

Разбор процедуры звонка

Запуск механизма

При загрузке страницы выполняется последовательно два процесса:

- авторизация на комет-сервере;
- инициализация видео-апи.

1. Авторизация

Выполняется код

```
auth();
```

...который запускает процесс авторизации пользователя, а именно:

- в **LocalStorage** браузера создаётся пользователь, либо читается оттуда, если уже был записан ранее;
- создаётся промис, который ждёт результата аякс-вызова php-скрипта `.../auth.php`
- Аякс-вызов запросом через оболочку CometQL добавляет переданного ему пользователя с кодом **user_id** и хешем **user_hash**:

```
mysqli_query(  
    $link,  
    "INSERT INTO users_auth (id, hash)VALUES('" . mysqli_real_escape_string( $link,  
$user_id ) . "', '" . mysqli_real_escape_string( $link, $user_key ) . "' );"  
);
```

Выполняется задача: поместить в область видимости Комет-сервера и Фрисвитч-сервера пользователя с определёнными данными. **Здесь и далее: взаимодействие серверов с внешним миром строится посредством наличия или отсутствия данных в базе данных.**

Про CometQL:

<https://comet-server.com/wiki/doku.php/comet:cometql#cometql>

Описание предназначение таблиц комет-сервера:

https://comet-server.com/wiki/doku.php/comet:cometql#%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5_%D1%82%D0%B0%D0%B1%D0%BB%D0%B8%D1%86

Список таблиц на данный момент:

Table	Action	Rows	Type	Collation	Size	Overhead
conference	★ Browse Structure Search Insert Empty Drop	20	InnoDB	utf8mb4_general_ci	16 KiB	-
log_event	★ Browse Structure Search Insert Empty Drop	30	InnoDB	utf8_general_ci	16 KiB	-
log_query	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci	16 KiB	-
pipes_settings	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci	48 KiB	-
pipe_messages	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci	64 KiB	-
revoked_tokens	★ Browse Structure Search Insert Empty Drop	0	InnoDB	ascii_bin	16 KiB	-
users_auth	★ Browse Structure Search Insert Empty Drop	19	InnoDB	utf8_general_ci	48 KiB	-
users_data	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci	48 KiB	-
users_messages	★ Browse Structure Search Insert Empty Drop	12	InnoDB	utf8_general_ci	80 KiB	-
users_time	★ Browse Structure Search Insert Empty Drop	2	MEMORY	utf8_general_ci	16 MiB	138
10 tables	Sum	83	InnoDB	utf8_general_ci	16.3 MiB	138

При внешней интеграции (например, с базой данных пользователей Битрикса):

user_id — код пользователя, его можно брать прямо из Битрикса либо строить соответствие, например *код пользователя при видео-звонке соответствует такому-то пользователю в Битриксе*;

user_hash — хеш, он же "секретное слово", он же "соль", он же "пароль". Отвечает за идентификацию того, кто коннектится с кодом **user_id**.

- Промис, дождавшись аякс, запускает механизм соединения к серверу

```
cometApi.start({
  node: "fs.centrmед.ua/comet-server",
  dev_id: 0,
  user_id:user_id,
  user_key:user_key,
})
```

Который фактически устанавливает вкладку в режим "Мастер" и запускает вебсокет-соединение, слушающий, что говорит сервер.

2. Инициализация видео-апи

Производится через вызов

```
cometVideoApi.start({...})
```

...где в качестве параметра передаётся объект с обработчиками событий:

- onCall — перед непосредственно вызовом (присоединением к конференции);
- onCallEnd — по завершении звонка;
- onCallAccept — одноразовый колбек при получении ответа от локатора;
- onCallStart — при установке соединения между звонящим и сервером;
- onOut — когда отсоединился от конференции;
- onIn — когда присоединился к конференции.

Сводится к функции

```
cometVideoApi.startCallWaiting
```

...в которой мы подписываемся на канал

```
"msg.sys_sipCall"
```

через вызов функции

```
cometApi.subscription("msg.sys_sipCall", function(callEvent) {...});
```

Вторым параметром в вызове выше передаём колбэк-функцию, которая дёргается во время активности при звонке (а буквально: что-то проскочило в канале "msg.sys_sipCall").

Пример объекта, который получает колбэк-функция при звонке (нам звонит сервер в ответ на наше желание присоединиться к конференции):

```

callEvent
▼ {data: {...}, server_info: {...}} ⓘ
  ▼ data:
    ▼ message:
      ► __proto__: Object
    ▼ sys:
      callKey: "e54fc520-3d77-11e8-a5c1-901b0e923966"
      callPipe: "web_syscall_vA_QaZNuXxUERBzyrmcLVJpAPqyAcA"
      caller_id: "854373"
      conference: true
      conference_name: "101"
      mode: "video"
      serverName: "fs.centrmед.ua"
      serverPort: "7443"
      sipNumber: "0*101*video"
      stream: 0
      ► __proto__: Object
    ► __proto__: Object
  ▼ server_info:
    event: "sys_sipCall"
    history: false
    marker: undefined
    pipe: "undefined"
    user_id: 0
    uuid: undefined
    ► __proto__: Object
  ► __proto__: Object

```

/// Если пришёл звонок в конференцию, к которой мы уже подключены

На скрине выше: название канала — в ключе

`callEvent.data.sys.callPipe`

== «`web_syscall_vA_QaZNuXxUERBzyrmcLVJpAPqyAcA`»

Если мы по этому каналу уже говорим, срабатывает условие

```

if(
  cometVideoApi._callPipes[callPipeName]
  && cometVideoApi.opt.activeCallEvent
  && cometVideoApi.opt.activeCallEvent.data.sys.callPipe == callPipeName)
{

```

/// Если нам пришёл звонок в канал, к которому мы ещё не присоединились.

Подписываемся на уведомления по этому каналу (в примере:

«`web_syscall_vA_QaZNuXxUERBzyrmcLVJpAPqyAcA`»)

через

```

callEvent._callPipeId = cometApi.subscription(callPipeName,
function(event) {...}

```

Второй параметр — опять же колбек, которая отслеживает сигналы и реагирует на них:

- out
- toggleMute
- toggleHold
- ping
- pong

- accept
- mediaStart
- + состояние `cometVideoApi._callPipes[pipeName].isAccept`

Взаимодействие с ВидеоАпи на примере отключения аудио

Взаимодействие с видео-апи производится через вебсокет и прослушиваемый канал.

Для примера разберём такое действие как отключить аудио.

Кнопка mute вызывает функцию

`MuteAudio()`

которая перескакивает на функцию

`cometVideoApi.muteAudio`

...которая:

- пробрасывает в текущую сессию браузера — на объект `RTCSession` (обслуживается библиотекой `//comet-server.com/jssip.js`) сигнал на то, чтобы отключить аудио-поток. Аудио отключается.
- через общий комет-апи "говорит" в канал событием "toggleMute":
`cometApi.multi_web_pipe_send(cometVideoApi.opt.activeCallEvent.data.sys.callPipe, "toggleMute", { isMute:setMute })`

В канале

«`web_syscall_vA_QaZNuXxUERBzyrmcLVJPdAPqyAcA`»

появляется сообщение с `event == "toggleMute"`, на который реагируют те, кто его слушает

(!!! Работает тогда, когда больше одного участника в конференции)

А именно срабатывает коллбек на прослушку и условие:

```
if(event.server_info.event == "toggleMute" && cometVideoApi._callPipes[pipeName].isAccept)
{
  console.info("callPipe", event.server_info.event, event)
  if(cometVideoApi.opt.onToggleMute)
  {
    cometVideoApi.opt.toggleMute({
      action:"toggleMute",
      uuid: event.server_info.uuid,
      user_id: event.server_info.user_id,
      isMute:event.data.isMute,
      status:event.data.isMute,
      callInfo: cometVideoApi._callPipes[pipeName].originalEvent,
      type: cometVideoApi._callPipes[pipeName].originalEvent.data.sys.type
    })
  }
}
```

Присоединиться к конференции

Кнопка присоединения к конференции (Join to conference) запускает функцию **StartCall**, внутри которой аякс-скрипт `.../call.php`

Задача: добавить в БД запись о новом участнике конференции.

```
$query = "INSERT INTO conference (name, user_id, caller_id, message, profile)VALUES('"
.mysql_real_escape_string($link, $room)."', '"
.mysql_real_escape_string($link, $user_id)."', '"
.mysql_real_escape_string($link, $caller_id)."', '"
.mysql_real_escape_string($link, $message)."', '"
.mysql_real_escape_string($link, $profile)."' );";
```

Описание конкретно данной таблицы (conference):

https://comet-server.com/wiki/doku.php/comet:video:api#cometql_api

То же, но глазами Фрисвитч-сервера: сервер следит за базами данных. В частности за таблицей *conference*. Если в таблице нет записи о том, что в какой-то конференции (по её номеру) нет участников, значит никто и не находится на связи. Как только появляется записи с участниками, фрисвитч понимает, что есть слушатели и начинает синхронизировать потоки между ними.

Как только *call.php* добавил запись про участника, фрисвитч подхватывает его и начинает обслуживать.

Обрыв звонка

Обрыв звонка (кнопка HangUp): запускается функция *EndCall*, которая по цепочке вызовов доходит до *_cometServerApi.prototype.socketArraySend*

По пути формируется объект с параметрами, который встраивается в очередь сообщений (*очередь сообщений* — это переписка, пинг, статистика или обмен чем-то ещё).

Непосредственно очередь сообщений тут:

_cometServerApi.prototype.messageHistory — массив

Пример очередей:

```
▼ 0:
  data: "statistics+{"url":"https://fs.centrimed.ua/videochat_test/", "dev_id": "0", "version": "3.31"}"
  time: 1523441965024
  ► __proto__: Object
▼ 1:
  data: "web_pipe2+web_syscall_vA_QaZNuXxUERBzyrmcLVJpAPqyAcA+ping+{"status": "ping", "_cometApi_uuid": "jsapiyby83cc6BXz"}"
  time: 1523441967782
  ► __proto__: Object
▼ 2:
  data: "web_pipe2+web_syscall_vA_QaZNuXxUERBzyrmcLVJpAPqyAcA+pong+{"status": "pong", "_cometApi_uuid": "jsapiApFAfAM4Kvr"}"
  time: 1523441970540
  ► __proto__: Object
▼ 3:
  data: "subscription+web_syscall_vA_QaZNuXxUERBzyrmcLVJpAPqyAcA+track_web_syscall_vA_QaZNuXxUERBzyrmcLVJpAPqyAcA"
  time: 1523441970592
```

Пример разобранного объекта при разрыве соединения:

```
{
  callKey :
    "d1a0e29e-3d71-11e8-ad0e-901b0e923966"
  status :
    "hangup"
```

```
_cometApi_uuid :  
  "jsapijd1i8tL11YF"  
}
```

Непосредственно строка, отправляемая в вебсокет для разрыва:

```
web_pipe2  
web_syscall_vA_QaZNuXxUERBzyrmcLVJpAPqyAcA  
out  
*  
{"status":"hangup","callKey":"d1a0e29e-3d71-11e8-ad0e-901b0e923966","_cometApi_uuid":"jsapijd1i8tL11YF"}
```

Отправка осуществляется через стандартный `socket.send(data)`;

Команды для обмена с сервером

Обработкой команд занимается сервер.

Их перечень:

```
(*****)
```

Ещё не разгрёб:

Со стороны сервера есть возможность направить запрос к
SELECT * FROM `conference_members`

Тут должна быть информация вытаскиваемая непосредственно в момент вызова из FreeSwitch о продолжительности звонка и участниках

Есть ещё возможность из FreeSwitch отправлять к вам разные события.

Можно внедрить код в dialplan перед подключением к конференции

https://comet-server.com/wiki/doku.php/comet:video:api#%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0_dialplan_defaultxml_%D0%BD%D0%B0_freeswitch

Фактически там можно добавить разные инструкции в том числе и вызов скриптов. Это уже надо смотреть документацию на FreeSwitch

Настройки конференций

https://freeswitch.org/confluence/display/FREESWITCH/mod_conference

(очень рекомендую изучить все)

но **max-members** позволит ограничить количество участников. У вас их 2 и тогда не будет ещё слушателей

auto-record будет начинать запись в файл вашего звонка или можно этой командой начать стриминг rtmp потока в nginx для записи его как live стрима на сервере с nginx который будет не обязательно на машине с фриисвитчем

Если auto-record будет писать в файл то потом этот файл с диска можно будет выкачать любым удобным способом.

Если это будет rtmp стрим то он будет доступен на сервере с nginx до тех пор пока его от туда не удалить.

Ещё момент. Задержка на подключение к видеосвязи занимает обычно от 3 до 15 секунд. В среднем 5 секунд. Есть возможность перед началом звонка выбрать качество связи.

Соответственно если интернет слабый или выбрано высокое качество видео то задержка от начала подключения до того как мы увидим абонента растёт.

Если интернет не тянет высокое качество видео то будет проседать fps и звук а не качество картинки.

Ещё в период работы на телемедициной была неожиданность которая тоже отняла время. Надо иметь возможности по восстановлению звонка после обрыва связи.

У вас в описании всё просто доктор и пациент ждёт в комнате. У меня было сложнее и доктора назначало пациенту.

Но если кто то из них обновит страницу или у кого то интернет пропадёт то предусмотреть восстановление звонка надо будет на уровне приложения. Апи видеочатов у меня такое не делает.