



Documento de integração

WebService USCall UC

Sumário

0.0 Objetivo.....	3
0.1 Descrição.....	3
1.0 Funções.....	5
1.1 Função <i>getAgent</i>	5
1.2 Função <i>startCall</i>	5
1.3 Função <i>cdrSearch</i>	5
1.4 Função <i>audioDownload</i>	6
1.5 Função <i>agentLogin</i>	7
1.6 Função <i>agentLogout</i>	7
1.7 Função <i>agentStatus</i>	7
1.8 Função <i>getPauses</i>	8
1.9 Função <i>startPause</i>	9
1.10 Função <i>endPause</i>	9
1.11 Função <i>hangupCall</i>	9
1.12 Função <i>monitExten</i>	10
1.13 Função <i>getExtens</i>	10
1.14 Função <i>extenStatus</i>	11
1.15 Função <i>getQueue</i>	11
1.16 Função <i>queueStatus</i>	12
1.17 Função <i>cdrTariff</i>	12
1.18 Função <i>listTrunk</i>	13
1.19 Função <i>listURA</i>	14
1.20 Função <i>startCallura</i>	15
1.21 Função <i>cdrURA</i>	15



0.0 Objetivo

Este documento tem por objetivo descrever a forma de integração de softwares de terceiros através do Webservice USCall UC.

0.1 Descrição

O Webservice USCall é utilizado para integração e comunicação entre diferentes aplicações com o software UScall Cloud UC.

A comunicação é realizada através do formato XML/HTTPS utilizando o protocolo SOAP (Simple Object Access Protocol). As funções do Webservice podem ser acessadas pelo documento WSDL (Web Services Description Language) através da URL:

<https://{endereço IP ou DNS}/aplicativos/webservice/wsuscall.php?wsdl>

As respostas são retornadas no formato JSON (JavaScript Object Notation). Todas as funções devem ser acessadas utilizando um token de autenticação, este token é gerado e mantido pelo administrador da empresa, e deve ser solicitado ao mesmo.

Abaixo um exemplo de conexão cliente, iniciando a função startCall para efetuar uma ligação, linguagem PHP:

```
<?php
ini_set("soap.wsdl_cache_enabled", 0);
$context = stream_context_create([
'ssl' => [
// set some SSL/TLS specific options
'verify_peer' => false,
'verify_peer_name' => false,
'allow_self_signed' => true ]
]);
$client = new SoapClient('https://172.20.0.31/aplicativos/webservice/wsuscall.php?
wsdl', ['stream_context' => $context ]);
$result = $client->startCall('$1$gcNiVTZp$5IqX.EhuJyLNG7pBc5o4..', 'clicktocall',
'1139952800', '2800', '1');
var_dump($result);
?>
```

1.0 Funções

1.1 Função *getAgent*

Esta função é utilizada para listar os agentes de call center cadastrados na empresa. São retornados os dados: Número do agente, Nome do agente, Senha MD5 e Fila Chave. Parâmetros necessários: **Token**

1.2 Função *startCall*

Esta função é utilizada para efetuar uma ligação pelo USCall. Existem duas forma de discagem, outbound ou clicktocal que deve ser informado no parâmetro 'tipo'. **outbound**: É utilizado para efetuar uma ligação ativa por um ramal, no parâmetro origem deve ser informado o número do ramal, e no parâmetro destino o número que será discado.

clicktocal: É utilizado para efetuar uma discagem para um número e direcionar para um DDR configurado no USCall. No parâmetro origem deve ser informado o número que será discado, e no parâmetro destino o número de DDR que irá receber a ligação.

idcti: ID gerado pelo integrador para buscas de gravações (opcional). Para todos os tipos é necessário informar o parâmetro de categoria, que é utilizado para informar a regra de discagem que será utilizada da empresa. O formato do número de discagem deve estar de acordo com as regras cadastradas na categoria informada. Parâmetros necessários: Token, Tipo(outbound ou clicktocal), origem, destino, categoria

1.3 Função *cdrSearch*

Esta função é utilizada para efetuar uma busca dos registros de ligações do PBX. Abaixo descrição dos campos para efetuar os filtros:

dataini: Data e horário inicial do filtro para a busca. Formato deve ser YYYY-MM-DD HH:i:s. Caso NULO, será efetuado o filtro da data atual da busca, com o horário: 00:00:00.

datafin: Data e horário final do filtro para a busca. Formato deve ser YYYY-MM-DD HH:i:s. Caso NULO, será efetuado o filtro da data atual da busca, com o horário: 23:59:59.

origem: Número de origem do filtro para a busca (que originou a ligação). Pode ser NULO.

destino: Número de destino do filtro para a busca (que recebeu a ligação). Pode ser NULO.

idunico: ID do sistema ou cti para buscar um registro em específico.

Parâmetros necessários: **Token**



Campos de retorno:

bilheteunico: ID único da ligação (usado para buscar gravações).

Audio: Nome do arquivo de gravação.

datahora: Data e horário do ligação.

tipo: Tipo da ligação, podem ser: Recebida, Efetuada, Interna, Discador e Click to call.

DDR: Número DDR que a ligação foi recebida.

origem: Número que originou a ligação.

destino: Número de destino da ligação.

tronco: Tronco/Operadora de entrada ou saída da ligação.

espera: Tempo de espera até ser atendida a ligação, formato HH:i:s.

duracao: Tempo total da ligação, formato HH:i:s.

status: Status da ligação podem ser: COMPLETADA, NÃO ATENDIDA, TRANSFERENCIA, OCUPADO, ABANDONO, MSG.OPERADORA, CHAMANDO, ATENDIDO, TELEFONE INDISPONIVEL, FILA VAZIA

idcti: ID gerado pelo integrador para buscas.

1.4 Função audioDownload

Esta função é utilizada para efetuar busca de gravação de uma ligação no PBX. O áudio é retornado codificado em base64 no campo **arquivo_base_64**. Abaixo descrição dos campos:

idunico: ID único da ligação, utilize a função **cdrSearch** para buscar o id da ligação. Ou utilize o idcti gerado pela função startCall.

Formato: Formato do arquivo de áudio para retorno, caso não informado será retornado o formato padrão: WAV (16 bits, 8000Hz). Os formatos disponíveis são mp3, ogg e wav.

Parâmetros necessários: **Token, idunico**

Campos de retorno:

Nomeaudio: Nome do arquivo de áudio no PBX

Data: Data e horário da ligação.

Tipo: Tipo da ligação, consulte a função **cdrSearch** para detalhes dos tipos.

Status: Status da ligação, consulte a função **cdrSearch** para detalhes dos status.

Origem: Número que originou a ligação.



Destino: Número de destino da ligação.

arquivo_base_64: Arquivo binário do áudio, codificado no formato base64

1.5 Função *agentLogin*

Esta função é utilizada para efetuar um login de agente no sistema. Abaixo descrição dos parâmetros:

agente: Número do agente criado no sistema

ramal: Número do ramal que o agente irá utilizar para login

senhamd5: Senha do agente no formato md5

filachave: Fila chave que o agente será associado, a fila deve ser do tipo Call Center, caso não informado será utilizado a fila chave configurada no sistema.

Parâmetros necessários: **Token, agente, ramal, senhamd5**

1.6 Função *agentLogout*

Esta função é utilizada para efetuar um logout de agente no sistema. Abaixo descrição dos parâmetros:

agente: Número do agente logado no sistema

Parâmetros necessários: **Token, agente**

1.7 Função *agentStatus*

Esta função é utilizada para consultar o status dos agentes. Abaixo descrição dos parâmetros:

agente: Número do agente para consulta, caso não informado serão retornados todos agentes criados no sistema

Parâmetros necessários: **Token**



Campos de retorno:

num: Número do agente

status: Status atual do agente, status disponíveis: em pausa, ocupado, tocando, ocioso, ramal inoperante, desconhecido, erro login, nao logado

pausa: Pausa atual do agente

campanha: Campanha atual do agente

ramal: Ramal em que o agente está logado

numero: Número da ligação atual do agente

tempo: Tempo de ligação atual do agente

filas: Filas que este agente está logado

recebidas: Total de ligações recebidas do dia deste agente

efetuadas: Total de ligações efetuadas do dia deste agente

atendfila: Fila atendida da ligação atual do agente (Ligações efetuadas será retornado como Ativo)

idcall: Fila atendida da ligação atual do agente

ddr: Número do DDR da ligação atual do agente

1.8 Função *getPauses*

Esta função é utilizada para consultar as pausa cadastradas no sistema. Abaixo descrição dos parâmetros:

Parâmetros necessários: **Token**

Campos de retorno:

id: ID único da pausa

pause: Nome da pausa

tempo: Tempo máximo em segundos que um agente pode permanecer na pausa (0 para ilimitado)

filas: Fila permitidas para utilizar esta pausa, agentes não logados nestas filas não poderão utilizar a pausa (ALL para todas filas)

1.9 Função *startPause*

Esta função é utilizada para pausar um agente no sistema. Abaixo descrição dos parâmetros:

agente: Número do agente logado no sistema

idpausa: ID da pausa que será efetuada no agente. Utilize a função *getPauses* para consultar o id da pausa

Parâmetros necessários: **Token, agente, idpausa**

1.10 Função *endPause*

Esta função é utilizada para des-pausar um agente no sistema. Abaixo descrição dos parâmetros:

agente: Número do agente logado no sistema

Parâmetros necessários: **Token, agente**

1.11 Função *hangupCall*

Esta função é utilizada para desconectar uma ligação em curso de um ramal e/ou agente.

Abaixo descrição dos parâmetros:

ramal: Número do ramal ou agente que a ligação será desconectada

agente: Informar se o valor no parâmetro ramal é um agente, use o valor **yes** para agente.

Caso não informado será considerado um ramal.

Parâmetros necessários: **Token, ramal**

1.12 Função *monitExten*

Esta função é utilizada para monitorar uma ligação em curso de um ramal e/ou agente.

Abaixo descrição dos parâmetros:

srcramal: Número do ramal que iniciará a monitoria

dstramal: Número do ramal/agente que será monitorado

agente: Informar se o valor no parâmetro dstramal é um agente. Use o valor **yes** para agente, caso não informado será considerado um ramal.

tipo: Tipo da monitoria, os tipos disponíveis são: escuta (monitoria silenciosa), monitor (Monitoria em que o ramal monitorado consegue ouvir o áudio do ramal que iniciou a monitoria), conf (Monitoria em que todos os canais escutam o áudio). Caso nenhum tipo seja informado será usado a monitoria escuta. Parâmetros necessários: **Token, srcramal, dstramal**

1.13 Função *getExtens*

Esta função é utilizada para listar os ramais cadastrados no sistema. Abaixo descrição dos parâmetros:

ramal: Número do ramal para listar, caso não informado serão listados todos ramais.

Parâmetros necessários: **Token**

Campos de retorno:

num: Número do ramal

tipo: Tipo do ramal, tipos disponíveis: callcenter e pbx

nome: Nome do ramal

1.14 Função extenStatus

Esta função é utilizada para visualizar o status dos ramais do sistema. Abaixo descrição dos parâmetros:

tipo: Tipo do(s) ramal(s) para visualização, os tipos disponíveis são: callcenter, pbx, all(todos). Caso não informado serão visualizados todos os tipos de ramais. **ramal:** Número do ramal para visualizar, caso não informado serão visualizados todos ramais.

Parâmetros necessários: **Token**

Campos de retorno:

ramal: Número do ramal

status: Status do ramal, os status disponíveis são: disponível, indisponível, ocupado, tocando, discando

sigame: Número de desvio do ramal

ip: Endereço IPv4 do dispositivo autenticado neste ramal

numero: Número da ligação atual do ramal

tempo: Tempo de ligação atual do ramal

1.15 Função getQueue

Esta função é utilizada para listar as filas cadastradas no sistema. Abaixo descrição dos parâmetros:

fila: Número da fila para listar, caso não informado serão listadas todas as filas. Parâmetros necessários: Token

Campos de retorno:

fila: Número da fila

tipo: Tipo da fila, os tipos disponíveis são: callcenter e pbx

nome: Nome da fila

1.16 Função queueStatus

Esta função é utilizada para listar as ligações em espera nas filas e seus status. Abaixo descrição dos parâmetros:

fila: Número da fila para listar, caso não informado serão listadas todas as filas. Parâmetros necessários: **Token**

Campos de retorno:

fila: Número da fila

status: Status da fila, os status disponíveis são: ok (fila vazia) e busy (ligações em espera na fila)

espera: Quantidade de ligações em espera na fila

numeros: Lista de números em espera na fila (numero: identificação do número em espera, tempo: tempo que o número está em espera na fila)

1.17 Função cdrTariff

Esta função é utilizada para listar os registros de ligações de saída, para uso com tarifadores. Abaixo descrição dos parâmetros:

Dataini: Data e horário inicial para filtro por data, formato para envio de data e horário: YYYY-mm-dd HH:II:SS, caso não informado será usado data do dia da consulta, às 00:00hs.

Datafin: Data e horário final para filtro por data, formato para envio de data e horário: YYYY-mm-dd HH:II:SS, caso não informado será usado data do dia da consulta, às 23:59hs.

Parâmetros necessários: **Token**

Campos de retorno:

datahorario: Data e horário da ligação

ramalnum: Número do ramal que efetuou a ligação.

ramalnome: Nome do ramal que efetuou a ligação.

ramalcanal: Canal utilizado pelo ramal para efetuar a ligação.

numero: Número discado pelo ramal.

tronconome: Nome do tronco por onde foi efetuada a ligação.

truncocanal: Canal utilizado do tronco por onde foi efetuada a ligação.

statuscall: Status da ligação.

durtotal: Duração total da ligação em segundos.

durcall: Duração da ligação em segundos, considerando somente o tempo após atendimento.

idcall: ID único da ligação.

1.18 Função listTrunk

Esta função é utilizada para listar os troncos criados no PBX. Abaixo descrição dos parâmetros:

Parâmetros necessários: **Token**

Campos de retorno:

nome: Nome do tronco

tipo: Tipo do tronco.

canal: Canal usado para discagem pelo tronco.

1.19 Função listURA

Esta função é utilizada para listar as URAS dinâmicas (com sintetizado de voz) criados no PBX. Abaixo descrição dos parâmetros:

idura: ID único da URA para consulta.

Parâmetros necessários: **Token**

Campos de retorno:

id:ID único da URA

nome:Nome da URA.

descricao:Descrição da URA.

fluxos: Fluxos de atendimento da URA, retorno em json de acordo com a quantidade de fluxos cadastrados. Abaixo descrição dos campos do fluxo:

Campos de retorno fluxo:

fluxo: Número único do fluxo.

descricao: Descrição do fluxo.

audio: Áudio iniciado no fluxo de atendimento, pode ser do tipo texto (text), ou arquivo (file). Exemplo de retorno: tipo:texto/arquivo. Para áudio do tipo texto, é possível informar variáveis dinâmicas para uso com a função startCallura, caso o texto esteja formatado para isto, como {CTI_INFO1},{CTI_INFO2} e etc.

Timeout: Tempo limite que será aguardado o dígito ou áudio.

repetir: Quantidade de repetição para tentativas inválidas (não encontradas).

destino_invalido: Destino para entradas inválidas, pode ser um fluxo ou uma entrada no PBX.

tipo_fluxo: Tipo do fluxo, poder ser voz ou dígito.

palavras: Palavras/dígitos cadastrados para direcionar de acordo com as entradas informadas pela origem, retorno em json de acordo com a quantidade de palavras cadastradas. Abaixo descrição dos campos das palavras:

Campos de retorno palavras:

palavra: Palavra ou dígito cadastrado, para mais de uma palavra é utilizado o caractere ',' (vírgula). O caractere '*' (asterisco), representa um armazenamento de informação somente.

destino: Destino para palavras/dígitos que coincidem com as palavras cadastradas, pode ser um fluxo ou uma entrada no PBX.

1.20 Função startCallura

Esta função é utilizada para efetuar uma discagem para um número e direcionar para uma URA de atendimento dinâmica. Abaixo descrição dos parâmetros:

idura: ID único da URA que será direcionado a ligação.

destino: Número telefônico que será discado, deve estar no formato configurado na categoria.

categoria: Número da categoria usada para consultar e direcionar para a regra de discagem.

ldcti: ID gerado pelo integrador para buscas (opcional).

cti_info{numero}: Variáveis que podem ser informadas (de 1 -10) para uso com URA's que tenham sintetizador de voz (tts), onde o que for informado na variável será falado para o número discado, de acordo com o fluxo configurado. Os valores serão interpretados de acordo com o texto do fluxo formatado. Consulte a função **listURA** para mais informações. Parâmetros necessários: **Token, idura, destino, categoria**

1.21 Função cdrURA

Esta função é utilizada para consultar os registro de ligações que entraram nas URA's de atendimento dinâmicas. Abaixo descrição dos parâmetros:

dataini: Data e horário inicial do filtro para a busca. Formato deve ser YYYY-MM-DD HH:i:s. Caso NULO, será efetuado o filtro da data atual da busca, com o horário: 00:00:00.

datafin: Data e horário final do filtro para a busca. Formato deve ser YYYY-MM-DD HH:i:s. Caso NULO, será efetuado o filtro da data atual da busca, com o horário: 23:59:59.



idura: ID único da URA de atendimento para busca.

idunico: ID do sistema ou cti para buscar um registro em específico.

Parâmetros necessários: **Token**

Campos de retorno:

datahora: Data e horário do último registro.

ura: Identificação da URA de atendimento dinâmica.

bilheteunico: ID único da ligação (usado para buscar gravações).

ddr: Número DDR que a ligação foi recebida.

origem: Identificação da origem que iniciou a URA de atendimento.

destino: Identificação do destino que foi direcionado este evento, com base nas configurações de URA

tronco: Identificação do tronco/operadora em que iniciou a ligação.

duracao: Duração deste evento

fluxo: Identificação do fluxo deste evento

palavra_origem: Palavra/dígito informado pela origem solicitado pela URA de atendimento.

palavra_dest: Palavra/dígito que coincide nas palavras configuradas da URA de atendimento

evento: Identificação do evento detalhado deste registro antes de sair da URA de atendimento.

status: Status do último registro antes de sair da URA de atendimento

eventos: Lista detalhada em JSON de todos os registros da ligação, enquanto a mesma estava na URA de atendimento.

idcti: ID gerado pelo integrador