

Webservices Expand

API

Versión	Fecha	Ultimo cambio
1.8	22-Feb-2019	Agregado de Dialer: listado de todas las campañas

Índice

Introducción	5
Webservices SOAP	6
Punto de Acceso.....	6
Autenticación	6
PBX: Manejo de llamadas y central telefónica	7
Originación de llamadas	7
Crear una llamada	8
Obtener estado de una llamada	8
Cancelar una llamada originada.....	9
Ejemplo.....	10
Extensiones existentes en una PBX	10
Acceso a Grabaciones	12
Ejemplo.....	12
Acceso a detalle de llamadas de PBX	12
Ejemplo.....	14
Acceso a detalle de llamadas de PBX Customizado.....	14
Ejemplo.....	15
Callcenters: operativa callcenter y agentes	16
Estado de las colas y agentes asociados.....	16
Ejemplo.....	17
Actividad de una cola de Callcenter	18
Ejemplo.....	20
Actividad de Agentes	21
Ejemplo.....	22
Login de agentes	23
Logout de agentes	24

Pausar agente.....	25
Despausar agente.....	25
Obtener información de agente.....	26
Obtener nombre de usuario	26
Obtener información del usuario	27
Ejemplo.....	27
Obtener tiempos de logueo y pausa.....	28
Ejemplo.....	28
Obtener detalles de sesión de agentes	29
Ejemplo.....	30
Obtener estado de las extensiones	31
Ejemplo.....	32
Gestión de mensajes SMS.....	34
Envío de SMS	34
Estado de SMS enviado	34
API REST	36
Módulo: Expand Dialer.....	38
Agregar contacto	38
Obtener un contacto.....	39
Editar contacto	42
Obtener todas las campañas.....	43
Obtener datos sobre una campaña	44
Obtener todos los contactos de una campaña.....	44
Obtener actividad de la ejecución de una campaña.....	46
Listas	48
API: Listas Simples.....	48
API: Elementos de Listas Simples	49
Ejemplos de uso	51
Panel de agente: llamadas salientes pendientes.....	53
API: Manipulación de llamadas salientes pendientes	54

Ejemplos de uso.....	56
Formularios para Agentes.....	59
API: Formularios para Agentes.....	59
Ejemplos de uso.....	60
Webservices externos.....	62
Mapeo de Número de teléfono a nombre	62
Captura de Datos de IVR.....	63
Implementación de Popups en Callcenter	64

Introducción

eXpand provee un conjunto de Webservices públicos que permiten a sistemas externos (CRM, ERPs, sistemas de backend en general) interactuar con Expand a los efectos de implementar automatismos, acceso programático a datos y reportes, etc.

Este documento especifica estos WebServices (APIs). Utilizando estos es posible acceder a funcionalidades de eXpand, así como hacer que eXpand realice tareas de forma automática, programando la interacción entre eXpand y un programa o un sistema de gestión.

eXpand tiene 2 conjuntos de WebServices, de tipo SOAP (también llamados XML Webservices) y de tipo RESTful (o HTTP Webservices).

Webservices SOAP

Este conjunto de APIs está compuesto de una serie de Webservices de tipo SOAP que abarcan temas como:

- PBX: Acceso a reportes, generación (Origen) de llamadas y acceso a grabaciones de llamadas
- Callcenters: acceso a registros de actividad de la operativa de colas de callcenter y agentes
- SMS: Envío y gestión de mensajes SMS

Punto de Acceso

Los webservices se acceden mediante la URI

`http://IP-SERVER/apiws/2/apiws.php`

El WDSL que los describe se encuentra en

`http://IP-SERVER/apiws/2/apiws.php?wsdl`

En ambos casos *IP-SERVER* corresponde al IP o nombre DNS asignado a la instalación de Expand dentro de su organización.

Autenticación

El acceso a los Webservices requiere el uso de una clave de acceso llamada *authorizedKey*.

Para cada aplicación que acceda a esta API debe generarse una clave de acceso. Esta se genera en la interfaz web de administración, sección Configuración/Sistema/Claves. A cada clave generada se le asocia una descripción que describe el sistema que hace uso de la misma.

PBX: Manejo de llamadas y central telefónica

Permite el acceso a reportes sobre actividad de la PBX, acceso a llamadas grabadas y originar llamadas nuevas.

Originación de llamadas

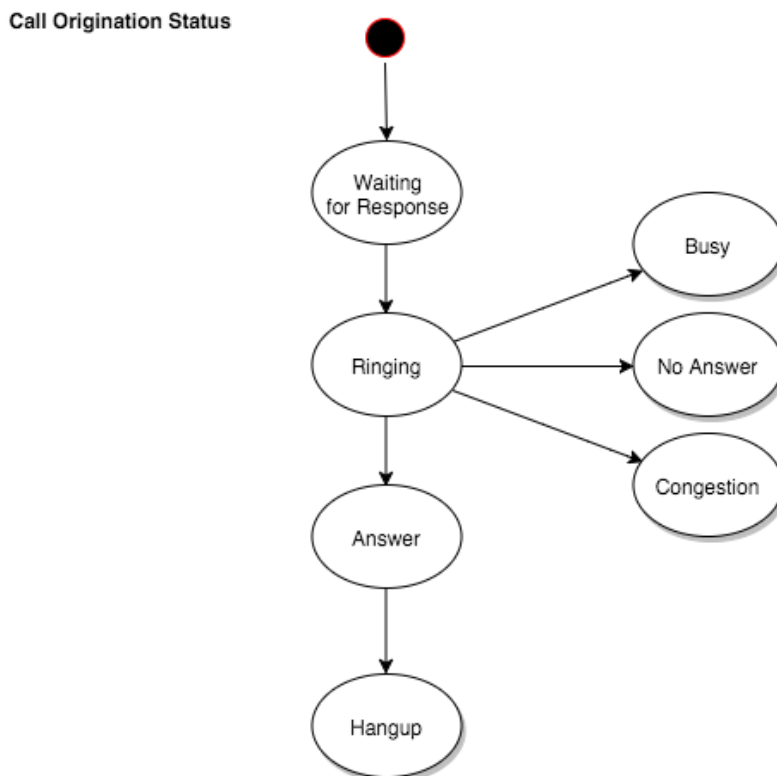
Este conjunto de servicios permiten solicitar la originación (creación) de una llamada, consultar su estado o cancelarlo.

El ciclo de vida típico consiste en:

- Solicitar la originación. Se devuelve un ID de trabajo
- Consultar por el estado de un trabajo hasta que el mismo finaliza

Opcionalmente, es posible cancelar un trabajo en curso.

El siguiente diagrama muestra los estados por los que pasa una llamada creada:



Crear una llamada

```
originateCall(authorizedKey, src, srcCallerID, dst, dstCallerID, timeout, callinfo)
```

Recibe una solicitud de originación de llamada entre src y dst, que quedan registrados con srcCallerID y dstCallerID respectivamente. Se disca por timeout segundos, y es posible asociar información adicional al registro (utilizable en reportes dinámicos) usando el parámetro callinfo.

Ã

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
src	Extensión o número que origina la llamada
srcCallerID	CallerIDName de quien origina la llamada
dst	Extensión o número al que se debe destinar la llamada
dstCallerID	CallerIDName del destino de la llamada
timeout	Tiempo en segundos antes de dar timeout
callinfo	Datos adicionales a registrar junto a esta llamada, lista de atributos clave valor separados por coma.

Devuelve

-1	Error authorizedKey incorrecta.
originatedID	String de ID con el que queda ingresado el trabajo, utilizado para consultar su estado

Obtener estado de una llamada

```
originateCallStatus(authorizedKey, originatedID)
```

Obtiene el estado actual de una solicitud de originación. El estado de un trabajo se mantiene durante el lapso del mismo y por un tiempo adicional configurable en el sistema (típicamente 60 segundos)

Parámetros

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
originatedID	String de ID de trabajo sobre el que se consulta

Devuelve

Retorna un valor numérico indicando el estado de la llamada.

-1	Error authorizedKey incorrecta
0	Error inesperado o ID de trabajo inexistente
1	RINGING: Llamada siendo discada
2	ANSWER: llamada contestada, en curso
3	HUNGUP: Llamada finalizada normalmente
4	BUSY: Llamada realizada y dio ocupado. Trabajo terminado.
5	NO ANSWER: Llamada realizada y no contesto nadie. Trabajo terminado.
6	CONGESTION o NOTAVAILABLE: Llamada no pudo ser realizada por falla en canal saliente o quien la recibe la rechazó. Trabajo terminado.
7	OFFHOOK: Llamada no pudo ser realizada porque la parte originadora tenía la extensión en estado OFFHOOK. Trabajo terminado.

Cancelar una llamada originada

`originateCallCancel(authorizedKey, originatedID)`

Permite la cancelación de una originación de llamada en curso

Parámetros

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
---------------	--

originatedID	String de ID de trabajo sobre el que se consulta
--------------	--

Devuelve

Retorna un valor un numérico indicando el estado de la cancelación

-1	Error authorizedKey incorrecta
0	Cancelación fallida: no existe el originatedID o llamada ya terminada
1	OK: llamada cancelada

Ejemplo

El siguiente ejemplo solicita una llamada entre el interno 701 y el número de Netgate y luego consulta el estado de ese trabajo. Se asocia el dato '01-43-1' a la llamada, para que el mismo figure en los reportes (exportación de datos) posteriores.

```
originateCall('319a67432f51ed53938542b809320dd2', "701", "Juan Perez",
"927126666", "Netgate", 20, info="01-43-1")
```

Retorna "e68af1aa"

```
originateCallStatus('319a67432f51ed53938542b809320dd2', "e68af1aa")
```

Extensiones existentes en una PBX

Permite acceder a la lista de extensiones definidas en una PBX

```
extensionPbx(authorizedKey)
```

Parámetros

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
---------------	--

Devuelve un XML con la lista de extensiones, informando para cada extensión el canal y CallerID asociado.

```
<?xml version="1.0" encoding="UTF-8"?>
<extensions>
```

```
<extensioninfo extension="101" channel="SIP/101" calleridName="Juan Perez"/>  
<extensioninfo extension="102" channel="SIP/102" calleridName="Pedro Gonzalez"/>  
<extensioninfo extension="103" channel="SIP/103" calleridName="Esteban  
Rodriguez"/>  
<extensioninfo extension="321" channel="SIP/321" calleridName="321"/>  
</extensions>
```

Acceso a Grabaciones

Permite obtener el audio de una llamada a partir de su uniqueid.

```
recordingPbx(authorizedKey, uniqueId)
```

Parámetros

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
uniqueId	Identificador único de una llamada generado por Expand

Devuelve un XML conteniendo un status code:

-1	Error authorizedKey incorrecta o no existe el UniqueID
0	Uniqueid valido pero grabación no disponible
1	Grabación disponible e incluida en XML

Si status es 1, el XML incluye el UniqueID y la grabación de la llamada especificada, codificada en formato MP3 y luego serializada en Base64 para poder ser transmitida sobre SOAP (ya que no es posible transmisión binaria sobre SOAP)

Ejemplo

```
<?xml version="1.0" encoding="UTF-8" ?>
<record>
  <recording status="1" uniqueid="12548543.8" encoding="base64" >.....</recording>
</record>
```

Acceso a detalle de llamadas de PBX

Permite acceder a la lista de llamadas generadas en la central ó acceder al detalle de una llamada en particular.

```
cdrData(authorizedKey,uniqueid,inicio,fin)
```

Parámetros

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
uniqueID	Identificador único de una llamada. En caso de no especificarlo se accede al listado de llamadas para el período indicado entre la fecha de inicio y la fecha de fin.
inicio	Fecha de inicio para obtener el listado de llamadas
fin	Fecha fin para obtener el listado de llamadas

Devuelve un XML con la lista de llamadas con los siguientes atributos para cada llamada:

UNIQUEID	Identificador único de llamada
SRC	Número que originó la llamada
DST	Número destino de la llamada
DCONTEXT	Contexto de la llamada (típicamente el Departamento)
CLID	Callerid de llamada
CHANNEL	Canal por el cual se estableció la llamada
DSTCHANNEL	Canal usado en la llamada destino
LASTAPP	Última función o procedimiento ejecutado por eXpand
LASTDATA	Último dato utilizado por la función o procedimiento de Lastapp
CALLDATE	Fecha en la cual se originó la llamada
DURATION	Duración de llamada contemplando el tiempo de ringing
BILLSEC	Duración de la llamada desde que se estableció la comunicación
DISPOSITION	Estado final de la llamada por ejemplo: ANSWER, BUSY, CONGESTION, CANCEL

Ejemplo

```
<?xml version="1.0" encoding="UTF-8" ?>
<cdr>
  <call uniqueid="1333249364.937" src="103" dst="211" dcontext="tecnico"
clid="103 (103)" channel="SIP/103-00000256" dstchannel="" lastapp="Hangup"
lastdata="" calldate="2012-04-01 00:02:44" duration="4" billsec="4"
disposition="ANSWERED" />
  <call uniqueid="1333249485.938" src="993343443" dst="q_tecnico"
dcontext="queues" clid=" Tecnico (993343443)" channel="IAX2/tcc-1523"
dstchannel="SIP/103-00000258" lastapp="AGI" lastdata="GET" calldate="2012-04-01
00:04:45" duration="150" billsec="150" disposition="ANSWERED" />
</cdr>
```

Acceso a detalle de llamadas de PBX Customizado

Permite acceder a la lista de llamadas generadas en la central ó acceder al detalle de una llamada en particular, con customizaciones específicas de una instalación eXpand

```
cdrCustom(authorizedKey, inicio, fin)
```

Parámetros

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
inicio	Fecha de inicio para obtener el listado de llamadas
fin	Fecha fin para obtener el listado de llamadas

Devuelve un XML con la lista de llamadas con los siguientes atributos para cada llamada:

ID	
UNIQUEID	Identificador único de llamada
SRC	Número del abonado que origina la llamada, si está disponible (Caller ID)
DNIS	Número que identifica el servicio al que llamó el abonado. Por ejemplo 1710

CALLDATE	Timestamp de inicio de la llamada, en formato YYYY-MM-DD HH:MM:SS
CONTEXT	Nombre del contexto en expand donde finaliza la llamada
DURATION	Duración completa de llamada en segundos, contemplando el tiempo de establecimiento
ESTADO	Estado en que finalizó la llamada
OBSERVACIONES	Observaciones inherentes al estado de la llamada

Los campos de ESTADO y OBSERVACIONES son particulares de la instalación de eXpand, y su semántica está dada por requerimientos específicos de cada cliente.

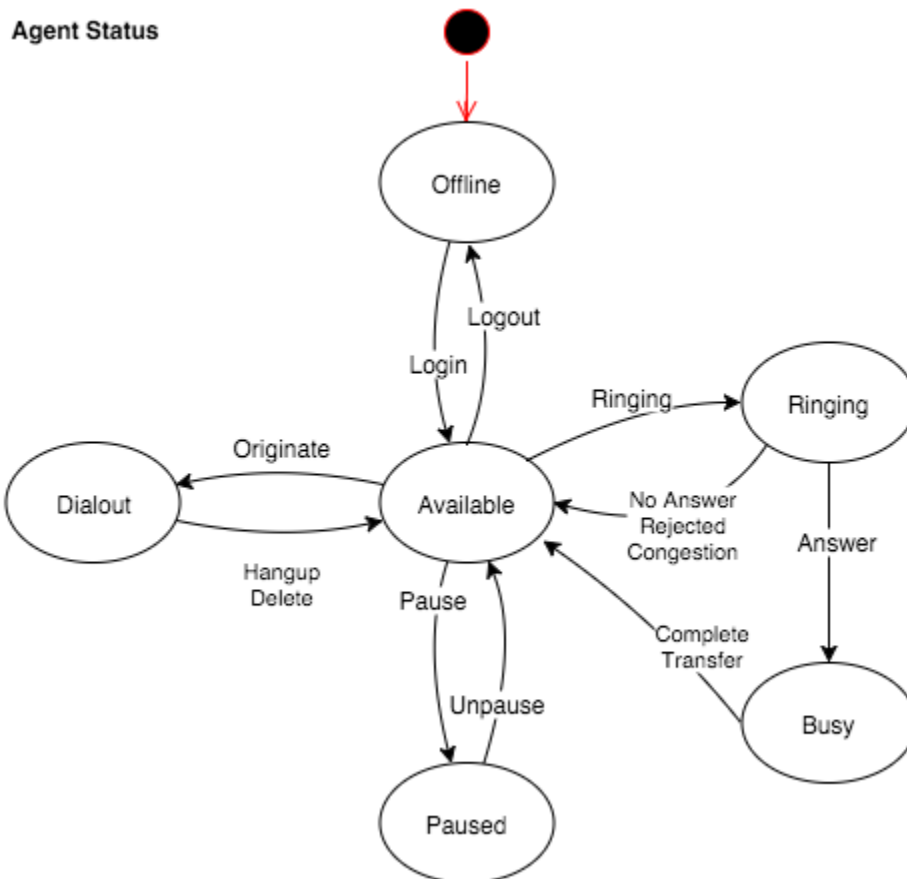
Ejemplo

```
<call uniqueid="1475757732.921" src="23083638" dnis="23207769"
context="nombreContextoExpand"
calldate="2016-10-06 09:42:12" duration="19" estado="xxx" observaciones="xxx" />
```

Callcenters: operativa callcenter y agentes

Este conjunto de webservices permiten obtener reportes de actividad de callcenters, agentes y colas, así como realizar acciones sobre los agentes.

A modo de introducción, un agente vinculado a una cola recibe llamadas derivadas a la cola (también llamada *skill*), siguiendo el siguiente diagrama de estados:



Estado de las colas y agentes asociados

Permite obtener un xml representando el estado actual del módulo de callcenters.

`statusQueues(authorizedKey)`

Parámetros

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
---------------	--

Devuelve un XML con la lista colas, agentes, estados y llamadas en curso.

Ejemplo

```
<?xml version="1.0" encoding="iso-8859-1"?>
<panel generated="1275586846.78" >
  <agents>
    <agent lastcongestion="0" laststatuschange="0" name="Juan Perez"
status="offline" warning="0">104</agent>
    <agent lastcongestion="0" laststatuschange="1275585158" name="Facundo
Martinez" status="busyout" warning="0">105</agent>
    <agent lastcongestion="0" laststatuschange="1275585158" name="Elisa
Rodriguez" status="offline" warning="0">106</agent>
  </agents>
  <queues>
    <queue qname="q_call" name="Callcenter">
      <qstatus>
        <abandoned>0</abandoned>
        <completed>0</completed>
        <counterabn>0</counterabn>
        <counterans>0</counterans>
        <numongoing>3</numongoing>
        <numwaiting>1</numwaiting>
        <ratio>0</ratio>
      </qstatus>
      <qagents>
        <qagent logged="true" paused="false" status="waiting"
pausetype="">104</qagent>
        <qagent logged="true" paused="false" status="waiting"
pausetype="">105</qagent>
        <qagent logged="true" paused="true" status="waiting"
pausetype="Almuerzo">106</qagent>
      </qagents>
    </queue>
    <qcalls>
      <call type="in" callerid="5062934" calleridname=""
connecttime="1275586803" intime="1275586803" uniqueid="1275586803.1502" />104</call>
```

```
<call type="out" callerid="5062934" calleridname=""
connecttime="1275586803" intime="1275586803" uniqueid="1275586803.1502" />105</call>
<call type="out" callerid="5062934" calleridname=""
connecttime="1275586803" intime="1275586803" uniqueid="1275586803.1502" />0</call>
</qcalls>
</queue>
</queues>
</panel>
```

Actividad de una cola de Callcenter

Permite acceder a un XML con registros de actividad de una cola de callcenter

```
queueData(Cola, [Agente], FechaInicio, FechaFin)
```

Parámetros

Cola	ID de la cola a analizas
Agente	Retorna solo la actividad del ID de agente indicado. Invocar con string vacío para obtener la actividad de todos los agentes
FechaInicio	Fecha y hora de inicio del reporte en formato YYYY-MM-DD HH:MM:SS
FechaFin	Fecha y hora de finalización del reporte en formato YYYY-MM-DD HH:MM:SS

Devuelve un XML conteniendo eventos de actividad de esa cola entre las fechas dadas. Cada evento contiene:

Fecha del evento
Hora del evento
Actividad registrada
Parámetros de la actividad

Las actividades y sus parámetros se desglosan en:

ENTERQUEUE(UniqueID, CallerID)	La llamada indicada entró a la cola
RINGING(UniqueID, Agent)	La llamada indicada le está sonando al agente
CONNECT(UniqueID, Agent, WaitSec)	La llamada conectó con el Agente luego de *WaitSec* segundos de espera
NOANSWER(UniqueID, Agent)	El agente no contestó la llamada luego de los segundos de discado configurados
CONGESTION(UniqueID, Agent)	El agente rechazó la llamada inmediatamente
COMPLETECALLER(UniqueID, Agent, TalkSec)	La llamada indicada con el Agente dado fue completada y cortada por el llamante luego de *TalkSec* segundos
COMPLETEAGENT(UniqueID, Agent, TalkSec)	Idem, completada por el Agente
COMPLETETRANSFER(UniqueID, Agent, TalkSec, Dest)	Idem, completada con una transferencia al destino indicado
ABANDON(UniqueID, WaitSec)	La llamada indicada es abandonada luego de *WaitSec* segundos
DIALOUT(UniqueID, Agent, Dest)	El agente inicia una llamada saliente
DIALOUTEND(UniqueID, Agent, TalkSec)	Se completa la saliente luego de *TalkSec* segundos
LOGIN(Agent)	El Agente dado se loguea a la cola
LOGOUT(Agent)	El Agente dado se desloguea de la cola
PAUSE(Agente, Cause)	El Agente dado inicia una pausa, con la razón cause dada

UNPAUSE(Agente)	El Agente dado termina una pausa
-----------------	----------------------------------

Ejemplo

```
<?xml version="1.0" encoding="UTF-8" ?>
<queue_activity>
  <activity date="2012-05-02 09:13:17" queue="q_ven" uniqueid="NONE"
action="LOGIN" callerid="" agent="157" waitsec="" talksec="" dest="" cause="" />
  <activity date="2012-05-02 09:18:06" queue="q_ven"
uniqueid="1335961066.118" action="RINGING" callerid="" agent="157" waitsec=""
talksec="" dest="" cause="" />
  <activity date="2012-05-02 09:18:11" queue="q_ven"
uniqueid="1335961066.118" action="CONNECT" callerid="" agent="157" waitsec="5"
talksec="" dest="" cause="" />
  <activity date="2012-05-02 09:20:02" queue="q_ven"
uniqueid="1335961066.118" action="COMPLETECALLER" callerid="" agent="157" waitsec=""
talksec="111" dest="" cause="" />
  <activity date="2012-05-02 09:26:12" queue="q_ven" uniqueid="NONE"
action="PAUSE" callerid="" agent="157" waitsec="" talksec="" dest="" cause="Tareas
administrativas" />
  <activity date="2012-05-02 09:29:44" queue="q_ven" uniqueid="NONE"
action="UNPAUSE" callerid="" agent="157" waitsec="" talksec="" dest="" cause="" />
</queue_activity>
```

Actividad de Agentes

Retorna registros de actividad de agentes.

`agentData(Cola, FechaInicio, FechaFin)`

Parámetros:

Cola	ID de la cola a analizar
FechaInicio	Fecha y hora de inicio del reporte en formato YYYY-MM-DD HH:MM:SS
FechaFin	Fecha y hora de finalización del reporte en formato YYYY-MM-DD HH:MM:SS

Devuelve un XML con registros de los agentes con actividad en la cola dada entre las fechas indicadas.

Cada registro contiene:

Agente	Número y nombre de agente
Logueado	Tiempo total logueado expresado en formato HH MM' SS"
LogueadoSeg	Tiempo total logueado expresado en segundos
Pausado	Tiempo total pausado, expresado en formato HH MM' SS"
PausadoSec	Tiempo total pausado, expresado en segundos
Promedio	Duración promedio de las llamadas, en formato HH MM' SS"
PremedioSeg	Duración promedio de las llamadas, expresado en segundos
Hablado	Tiempo total hablado en llamadas entrantes de cola, expresado en HH MM' SS"
HabladoSeg	Tiempo total hablado en llamadas entrantes de cola, expresado en segundos
HabladoSaliente	Tiempo total hablado en llamadas salientes de cola, expresado en HH MM' SS"

HabladoSeg	Tiempo total hablado en llamadas salientes de cola, expresado en segundos
Disponible	Tiempo total que el agente estuvo disponible, expresado en HH MM' SS"
DisponibleSeg	Tiempo total que el agente estuvo disponible, expresado en segundos
Atendidas	Cantidad de llamadas atendidas
Perdidas	Cantidad de llamadas perdidas por el agente (ofrecidas no atendidas)
Presentadas	Cantidad de llamadas presentadas al agente
Salientes	Cantidad de llamadas salientes realizadas por el agente
Entrantes	Cantidad de llamadas entrantes
EntrantesTransferidas	Cantidad de llamadas entrantes transferidas
AtendidasCortas	Cantidad de llamadas atendidas cortas
PromedioEntranteYSaliente	Tiempo promedio total de llamadas entrantes y salientes

Ejemplo

```
<?xml version="1.0" encoding="UTF-8" ?>
<agents>
  <agent agente="(152) Angela Batto" logueado="07h 10' 40'"
logueadoSeg="25840" pausado="00h 58' 03'" pausadoSeg="3483"
  promedio="00h 02' 40'" promedioSeg="159.76923076923" hablado="00h 34'
37'" habladoSeg="2077"
  habladoSaliente="00h 00' 00'" habladoSalienteSeg="0" disponible="05h
38' 00'" disponibleSeg="20280"
  atendidas="13" perdidas="0" presentadas="13" salientes="8" entrantes="0"
entrantesTransferidas="0"
  atendidasCortas="0" promedioEntranteYSaliente="0"/>
  <agent agente="(155) Elisa Perez" logueado="05h 45' 50'"
logueadoSeg="20750" pausado="01h 17' 37'" pausadoSeg="4657"
  promedio="00h 03' 24'" promedioSeg="203.76923076923" hablado="00h 44'
09'" habladoSeg="2649"
```

```

        habladoSaliente="00h 00' 00'" habladoSalienteSeg="0" disponible="03h
44' 04'" disponibleSeg="13444"
        atendidas="13" perdidas="1" presentadas="14" salientes="23"
entrantes="0" entrantesTransferidas="0" atendidasCortas="0"
        promedioEntranteYSaliente="0" />
        <agent agente="(157) Horacio Fraga" logueado="06h 47' 05'"
logueadoSeg="24425" pausado="01h 49' 32'" pausadoSeg="6572"
        promedio="00h 01' 24'" promedioSeg="84.071428571429" hablado="00h 19'
37'" habladoSeg="1177"
        habladoSaliente="00h 00' 00'" habladoSalienteSeg="0" disponible="04h
37' 56'" disponibleSeg="16676"
        atendidas="14" perdidas="0" presentadas="14" salientes="7" entrantes="0"
entrantesTransferidas="0" atendidasCortas="0"
        promedioEntranteYSaliente="0" />
        <agent agente="(153) Karina Nasif" logueado="07h 00' 07'"
logueadoSeg="25207" pausado="01h 06' 18'" pausadoSeg="3978"
        promedio="00h 02' 29'" promedioSeg="149.1875" hablado="00h 39' 47'"
habladoSeg="2387"
        habladoSaliente="00h 00' 00'" habladoSalienteSeg="0" disponible="05h
14' 02'" disponibleSeg="18842"
        atendidas="16" perdidas="0" presentadas="16" salientes="75"
entrantes="0" entrantesTransferidas="0" atendidasCortas="0"
        promedioEntranteYSaliente="0" />
        <agent agente="(154) Maria Ines Rodriguez" logueado="06h 58' 32'"
logueadoSeg="25112" pausado="01h 47' 24'" pausadoSeg="6444"
        promedio="00h 04' 05'" promedioSeg="244.71428571429" hablado="00h 57'
06'" habladoSeg="3426"
        habladoSaliente="00h 00' 00'" habladoSalienteSeg="0" disponible="04h
14' 02'" disponibleSeg="15242"
        atendidas="14" perdidas="0" presentadas="14" salientes="30"
entrantes="0" entrantesTransferidas="0" atendidasCortas="0"
        promedioEntranteYSaliente="0" />
    </agents>

```

Login de agentes

Permite loguear un agente de callcenter a todas las colas en las cuales se encuentre vinculado.
`loginAgent(authorizedKey, agent, extension)`

Parámetros:

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
agent	Usuario correspondiente al agente a ser logueado
extension	Interno al cual se intenta loguear el agente

Devuelve el valor TRUE en caso de éxito, por contrario retorna FALSE.

Logout de agentes

Permite desloguear un agente de callcenter de todas las colas en las cuales se encuentre vinculado.

`logoutAgent(authorizedKey, agent)`

Parámetros:

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
agent	Usuario correspondiente al agente a ser deslogueado
extension	Interno al cual se intenta loguear el agente

Retorna el valor TRUE en caso de éxito, por contrario retorna FALSE.

Pausar agente

Permite pausar un agente de callcenter en las colas en las cuales se encuentre logueado.

```
pauseAgent(authorizedKey, agent, queues, pauseDTMF)
```

Parámetros:

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
agent	Usuario correspondiente al agente a ser pausado
queues	Lista de colas en las cuales se debe pausar al agente, en caso de no especificarlas se pausa en todas las colas que el agente se encuentre logueado
pauseDTMF	Indica el motivo de la pausa, en caso de no ser especificado se realiza una pausa sin motivo

Retorna el valor TRUE en caso de éxito, por contrario retorna FALSE.

Despausar agente

Permite despausar un agente de callcenter en las colas en las cuales se encuentre pausado.

```
unpauseAgent(authorizedKey, agent, queues)
```

Parámetros:

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
agent	Usuario correspondiente al agente a ser pausado
queues	Lista de colas en las cuales se debe despausar al agente, en caso de no especificarlas se despausa de todas las colas que el agente se encuentre pausado

Retorna el valor TRUE en caso de éxito, por contrario retorna FALSE.

Obtener información de agente

Retorna información de un agente dado cuando el mismo se encuentra logueado.

```
getAgentLoggedInfo(authorizedkey, username)
```

Retorna en caso de encontrarse logueado un XML, de lo contrario retorna 0.

Ejemplo

XML devuelto si el agente se encuentra logueado:

```
<?xml version="1.0" encoding="UTF-8" ?>
<agentAgent>
  <user nombre="Administrador" email="admin@expand.com.uy" logueado="1"
  agente="admin" nroAgente="12345" extension="803" />
</agentAgent>
```

Códigos de error devueltos:

-1	authorizedKey incorrecta
-2	No se especificó un nombre de agente
-3	Error desconocido

Obtener nombre de usuario

A partir del id de un agente permite obtener el nombre del usuario del agente.

```
getUserByAgent(authorizedKey, agent)
```

Parámetros:

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
agent	Número de agente

Retorna el nombre de usuario asociado al id de agente, de no existir retorna un string vacío.
En caso de error retorna los siguientes códigos:

-1	authorizedKey incorrecta
-2	parámetros incorrectos
-3	Error desconocido

Obtener información del usuario

A partir del username se obtiene información del usuario, nombre, email, perfil, supervisor, agente y nro de agente.

```
getUserInfo(authorizedKey, username)
```

Parámetros:

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
username	Nombre del usuario

Retorna un xml con la información de usuario, en caso de no existir el username retorna un string vacío.

Ejemplo

```
<?xml version="1.0" encoding="UTF-8" ?>
  <userInfo>
    <user nombre="Natalia Garcia" email="ngarcia@garcia.com.uy"
    perfil="operador_tecnico" supervisor="0" agente="1" nroAgente="118" />
  </userInfo>
```

En caso de error retorna los siguientes códigos:

-1	authorizedKey incorrecta
-2	parámetros incorrectos
-3	Error desconocido

Obtener tiempos de logueo y pausa

Permite obtener los tiempos de logueo y pausa de los agentes en un período de tiempo determinado, pudiendo filtrar por cola y agente.

```
getAgentTimes(authorizedKey, inicio, fin, agente, cola)
```

Parámetros:

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
inicio	Fecha a partir de la que se desea obtener los datos en formato YYYY-MM-DD HH:MM:SS
fin	Fecha hasta la cual se desea obtener los datos en formato YYYY-MM-DD HH:MM:SS
agente	el nombre de usuario de un agente determinado, o vacío si se desea obtener los datos de todos los agentes
cola	el nombre de la cola por la que filtrar, o vacío si se desea obtener los datos de todas las colas

Retorna un xml con los tiempos de login y pausa de los agentes en las colas, y la información de las colas de cada uno.

Ejemplo

```
<?xml version="1.0" encoding="UTF-8" ?>
<agents>
  <agent agent="123" user="fmega" name="Franco Mega" logged="593714"
paused="56686" >
    <queueagent name="q_tcc" description="tccadsl" />
    <queueagent name="q_tec" description="Callcenter Tecnico" />
    <queueagent name="q_tra" description="CallCenter C" />
  </agent>
  <agent agent="126" user="pgandini" name="Paula Gandini"
logged="522215" paused="61343" >
    <queueagent name="q_fmn" description="Fabian Maitzman" />
    <queueagent name="q_tcc" description="tccadsl" />
    <queueagent name="q_tec" description="Callcenter Tecnico" />
    <queueagent name="q_tra" description="CallCenter C" />
  </agent>
```

```
<agent agent="130" user="scarrion" name="Santiago Carrion"
logged="542030" paused="50058" >
  <queueagent name="q_fmn" description="Fabian Maitzman" />
  <queueagent name="q_tcc" description="tccadsl" />
  <queueagent name="q_tec" description="Callcenter Tecnico" />
  <queueagent name="q_tra" description="CallCenter C" />
</agent>
</agents>
```

En caso de error retorna los siguientes códigos:

-1	authorizedKey incorrecta
-2	parámetros incorrectos
-3	Error desconocido
-4	Fecha de inicio incorrecta
-5	Fecha de fin incorrecta
-6	Cola no existente
-7	Agente no existente
-8	Rango de fechas no válido
-9	Rango de fechas posterior a la fecha actual
-10	No se encontraron registros

Obtener detalles de sesión de agentes

Permite crear un reporte de logins y pausas de agentes de Callcenter con desglose por tipo de pausa y duración de cada una.

```
agentSessionDetail(authorizedKey, inicio, fin, [userID], [cola])
```

Parámetros:

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
---------------	--

inicio	Fecha a partir de la que se desea obtener los datos en formato YYYY-MM-DD HH:MM:SS
fin	Fecha hasta la cual se desea obtener los datos en formato YYYY-MM-DD HH:MM:SS
userID	ID de usuario sobre el que se desea reportar, o vacío si se desea obtener los datos de todos los usuarios
cola	el nombre de la cola por la que filtrar, o vacío si se desea obtener los datos para todas las colas

Devuelve un reporte codificado en XML con datos sobre las sesiones Login-Logout completas dentro del periodo indicado, para el agente indicado. En caso de no proporcionar agentID, se retorna una colección de reportes con información de todos los agentes con actividad de login/logouts dentro del período.

Ejemplo

```
<?xml version="1.0" encoding="UTF-8"?>
<agents>
  <agent userID="ngarcia" agentID="711">
    <queue queueName="q_bps" logged="43200" paused="4320">
      <pauses>
        <pause type="smoking" duration="600" />
        <pause type="lunch" duration="3600" />
        <pause type="bathroom" duration="120" />
      </pauses>
      <sessions>
        <session start="2014-01-01 00:00:00" end="2014-01-01
12:00:00" duration="43200">
          <pause type="bathroom" start="2014-01-01 02:00:00"
end="2014-01-01 02:02:00" duration="100"/>
          <pause type="bathroom" start="2014-01-01 02:05:00"
end="2014-01-01 02:05:20" duration="20"/>
          <pause type="smoking" start="2014-01-01 02:10:00"
end="2014-01-01 02:20:00" duration="600" />
          <pause type="lunch" start="2014-01-01 09:00:00"
end="2014-01-01 10:00:00" duration="3600" />
        </session>
      </sessions>
    </queue>
  </agent>
</agents>
```

```

<queue queuename="q_1199" logged="4200" paused="240">
  <pauses>
    <pause type="" time="240" />
  </pauses>
  <sessions>
    <session start="2014-01-01 10:00:00" end="2014-01-02
11:00:00" time="3600">
      <pause type="" start="2014-01-01 10:30:00"
end="2014-01-01 10:34:00" time="240" />
    </session>
    <session start="2014-01-01 11:30:00" end="2014-01-02
11:40:00" time="600">
    </session>
  </sessions>
</queue>
</agent>
</agents>

```

En caso de error retorna los siguientes códigos:

-1	authorizedKey incorrecta
-2	Error desconocido
-3	Fecha de inicio incorrecta
-4	Fecha de fin incorrecta
-5	Cola no existente
-6	Usuario no existente

Obtener estado de las extensiones

Permite obtener el estado de las extensiones, e información sobre los agentes y colas involucradas.

`getAgentLogin(authorizedKey, extensions)`

Parámetros:

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
---------------	--

extensions	Conjunto de extensiones separadas por coma (,)
------------	--

Devuelve un xml con las extensiones, los agentes logueados, estado de pausa y las colas involucradas.

Ejemplo

```
<extensions>
  <extension number="101" channelType="SIP" channelId="101" >
    <agent number="711" user="nagarcia" name="Nazaret Garcia" />
    <queue name="q_tra" description="CallCenter C"
is_logged="true" logged_date="2014-09-11 15:29:27" is_paused="false" pause_type="" />
  </extension>
  <extension number="151" channelType="SIP" channelId="151" >
    <agent number="705" user="darioj" name="Dario"
is_logged="true" logged_date="2014-09-11 15:29:27" is_paused="false" pause_type="" />
    <queue name="q_tra" description="CallCenter C"
is_logged="true" logged_date="2014-09-11 15:29:27" is_paused="true"
pause_type="Almuerzo" />
    <queue name="q_fmn" description="Fabian Maitzman"
is_logged="true" logged_date="2014-09-11 15:29:27" is_paused="false" pause_type="" />
    <queue name="q_ven" description="Callcenter Ventas"
is_logged="true" logged_date="2014-09-11 15:29:27" is_paused="false" pause_type="" />
    <queue name="q_age" description="Agesic" is_logged="true"
logged_date="2014-09-11 15:29:27" is_paused="true" pause_type="Descanso"/>
    <queue name="q_tec" description="Callcenter Tecnico"
is_logged="true" logged_date="2014-09-11 15:29:27" is_paused="true"
pause_type="Descanso" />
    <queue name="q_emp" description="Empanadas Rodeo"
is_logged="true" logged_date="2014-09-11 15:29:27" is_paused="false" pause_type="" />
  </extension>
</extensions>
```

En caso de error retorna los siguientes códigos:

-1	authorizedKey incorrecta
-2	parámetros incorrectos
-3	Error desconocido
-4	No se proporcionaron extensiones para filtrar

-5	Alguna de las extensiones no es correcta
----	--

Gestión de mensajes SMS

Estos Webservices están disponibles cuando su eXpand tiene habilitado y funcionando el módulo de envío y gestión de mensajes SMS

Envío de SMS

Permite el envío de un SMS desde aplicaciones externas a la central a uno o múltiples destinatarios.

`sendSms(authorizedKey, msgId, msgData, msgRecip)`

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
msgId	Identificador del mensaje. En el uso desde sistemas externos el valor de msgId debe ser 0
msgData	Texto a ser enviado en el mensaje. El mismo no debe superar los 160 caracteres. Notar que el soporte de caracteres extendidos muchas veces no es soportado por los carriers.
msgRecip	Lista de números destinatario del SMS. Puede ser un único elemento o en caso de ser más de uno, deben ir separados por espacio o coma.

Devuelve:

-1	La authorizedKey no es correcta
-2	Se excede la cantidad de mensajes enviados por segundo. (Dicho parámetro es configurable desde la administración de eXpand)
0	No fue posible enviar el mensaje
IdSMS	En caso de Éxito devuelve un ID numérico mayor a 0, representando el trabajo para su posterior seguimiento.

Estado de SMS enviado

Permite realizar el seguimiento sobre el estado de un sms enviado por medio de *sendSms*

`statusSms(authorizedKey, idSMS)`

authorizedKey	Clave de acceso de los Webservices usada por esta aplicación
idSMS	Identificador del trabajo de envío sobre el que se desea consultar el estado

Devuelve un string con el estado del trabajo:

String vacío	en caso de no encontrar un status para el idSMS
ERROR	Ocurrió un error al intentar el envío. Trabajo cancelado
SENT	Mensaje enviado y procesado. Trabajo completado. Depende del carrier si el mismo es o no entregado al destinatario.
PENDING	Pendiente de envío
DRAFT	SMS almacenado como borrador

API REST

La API REST de eXpand Web permite interactuar con los módulos de eXpand desarrollados a partir de la versión 3.0 de eXpand Web.

Autenticación

El acceso a los métodos está controlado por autenticación tipo HTTP Basic Auth usando usuario/contraseña específico para el tipo de acceso requerido.

Datos

Todos los datos intercambiados son de tipo JSON, salvo excepciones específicamente indicados.

Códigos de respuesta

La API responde con códigos de respuesta HTTP de acuerdo al resultado general de la solicitud:

2XX - Solicitud exitosa

4XX - Error en la solicitud, error del cliente

5XX - Error en el servidor que no logra procesar la solicitud

Status Code	Description
200	OK
201	Created
202	Accepted (Request accepted, and queued for execution)
400	Bad request
401	Authentication failure
403	Forbidden
404	Resource not found
405	Method Not Allowed
409	Conflict

412	Precondition Failed
413	Request Entity Too Large
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

Módulo: Expand Dialer

URL Base del módulo:

<http://{{expand-hostname}}/expandweb3/api/campaign/external/> + metodo

Método y endpoint	
POST /campaigns/<id campaña> /contacts	Alta contacto
GET /campaigns/<id campaña>/contacts/<id contacto>	Obtiene contacto
PUT /campaigns/<id campaña>/contacts/<id contacto>	Edita contacto
GET /campaigns	Lista campañas
GET /campaigns/<id campaña>	Datos una campaña
GET /campaigns/<id campaña>/contacts	Contactos de una campaña
GET /campaigns/<id campaña>/contacts/activity	Actividad de todos los contactos de una campaña

Agregar contacto

```
POST /campaigns/<id campaña>/contacts
```

id campaña es un número entero, usado en el HEADER, que identifica la campaña a la que se le agrega el contacto.

En el cuerpo debe incluirse un diccionario JSON con los datos del contacto:

- ☒ **name** y **numcall** que son los datos del contacto a dar de alta . numcall normalmente se da de alta de forma canónica, sin el 9 usado para tomar línea
- ☒ **enable** debe ser 'true' para indicar que el contacto debe ser llamado
- ☒ **info** es un string conteniendo un JSON con datos adicionales del contacto a ser utilizados por Dialer. Es opcional

Respuestas

Status	Response
201	Contacto agregado. Retorna un JSON con el contacto completo, incluyendo un id que lo identifica para futuras operaciones
400	Error en los datos de entrada { "errorCode" :"Unknown_data_format"}

Ejemplo

Dialer: Agregar un contacto

EXAMPLE REQUEST

POST

http://{{expand-hostname}}/expandweb3/api/campaign/external/campaigns/313/contacts

Params

Headers (2)

Body

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

```

1 {
2   "name": "Rodrigo",
3   "numcall": "27111200",
4   "enable": true,
5   "info": {
6     "ci": "3123456-0",
7     "correo": "rangles@expand.com.uy"
8   }
9 }
10

```

EXAMPLE RESPONSE

Body

Headers (5)

Status 201 CREATED

Pretty

Raw

Preview

JSON

```

1 {
2   "attempttimestamp": "",
3   "enable": true,
4   "id": 1698,
5   "idcall": 1698,
6   "info": {
7     "ci": "3123456-0",
8     "correo": "rangles@expand.com.uy"
9   },
10  "lastattempt": "",
11  "name": "Rodrigo",
12  "numcall": "27111200",
13  "state": "Pending",
14  "statelog": []
15 }

```

Obtener un contacto

GET /campaigns/<id campaña>/contacts/<id contacto>

id campaña es un número entero, usado en el HEADER, que identifica la campaña a la que

pertenece el contacto.

Id contacto es un número entero que identifica al contacto que fue obtenido cuando se agregó

Respuestas

Status	Response
200	Contacto encontrado En el cuerpo de la respuesta se incluye un JSON con los datos del contacto, estado e histórico de intentos de conectar si correspondiera
400	Error en los datos de entrada <pre>{“errorCode”:”Unknown_data_format”}</pre>
404	Contacto no encontrado

Ejemplo

Dialer: Obtener un contacto

EXAMPLE REQUEST

GET

http://{{expand-hostname}}/expandweb3/api/campaign/external/campaigns/313/contacts/1698

Params

Headers (2)

Body

	Key	Value	Description	...	Bulk Edit	Presets ▼
	Authorization	Basic Y29udGFjdGNsaWVudDpGd2N6T1VyQk9t				
☒	Content-Type	application/json				
	New key	Value	Description			

EXAMPLE RESPONSE

Body

Headers (5)

Status 200 OK

Pretty

Raw

Preview

JSON

```

1 {
2   "attempttimestamp": "",
3   "enable": true,
4   "id": 1698,
5   "idcall": 1698,
6   "info": {
7     "ci": "3123456-0",
8     "correo": "rangles@expand.com.uy"
9   },
10  "lastattempt": "",
11  "name": "Rodrigo",
12  "numcall": "27111200",
13  "state": "Pending",
14  "statelog": [
15    {
16      "timestamp": "2017-11-10 18:50:13",
17      "transition": "_Source_API_Api"
18    }
19  ]
20 }
```

Editar contacto

```
PUT /campaigns/<id campaña>/contacts/<id contacto>
```

id campaña es un número entero, usado en el HEADER, que identifica la campaña a la que se le edita el contacto.

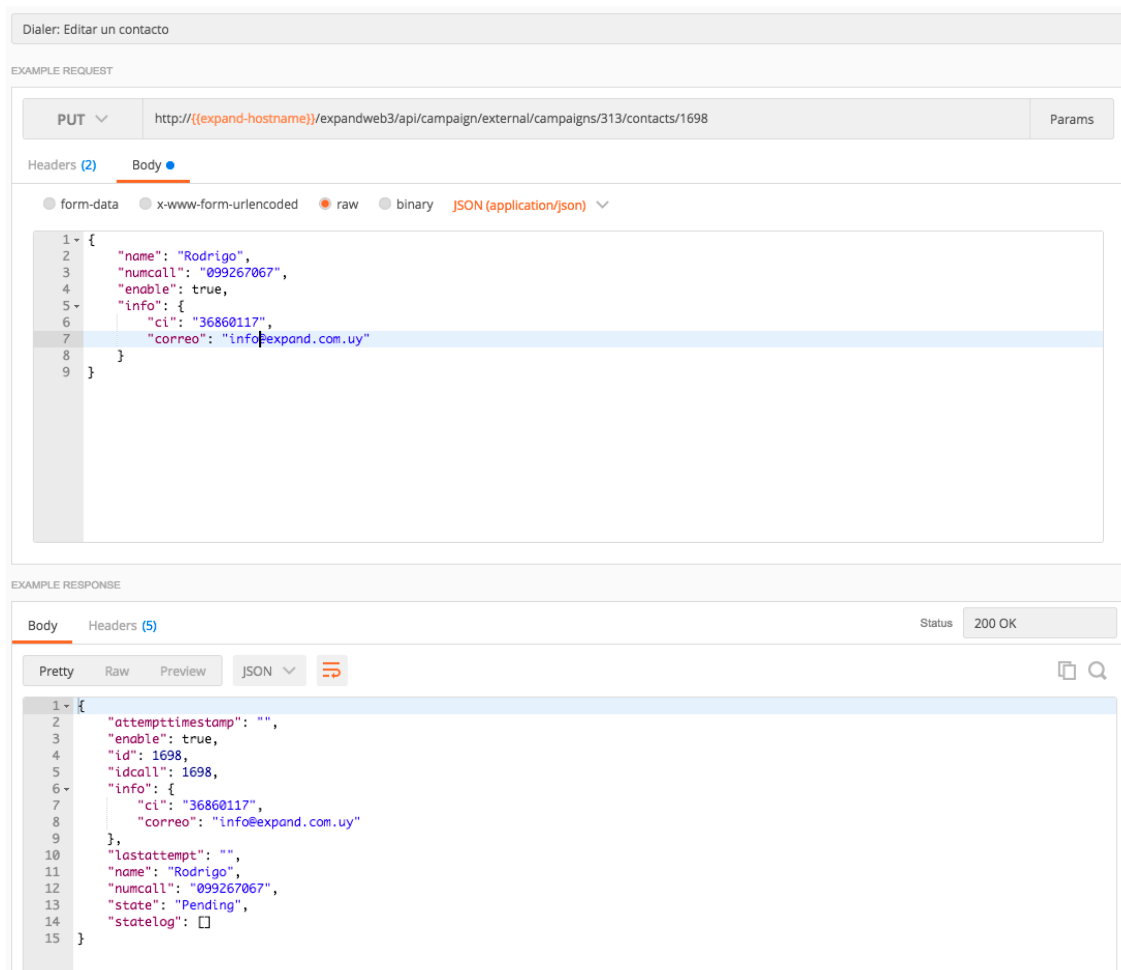
Id contacto es un número entero que identifica al contacto que fue obtenido cuando se agregó

En el cuerpo debe incluirse un diccionario JSON con los datos nuevos del contacto:

- **name** y **numcall** que son los datos del contacto a dar de alta. Numcall normalmente se da de alta de forma canónica, sin el 9 usado para tomar línea
- **enable** debe ser 'true' para indicar que el contacto debe ser llamado
- **info** es un string conteniendo un JSON con datos adicionales del contacto a ser utilizados por Dialer. Es opcional

Respuestas

Status	Response
200	Contacto modificado. Retorna un JSON con el contacto completo
400	Error en los datos de entrada { "errorCode" : "Unknown_data_format" }
404	Contacto a modificar no encontrado



Obtener todas las campañas

GET `/campaigns`

Retorna un listado con todas las campañas sobre las que se puede trabajar en un JSON

El método permite filtros los resultados mediante el uso de operadores en el HEADER:

`State = Incomplete, Active, Finalized, Stored`

Permite filtrar por estado de las campaña

`order=asc, desc`

Define el orden de las campañas ascendientemente (**asc**) o descendientemente (**desc**)

order_by=id, name

Permite definir la columna por la que ordena:

- id de la campaña (orden natural) (**id**)
- nombre del contacto (**name**)

Obtener datos sobre una campaña

```
GET /campaigns/<id campaña>
```

id campaña es un número entero, usado en el HEADER, que identifica la campaña.

Retorna un JSON con Nombre, tipo de agendamiento, fechas de operación, reglas de rediscado, etc.

Obtener todos los contactos de una campaña

```
GET /campaigns/<id campaña>/contacts
```

id campaña es un número entero, usado en el HEADER, que identifica la campaña.

Respuestas

Status	Response
200 OK	Contactos recuperados Se retorna un JSON conteniendo un diccionario con • recordsTotal: número que indica cantidad de contactos existentes • data: lista de contactos
404	Campaña no encontrada

El método permite filtrar los resultados mediante el uso de operadores en el HEADER:

'start', 'limit': permite filtrar los contactos entre el registro start y limit registros

'order', 'order_by': permite definir el orden en que se obtienen. order puede ser **asc** (Ascendente) o **desc** (Descendente). order_by puede ser **numcall** (ordena por el número telefónico del contacto) o **name** (ordena por el nombre del contacto)

'like': obtiene solo aquellos contactos que tienen en su número o en el nombre el valor indicado. Ej: like=Rodrigo obtiene los contactos que contienen el texto "Rodrigo" en el nombre

'state': Permite obtener solo contactos con el estado **Pending**, **Completed**, **Rejected** o **Canceled**

Ejemplos:

NAME

Dialer: Obtener contactos

EXAMPLE REQUEST

GET ▼ http://{{expand-hostname}}/expandweb3/api/campaign/external/campaigns/313/contacts Params

Headers (2) Body

Key	Value	Description	...	Bulk Edit	Presets ▼
Authorization	Basic Y29udGFjdGNsaWVudDpGd2N6T1VyQk9t				
☒ Content-Type	application/json				
New key	Value	Description			

EXAMPLE RESPONSE

Body Headers (7) Status 200 OK

Pretty Raw Preview JSON ▼

```

1 {
2   "recordsTotal": 10,
3   "data": [
4     {
5       "attempttimestamp": "2017-07-12 15:43:40",
6       "enable": true,
7       "id": 1251,
8       "idcall": 1251,
9       "info": {
10        "ci": "3123456-0",
11        "correo": "rangles@expand.com.uy",
12        "nombre": "Rodrigo Angles",
13      },
14      "lastattempt": "Answer",
15      "name": "Rodrigo",
16      "numcall": "27111200",
17      "state": "Completed",
18      "statelog": [
19        {
20          "timestamp": "2017-07-12 15:43:40",
21          "transition": "Answer"
22        }
23      ],
24      "statetimestamp": "2017-07-12 15:44:18"
25    },
26    {
27      "attempttimestamp": "2017-07-12 15:45:43",
28      "enable": true,
29      "id": 1343,
30      "idcall": 1343,
31      "info": {},
32      "lastattempt": "Answer"
33    }
34  ]
35 }

```

Obtener actividad de la ejecución de una campaña

Retorna información similar a la que se puede obtener mediante el reporte “Exportar Actividad” en el *dashboard* de la campaña.

```
GET / external/campaigns/<id campaña>/contacts/activity
```

`id_campaña` es un número entero que identifica la campaña de la que se quiere extraer la actividad generada

El método permite filtrar los resultados mediante el uso de operadores en el HEADER:

`State = Pending, Completed, Rejected, Canceled`

Permite filtrar por estado del contacto, retornando únicamente los pendientes (Pending), completados (Completed), rechazados que cumplieron todos los reintentos programados (Rejected) cancelados (Canceled)

`sstimestamp, estimestamp`

Filtra contactos por el último cambio de estado, pudiendo establecer un inicio (`sstimestamp`) y fin (`estimestamp`)

`saddtimestamp, eaddtimestamp`

Filtra contactos por la fecha en que fue agregado a la campaña, pudiendo establecer un inicio (`saddtimestamp`) y fin (`eaddtimestamp`)

Una vez aplicado un filtro puede además limitarse a un subconjunto de registros:

`start=S, limit=L`

Obtiene solo `L` contactos a partir del `S`-avo resultado, usado típicamente en paginación

Y definir el orden en que se desean obtener:

`order=asc, desc`

Define el orden de los contactos obtenidos ascendientemente (`asc`) o descendientemente (`desc`)

`order_by=name, numcall, timestampadd, statetimestamp`

Permite definir la columna por la que ordena:

- nombre del contacto (`name`)
- número del contacto (`numcall`)
- fecha en que fue agregado el contacto (`timestampadd`)
- fecha del último cambio de estado (`statetimestamp`)

Respuestas

Status	Response
200 OK	Contactos recuperados Se retorna un JSON conteniendo un diccionario con - recordsTotal: número que indica cantidad de contactos existentes - data: lista de contactos
404	Campaña no encontrada

Ejemplos:

`http://expand/expandweb3/api/campaign/
external/campaigns/427/contacts/activity?order_by=name&order_asc&state=Pending`

Obtiene contactos pendientes de la campaña 427, y los ordena ascendentemente por Nombre

`http://expand/expandweb3/api/campaign/
external/campaigns/427/contacts/activity?saddtimestamp='2019-02-01'&eaddtimestamp='2019-02-28'`

Obtiene contactos de la campaña 427 que hayan sido agregados entre el 01 de Febrero de 2019 y el 28 de Febrero de 2019

Listas

Las listas de eXpand permiten almacenar datos en forma de clave-valor. Se admiten claves simples o de tipo rangos. Las de tipo simple permiten por ejemplo asociar un número con un nombre. Las de tipo rango permiten por ejemplo asociar un conjunto de números a un mismo valor.

API: Listas Simples

<code>/expandweb3/api/external/lists/simple</code>	
Método	Acción
GET	Obtiene datos de todas las listas

Respuestas

200	Ok
4xx	Error

<code>/expandweb3/api/external/lists/simple/<nombre lista></code>	
Método	Acción
GET	Obtiene datos de la lista identificada por <nombre lista>

Respuestas

200	Ok
4xx	Error

API: Elementos de Listas Simples

/expandweb3/api/external/lists/simple/<id lista>/elements	
Método	Acción
GET	Obtiene todos los elementos de la lista identificada por <id lista>
POST	Crea (agrega) un elemento a la lista identificada por <id lista> El BODY debe incluir un JSON con el elemento: <pre>{ "key": "5679", "value": "Un elemento" }</pre>

Respuestas

201	Elemento agregado correctamente. Se incluye JSON del elemento.
4xx	Error genérico
409	Error al agregar el elemento, ya existe un elemento con esa clave

<code>/expandweb3/api/external/lists/simple/<lista>/elements/<key></code>	
Método	Acción
GET	Obtiene elemento con clave <code><key></code> de la lista identificada por <code><lista></code>
PUT	Actualiza el elemento <code><key></code> indicado de la lista <code><lista></code>. El BODY debe incluir un JSON con el nuevo valor a asociar al elemento: <pre>{ "value": "Valor nuevo del elemento" }</pre>
DEL	Elimina el elemento con clave <code><claveelem></code> de la lista <code><lista></code>

Respuestas

200	Ok. Se incluye el JSON del elemento indicado.
4xx	Error genérico
409	Error al agregar el elemento, ya existe un elemento con esa clave

Ejemplos de uso

Obtener Listas

GET /expandweb3/api/external/lists/simple HTTP/1.1

Host: cliente.expand

Authorization: Basic ZXh0ZXJuYXZlcnRlcjoxZGVEOTljZjk=

Content-Type: application/json

```

1 {
2   "data": [
3     {
4       "alias": "Vendedores",
5       "description": "Asocia CallerID de llamantes con extensiones de ejecutivos de venta. Mantenido por SAP.",
6       "id": 1,
7       "name": "vendedores",
8       "type": "simple"
9     },
10    {
11      "alias": "Cartera",
12      "description": "Asocia CallerID de llamantes con la extension del ejecutivo de cartera que debe atenderlo. Mantenido por SAP",
13      "id": 2,
14      "name": "cartera",
15      "type": "simple"
16    }
17  ]
18 }
```

Obtener elementos de una lista

GET /expandweb3/api/external/lists/simple/cartera/elements HTTP/1.1

Host: cliente.expand

Authorization: Basic ZXh0ZXJuYXZlcnRlcjoxZGVEOTljZjk=

Content-Type: application/json

```

1 {
2   "data": [
3     {
4       "id": 9,
5       "key": "6438331",
6       "value": "200"
7     }
8   ]
9 }
```

Agregar un elemento a una lista

POST /expandweb3/api/external/lists/simple/cartera/elements HTTP/1.1

Host: cliente.expand

Authorization: Basic ZXh0ZXJuYXZlcnRlcjoxZGVEOTljZjk=

Content-Type: application/json

```

{
  "key": "27126666",
  "value": "112"
}
```

```
1 {  
2   "id": 17,  
3   "key": "27126666",  
4   "value": "112"  
5 }
```

Actualizar un elemento de una lista

PUT /expandweb3/api/external/lists/simple/cartera/elements/27126666 HTTP/1.1

Host: cliente.expand

Authorization: Basic ZXh0ZXJuYWxfZdXNlcjoxZGVEOTljZjk=

Content-Type: application/json

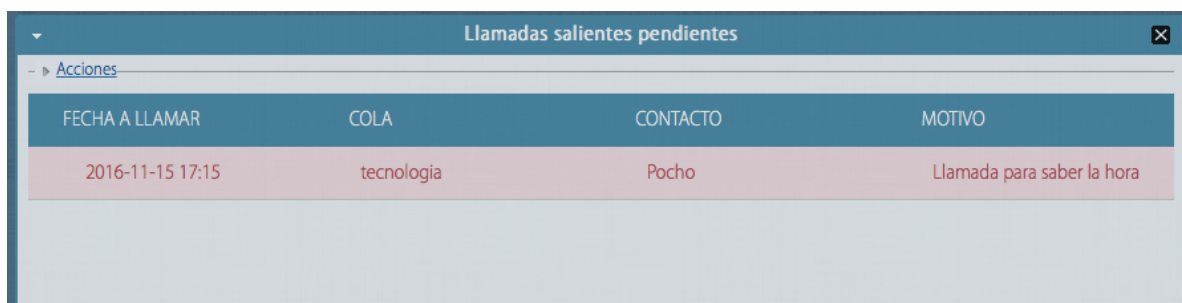
```
{  
  "value": "valor nuevo"  
}
```

```
1 {  
2   "id": 17,  
3   "key": "27126666",  
4   "value": "valor nuevo"  
5 }
```

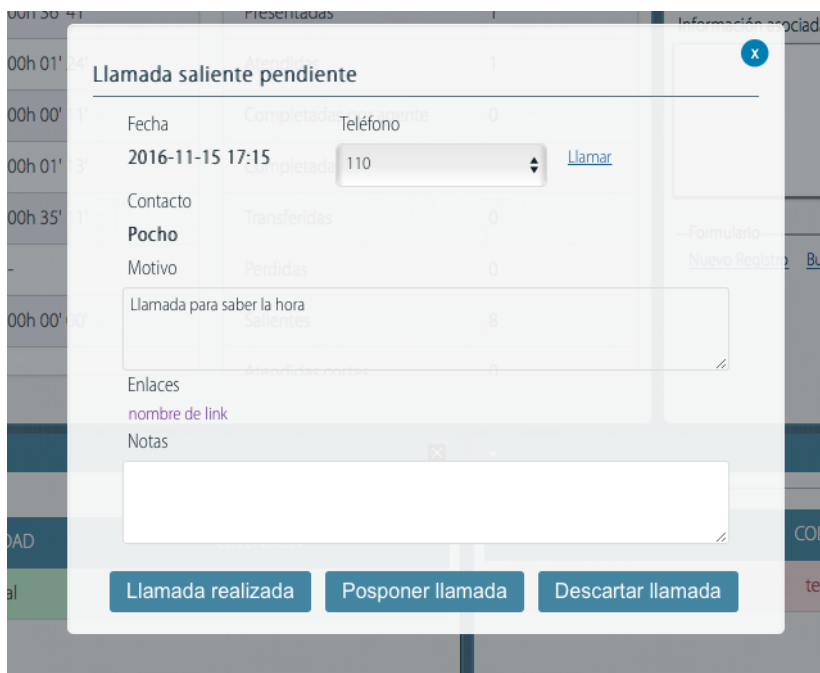
Panel de agente: llamadas salientes pendientes

Los agentes de un callcenter cuentan con un panel customizable, en el cual entre otros módulos de información, pueden incluir un panel de **llamadas salientes pendientes**. Este panel muestra llamadas pendientes que el callcenter debe realizar.

Cada llamada está asociada a un callcenter y a una cola específica. En su funcionamiento más simple, todos los agentes de un callcenter ven y pueden realizar esa llamada pendiente.



FECHA A LLAMAR	COLA	CONTACTO	MOTIVO
2016-11-15 17:15	tecnologia	Pocho	Llamada para saber la hora



Llamada saliente pendiente

Fecha: 2016-11-15 17:15

Teléfono: 110 [Llamar](#)

Contacto: Pocho

Motivo: Llamada para saber la hora

Enlaces: nombre de link

Notas:

[Llamada realizada](#) [Posponer llamada](#) [Descartar llamada](#)

Las llamadas cuentan a su vez con atributos que la describen y modifican el comportamiento de la aplicación:

Atributo	Descripción
----------	-------------

Contact	Nombre de la persona o el contacto al que debe llamarse
Subject	Describe el motivo de la llamada
Phone	Uno o una lista de teléfonos a los que debe discarse para completar la llamada
Agent	Asigna la llamada a un determinado agente de ese callcenter. Solo ese agente verá y puede realizar la llamada.
Calldate	Fecha y hora en la que debe realizarse la llamada. Al aproximarse la hora se destaca la llamada en el panel mediante el uso de colores. Pasada esta fecha la misma se señala en rojo como pendiente pasado.
Expiration	Cantidad de segundos a partir de la hora establecida. Durante ese tiempo se mantiene el aviso (variación de colores) para el agente. La llamada puede realizarse igualmente luego de pasado dicho plazo. En caso de no proveerse, la llamada no caduca.
Link	Enlace que aparece junto con el contacto y permite por ejemplo vincular este contacto con el CRM

Un agente puede realizar la llamada clickeando sobre el link, y en función de cómo transcurra la llamada puede marcar la misma como:

- Descartar, la llamada es eliminada de los pendientes
- Asignar, la llamada puede asignarse a otra cola o agente
- Posponer, estableciendo una nueva fecha y hora en la que debe ser realizada
- Completar, queda realizada y eliminada de los pendientes

API: Manipulación de llamadas salientes pendientes

```
/expandweb3/api/external/outgoingcalls/call
```

Método	Acción
GET	Obtiene datos de todas las llamadas pendientes Pueden aplicarse filtros incluyendo en el el Request valores por los cuales filtrar, como ser: • queue • phone • startdate, enddate (fechas en ISO8601) • agent • status (PENDING, COMPLETED, CANCELED, EXPIRED)
POST	Crea una nueva llamada pendiente. Debe proveerse en formato JSON al menos los siguientes atributos: • queue • phone • contact • subject • calldate • agent

Respuestas

201	Ok
4xx	Error: Cola invalida, Agente invalido, Fecha invalida

/expandweb3/api/external/outgoingcalls/call/<callid>	
Método	Acción
PUT	Actualiza o modifica una llamada pendiente. Deben proveerse en el cuerpo del request todos los campos.
DEL	Cancela una llamada pendiente. La misma es borrada lógicamente y no es presentada a los agentes.

Ejemplos de uso

Agregar una llamada

POST <http://ip/expandweb3/api/external/outgoingcalls/call>

```
{
  "queue": "tecnologia",
  "phone": ["21232322", "0999999999"],
  "contact": "Pedro",
  "subject": "Confirmacion contrato ADSL",
  "calldate" : "2016-11-10 15:00",
  "expiration" : 0,
  "agent" : "pperez",
  "link": {
    "nombre de link": "http://www.link.com"
  }
}
```

Response:
HTTP/1.1 201 OK
Content-Type: application/json

```
{
  "callid": 1
}
```

Obtener una llamadas pendiente dado su ID

GET <http://ip/expandweb3/api/external/outgoingcalls/call/1>

Response:
HTTP/1.1 200 OK
Content-Type: application/json

```
{
  "id": 1,
  "queue": "tecnologia",
  "phone": ["21232322", "0999999999"],
  "contact": "Pedro",
  "subject": "Confirmacion contrato ADSL",
  "calldate" : "2016-11-10 15:00",
  "expiration" : 0,
  "agent" : "pperez",
}
```



```
"status": "PENDING",
"link": {
  "nombre de link": "http://www.link.com"
}
}
```

Modificar datos de una llamada en base a su ID (1)

PUT http://<ip>/expandweb3/api/external/outgoingcalls/call/1

```
{
  "queue": "tecnologia",
  "phone": ["21232322", "099999999"],
  "contact": "Pedro",
  "subject": "Confirmacion contrato ADSL",
  "calldate": "2016-11-10 15:00",
  "expiration": 86400,
  "agent": "pperez",
  "link": {
    "nombre de link": "http://www.link.com"
  }
}
```

Response:
HTTP/1.1 200 OK
Content-Type: application/json

```
{
  "callid": 1
}
```

Eliminar una llamada pendiente dado su ID (1)

DEL http://<ip>/expandweb3/api/external/outgoingcalls/call/1

Response:
HTTP/1.1 200 OK
Content-Type: application/json
"Ok"

Obtener todas las llamadas pendientes

GET http://<ip>/expandweb3/api/external/outgoingcalls/call/

Response:
HTTP/1.1 200 OK
Content-Type: application/json

```
{
  "data" : [
    {
      "id": 1,
      "queue": "tecnologia",
      "phone": ["21232322","0991271878"],
      "contact": "Diego",
      "subject": "Confirmacion contrato ADSL",
      "calldate" : "2016-11-10 15:00",
      "expiration" : 86400,
      "agent" : "pperez",
      "status": "PENDING",
      "link": {
        "nombre de link": "http://www.link.com"
      }
    },
    {
      "id": 2,
      "queue": "tecnologia",
      "phone": ["3344343","0991878"],
      "contact": "Pepe",
      "subject": "Problema ADSL",
      "calldate" : "2016-11-10 15:00",
      "expiration" : 0,
      "agent" : "daguirre",
      "status": "COMPLETED",
      "link": {
        "nombre de link": "http://www.link.com"
      }
    }
  ]
}
```

Formularios para Agentes

Este módulo permite definir Formularios para captura de datos, asociarlos a colas de callcenter y permitir que los agentes los invoquen a los efectos de capturar datos relacionados con la llamada en curso. Los formularios también permiten al agente realizar la grabación de una llamada mientras captura los datos, a los efectos de contar con un respaldo en un audio de los datos capturados (típicamente utilizado para la firma verbal de contratos de servicio)

Los formularios se definen en el área de Administración/Ajustes. Allí se establece un nombre, los campos del formulario y las colas de llamadas a las que estará asociado.

Una vez configurado, los agentes invocan el formulario desde el portal del agente cuando tiene una llamada en curso, donde puede visualizar el Mensaje que debe reproducir y una instancia del formulario para llenar los datos.

La extracción de datos resultantes de las captura realizadas por los agentes se puede realizar mediante los siguientes métodos de la API eXpand

API: Formularios para Agentes

/expandweb3/api/external/dynamicforms/<name_form>/entries	
Método	Acción
GET	Obtiene todos los registros del formulario especificado. Pueden aplicarse filtros incluyendo en la URI del Request: • start_date: filtrar por formularios ingresados luego de esta fecha. Fecha en formato ISO8601 • end_date: Filtrar por formularios ingresados antes de esta fecha. Fecha en formato ISO8601

Respuestas

201	Ok. Genera una lista de registros de formularios codificados en JSON. Cada formulario cuenta con un id.
-----	---

4xx	Error: Nombre de formulario invalido, Fecha invalida
------------	--

/expandweb3/api/external/dynamicforms/<name_form>/entries/<idcall>	
Método	Acción
GET	Obtiene el registro de formulario asociado al <i>IDCall</i> indicado

Respuestas

201	Ok. En el cuerpo se genera un JSON conteniendo el formulario solicitado
4xx	Error: Nombre de formulario invalido, ID inválido, Fecha invalida

Ejemplos de uso

Obtener todos los formularios de la cola 'ope'

GET http://ip/expandweb3/api/external/dynamicforms/<name_form>/entries

```
[
  {
    "_id": "59c14bc9d04d72233046dc3f",
    "agent": {
      "number": 303,
      "username": "cguerra"
    },
    "audio_filename": "audio_91662_1505840066.63",
    "call_chunk": "55a6bc9a43564848ae433fecf3ec0a9115058400249",
    "data": {
      "codigofeliz": [
        {
          "label": "codigofeliz",
          "value": "NG10WF917"
        }
      ]
    },
    "dynamic_form_id": 1,
    "extension": "SIP/303",
    "incomingcall_id": 91662,
    "last_modified_timestamp": "2017-09-19T13:54:33.824000-03:00",
    "queue": "ope",
    "session_id": "5c1df070-19eb-4e6b-8a27-96b28ec51cc8-150582807997",
    "timestamp": "2017-09-19T13:54:33.824000-03:00",
    "uniqueid": "1505840001.827"
  }
]
```

```
}
]
```

Obtener un registro

GET http://ip:/expandweb3/api/external/dynamicforms/<name_form>/entries/91662

```
{
  "_id": "59c14bc9d04d72233046dc3f",
  "agent": {
    "number": 303,
    "username": "cguerra"
  },
  "audio_filename": "audio_91662_1505840066.63",
  "call_chunk": "55a6bc9a43564848ae433fecf3ec0a9115058400249",
  "data": {
    "codigofeliz": [
      {
        "label": "codigofeliz",
        "value": "NG10WF917"
      }
    ]
  },
  "dynamic_form_id": 1,
  "extension": "SIP/303",
  "incomingcall_id": 91662,
  "last_modified_timestamp": "2017-09-19T13:54:33.824000-03:00",
  "queue": "ope",
  "session_id": "5c1df070-19eb-4e6b-8a27-96b28ec51cc8-150582807997",
  "timestamp": "2017-09-19T13:54:33.824000-03:00",
  "uniqueid": "1505840001.827"
}
```

Webservices externos

En entornos de callcenters es posible que Expand consulte a servicios externos a los efectos de enriquecer la información con la que cuenta Expand sobre una llamada, y aportarle esos datos a los registros de llamadas de Expand y a los agentes de un callcenters (típicamente mediante un popup)

En este caso se invierten los roles y es Expand quien consume Webservices provistos por el sistema de backend de su empresa.

Para poder hacer uso de esta funcionalidad es necesario que:

- El software de backend (CRM, ERP, sistemas de backend en general) implemente los webservices indicados a continuación. Esta programación debe hacerla el cliente en sus propios sistemas. Es posible usar REST o SOAP, según el contexto de los sistemas del cliente.
- Modificar la programación de dialplan (IVR) para hacer uso de estos webservices. Esta tarea la realiza el equipo técnico de Expand a pedido y de manera coordinada con su área técnica o de desarrollo

A continuación se describen los Webservices que típicamente son utilizados, con una interfaz sugerida:

Mapeo de Número de teléfono a nombre

Expand cuenta con el CallerID del llamante (número telefónico que origina la llamada entrante). Si el sistema de backend tiene cargados los números de sus clientes, es habitual usar esa información y presentarla a quien la recibe.

```
string getCallerIDName(callerIdNum: string)
```

Este webservice debe retornar un string conteniendo el CallerID Name (Nombre) que se desea asociar a una llamada. Es habitualmente invocado por cada llamada entrante a Expand, y permite que en los usos sucesivos, la llamada sea identificada con ese nombre. El mismo será visible en los reportes de monitoreo de eXpand, para el Agente de Callcenter y en el display de los teléfonos IP utilizados. Puede consistir en el nombre del cliente o cualquier otro identificador que resulte útil.

Captura de Datos de IVR

Durante el pasaje de una llamada por el IVR implementado, es posible recolectar datos que se van generando, como ser su número de cliente, razón por la que llama, etc. Esta información puede ser luego reportada a su sistema de backend al conectar con un agente o en otras instancias (ejemplo: para un registro de contacto).

Existen 2 formas de ir generando esta información:

- Mediante claves estáticas en el IVR: En la programación del IVR se identifican nodos con etiquetas denominados IVRCodes. Estos los define el cliente y son implementados por el equipo de Soporte de Expand durante la programación del IVR
- Mediante el uso de un Webservice externo

En el 2do caso, puede usarse un WS como el que sigue:

```
string IVRCode(UniqueID, callerIdNum, IVRCode, IVRData)
```

Reportara un ID de llamada, el callerIDNum de la misma, el nodo en que se encuentra durante el IVR (IVRCode) y el IVRData actual. El string devuelto es anexado al IVRData que se va construyendo.

Implementación de Popups en Callcenter

Es deseable que un agente que atiende una llamada de un cliente pueda contar con la información de ese cliente en pantalla en el momento de atenderlo. Esto permite optimizar los tiempos de atención y evita al agente tener que identificar la persona con la que va a hablar. Para ello, Expand puede informar al sistema de backend (CRM, ERP, etc.) en el momento en que una llamada conecta con un agente, y que de esa manera el CRM pueda desplegar esa información.

En caso de que el sistema de backend utilizado no disponga de esta funcionalidad, Expand incorpora un panel de agente que cumple el rol de popup básico para suplantarlos.

```
bool pushAgentInfo(uniqueid, queueName, callerid, agentchan, ivrdata)
```

Este webservice es invocado por Expand en el momento en que una llamada de una cola de Callcenter es asignada y contestada por un agente. Informa al sistema de backend sobre el agente que la recibe y los datos disponibles de esa llamada.

Los datos informados por el mismo son:

Uniqueld	String que identifica de manera unívoca la llamada dentro de eXpand. Ejemplo: "1275586803.1502"
queueName	String que contiene el nombre de la cola por la cual esa llamada fue asignada al agente (un agente puede participar simultáneamente de varias colas). Por ejemplo "q_cli"
CallerID	String de forma "Nombre <Número>" donde Nombre es el CallerIDName (ver webservice getCallerIDName) y Número es el identificador de número de origen (número del llamante) reportado por el operador de PSTN cuando la llamada ingresó a Expand.
agentChan	String de la forma "Tecnología/Identificador" indicando la extensión a la cual fue asignada la llamada, por ejemplo "SIP/107"
IVRData	String que contiene datos formateados de forma arbitraria durante el pasaje de la llamada del cliente por el IVR. Puede ser vacío, y típicamente se utiliza para indicar otros datos que fueron capturados antes de que la llamada sea derivada a la cola. Ejemplos posibles son las

	opciones del menú que digitó, número de cliente o servicio obtenido en el IVR o cualquier otro dato digitado por el cliente.
--	--

Nota: eXpand provee su propio sistema de popup para clientes que no cuentan con un sistema de backoffice. En caso de utilizar este webservice para integrarse con sistemas externos, el mismo queda en desuso en favor del sistema propio del cliente.