setting	Value
URL	https://api.tyntec.com/chat-api/v2/applications/default
Request Type	PATCH
Authorization	apikey Value = the generated API Key from Tyntec
Additional Headers	Accept application/problem+json
Additional Headers	Content-Type application/json

Body	`{ "webhooks": [{ "events": ["MoMessage"], "callbackUrl": "https://jtel-portal/incoming" }, { "events": ["MessageStatus::accepted", "MessageStatus::delivered", "MessageStatus::seen", "MessageStatus::failed", "MessageStatus::channelFailed", "MessageStatus::deleted", "WhatsAppGroupEvent::userJoined", "WhatsAppGroupEvent::userLeft", "WhatsAppGroupEvent::subjectChanged", "WhatsAppGroupEvent::descriptionChanged"], "callbackUrl": "https://jtel-portal/delivery" }] }`

Chat Connector Configuration

The following additional parameters are configured in the Chat Connector:

- The check mark Whatsapp is set.
- The TynTec API Key is entered.
- The incoming service number for WhatsApp is configured and in E.164 and associated with the connector

Edit Chat Connector

Client ID :	4
Name :	SUPPORTCHAT
Title :	Support Chat
Whatsapp :	☑
Whatsapp ApiKey :	[redacted]
Whatsapp Number :	[redacted]