



Инструкция по настройке модуля
интеграции Asterisk с CRM-системой
retailCRM

Оглавление

1. Настройка модуля.....	3
1.1. Настройка интеграции в retailCRM.....	3
1.2. Настройка в личном кабинете модуля.....	4
1.3. Настройка внутренних телефонов сотрудников.....	5
1.4. Настройки телефонии.....	6
1.5. Описание API и параметр clientId	7
2. Настройка asterisk.....	7
2.1. Настройка AMI	7
2.2. Описание API	9
2.2.1. Получение имени клиента и ответственного менеджера.....	9
2.2.2. Создание событий звонка в CRM-системе и выгрузка информации по звонку в CRM.....	10
2.3. Описание AGI-скрипта event.sh	12
3. Техподдержка.....	13

1. Настройка модуля

1.1. Настройка интеграции в retailCRM

Для настройки интеграции Asterisk с retailCRM зайдите в панель управления retailCRM. В разделе Администрирование / Интеграция / Ключи доступа к API нажмите кнопку «Добавить».

Необходимые изменения:

1. **Активность** – checkbox должен быть установлен;
2. **Тип доступа** – ко всем магазинам либо только определенным (если их несколько);
3. **Разрешенные методы API** – для модуля необходимы следующие методы:
 - a. /api/users
 - b. /api/customers
 - c. /api/integration-modules/{code}
 - d. /api/integration-modules/{code}/edit
 - e. /api/telephony/call/event
 - f. /api/telephony/calls/upload
 - g. /api/telephony/manager

Главная → Интеграция → Ключи доступа к API →

Новый ключ API

Ключ *

Активность

Комментарий

Тип доступа *

Магазины

- Демо-магазин
- asterisk-api-alice-trade

Разрешенные методы API *

Заказы

- /api/orders
- /api/orders/create

Скопируйте ключ и нажмите на кнопку «Сохранить».

1.2. Настройка в личном кабинете модуля

Нажмите кнопку “Подключить” в описании модуля Маркетплейса или зайдите в личный кабинет сервиса по адресу: <https://asterisk-api.alice-trade.ru/api/account/>



Авторизация в сервисе asterisk-api

Адрес CRM:*

API ключ сервиса:*

1. В поле «Адрес CRM» укажите URL панели управления retailCRM в виде: https://ваш_магазин.retailcrm.ru/;
2. В поле «API ключ сервиса» сохраненный ключ с предыдущего шага;
3. Нажмите кнопку «Вход».

1.3. Настройка внутренних телефонов сотрудников

На первой вкладке «Ваши телефоны» личного кабинета список ваших сотрудников из retailCRM. Укажите внутренний номер нужных сотрудников в поле «Внутренний добавочный номер» и нажмите на кнопку «Сохранить».

После заполнения всех телефонов нажмите кнопку «Сохранить настройки».

#	Сотрудник	Внутренний добавочный номер	Операции
2	asterisk-retailcrm (████████████████████)	<input type="text" value="01"/>	Сохранить
3	Инна (████████████████████)	<input type="text" value="02"/>	Сохранить
4	Александра (████████████████████)	<input type="text" value="03"/>	Сохранить
5	Юлий (████████████████████)	<input type="text" value="04"/>	Сохранить
6	Зинаида (████████████████████)	<input type="text" value="05"/>	Сохранить
7	Юлий (████████████████████)	<input type="text" value="06"/>	Сохранить

[Сохранить настройки](#)

P.S.: После изменения внутреннего номера менеджера, пожалуйста, нажмите на кнопку "Сохранить", и только после всех изменений данной вкладки - кнопку "Сохранить настройки". Настройки не применяются в CRM до нажатия на кнопку "Сохранить настройки".

В панели управления retailCRM модуль активируется, при переходе на него будет доступна кнопка «Перейти в личный кабинет» для быстрого перехода в личный кабинет модуля для настройки. Checkbox «Активность» модуля позволяет временно деактивировать или повторно активировать данный модуль. При деактивизации функционал телефонии в панели управления retailCRM отключается.

Главная → Интеграция → Маркетплейс →

Настройки интеграции с Asterisk

Основные **Телефония**

[Перейти в личный кабинет](#)

Активность

Список доступных стран Россия [+7]

1.4. Настройки телефонии

Ваши телефоны
Настройки Asterisk
API

Настройки телефонии

Префикс CallerID(name):

Префикс CallerID(name)

Укажите префикс, который будет показываться до имени CallerID(name) - [Prefix]: [CallerID(name)]

Контекст и приоритет исходящих звонков:*

default,1

Укажите наименование контекста и приоритета через запятую: context[,priority]

CallerID(name) звонка менеджеру:

CRM

Введите CallerID(name) для звонка менеджеру от Asterisk

Настройки доступа к АМІ

Имя / адрес АМІ сервера:*

127.0.0.1

Укажите имя или ip-адрес сервера где находится asterisk-сервер с настроенным АМІ

Порт АМІ:*

8088

Введите порт АМІ

Логин АМІ:*

asterisk

Введите имя пользователя АМІ

Пароль АМІ:*

Укажите пароль пользователя АМІ

Введите пароль пользователя АМІ

Сохранить настройки

В данном разделе доступны следующие настройки:

1. **Префикс CallerID(name)** – по умолчанию «пустое значение», при заполнении данная строка будет показана перед именем входящего звонка. При пустом значении префикс не показывается;
2. **Контекст и приоритет исходящих звонков** – по умолчанию «default,1». При звонке из интерфейса retailCRM звонок будет переведен в данный контекст с данным приоритетом;
3. **CallerID(name) звонка менеджеру** – по умолчанию «CRM», при звонке клиенту из интерфейса retailCRM данный звонок будет идентифицироваться этим именем для менеджера;
4. **Имя / адрес АМІ сервера** – для инициализации звонка клиенту на asterisk должен быть настроен АМІ (Asterisk Management Interface). В данном поле должен быть указан внешний адрес вашего сервера asterisk;
5. **Порт АМІ** – по умолчанию «8088», в данном поле указывается TCP порт на котором слушает http-служба с АМІ сервисом;
6. **Логин АМІ** – по умолчанию «asterisk», в данном поле указывается имя пользователя настроенного в АМІ;
7. **Пароль АМІ** – в данном поле указывается пароль для пользователя АМІ.

При сохранении настроек будет проверен доступ к АМІ серверу и произведена проверка авторизации. В случае отсутствия доступа к АМІ будет выведена ошибка: «Нет доступа до АМІ: проверьте адрес, порт, логин или пароль» и изменения не будут сохранены.

Вопросы по настройке АМІ на вашем сервере Asterisk будут рассмотрены подробнее в разделе «Настройка АМІ».

1.5. Описание API и параметр clientId

[Ваши телефоны](#)
[Настройки Asterisk](#)
[API](#)

Параметры API

Параметр clientId:

5494143d6ad0b09976b

Этот параметр используется во всех запросах к API

Описание API и настройка AMI и AGI интерфейсов

API: /api/manager :: получение имени потенциального клиента и ответственного менеджера по входящему номеру телефона

API: /api/event :: создание событий звонка в CRM-системе и загрузка информации по звонку в CRM

AGI: event.sh :: AGI-скрипт логирования события и загрузки файла записи на сервис

AMI: call :: настройка AMI-интерфейса для выполнение звонков из CRM

В разделе API можно узнать основной параметр - clientId который используется для авторизации во всех API-запросах, а так же описание методов API и настройка сервисов AMI и AGI.

2. Настройка asterisk

2.1. Настройка AMI

1. Для начала откроем конфигурационный файл `/etc/asterisk/manager.conf` и внесем следующие изменения:

```
[general]
enabled = yes
port = 5038
bindaddr = 0.0.0.0
webenabled = yes
httptimeout = 60
```

```
[asterisk]
secret = указываем сложный пароль
deny = 0.0.0.0/0.0.0.0
permit = 84.47.136.122/255.255.255.255 ; разрешаем запросы только с
адреса сервиса
read = cdr,reporting,originate
write = reporting,originate
```

2. Теперь нам нужно опубликовать аям интерфейс на стандартном порту 8088 используя http. Для этого редактируем файл `/etc/asterisk/http.conf` (а если его нет то создаем)

```
[general]
enabled=yes
enablestatic=yes
bindaddr=0.0.0.0
bindport=8088
prefix=asterisk
```

3. Далее, в консоли Asterisk мы должны выполнить команду: `core restart now;`
4. Затем для проверки HTTP статуса, так же в консоли Asterisk выполним: `http show status` и если все сделано правильно должны увидеть примерно следующее:

```
CLI> http show status
HTTP Server Status:
Prefix: /asterisk
Server: Asterisk/16.1.1
Server Enabled and Bound to 0.0.0.0:8088

Enabled URI's:
/asterisk/httpstatus => Asterisk HTTP General Status
/asterisk/phoneprov/... => Asterisk HTTP Phone Provisioning Tool
/asterisk/amanager => HTML Manager Event Interface w/Digest
authentication
/asterisk/arawman => Raw HTTP Manager Event Interface w/Digest
authentication /asterisk/manager => HTML Manager Event Interface
/asterisk/rawman => Raw HTTP Manager Event Interface
/asterisk/static/... => Asterisk HTTP Static Delivery
/asterisk/amxml => XML Manager Event Interface w/Digest
authentication
/asterisk/mxml => XML Manager Event Interface
/asterisk/ari/... => Asterisk RESTful API
/asterisk/ws => Asterisk HTTP WebSocket

Enabled Redirects:
None.
```

5. Проверяем наличие доступа к порту ајам интерфейса (tcp, 8088) в файрволе для адреса сервиса 84.47.136.122;
6. Указываем внешний адрес хоста на котором доступен порт ајам интерфейса (tcp, 8088) во вкладке "Настройки Asterisk" и проверяем наличие доступа Сохранением настроек. Если доступа нет, то при сохранении появиться сообщение "Нет доступа до AMI: проверьте адрес, порт, логин или пароль."

2.2. Описание API

2.2.1. Получение имени клиента и ответственного менеджера

Формат запроса:

`https://asterisk-api.alice-trade.ru/api/manager?clientId=<api-токен>&phone=<номер-телефона>&type=<тип-результата>`

Данный вызов необходимо выполнять в контексте входящего звонка до вызова голосового меню. После получения результата запроса в формате json или text результаты присваиваются переменной CALLERID(name) и производится переадресация на ответственного менеджера, если он определен.

Параметры:

clientId (string, обязательный параметр): API-токен, его можно получить на вкладке API личного кабинета;

type (string, обязательный параметр): в каком формате вернуть результат. Варианты: **json** - в json формате, **text** - в текстовом виде: <CallerID(name)>, <code> , где <CallerID(name)> - найденное имя клиента, если клиент не найден, то пустое значение. <code> - внутренний номер ответственного менеджера. Разделитель полей - запятая.

phone (string, обязательный параметр): номер телефона, для которого нужно найти имя и ответственного менеджера.

Пример использования в Asterisk (тип json, нужна поддержка JSON в asterisk):

```
[init-channel]
exten => s,1,NoOp(Context: init-channel ${CALLERID(num)})
same => n,Set(C_RESULT=${CURL(https://asterisk-api.alice-trade.ru/api/manager?clientId=xxxxxxx&type=json&phone=${CALLERID(num)})})
same => n,Set(CALLERID(name)=${JSONELEMENT(C_RESULT,result/callerId)})
same => n,Set(TOEXT=${JSONELEMENT(C_RESULT,result/phone)})
same => n,GotoIf(${TOEXT}?:unknow-client,s,1) ; клиент не известен, кидаем на голосовой автоответчик
same => n,Goto(manager-call,s,1) ; нашли ответственного менеджера
```

Пример использования в Asterisk (тип text):

```
[init-channel]
exten => s,1,NoOp(Context: init-channel ${CALLERID(num)})
same => n,Set(C_RESULT=${CURL(https://asterisk-api.alice-trade.ru/api/manager?clientId=xxxxxxx&type=text&phone=${CALLERID(num)})})
same => n,Set(ARRAY(V_CALLERNAME,V_CODE)=${C_RESULT})
same => n,Set(CALLERID(name)=${V_CALLERNAME})
same => n,Set(TOEXT=${V_CODE})
same => n,GotoIf(${TOEXT}?:unknow-client,s,1) ; клиент не известен, кидаем на голосовой автоответчик
same => n,Goto(manager-call,s,1) ; нашли ответственного менеджера
```

Контекст вызова ответственного менеджера:

```
[manager-call]
exten => s,1,NoOp(Context manager-call)
same => n,Macro(record-call) ; макрос начала записи звонка
same => n,Macro(start-in-call,${TOEXT}) ; Макрос отправки события входящего звонка [type=in]
same => n,Dial(SIP/${TOEXT},60,tr) ; передаем звонок ответственному менеджеру
same => n,Goto(queue-managers,s,1) ; если менеджер не ответил в течении 60 сек., то переводим на контекст очереди
same => n,Hangup()
```

2.2.2. Создание событий звонка в CRM-системе и выгрузка информации по звонку в CRM

Формат запроса:

`https://asterisk-api.alice-trade.ru/api/event?clientId=<api-токен>&type=<тип-события>&uniqueid=<уникальный-идентификатор-звонка>&phone=<номер-телефона>&code=<внутренний-номер>&status=<статус>&direct=<направление>&DATE=<дата>&hold=<время-ожидания>&duration=<длительность>&path=<путь-к-записи>`

Данный вызов API позволяет передать событие входящего или исходящего звонка в CRM, а так же событие завершения звонка и передачу данных о разговоре.

Параметры:

clientId (string, обязательный параметр): API-токен, его можно получить на вкладке API личного кабинета;

type (string, обязательный параметр): тип события, возможные варианты in - событие начала входящего звонка, out - событие начала исходящего звонка, hangup - событие завершения звонка, upload - загрузка записи разговора на сервис. По событию hangup происходит загрузка в CRM информации по разговору;

uniqueid (string, обязательный параметр): уникальный идентификатор разговора, как правило для этого используется переменная `{UNIQUEID}` asterisk;

phone (string, обязательный параметр): входящий номер телефона в случае события in и исходящий в случае события out. То же самое и для события hangup, только в этом случае дополнительно направление указывается в параметре direct;

code (string, array, обязательный параметр): внутренние номера сотрудников/сотрудника, которые должны быть проинформированы о входящем звонке в случае событий in. При этом у этих менеджеров появляется всплывающее окно с информацией о клиенте. В этом случае поле code содержит все нужные внутренние номера телефонов разделенные запятой. При событии out и hangup этот параметр должен содержать единственный внутренний телефонный номер;

status (string, для события hangup): в случае события hangup в этом параметре передается статус завершения разговора. Возможные варианты: ANSWER и ANSWERED - на звонок ответили, CANCEL - звонок был отменен, BUSY - была занята линия;

direct (string, для события hangup): в случае события hangup в этом параметре передается направление совершенного звонка. in - входящий, out - исходящий;

date (string, для события hangup): в случае события hangup в этом параметре передается дата и время совершенного звонка. Формат: YYYY-MM-DD HH:MM:SS;

hold (int, для события hangup): в случае события hangup в этом параметре передается время ожидания ответа менеджера в секундах;

duration (int, для события hangup): в случае события hangup в этом параметре передается время разговора в секундах;

path (string, для события hangup): если вы храните записи разговоров локально у себя, то необходимо передать в этом параметре URL на эту запись, например

<https://login:password@my-server.ru/asterisk/273324.wav> для того, что бы CRM могла воспроизвести ее;

Пример макроса для события in:

```
[macro-start-in-call]
exten => s,1,NoOp(Context macro-start-call code: ${ARG1})
same  => n,Set(T=${CURL(https://asterisk-api.alice-trade.ru/api/event?clientId=XXXXXXXX&phone=${CALLERID(num)}&code=${ARG1}&uniqueid=${UNIQUEID}&type=in)})
```

Пример контекста для события hangup и с переводом на приложение Dial():

```
[incoming-call]
exten => s,1,NoOp(Context incoming-call)
same  => n,Macro(record-call) ; включаем запись звонка
same  => n,Macro(start-in-call,${TOEXT}) ; указывается внутренний номер, которому
будет показана карточка входящего звонка
same  => n,Dial(SIP/${TOEXT},60,tr) ; соединяем с ответственным менеджером
same  => n,Goto(queue-managers,s,1) ; ответственный менеджер спит, передаем
вызов в общую очередь менеджеров
same  => n,Hangup() ; закрываем входящий канал

exten => h,1,DeadAGI(event.sh,
${MIXMONITOR_FILENAME},${UNIQUEID},${CALLERID(num)},{TOEXT},{STRFTIME({EPOCH},,%Y-%m-%d
%H:%M:%S)},"in",{CDR(disposition)},{CDR(billsec)}) ; логируем событие
```

Пример контекста для события hangup и с переводом на приложение Queue():

```
[incoming-call]
exten => s,1,NoOp(Context incoming-call)
same  => n,Macro(record-call) ; включаем запись звонка
same  => n,Macro(start-in-call,"XX,YY,ZZ") ; можно указать список внутренних номеров
у которых всплывет подсказка, пустое значение переменной – всем показать событие входящего звонка
same  => n,Queue(queue_managers,tr,,60) ; помещаем звонок в очередь менеджеров
same  => n,Hangup() ; закрываем входящий канал

exten => h,1,NoOp(Context hangup-queue)
same  => n,Set(TO_EXT=${CUT(MEMBERINTERFACE,/,2)}) ; получаем номер ответившего менеджера
same  => n,DeadAGI(event.sh,
${MIXMONITOR_FILENAME},${UNIQUEID},${CALLERID(num)},{TO_EXT},{STRFTIME({EPOCH},,%Y-%m-%d
%H:%M:%S)},"in",{CDR(disposition)},{CDR(billsec)}) ; логируем событие
```

Для логирования hangup-события используется AGI (Asterisk Gateway Interface) и bash скрипт который анализирует параметры и создает нужные API-запросы. Описание bash-скрипта event.sh будет рассмотрено в следующем разделе.

2.3. Описание AGI-скрипта event.sh

Порядок установки и настройки:

1. Для начала файл необходимо скачать скрипт event.sh с адреса <https://asterisk-api.alice-trade.ru/data/event.sh> , после чего разместить его в директории /var/lib/asterisk/agi-bin или в директории, которая указана в переменной astagidir конфигурационного файла asterisk.conf;
2. Дать права для исполнения для пользователя из под которого запущен Asterisk;
3. Откройте файл event.sh для редактирования и найдите секцию настроек. Укажите в переменной CLIENTID ваше значение api-токена;
4. Если вы используете приложение Queue(), то укажите в переменной QUEUE_LOG путь до файла статистики, как правило по умолчанию это: /var/log/asterisk/queue_log;
5. В первое время рекомендуется включить дебаг, указав в переменной DEBUG_LOG путь до него. Необходимо проверить что данный файл будет иметь права на запись от пользователя из под которого запущен Asterisk. В последствии дебаг можно отключить сделав эту переменную пустой - "";
6. Если вы хотите хранить записи разговоров на нашем сервисе, то необходимо в переменной SEND_RECORD_TO_SERVICE указать yes, иначе пустое значение - "";
7. В случае использования приложения Queue(), а так же Dial() с несколькими вызываемыми каналами, переменная длительности звонка выдает неверное значение, поэтому рекомендуется использовать утилиту soxi для получения информации по длительности записи. Для этого необходимо установить эту утилиту и в переменной SOXI_PATH указать путь до нее. Для того что бы не использовать утилиту soxi сделайте переменную пустой - "";
8. Если вы желаете хранить записи разговоров на своем хранилище, то для прослушивания записей из интерфейса CRM необходимо предопределить переменную RECORD_PATH (искать поиском по скрипту) передав путь до файла записи. Формат пути должен быть таким к примеру: http://my-server.ru/asterisk/YYYY-MM-DD/UUUUUU.wav. В качестве протокола можно так же использовать ftp и https. А так же указывать логин и пароль для доступа к записи.

Примечания и комментарии по скрипту event.sh:

- Если вы используете приложение Dial() с одним вызываемым каналом, то в экстеншене h (событие закрытия каналов) статус звонка будет доступен в переменной \${DIALSTATUS}, а длительность разговора соответственно в переменной \${ANSWEREDTIME}. Итого команда логирования в dialplane может выглядеть так: same => n,DeadAGI(event.sh, \${MIXMONITOR_FILENAME},\${UNIQUEID},\${TO_EXT},\${CALLERID(num)},{STRFTIME (\${EPOCH},,%Y-%m-%d %H:%M:%S)},"in",\${DIALSTATUS},\${ANSWEREDTIME});
- Но если приложение Dial() используется с несколькими вызываемыми каналами, то в этом случае поймать номер телефона кто ответил на звонок можно из переменной \${DIALEDPEERNUMBER}, длительность разговора утилитой soxi, а вот статус звонка к сожалению пока не ясно как получить;

- В случае использования очереди - Queue() все становится еще сложнее. Для начала нужно получить в экстеншене h (событие закрытия каналов) внутренний номер кто ответил на этот разговор. Это можно получить из переменной `#{CUT(MEMBERINTERFACE/,2)}`. Но чтобы эта переменная была заполнена у параметров очереди необходимо прописать следующий параметр: `setinterfacevar=yes`. Несмотря на решение проблемы с тем кто ответил на звонок - данных по статусу и длительности разговора и ожидания в переменных нет. Но их можно найти в лог-файле очереди. Этот параметр `queue_log_name` конфигурационного файла `logger.conf`. По `UNIQUEID` ищем статусы `COMPLETEAGENT` или `COMPLETECALLER` и получаем длительность и статус разговора.

3. Техподдержка

Все вопросы по настройке и работе модуля можно адресовать на e-mail:
asterisk-retailcrm@alice-trade.ru