Servicios profesionales para:

LEFEBVRE EL DERECHO

PROPUESTA DESARROLLO DE API EXTERNA CON AUTENTICACIÓN JWT PARA JITSI MEET

Propuesta Tecnológica

CONTENIDO

1 CONTROL DE VERSIONES

FECHA	VERSIÓN	OBSERVACIONES	AUTOR
10-09-2020	Versión inicial v01	Funciones disponibles para jitsi	Kelvin Castillo

2 SERVICIO

2.1 Introducción

El propósito de este documento es detallar cada una de las funcionalidades que tiene incorporadas Jitsi Meet.

Por nuestra amplia experiencia, estamos convencidos que podemos aportar en un solo equipo, expertos en las tecnologías de este aplicativo; a continuación, describiremos en los siguientes apartados de este documento las funciones permitias con la aplicación de Jitsi Meet.

2.2 Objetivos

- Proporcionar una visibilidad de las funciones permitidas con la aplicación de Jitsi Meet
- Describir el alcance y las limitaciones de las funciones nativas de la aplicación de Jitsi Meet.
- Determinar las mejoras posibles que se pueden realizar a las funciones nativas de la aplicación de Jitsi Meet.

2.3 JITSI MEET

Jitsi Meet es un plataforma de código abierto que facilita las videoconferencias grupales con un máximo de 200 interlocutores, y no solo se queda en una app o web de videollamadas, sino que nos permite ir más allá.

Al ser de código abierto, sus servicios e integración en una aplicación propia, se ofrecen de forma gratuita y utiliza Jitsi Videobridge para proporcionar videoconferencias escalables, seguras y de alta calidad

Las llamadas pueden ser de vídeo o de voz, y ofrece la posibilidad de grabar la llamada con Dropbox o retrasmitirla en directo en YouTube. Además ofrece la posibilidad de conectar con los calendarios de Google y Office 365, para poder programar reuniones y notificar de cuándo van a llevarse a cabo.

Ofrece opciones como un chat de texto o la opción de levantar la mano de forma virtual para hablar ordenadamente. Permite silenciar a todos los usuarios mientras un usuario esta hablando y hay un contador para saber cuánto tiempo está hablando cada persona.

Para la gran mayoría de opciones que se han mencionado, se precisa de instalar y configurar Jibri en el PC a través de Google Chrome.

2.4 API EXTERNA IFRAME

Se puede utilizar el API externa de Jitsi Meet para integrar Jitsi Meet en una aplicación propia, la cual permite generar un enlace para crear una llamada desde una aplicación propia, con las configuraciones iniciales necesarias como: nombre de la sala, listado de usuarios permitidos, usuario anfitrion, listado de dispositivos permitidos, etc. Para poder hacer uso de esta API se debe integrar la biblioteca:

<script src='https://<your-domain>/external_api.js'></script>

Esta API externa se puede implementar utilizando el objeto API de Jitsi Meet el cual se puede instanciar de la siguiente manera:

api = new JitsiMeetExternalAPI(domain, options)

El constructor de este objeto tiene varias opciones:

- **domain**: dominio utilizado para crear la URL de la conferencia.
- **options**: objeto con propiedades - los argumentos opcionales:
  - **roomName**: (opcional) nombre de la sala para unirse.
  - **width**: (opcional) ancho para el iframe que se creará. Si se especifica un número, se trata como unidades de píxeles. Si se especifica una cadena, el formato es número seguido de 'px', 'em', 'pt' o '%'.
  - **height**: (opcional) altura para el iframe que se creará. Si se especifica un número, se trata como unidades de píxeles. Si se especifica una cadena, el formato es número seguido de 'px', 'em', 'pt' o '%'.
  - **parentNode**: (opcional) Elemento DOM HTML donde el iframe se agregará como hijo.
  - **configOverwrite**: objeto JS (opcional) con anulaciones para las opciones definidas en config.js .
  - **interfaceConfigOverwrite**: objeto JS (opcional) con anulaciones para las opciones definidas en interface_config.js .
  - **noSSL**: (opcional, el valor predeterminado es falso) Booleano que indica si se debe contactar al servidor mediante HTTP o HTTPS.
  - **jwt**: token JWT (opcional) .
  - **onload**: controlador (opcional) para el evento onload de iframe.
  - **invitees**: (opcional) Matriz de objetos que contienen información sobre los nuevos participantes que serán invitados a la convocatoria.
  - **devices**: (opcional) un mapa que contiene información sobre los dispositivos iniciales que se utilizarán en la llamada.
  - **userInfo**: (opcional) objeto JS que contiene información sobre el participante que abre la reunión, como email.

Esta API viene deshabilitada por default, para poder habilitarla se debe hacer desde la configuración en la imagen del VideoBridge (JVB).

2.5 API REST STATS

Jitsi Meet incluye el API rest stats que permite obtener las estadisticas de las llamadas realizadas (participantes, duración, etc.), para poder hacer uso de esta API, debe estar previamente habilitada en la imagen del Videobridge (JVB) y tener el puerto 8080 expuesto (si el puerto no fue modificado) para poder consumir el API.

Los informes que provee Jitsi Videobridge contienen las siguientes estadísticas (y más):

1. Número de subprocesos utilizados por la JVM.
2. Tasa de bits actual, tasa de paquetes y tasa de pérdida de paquetes.
3. Número actual de canales de audio y video y conferencias.
4. Número estimado actual de transmisiones de video.
5. El tamaño de la conferencia más grande en curso.
6. La distribución de los tamaños de las conferencias en curso.
7. Agregados de RTT y jitter en todos los usuarios.
8. El número total de conferencias creadas, completadas, fallidas y parcialmente fallidas.
9. El número total de mensajes enviados y recibidos a través de canales de datos WebRTC y sockets web COLIBRI.
10. La duración total de todas las conferencias completadas.
11. El número de sesiones ICE establecidas sobre UDP o TCP.

Jitsi Videobridge utiliza los siguientes nombres de estadísticas en los informes:

- current_timestamp: el valor es la fecha y hora en que se generan las estadísticas (en UTC).
- threads: el número de subprocesos de Java que utiliza el puente de vídeo.
- bit_rate_download / bit_rate_upload: la tasa de bits total entrante y saliente (respectivamente) para el puente de video en kilobits por segundo.
- packet_rate_download / packet_rate_upload - la tasa total de paquetes entrantes y salientes (respectivamente) para el puente de video en paquetes por segundo.
- loss_rate_download: la fracción de paquetes RTP entrantes perdidos. Esto se basa en números de secuencia RTP y es relativamente preciso.
- loss_rate_upload: la fracción de paquetes RTP salientes perdidos. Esto se basa en los informes del receptor RTCP entrantes y en un intento de restar la fracción de paquetes que no se enviaron (es decir, se perdieron antes de llegar al puente). Además, esto se promedia sobre todos los flujos de todos los usuarios en lugar de todos los paquetes, por lo que no se pondera correctamente. Esto no es exacto, pero puede ser una métrica útil de todos modos.
- rtp_loss: obsoleto. La suma de loss_rate_download y loss_rate_upload .

- jitter_aggregate: experimental. Un valor promedio (en milisegundos) de la fluctuación calculada para los flujos entrantes y salientes. Esto no se ha probado y actualmente no se sabe si los valores son correctos o no.
- rtt_aggregate: un valor promedio (en milisegundos) del RTT en todas las transmisiones.
- largest_conference: el número de participantes en la conferencia más grande que se celebra actualmente en el puente.
- conference_sizes: la distribución de los tamaños de conferencia alojados en el puente. Es una matriz de números enteros de tamaño 15, y el valor en el índice (de base cero) i es el número de conferencias con i participantes. El último elemento (índice 14) también incluye conferencias con más de 14 participantes.
- audiochannels: el número actual de canales de audio.
- videochannels: el número actual de canales de video.
- conferences: el número actual de conferencias.
- participants: el número actual de participantes.
- videostreams: una estimación del número de secuencias de vídeo actuales reenviadas por el puente.
- total_loss_controlled_participant_seconds: el número total de segundos-participante que están controlados por pérdidas.
- total_loss_limited_participant_seconds: el número total de segundos-participante que están limitados por pérdidas.
- total_loss_degraded_participant_seconds: el número total de segundos de participante que están degradados por pérdida.
- total_conference_seconds: la suma de la duración de todas las conferencias completadas, en segundos.
- total_conferences_created: el número total de conferencias creadas en el puente.
- total_failed_conferences: el número total de conferencias fallidas en el puente. Una conferencia se marca como fallida cuando todos sus canales han fallado. Un canal se marca como fallido si no tiene actividad de carga útil.
- total_partially_failed_conferences: el número total de conferencias parcialmente fallidas en el puente. Una conferencia se marca como parcialmente fallida cuando algunos de sus canales han fallado. Un canal se marca como fallido si no tiene actividad de carga útil.
- total_data_channel_messages_received / total_data_channel_messages_sent: el número total de mensajes recibidos y enviados a través de canales de datos.
- total_colibri_web_socket_messages_received / total_colibri_web_socket_messages_sent: el número total de mensajes recibidos y enviados a través de los conectores web COLIBRI.

Las estadísticas están disponibles a través del endpoint /colibri/stats en la interfaz REST privada (si se ha habilitado) en formato JSON:

```
{
  "audiochannels": 0,
  "bit_rate_download": 0,
  "bit_rate_upload": 0,
  "conference_sizes": [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0 ],
  "conferences": 0,
  "current_timestamp": "2019-03-14 11:02:15.184",
  "graceful_shutdown": false,
  "jitter_aggregate": 0,
  "largest_conference": 0,
  "loss_rate_download": 0,
  "loss_rate_upload": 0,
  "packet_rate_download": 0,
  "packet_rate_upload": 0,
  "participants": 0,
  "region": "eu-west-1",
  "relay_id": "10.0.0.5:4096",
  "rtp_loss": 0,
  "rtt_aggregate": 0,
  "threads": 59,
  "total_bytes_received": 257628359,
  "total_bytes_received_octo": 0,
  "total_bytes_sent": 257754048,
  "total_bytes_sent_octo": 0,
  "total_colibri_web_socket_messages_received": 0,
  "total_colibri_web_socket_messages_sent": 0,
  "total_conference_seconds": 470,
  "total_conferences_completed": 1,
  "total_conferences_created": 1,
  "total_data_channel_messages_received": 602,
  "total_data_channel_messages_sent": 600,
  "total_failed_conferences": 0,
  "total_ice_failed": 0,
  "total_ice_succeeded": 2,
  "total_ice_succeeded_tcp": 0,
  "total_loss_controlled_participant_seconds": 847,
  "total_loss_degraded_participant_seconds": 1,
  "total_loss_limited_participant_seconds": 0,
  "total_packets_dropped_octo": 0,
  "total_packets_received": 266644,
  "total_packets_received_octo": 0,
  "total_packets_sent": 266556,
  "total_packets_sent_octo": 0,
  "total_partially_failed_conferences": 0,
  "total_participants": 2,
  "videochannels": 0,
  "videostreams": 0
}
```

Las estadísticas también se pueden publicar periódicamente a través de XMPP (lo que permite a jicofo monitorear un conjunto de puentes y realizar el balanceo de carga). En este caso las estadísticas se representan en formato XML con un elemento stats como este:

```
<stats xmlns=' http://jitsi.org/protocol/colibri'>
        <stat value='2014-07-30 10:13:11.595' name='current_timestamp'/>
        <stat value='229' name='threads'/>
        <stat value='689.0096' name='bit_rate_download'/>
        <stat value='0.00299' name='rtp_loss'/>
        <stat value='4' name='audiochannels'/>
        <stat value='700.9024' name='bit_rate_upload'/>
        <stat value='2' name='conferences'/>
        <stat value='4' name='videochannels'/>
        <stat value='4' name='participants'/>
        <stat value='1' name='total_failed_conferences'/>
        <stat value='1' name='total_partially_failed_conferences'/>
        <stat value='1' name='total_no_payload_channels'/>
        <stat value='2' name='total_no_transport_channels'/>
        <stat value='8' name='total_channels'/>
</stats>
```

La funcionalidad de informes de estadísticas se puede configurar con las siguientes propiedades:

- org.jitsi.videobridge.ENABLE_STATISTICS - propiedad booleana. El valor predeterminado es false
- org.jitsi.videobridge.STATISTICS_TRANSPORT - propiedad de cadena. Una lista de transportes separados por comas. Los transportes admitidos son "muc" y "callstats.io".
- org.jitsi.videobridge.STATISTICS_INTERVAL - propiedad de entero. Esta propiedad especifica el tiempo de informe en milisegundos entre la generación de las estadísticas. Por defecto, el intervalo es de 1000 milisegundos.

2.6 GRABAR Y TRANSMITIR LLAMADAS (JIBRI)

Jibri proporciona servicios para grabar o transmitir una conferencia Jitsi Meet.

Funciona lanzando una instancia de Chrome renderizada en un framebuffer virtual y capturando y codificando la salida con ffmpeg. Está diseñado para ejecutarse en una máquina separada (o una VM), sin otras aplicaciones que utilicen la pantalla o los dispositivos de audio. Solo se admite una grabación a la vez en un solo jibri.

Jibri requiere que se habiliten algunas configuraciones dentro de una configuración de Jitsi Meet. Estos cambios incluyen hosts virtuales y cuentas en Prosody, configuraciones para jitsi meet web (dentro de config.js) así como jicofo sip-communicator.properties.

Estas configuraciones permiten grabar las llamadas realizadas y poder transimitir directamente a YouTube.

2.7 AUTENTICACION

Jitsi permite manejar la autenticación de los usuarios con diferentes configuraciones, La autenticación se puede controlar con las variables de entorno configuradas. Si el acceso de invitados está habilitado, los usuarios no autenticados deberán esperar hasta que un usuario se autentique antes de poder unirse a una sala. Si el acceso de invitados no está habilitado, todos los usuarios deberán autenticarse antes de poder unirse.

Autenticación interna

Es el modo de autenticación predeterminado (internal) utiliza credenciales XMPP para autenticar a los usuarios. Para habilitarlo, debe habilitar la autenticación con ENABLE_AUTH y establecerlo AUTH_TYPE en internal, este tipo de autenticacion requiere que los usuarios esten registrados en prosody.cfg.

Autenticación mediante LDAP

Jitsi permite utilizar LDAP para autenticar usuarios. Para habilitarlo, debe habilitar la autenticación con ENABLE_AUTH y establecerlo AUTH_TYPE en ldap, luego configure los ajustes que puede ver a continuación.

Variable	Descripción	Ejemplo
LDAP_URL	URL para la conexión ldap	ldaps: //ldap.domain.com/
LDAP_BASE	DN base LDAP. Puede estar vacío.	DC = ejemplo, DC = dominio, DC = com
LDAP_BINDDN	DN de usuario LDAP. No especifique este parámetro para el enlace anónimo.	CN = binduser, OU = usuarios, DC = ejemplo, DC = dominio, DC = com
LDAP_BINDPW	Contraseña de usuario LDAP. No especifique este parámetro para el enlace anónimo.	LdapUserPassw0rd
LDAP_FILTER	Filtro LDAP.	(sAMAccountName =% u)

Variable	Descripción	Ejemplo
LDAP_AUTH_METHOD	Método de autenticación LDAP.	enlazar
LDAP_VERSION	Versión del protocolo LDAP	3
LDAP_USE_TLS	Habilitar LDAP TLS	1
LDAP_TLS_CIPHERS	Establecer lista de cifrados TLS para permitir	SECURE256: SECURE128
LDAP_TLS_CHECK_PEER	Requerir y verificar el certificado del servidor LDAP	1
LDAP_TLS_CACERT_FILE	Ruta al archivo de certificado de CA. Se usa cuando la verificación del certificado del servidor está habilitada	/etc/ssl/certs/ca-certificates.crt
LDAP_TLS_CACERT_DIR	Ruta al directorio de certificados de CA Se utiliza cuando la verificación del certificado del servidor está habilitada.	/ etc / ssl / certs
LDAP_START_TLS	Habilite START_TLS, requiere LDAPv3, la URL debe ser ldap: // no ldaps: //	0

Autenticación mediante tokens JWT

Jitsi permite utilizar tokens JWT para autenticar usuarios. Para habilitarlo, debe habilitar la autenticación con ENABLE_AUTH y establecer AUTH_TYPE en jwt, luego configure los ajustes que puede ver a continuación.

Variable	Descripción	Ejemplo
JWT_APP_ID	Identificador de aplicación	my_jitsi_app_id
JWT_APP_SECRET	Secreto de aplicación conocido solo por su token	my_jitsi_app_secret
JWT_ACCEPTED_ISSUERS	(Opcional) Establecer asap_accepted_issuers como una lista separada por comas	my_web_client, my_app_client

Variable	Descripción	Ejemplo
JWT_ACCEPTED_AUDIENCES	(Opcional) Configure asap_accepted_audiences como una lista separada por comas	mi_servidor1, mi_servidor2
JWT_ASAP_KEYSERVER	(Opcional) Establezca asap_keyserver en una URL donde se puedan encontrar claves públicas	https://example.com/asap
JWT_ALLOW_EMPTY	(Opcional) Permita usuarios anónimos sin JWT mientras valida los JWT cuando se proporcionen	0
JWT_AUTH_TYPE	(Opcional) Controla qué módulo se utiliza para procesar los JWT entrantes	simbólico
JWT_TOKEN_AUTH_MODULE	(Opcional) Controla qué módulo se utiliza para validar los JWT	token_verification

2.8 SEGURIDAD Y PRIVACIDAD

En muchos aspectos, las reuniones de Jitsi son simplemente privadas por diseño. Para empezar, todas las salas de reuniones son efímeras: solo existen mientras la reunión tiene lugar. Se crean cuando el primer participante se une y se destruyen cuando el último se va. Si alguien vuelve a unirse a la misma sala, se crea una nueva reunión con el mismo nombre y no hay conexión con ninguna reunión anterior que se haya celebrado con el mismo nombre.

Todo esto es muy importante. Algunos de los sistemas que permiten a las personas "pre-crear" salas, tienen indicaciones sutiles que permiten a un atacante potencial distinguir las reuniones reservadas de las no reservadas, lo que hace que las reuniones reservadas sean más fáciles de identificar y apuntar.

Se debe contar con un generador de nombres de reuniones al azar. Ofrecer nombres que sean fáciles de recordar y leer en voz alta en una llamada telefónica. Esto se puede considerar como un punto minimo pero que agrega seguridad a las reuniones que se generan con nombres automáticamente.

Es recomendable también que los usuarios anfitrion establezcan una contraseña para la reunión.

Jitsi permite una autenticación sólida, por lo que solo los usuarios autorizados serán moderadores, esto debe ser previamente configurado en la instalación realizada.

Las reuniones de Jitsi en general funcionan de 2 formas: peer-to-peer (P2P) o mediante Jitsi Videobridge (JVB). Esto es transparente para el usuario. El modo P2P solo se usa para reuniones 1 a 1. En este caso, el audio y el vídeo se cifran mediante DTLS-SRTP desde el remitente hasta el receptor, incluso si atraviesan componentes de red como servidores TURN.

En el caso de reuniones de varios participantes, todo el tráfico de audio y video todavía está encriptado en la red (nuevamente, usando DTLS-SRTP). Esta capa externa de cifrado DTLS-SRTP se elimina mientras los paquetes atraviesan Jitsi Videobridge; sin embargo, nunca se almacenan en ningún almacenamiento persistente y solo viven en la memoria mientras se enrutan a otros participantes de la reunión.

Es muy importante tener en cuenta que cuando los paquetes también se cifran de extremo a extremo, esta segunda capa de cifrado nunca se elimina (ni puede ser)

Dado que Jitsi está construido sobre WebRTC, una mirada más profunda a su arquitectura de seguridad es muy importante al evaluar los aspectos de seguridad de Jitsi.

De forma predeterminada, Jitsi Meet no requiere que los usuarios creen cuentas. Cualquier información que decidan ingresar, como su nombre o dirección de correo electrónico, es puramente opcional y solo se comparte con otros participantes de la reunión.