Vonix
Desenvolvedores

API REST Alimentação/Retorno

Versão 1.5.0 (17/06/2021)

Vonix Av. Dr. Paulo Machado, 489 Campo Grande MS 79021-300 **T** 67 33180700 **F** 67 33180701 contato@vonix.com.br www.vonix.com.br

Índice

Funcionamento Básico

Formato

A API de alimentação e retorno de contatos para as filas que usam discador é totalmente RESTful implementada em cima dos verbos HTTP para uma forma extremamente simples e para uso em qualquer linguagem de programação. Existem algumas URLs que você vai acessar para enviar contatos, excluir contatos que estão na fila e você não quer mais, e pegar os retornos das discagens feitas para os contatos.

Forma de Acesso

O acesso a URL deve ser autenticado. A autenticação a HTTP Basic. O usuário e senha para autenticação é criado para cada sistema integrante. Através deste usuário e senha, o sistema devolverá os retornos somente dos contatos alimentados por este usuário. É possível múltiplas alimentações em paralelo, de sistemas heterogêneos.

A URL para acesso geralmente é baseada no host feeder mais o domínio do cliente (ex: http://feeder.vonix.com.br/). Neste manual, daremos exemplo usando a URL “feeder.vonix.com.br”, usando a ferramenta “curl”, que é um software livre para acesso a URLs, bastante utilizado para testes no mundo inteiro (http://curl.haxx.se/).

Informações sobre as Filas

Você pode consultar as filas que o usuário pode alimentar acessando a URL /queues. Nesta URL, você terá como resposta a lista de filas com discagem automática disponíveis para alimentação.

Endereço (URL)

http://feeder.vonix.com.br/queues

Verbo

GET

Parâmetros

Nenhum

Exemplo

GET http://feeder.vonix.com.br/queues

Exemplo usando curl

curl -i -u user:password http://feeder.vonix.com.br/queues

Resposta (XML)

O retorno é recebido em formato XML.

A tag raiz “queues” contém tags “queue” com cada fila disponível.

Dentro de “queue” você terá as tags “id”, “name” e “description” com as informações sobre a fila.

Segue exemplo de um retorno pego pela URL /queues:

Exemplo de resposta

```
<?xml version="1.0" encoding="UTF-8"?>
<queues>
  <queue id="itaufase1">
    <id>itaufase1</id>
    <name>Itau Fase 1</name>
    <description>Fila de cobrança do Itaú, fase 1.</description>
  </queue>
  <queue id="bmg">
    <id>bmg</id>
    <name>Banco BMG</name>
    <description>Fila de atendimento inicial do Banco BMG.</description>
  </queue>
</queues>
```

Informações sobre uma Fila

Você pode consultar o estado de uma fila de discagens acessando a URL /queue/#ID_DA_FILA/status. Nesta URL, você terá como resposta alguns dados de estado da fila como o número de contatos e quando recebeu contatos pela última vez.

Endereço (URL)

http://feeder.vonix.com.br/queue/#ID_DA_FILA/status

Verbo

GET

Parâmetros

Nenhum

Exemplo

GET http://feeder.vonix.com.br/queue/desenvolvimento/status

Exemplo usando curl

curl -i -u user:password http://feeder.vonix.com.br/queue/desenvolvimento/status

Resposta (XML)

O retorno é recebido em formato XML.

Uma tag "queue" é a raiz, com o parâmetro o ID da fila.

Dentro de “queue” você terá as tags “status”, “stored_contacts” e “last_feed” com as informações sobre a fila naquele instante.

Segue exemplo de um retorno pego pela URL /queue/desenvolvimento/status:

Exemplo de resposta

```
<?xml version="1.0" encoding="UTF-8"?>
<queue id="desenvolvimento">
  <status>DIALING</status>
  <stored_contacts>1210</stored_contacts>
  <last_feed>2012-05-28 15:34:12</last_feed>
</queue>
```

Adicionar Contato

Para adicionar contato, você deve enviar um POST na URL com os dados do contato. Um contato pertence a uma fila de trabalho e pode ter vários telefones. Os dados do contato são passados como parâmetros do request.

Os telefones são passados como parâmetros do tipo array, onde a chave é o ID do telefone e o valor é o número do telefone em formato NACIONAL, ou seja DDD+NUMERO. Por ex: to[1234]=6733180700, to[1235]=6733180701. Para se ordenar a prioridade de discagem entre eles, pode ser adicionado ao número um traço e um inteiro, sendo que quanto menor o número, maior a prioridade de discagem. Caso não seja colocada a prioridade, a ordem será discada de acordo com a ordem enviada dos parâmetros. Um request com prioridade usaria to[1234]=6733180700-1, to[1235]=6733180701-2, neste caso sendo discado primeiramente ao 6733180700 e somente se este não atendesse, discado para o 6733180701.

Para solicitar prioridade ao contato, basta passar o parâmetro **`priority`**. Os contatos são discados na ordem de maior prioridade, ou seja, quanto maior a prioridade, mais cedo ele será discado. Por exemplo, se a fila tiver 100 contatos com prioridade 1, no caso de envio de contatos com prioridade 2, eles serão discados antes dos que já estavam na fila com prioridade 1. É uma forma de “furar a fila”, colocando contato para serem discados imediatamente.

Para agendar o horário de discagem do contato, você deve passar o parâmetro **`schedule_at`**. Caso o contato tenha este parâmetro, será discado na data/hora passada. Caso não tenha, entrará na fila. O formato é timestamp.

Para programar renitência ao contato (retentativas automáticas de discagem para este contato no caso de não conseguirmos atendimento), você deve usar o parâmetro **`retry`**. Neste parâmetro você vai definir quantas vezes devemos retentar o contato, qual o intervalo de tempo entre as tentativas e quais os números que devem ser rediscados. São três informações, sendo que <> é obrigatório e [] é opcional: `retry=<VEZES>,[INTERVALO],[CAUSAS_DAS_CHAMADAS]`, onde VEZES é o número de retentativas de trabalho para este contato (veja, não é para o telefone, mas o contato como um todo), INTERVALO é o tempo de espera entre as tentativas de trabalho para este contato em segundos, e CAUSAS_DAS_CHAMADAS são quais os retornos que os telefones devem ter para entrarem na próxima tentativa, numa lista separada por ponto-e-vírgula (Este parâmetro é auto-filtrado, ou seja, na primeira vez tentamos 5 telefones dele, e 3 retornaram causas que vamos rediscar. Na segunda tentativa, tentamos estes 3 novamente, mas desta vez somente 2 voltaram causas que vamos resdicar. Na terceira, tentamos os 2, e assim sucessivamente. Isto é chamado de auto-higienização do contato, para evitar gasto de retentativas com telefones que não teriam resposta). No caso do intervalo, caso não seja passado, será usado o padrão da fila. No caso das causas, caso não seja passado, serão usadas as causas padrões da fila (não responde, ocupado, fora de área, chamada rejeitada, caixa postal de voz). Esta é a renitência dos telefones que não conseguimos atendimento, mas chegamos no telefone de destino. A renitência dos telefones que não completaram por problema NA REDE DA OPERADORA são retentadas imediatamente várias vezes dentro do intervalo de 10 segundos sempre, sem necessidade de uso deste parâmetro.

Para passar dados adicionais do contato para uso ao longo da chamada (apresentação em screen pop-up, fonação de outros dados em filas de fonação, etc), você deve usar um tipo array no parâmetro contact_data, da mesma forma que passou os telefones do contato. A chave será o nome do dado adicional e o valor preenchido com o valor deste dado, por exemplo: contact_data[cpf]=11111111111, contact_data[idade]=30. Não existe um limite de número de dados adicionais para um contato.

Endereço (URL)

http://feeder.vonix.com.br/contact/#ID

Verbo

POST

Parâmetros

contact_name: Nome do contato enviado. Este nome aparecerá no softphone do atendente quando receber a chamada.
queue: ID da fila de trabalho para este contato
billing_group_id: ID do centro de custo/campanha para este contato
schedule_at: Momento em que o contato deve ser discado, em formato TIMESTAMP (parâmetro opcional, caso não passado, seguirá a ordem de entrada na fila)
priority: Prioridade do contato (parâmetro opcional, caso não passado, prioridade comum com todos os demais contatos na fila)
retry: Regra de renitência para o contato. Formato <VEZES>,[INTERVALO_EM_SEGUNDOS], [CAUSAS_DAS_CHAMADAS_SEPARADAS_POR_PONTO_VIRGULA], sendo que apenas VEZES é obrigatório (parâmetro opcional, caso não passado, seguirá o default configurado na fila para todos os contatos)
contact_data[#CHAVE]: Um dado adicional do contato para ser usado como informação complementar (geralmente usado para informar dados no screen pop-up ou focar dados adicionais numa fila de fonação, por exemplo)
to[#IDDOTELEFONE]: Um número de telefone do contato, no formato NACIONAL (DDD+NUMERO, ex: 1130121234). O id do telefone é a chave e o número é o valor. Pode-se ter vários deste parâmetro num mesmo contato. Se sucedido de "-" e um número inteiro, será tratado como prioridade de discagem. Quanto menor o valor da prioridade, maior prioridade terá o número, sendo o telefone discado antes

Exemplo

POST http://feeder.vonix.com.br/contact/12345?
contact_name=Joao&queue=desenvolvimento&billing_group_id=1&to[123]=1130121234&to[124]=11981341010&contact_data[cpf]=11111111111&contact_data[idade]=18

Exemplo usando curl

curl -i -u user:password -d
"contact_name=Joao&queue=desenvolvimento&billing_group_id=1&to[123]=1130121234&to[124]=11981341010&contact_data[cpf]=11111111111&contact_data[idade]=18" http://feeder.vonix.com.br/contact/12345

(obs: não é necessário colocar tipo POST porque é o default no curl para envio de parametros)

Adicionar Vários Contatos a uma Fila

Para adicionar uma lista de contatos em uma fila de uma única vez, você deve enviar um POST na URL com a lista de contatos em formato JSON. Todos os contatos serão inseridos na fila, com todos os seus dados, e a API retornará o resultado da importação. No caso de um ou mais contatos não terem sido importados por algum erro no contato, a API retornará a lista de contatos não importados.

Deve ser enviado o cabeçalho de Content-Type “application/json“. O limite de contatos por POST é de 1.000 contatos.

Endereço (URL)

http://feeder.vonix.com.br/contacts/#ID_DA_FILA

Verbo

POST

Content-Type

application/json

Corpo do Request

Uma array de objetos. Cada objeto é um contato. Para cada objeto, deve haver os seguintes parâmetros:

id: ID único do contato
contact_name: Nome do contato (Este nome aparecerá no softphone do atendente quando receber a chamada)
billing_group_id: ID do centro de custo/campanha
schedule_at: Momento em que o contato deve ser discado, em formato TIMESTAMP (parâmetro opcional, caso não passado, seguirá a ordem de entrada na fila)
priority: Prioridade do contato (parâmetro opcional, caso não passado, segue ordem de chegada dos contatos na lista e demais contatos na fila)
contact_data: Objeto com dados adicionais do contato para serem usados como informação complementar (geralmente usado para informar dados no screen pop-up ou focar dados adicionais numa fila de fonação, por exemplo)
retry: Regra de renitência para o contato. Formato <VEZES>,[INTERVALO_EM_SEGUNDOS], [CAUSAS_DAS_CHAMADAS_SEPARADAS_POR_PONTO_VIRGULA], sendo que apenas VEZES é obrigatório (parâmetro opcional, caso não passado, seguirá o padrão configurado na fila para todos os contatos)
to: Array com telefones do contato. Cada telefone é um objeto contendo os parâmetros **id** (id do telefone) e **number** (número do telefone em formato NACIONAL, ou seja, DDD+NUMERO. A ordem de discagem (prioridade) entre os telefones será a ordem do array (primeiro será discado primeiro, segundo será segundo, etc)

Exemplo

POST http://feeder.vonix.com.br/contacts/desenvolvimento

```
[
   {
      "id":"976510",
      "contact_name":"Joao da Silva",
      "billing_group_id":"1",
      “priority":"0",
      “contact_data”: {
        “cpf”:”11111111111”,
        “idade”:”30"
      },
      “to": [
         {
            “id":"1",
            "number":"51992944921"
        },
         {
            “id":"2",
            "number":"5132104252"
        },
         {
            “id":"1_51984574794",
            "number":"51984574794"
        }
      ]
   },
   {
      "id":"976471",
      "contact_name":"Rodrigo Santos Souza",
      "billing_group_id":"1",
      "priority":"0",
      “retry”:"3,3600",
      “to": [
         {
            “id":"CEL1",
            "number":"61999062358"
        },
         {
            “id”:”COM1",
            "number":"6133110125"
        }
     ]
   }
]
```

Exemplo usando curl

curl -u user:password -H "Content-Type: application/json" -X POST -d '[{"id":"1234","contact_name":"paulo","billing_group_id":"1","priority":"1","to":[{"id":"1","number":"6733180700"}, {"id":"2","number":"6799298107"}]}]' http://feeder.vonix.com.br/contacts/desenvolvimento

Resposta (JSON)

O retorno é recebido em formato JSON, com um objeto com o resultado da importação dos contatos.

Dentro do objeto, você terá os parametros:

status: status final da importação, que pode ser “error” ou "imported"
message: No caso de erro, a mensagem de descrição do erro na importação
imported_count: Número total de contatos importados para a fila
not_imported_conut: Número total de contatos que não foram importados
malformed_contacts: Array com os contatos que não foram importados

Segue exemplo de um retorno pego pela URL /results:

```
{
   "status": "imported",
   "imported_count": 3
   "not_imported_count": 2,
   "malformed_contacts": [
      {
         "billing_group_id": "1",
         "id": "976510",
         "priority": "0",
         "contact_name": "Rodrigo",
         "to": []
      },
      {
         "billing_group_id": "1",
         "id": "976314",
         "priority": "0",
         "contact_name": “Jose dos Santos",
         "to": [
            {
               "id": "0_53984018006",
               "name": "53984018006"
            }
         ]
      }
   ],
}
```

Remover Contato

Para remover um contato que ainda está na fila de espera de discagem, você deve enviar um DELETE com o ID do contato em questão. O contato será removido da fila de discagens. A URL é a mesma da alimentação.

Endereço (URL)

http://feeder.vonix.com.br/contact/#ID

Verbo

DELETE

Parâmetros

Nenhum

Exemplo

DELETE http://feeder.vonix.com.br/contact/12345

Exemplo usando curl

curl -i -u user:password -X DELETE http://feeder.vonix.com.br/contact/12345

Remover Todos os Contatos de uma Fila

Para remover todos os contatos que ainda estão em uma fila, na espera de discagem, você deve enviar um DELETE com o ID da fila em questão. Todos os contatos desta fila serão removido da fila para discagens. A URL neste caso contém a palavra “contacts” no plural.

Endereço (URL)

http://feeder.vonix.com.br/contacts/#ID_DA_FILA

Verbo

DELETE

Parâmetros

Nenhum

Exemplo

DELETE http://feeder.vonix.com.br/contacts/desenvolvimento

Exemplo usando curl

curl -i -u user:password -X DELETE http://feeder.vonix.com.br/contacts/desenvolvimento

Pegar Resultados

Seguindo a mesma interface das eventos de envio e deleção de contatos, para buscar os resultados dos contatos, deve ser acessada a URL /results. Nesta URL, baixa-se a fila de resultados pendentes para o usuário.

Todo contato que o discador fizer, já coloca online o resultado do contato na fila. E assim ele vai enchendo a fila. Toda vez que você der um GET nesta URL vai pegar tudo que tem na fila pra você e zerar a fila. Assim, o discador vai cotninuar trabalhando os contatos e vai colocando os resultados online na fila. Quando você pegar novamente, vai ter tudo que está na fila (ou seja, todos os retornos que chegaram depois da ultima vez que você acessou a URL). Assim, funciona em forma de pilha, ele vai colocando na pilha em tempo real (quando vai ocorrendo) e voce pega quando preferir. Para pegar os resultados dos contatos discados que estão na fila, você deve enviar um GET na URL. Cada alimentador tem um usuário (que você autentica) e cada usuário só recebe os resultados dos seus contatos enviados.

Endereço (URL)

http://feeder.vonix.com.br/results

Verbo

GET

Parâmetros

Nenhum

Exemplo

GET http://feeder.vonix.com.br/results

Exemplo usando curl

curl -i -u user:password http://feeder.vonix.com.br/results

Resposta (XML)

O retorno é recebido em formato XML.

Uma tag "results" é a raiz, com o parâmetro o ID do alimentador (usuário).

Dentro de “results” você terá os contatos, que são tags "contact", que tem o seu ID, sua fila e seu status como atributos. O status do contato pode ser ***success*** no caso do contato ter sido contatado em algum telefone, ***ivrsuccess*** no caso do contato não ter sido contatado em nenhum dos telefones, mas foi deixado mensagem na caixa postal de algum dos números, ***failure*** no caso do contato não ter sido contatado em nenhum dos telefones depois de tentados todos, ou ***error*** no caso do contato ter algum erro (ex: id duplicado, fila inexistente, etc). No caso de ***failure***, existe um atributo `will_retry` que indica se o contato ainda vai ser retentado automaticamente pela regra de renitência.

Nos casos de ***success***, ***ivrsuccess*** ou ***failure***, a tag "contact" conterá tags "number", que é o resultado da discagem para cada número do contato. Cada tag “number” tem atributos como “id”, que é o ID do telefone que foi alimentado, “dialed_at”, que é a data de discagem “hangup_at”, que é a data de desligamento da chamada deste telefone. Nos casos

de atendimento, também existem os atributos “answered_at” que é a data de atendimento, e “duration” que é a duração de conversação da chamada. Dentro da tag está o código de retorno de resultado da chamada. Telefones não discados tem retorno 0.

No caso de ***error***, existirá uma tag "description" com a descrição do erro.

Segue exemplo de um retorno pego pela URL /results:

```
<?xml version="1.0" encoding="UTF-8"?>
<results feeder_id="sistema">
  <contact status="success" queue="desenvolvimento" id="12346">
    <number dialed_at="16/02/2011 14:28:32" hangup_at="16/02/2011 14:28:54" id="00001">17</number>
    <number duration="10" dialed_at="16/02/2011 14:28:55" hangup_at="16/02/2011 14:29:16" id="00002"
answered_at="16/02/2011 14:29:06">16</number>
  </contact>
  <contact status="success" queue="desenvolvimento" id="12345">
    <number duration="15" dialed_at="16/02/2011 14:28:31" hangup_at="16/02/2011 14:29:16" id="00005"
answered_at="16/02/2011 14:29:02">16</number>
    <number id="00007">0</number>
  </contact>
  <contact status="error" queue="desenvolvimento" id="12347">
    <description>Duplicate contact</description>
  </contact>
  <contact status="failure" will_retry=“yes” queue=“desenvolvimento" id="12348">
    <number dialed_at="16/02/2011 14:30:54" hangup_at="16/02/2011 14:31:54" id="0010">19</number>
    <number dialed_at="16/02/2011 14:32:10" hangup_at="16/02/2011 14:33:10" id="0011">19</number>
  </contact>
</results>
```

Anexo 1: Tabelas

Tabela 1 - Códigos de Causas de Término de Chamada

ID	Causa de Desligamento
0	Cancelado
1	Número Vago
2	Sem Rota Operadora
3	Sem Rota Destino
4	Número com Problema
5	Inclusão DDD Errada
6	Canal Inválido
8	Cancelado Operadora
16	Atendida (desligamento normal)
17	Ocupado
18	Não Responde
19	Não Responde
20	Fora de Área
21	Chamada Rejeitada pela Operadora
22	Número Mudou
23	A Cobrar Rejeitado
26	Término Inesperado
27	Destino com Defeito
28	Formato Inválido
29	Serviço Indisponível
31	Desconectado da Rede
34	Sem Canal Disponível
38	Falha de Rede
41	Falha Temporária
43	Negado pela Rede
53	Destino Bloqueado
55	Destino Bloqueado
91	DDD Inválido
102	Tempo Esgotado
127	Falha de Interconexão
201	Atendimento automático detectado (descarte sem detecção de tipo)
202	Mensagem de operadora
203	Caixa postal de Voz (deixamos recado na caixa)
204	Caixa postal de Voz descartada (não deixamos recado na caixa)

Tabela 2 - Estados das Filas

Estado	Descrição
DIALING	Em discagem
NOCONTACTS	Sem contatos
CLOSEDHOURS	Fora de horário
NOAGENTS	Sem agentes logados
AGENTBUSY	Todos os agentes ocupados
PAUSED	Em pausa (discagens suspensas)
PROCESSINGTTS	Processando fonação (somente filas de fonação)