

API de Integração JavaScript

Versão 1.6.0 (23/08/2017)

Visão Geral	1
Integração de ações e eventos de telefonia	1
Ações	1
Eventos	1
Funcionamento da API	2
Formato	2
Nomenclaturas	2
Como utilizar	2
Utilização do componente	2
Exemplos	2
Página de exemplo	3
Métodos Disponíveis na API	4
Funções de Ações	4
<i>doDial (to, name, queue_id, billing_group_id, action_id)</i>	4
<i>doTransfer (call_id, to)</i>	4
<i>doHangUp (call_id)</i>	4
<i>doPause (reason)</i>	4
<i>doUnpause ()</i>	4
<i>doLogin (location)</i>	5
<i>doLogoff ()</i>	5
<i>doTag (call_id, tag)</i>	5
<i>doStatus (queue_id)</i>	5
Funções de Eventos (agente)	5

<i>onDial (function)</i>	5
<i>onDialAnswer (function)</i>	5
<i>onDialFailure (function)</i>	6
<i>onReceive (function)</i>	6
<i>onReceiveAnswer (function)</i>	6
<i>onReceiveFailure (function)</i>	6
<i>onHangUp (function)</i>	7
<i>onLogin (function)</i>	7
<i>onLogoff (function)</i>	7
<i>onPause (function)</i>	7
<i>onUnpause (function)</i>	7
<i>onStatus (function)</i>	7
Funções de Eventos (controle)	8
<i>onConnect (function)</i>	8
<i>onError (function)</i>	8

Anexo 1: Tabelas	9
Tabela 1 - Motivos de Pausa	9
Tabela 2 - Códigos de Causas de Término de Chamada	9
Tabela 3 - Estados dos Agentes	10

Visão Geral

Integração de ações e eventos de telefonia

Este componente tem a finalidade de integrar, através de ações e eventos, a utilização da telefonia na página web integrante. Para realizar a integração, a comunicação ocorre através de um componente JavaScript adicionado como objeto a uma página web.

Através de métodos deste objeto Javascript na página, é possível enviar comandos a telefonia. Você também pode declarar métodos que serão automaticamente chamados quando eventos referentes ao agente ocorrerem na telefonia. Assim, é possível que se faça pedidos de ações na telefonia diretamente da página web (iniciar nova chamada para o agente, pausar o agente, deslogar o agente da fila, marcar uma chamada com uma tag, etc). e também reagir aos eventos de telefonia que ocorrem na posição de atendimento (mostrar algo no recebimento de uma chamada, desligamento de uma chamada, entrada em pausa do agente, saída do agente, etc).

O componente estabelece uma conexão de rede persistente ao servidor de integração Vonix utilizando o melhor método possível para o navegador utilizado de forma transparente para o programador (websocket nativo, xhr-streaming, etc) para mandar ações e receber eventos. Ele faz as duas direções de comunicação. A conexão se faz através da porta 8010/tcp.

Ações

A primeira direção da comunicação na integração é VOCÊ mandar pedidos de ações para o sistema de telefonia Vonix (são as **ações** citadas no manual). São os métodos iniciados com "do". Quando você executa o método, o sistema de telefonia recebe os dados e executa o pedido da ação na posição de atendimento. Por exemplo, a ação de iniciar uma chamada, que é o método "**doDial**". Toda vez que você executar este método, ele vai originar a chamada para o destino passado. O softphone vai discar essa chamada automaticamente, vinda pelo servidor. Outra ação seria a doPause, para pausar o agente. Quando você executar este método, ele vai colocar o agente em pausa, com o motivo passado como parâmetro no método, para não receber mais chamadas receptivas, contabilizar o tempo de pausa no sistema, etc.

Eventos

A segunda direção da comunicação é o sistema de telefonia sinalizar os eventos que estão ocorrendo na posição para o sistema integrante. São todos os **eventos** de telefonia que ocorrem na posição de atendimento que o componente manda para o seu sistema, se você quiser tomar alguma ação no momento em que o evento ocorre. Basta você passar uma função para os métodos iniciados com "on". Por exemplo, toda vez que chegar uma chamada receptiva na posição de atendimento, o componente vai executar a função que você passou para o método "**onReceive**", passando os parâmetros da chamada descritos neste manual. Isto serve, por exemplo, para que você possa abrir a ficha de um cliente que está entrando na posição de atendimento (já ter a ficha do cliente aberta com todo o histórico de atendimentos, etc). Você pode colocar qualquer função com qualquer código seu e ela executará automaticamente sempre que o evento ocorrer. Existem vários eventos, possibilitando que o seu sistema possa ter total controle sobre a telefonia. Alguns exemplos são quando o atendente inicia uma chamada, quando a chamada que ele está é desligada, quando entra ou sai de pausa, quando o agente loga ou desloga do sistema, etc. Se você não quiser tratar um evento específico, basta não utilizar o declarar método.

Funcionamento da API

Formato

A API é distribuída em forma de um arquivo javascript (.js). Não há nenhuma dependência ou biblioteca adicional para seu funcionamento. O objeto a ser instanciado na sua página é o **Vonix**, e nele estão todos os métodos que você precisará para controlar toda a telefonia (ações e eventos).

Nomenclaturas

A API segue uma convenção de nomenclatura para facilitar o entendimento de seus métodos. Os métodos iniciados por **'do'** são ações, ou seja alguma solicitação que seu sistema pode fazer ao sistema de telefonia. Estes métodos você pode chamar a qualquer momento na sua página (vincular a chamada a um link “a”, ou qualquer outro lugar). Os métodos iniciados por **'on'** aceitam uma função como parâmetro e são onde você passa o código que será executado automaticamente a qualquer momento pelo componente em resposta a um evento ocorrido na telefonia. Você só precisa chamar uma vez as funções iniciadas com **'on'** para passar o código que você queira que seja executado. Sempre que ocorrer, será executado o seu código.

Como utilizar

O componente pode ser instalado em qualquer página HTML. Basta adicionar a tag `<script>` com ele, e depois criar uma instância do objeto **Vonix** para utilizá-lo.

Utilização do componente

Inserir na tag `<HEAD>` do código HTML o script. É ele que cria o objeto **Vonix** que você vai utilizar na sua página. Se o caminho do arquivo não for no raiz da sua aplicação, colocar o caminho correto:

```
<script type="text/javascript" src="vonix-1.5.min.js"></script>
```

Depois disto, crie uma instância do objeto **Vonix** e pronto. Chame todos os métodos a partir dela ao longo da sua aplicação. Na criação do objeto, você deve passar dois parâmetros: o primeiro é o IP ou a URL do servidor de integração da Vonix e o segundo é a matrícula do agente/usuário da Vonix que você vai utilizar. Caso seja passado um endereço IP ou uma URL iniciada com "http", será usada conexão sem SSL. Caso seja passado uma URL iniciada com "https", será utilizada conexão com SSL.

Por exemplo:

```
vonix_agent = new Vonix('192.168.0.1', '1001');
```

Agora o componente passa a observar os eventos do agente, e disparar as funções declaradas nos métodos iniciados com "on" (por exemplo, *onDial* quando for feita alguma chamada pelo agente). Além disso, você já pode chamar qualquer método de ação (por exemplo *doDial*).

Exemplos

Para iniciar uma chamada para o número 6733180700, nome do cliente “Amilcar Silvestre”, na fila default do agente, por exemplo:

```
vonix_agent.doDial('6733180700', 'Amilcar Silvestre', '', 1, '12345');
```

Para pausar o agente no motivo Treinamento:

```
vonix_agent.doPause('2');
```

Para escrever no console toda vez que entrar uma chamada para o agente:

```
vonix.onReceive(  
  function(call_id, date, queue, from, to, callfilename, contact_name, action_id, additional_data) {  
    console.log('Chegou uma chamada do número ' + to);  
  }  
);
```

Página de exemplo

Para facilitar o entendimento, juntamente com esta documentação é fornecida uma página HTML de exemplo completamente funcional, com código-fonte.

Métodos Disponíveis na API

Funções de Ações

doDial (to, name, queue_id, billing_group_id, action_id)

Origina uma chamada para o agente

Parâmetros:

to: Número do telefone em formato nacional ddd e número (ex: 11998761010, 2133141234, 6733180700)

name: Nome do cliente/destino. Irá aparecer para o agente no softphone

queue_id: ID da fila pela qual a chamada será feita (OPCIONAL, caso em branco, usará a fila default do agente)

billing_group_id: ID do centro de custo da chamada. É um inteiro. Caso não use, mandar sempre 1

action_id: ID identificador do cliente/destino, para ser usado por você (pode ser qualquer string/numero/etc)

doTransfer (call_id, to)

Transfere uma chamada do agente em andamento para outro destino. Este método termina a chamada na posição atual, mas sem derrubar a chamada, que vai ser redirecionada para o destino solicitado. Como o agente pode ter mais de uma chamada em andamento, é possível passar o call_id da chamada que se deseja transferir.

Parâmetros:

call_id: ID da chamada que se deseja transferir (OPCIONAL, caso em branco, transfere qualquer chamada em andamento)

to: Número do destino da transferência (geralmente 4 dígitos)

doHangUp (call_id)

Desliga uma chamada do agente em andamento. Como o agente pode ter mais de uma chamada em andamento, é possível passar o call_id da chamada que se deseja desligar.

Parâmetros:

call_id: ID da chamada que se deseja desligar (OPCIONAL, caso em branco, desliga qualquer chamada em andamento)

doPause (reason)

Solicita que o agente entre em pausa. Caso o agente esteja em chamada corrente, a pausa é agendada para ocorrer imediatamente após ao desligamento da(s) chamada(s) em andamento.

Parâmetros:

reason: Código (inteiro) do motivo da pausa, veja tabela anexa a este manual

doUnpause ()

Retira o agente de pausa, voltando para o trabalho

Parâmetros:

Nenhum

doLogin (location)

Loga o agente no ramal citado, para trabalho em suas filas de trabalho. Para usar esta função, é importante passar o número correto do ramal, e que o ramal esteja em funcionamento, ONLINE.

Parâmetros:

location: Número do ramal onde o agente está sentado no momento (4 dígitos)

doLogoff ()

Desloga o agente de suas filas de trabalho.

Parâmetros:

Nenhum

doTag (call_id, tag)

Adiciona uma marcação (etiqueta ou chamada também de tag) a chamada. É útil para marcar as chamadas para facilitar a consulta futura pela operação. Exemplos de tags: “acordo”, “desconhecido” etc.

Parâmetros:

call_id: ID da chamada que se deseja adicionar a tag

tag: Texto com a marcação (tag)

doStatus (queue_id)

Solicita que o sistema de telefonia informe o estado do agente em uma determinada fila. A resposta a este pedido virá na execução da função callback **onStatus**, que será executada em seguida a chamada desta função. Como o agente pode estar em mais de uma fila de atendimento ao mesmo tempo, é possível passar o queue_id da fila que se deseja saber o status.

Parâmetros:

queue_id: ID da fila que se deseja saber o estado do agente (OPCIONAL, caso em branco, usará a fila default do agente)

Funções de Eventos (agente)

onDial (function)

Declara a função a ser executada quando o agente realiza uma chamada ativa. O evento ocorre em qualquer chamada iniciada pelo agente, sendo ela manual ou pela API.

Parâmetros passados a função (call_id, date, queue_id, from, to, callfilename, contact_name, action_id):

call_id: Id único da chamada dentro do sistema de telefonia

date: Data/hora do início da chamada

queue_id: Id da fila a qual a chamada pertence

from: Número de origem da chamada (ramal da posição de atendimento)

to: Número de destino da chamada, em formato nacional de 10 dígitos

callfilename: Nome do arquivo de audio da gravação desta chamada no sistema de telefonia Vonix

contact_name: Nome do contato (caso tenha sido passado)

action_id: Identificador da ação de início da chamada, caso tenha sido solicitada ou feita pelo discador

onDialAnswer (function)

Declara a função a ser executada quando a chamada em curso no agente é atendida. O evento ocorre em qualquer chamada iniciada pelo agente, sendo ela manual ou pela API.

Parâmetros passados a função (call_id, date):

call_id: Id único da chamada dentro do sistema de telefonia

date: Data/hora do início da chamada

onDialFailure (function)

Declara a função a ser executada quando a chamada em curso não é atendida. O evento ocorre em qualquer chamada iniciada pelo agente, sendo ela manual ou pela API.

Parâmetros passados a função (call_id, date, cause_id, cause_description):

call_id: Id único da chamada dentro do sistema de telefonia

date: Data/hora do fim da chamada

cause_id: Identificador da causa de desligamento (ver tabela anexa a este manual)

cause_description: Descrição da causa de desligamento

onReceive (function)

Declara a função a ser executada quando o agente recebe uma chamada em sua posição de atendimento. A chamada pode ser receptiva ou vinda do discador automático. O que diferencia as duas é que no caso da chamada receptiva, o parâmetro *action_id* vira vazio e na chamada vinda do discador, o *action_id* vai vir preenchido com o ID do contato enviado na alimentação do contato.

Parâmetros passados a função (call_id, date, queue_id, from, to, callfilename, contact_name, action_id, additional_data):

call_id: Id único da chamada dentro do sistema de telefonia

date: Data/hora do início da entrada da chamada para o agente

queue_id: Id da fila a qual a chamada pertence

from: Número de origem da chamada, em formato nacional de 10 dígitos

to: Número de destino da chamada (ramal da posição de atendimento)

callfilename: Nome do arquivo de audio da gravação desta chamada no sistema de telefonia Vonix

contact_name: Nome do contato, se disponível

action_id: Identificador do cliente, se disponível (caso tenha sido solicitada ou feita pelo discador, etc)

additional_data: Objeto com dados adicionais (ex: caso o cliente tenha digitado numa URA, etc)

onReceiveAnswer (function)

Declara a função a ser executada quando o agente atende a chamada recebida em sua posição de atendimento (tanto receptiva como vinda do discador).

Parâmetros passados a função (call_id, date, waiting_seconds):

call_id: Id único da chamada dentro do sistema de telefonia

date: Data/hora do início da chamada

waiting_seconds: Segundos que a chamada ficou na fila de espera até ser atendida

onReceiveFailure (function)

Declara a função a ser executada quando o agente não atende a chamada recebida que foi oferecida a ele (tanto receptiva como vinda do discador).

Parâmetros passados a função (call_id, date, waiting_seconds):

call_id: Id único da chamada dentro do sistema de telefonia

date: Data/hora do início da chamada

waiting_seconds: Segundos que a chamada ficou tocando no agente até que ele não atendesse

onHangUp (function)

Declara a função a ser executada no encerramento da chamada em curso, seja ela receptiva, ativa ou vinda do discador automático.

Parâmetros passados a função (call_id, date, cause_id, cause_description):

call_id: Id único da chamada dentro do sistema de telefonia

date: Data/hora do fim da chamada

cause_id: Identificador da causa de desligamento (ver tabela anexa a este manual)

cause_description: Descrição da causa de desligamento

onLogin (function)

Declara a função a ser executada quando o agente se loga para trabalhar.

Parâmetros passados a função (date, location):

date: Data/hora da entrada do agente

location: Ramal da posição de atendimento que o agente se logou

onLogoff (function)

Declara a função a ser executada quando o agente se desloga do trabalho.

Parâmetros passados a função (date, location, duration):

date: Data/hora da saída do agente

location: Ramal da posição de atendimento que o agente estava

duration: Duração total da sessão de trabalho do agente em segundos

onPause (function)

Declara a função a ser executada quando o agente entra em pausa.

Parâmetros passados a função (date, reason):

date: Data/hora da entrada em pausa do agente

reason: ID do motivo da pausa do agente

onUnpause (function)

Declara a função a ser executada quando o agente sai da pausa (volta ao trabalho)

Parâmetros passados a função (date, reason, duration):

date: Data/hora da saída de pausa do agente

reason: ID do motivo da pausa do agente

duration: Duração total da pausa do agente em segundos

onStatus (function)

Declara a função a ser executada indicando o estado do agente no momento, ou em resposta a ação **doStatus**.

Parâmetros passados a função (status, location, since, status_data):

status: Estado do agente (ver tabela anexa a este manual)

location: Posição de atendimento atual do agente

since: Desde quando o agente está neste estado

status_data: Objeto com dados adicionais sobre o estado (ex: motivo da pausa caso em pausa, dados da chamada caso em atendimento, etc)

Funções de Eventos (controle)

onConnect (function)

Ocorre quando a conexão com o servidor de integração é estabelecida com sucesso

Parâmetros passados a função (date):

date: Data/hora da conexão

onError (function)

Ocorre no caso de algum erro, tanto no componente como em alguma ação pedida

Parâmetros passados a função (date, action_id, description):

date: Data/hora da conexão

action_id: Identificador da ação que gerou o erro

description: Mensagem de erro

Anexo 1 : Tabelas

Tabela 1 - Motivos de Pausa

ID	Motivo
1	Avaliação
2	Treinamento
3	Lanche
4	Toaleta
5	Ginástica Laboral
6	Erro no Sistema

Tabela 2 - 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
-----	-----------------------

Tabela 3 - Estados dos Agentes

Estado	Descrição
ONLINE	Agente logado e disponível
OFFLINE	Agente deslogado
PAUSED	Agente em pausa
UNAVAILABLE	Agente indisponível (problema no ramal)
RINGING	Agente em oferecimento de chamada
ONTHEPHONE	Agente ao telefone