

<b>Document type:</b> Acuerdo de interfaz		<b>Title:</b> Acuerdo de Interfaz Servicio de Chat	
<b>Date:</b> 03/05/2016	<b>State: UNDER REVISION</b>	<b>Author:</b> Antonio M. García Ruiz	<b>Version:</b> 0.3

## ***Acuerdo de Interfaz Servicio de Chat***



<b>Version</b>	<b>Author</b>	<b>Date</b>
0.1	Antonio M. García Ruiz	13-05-2016
0.2	Antonio M. García Ruiz	24-05-2016
0.3	Ángel López-Ochoa Goicoechea	11-11-2016

<b>Document type:</b> Acuerdo de interfaz		<b>Title:</b> Acuerdo de Interfaz Servicio de Chat	
<b>Date:</b> 03/05/2016	<b>State: UNDER REVISION</b>	<b>Author:</b> Antonio M. García Ruiz	<b>Version:</b> 0.3

## 1. Introducción

A través de de este interfaz se podrá utilizar el servicio de chat incluido dentro de la plataforma Vivait-Suite. El interfaz de comunicación está basado en invocaciones a distintos webservices que deberán ser controlados desde la aplicación cliente. Consta de 4 webservices que serán definidos con precisión en el siguiente apartado.

## 2. Invocación y parámetros.

### 2.1 StartChatUsuario

Este webservice será utilizado para inicializar el canal de chat. Una vez inicializado el canal, se facilitarán los datos de la llamada y de la sesión.

#### 2.1.1 Invocación

Para poder invocar a dicho webservice, se deberá hacer una petición http a la siguiente url:

<http://xxx.xxx.xxx.xxx:xxxx/ChatWebService/startchatusuario>

siendo xxx.xxx.xxx.xxx:xxxx la ip y el puerto donde haya sido desplegado el webservice

Se deberán pasar los parámetros a través de get.

Un ejemplo de invocación sería la siguiente

<http://172.25.128.248:8180/ChatWebService/startchatusuario?service=chat1&name=minombre&origin=6177&destiny=60600&idUsuario=123>

Los distintos parámetros que podemos indicar en dicha invocación son los siguientes:

Nombre	Tipo	Obligatorio	Descripción
<a href="#"><u>service</u></a>	String	Si	Servicio de chat a utilizar
<a href="#"><u>name</u></a>	String	Si	Nombre del usuario que genera la petición de chat.
<a href="#"><u>origin</u></a>	String	Si	Identificador numérico del usuario que genera la petición de chat.(Número telefónico del cliente)
<a href="#"><u>destiny</u></a>	String	Si	Identificador de la cola de destino del chat
<a href="#"><u>idUsuario</u></a>	String	Si	Id del usuario

<b>Document type:</b> Acuerdo de interfaz		<b>Title:</b> Acuerdo de Interfaz Servicio de Chat	
<b>Date:</b> 03/05/2016	<b>State:</b> UNDER REVISION	<b>Author:</b> Antonio M. García Ruiz	<b>Version:</b> 0.3

### 2.1.2 Resultado y códigos de error

Si el webservice se ejecuta correctamente enviará el código de respuesta http **SC\_OK 200** acompañado de un JSON con los siguientes datos:

```
{ "result": "OK",
  "diag": "10002000091463068795",
  "session": "4D463E5289082EB500AC1D2D75221EDB",
  "comments": "Channel created" }
```

El campo **diag** contiene el ucid de la llamada, parámetro necesario en las llamadas a los distintos webservices.

Si el webservice no se ejecuta de forma correcta, puede devolver los siguientes códigos de error:

Error	Código	Descripción
<i>SC_NOT_FOUND</i>	404	
<i>SC_BAD_REQUEST</i>	400	Not enough parameters
<i>SC_CONFLICT</i>	409	Failed to create the channel
<i>SC_INTERNAL_SERVER_ERROR</i>	500	e.getMessage()
<i>SC_GATEWAY_TIMEOUT</i>	504	Error Timeout(NNNN) superado

## 2.2 StartChat

Este webservice será utilizado para inicializar el canal de chat. Una vez inicializado el canal, se facilitarán los datos de la llamada y de la sesión.

### 2.2.1 Invocación

Para poder invocar a dicho webservice, se deberá hacer una petición http a la siguiente url:

<http://xxx.xxx.xxx.xxx:xxxx/ChatWebService/startchat>

siendo xxx.xxx.xxx.xxx:xxxx la ip y el puerto donde haya sido desplegado el webservice

<b>Document type:</b> Acuerdo de interfaz		<b>Title:</b> Acuerdo de Interfaz Servicio de Chat	
<b>Date:</b> 03/05/2016	<b>State:</b> UNDER REVISION	<b>Author:</b> Antonio M. García Ruiz	<b>Version:</b> 0.3

Se deberán pasar los parámetros a través de get.

Un ejemplo de invocación sería la siguiente

<http://172.25.128.248:8180/ChatWebService/startchat?service=chat1&name=minombre&origin=6177&destiny=60600&codCliente=2516542>

Los distintos parámetros que podemos indicar en dicha invocación son los siguientes:

Nombre	Tipo	Obligatorio	Descripción
<u><a href="#">service</a></u>	String	Si	Servicio de chat a utilizar
<u><a href="#">name</a></u>	String	Si	Nombre del usuario que genera la petición de chat.
<u><a href="#">origin</a></u>	String	Si	Identificador numérico del usuario que genera la petición de chat.(Número telefónico del cliente)
<u><a href="#">destiny</a></u>	String	Si	Identificador de la cola de destino del chat
<u><a href="#">codCliente</a></u>	String	No	Código de cliente

### 2.2.2 Resultado y códigos de error

Si el webservice se ejecuta correctamente enviará el código de respuesta http **SC\_OK 200** acompañado de un JSON con los siguientes datos:

```
{
  "result": "OK",
  "diag": "10002000091463068795",
  "session": "4D463E5289082EB500AC1D2D75221EDB",
  "comments": "Channel created"
}
```

**El campo diag contiene el ucid de la llamada, parámetro necesario en las llamadas a los distintos webservices.**

Si el webservice no se ejecuta de forma correcta, puede devolver los siguientes códigos de error:

Error	Código	Descripción
<i><a href="#">SC_NOT_FOUND</a></i>	404	
<i><a href="#">SC_BAD_REQUEST</a></i>	400	Not enough parameters
<i><a href="#">SC_CONFLICT</a></i>	409	Failed to create the channel
<i><a href="#">SC_INTERNAL_SERVER_ERROR</a></i>	500	e.getMessage()
<i><a href="#">SC_GATEWAY_TIMEOUT</a></i>	504	Error TimeOut(NNNN) superado

<b>Document type:</b> Acuerdo de interfaz		<b>Title:</b> Acuerdo de Interfaz Servicio de Chat	
<b>Date:</b> 03/05/2016	<b>State:</b> UNDER REVISION	<b>Author:</b> Antonio M. García Ruiz	<b>Version:</b> 0.3

### 2.3 FinishChat

Este webservice será utilizado para finalizar el canal de chat.

#### 2.3.1 Invocación

Para poder invocar a dicho webservice, se deberá hacer una petición http a la siguiente url:

<http://xxx.xxx.xxx.xxx:xxxx/ChatWebService/finishchat>

siendo xxx.xxx.xxx.xxx:xxxx la ip y el puerto donde haya sido desplegado el webservice

Se deberán pasar los parámetros a través de get.

Un ejemplo de invocación sería la siguiente

<http://172.25.128.248:8180/ChatWebService/finishchat?ucid=10002000101463126744&service=chat1&name=minombre&origin=6177&destiny=60600>

Los distintos parámetros que podemos indicar en dicha invocación son los siguientes:

Nombre	Tipo	Obligatorio	Descripción
<u>ucid</u>	String	Si	Identificador del chat
<u>service</u>	String	No	Servicio de chat a utilizar
<u>name</u>	String	No	Nombre del usuario que genera la petición de chat.
<u>origin</u>	String	No	Identificador numérico del usuario que genera la petición de chat.(Número telefónico del cliente)
<u>destiny</u>	String	No	Identificador de la cola de destino del chat

#### 2.3.2 Resultado y códigos de error

Si el webservice se ejecuta correctamente enviará el código de respuesta http **SC\_OK 200**

<b>Document type:</b> Acuerdo de interfaz		<b>Title:</b> Acuerdo de Interfaz Servicio de Chat	
<b>Date:</b> 03/05/2016	<b>State: UNDER REVISION</b>	<b>Author:</b> Antonio M. García Ruiz	<b>Version:</b> 0.3

acompañado de un JSON con los siguientes datos:

```
{"result":"OK",
  "comments":"Channel closed"}
```

Si el webservice no se ejecuta de forma correcta, puede devolver los siguientes códigos de error:

Error	Código	Descripción
<i>SC_NOT_FOUND</i>	404	
<i>SC_BAD_REQUEST</i>	400	Not enough parameters
<i>SC_CONFLICT</i>	409	Failed to create the channel SessionID no valid
<i>SC_INTERNAL_SERVER_ERROR</i>	500	Failed updating database, rollback for the UCID e.getMessage()
<i>SC_GATEWAY_TIMEOUT</i>	504	Error TimeOut(NNNN) superado

## 2.4 SendMessage

Este webservice será utilizado para enviar un mensaje de chat.

### 2.4.1 Invocación

Para poder invocar a dicho webservice, se deberá hacer una petición http a la siguiente url:

<http://xxx.xxx.xxx.xxx:xxxx/ChatWebService/sendmessage>

siendo xxx.xxx.xxx.xxx:xxxx la ip y el puerto donde haya sido desplegado el webservice

Se deberán pasar los parámetros a través de get.

Un ejemplo de invocación sería la siguiente

<http://172.25.128.248:8180/ChatWebService/sendmessage?ucid=10002000101463126744&idsender=6177&message=texto a insertar en el chat&tipoContenido=10&service=chat1&name=minombre&origin=6177&destiny=60600>

Los distintos parámetros que podemos indicar en dicha invocación son los siguientes:

<b>Document type:</b> Acuerdo de interfaz		<b>Title:</b> Acuerdo de Interfaz Servicio de Chat	
<b>Date:</b> 03/05/2016	<b>State:</b> UNDER REVISION	<b>Author:</b> Antonio M. García Ruiz	<b>Version:</b> 0.3

Nombre	Tipo	Obligatorio	Descripción
<u>ucid</u>	String	Si	Identificador del chat
<u>idsender</u>	String	Si	Identificar numérico del usuario que genera el mensaje (cliente o agente)
<u>message</u>	String	Si	Mensaje a enviar
<u>tipoContenido</u>	String	Si	Tipo de contenido del mensaje (Texto(10) o Estado(30)) el mensaje de estado corresponde a mensajes de paciencia o indicaciones de acción(escribiendo...)
<u>service</u>	String	No	Servicio de chat a utilizar
<u>name</u>	String	No	Nombre del usuario que genera la petición de chat.
<u>origin</u>	String	No	Identificador numérico del usuario que genera la petición de chat.(Número telefónico del cliente)
<u>destiny</u>	String	No	Identificador de la cola de destino del chat

#### 2.4.2 Resultado y códigos de error

Si el webservice se ejecuta correctamente enviará el código de respuesta http **SC\_OK 200** acompañado de un JSON con los siguientes datos:

```
{
  "idMessage": "144524524",
  "result": "OK",
  "diag": "Up",
  "comment": "Message correctly register"}

```

Si el webservice no se ejecuta de forma correcta, puede devolver los siguientes códigos de error:

Error	Código	Descripción
<i>SC_NOT_FOUND</i>	404	
<i>SC_BAD_REQUEST</i>	400	Not enough parameters
<i>SC_CONFLICT</i>	409	Failed to record the message SessionID no valid
<i>SC_INTERNAL_SERVER_ERROR</i>	500	Database error SessionID is not valid Not session available e.getMessage()
<i>SC_GATEWAY_TIMEOUT</i>	504	Error TimeOut(NNNN) superado

<b>Document type:</b> Acuerdo de interfaz		<b>Title:</b> Acuerdo de Interfaz Servicio de Chat	
<b>Date:</b> 03/05/2016	<b>State:</b> UNDER REVISION	<b>Author:</b> Antonio M. García Ruiz	<b>Version:</b> 0.3

## 2.5 ReceiveMessage

Este webservice será utilizado comprobar si hay mensajes pendientes de recibir en el chat. Este webservice tiene que ser invocado periódicamente para mostrar los mensajes en base de datos a partir del mensaje enviado como parámetro.

### 2.5.1 Invocación

Para poder invocar a dicho webservice, se deberá hacer una petición http a la siguiente url:

<http://xxx.xxx.xxx.xxx:xxxx/ChatWebService/receivemessage>

siendo xxx.xxx.xxx.xxx:xxxx la ip y el puerto donde haya sido desplegado el webservice

Se deberán pasar los parámetros a través de get.

Un ejemplo de invocación sería la siguiente

<http://172.25.128.248:8180/ChatWebService/receivemessage?ucid=10002000101463126744&idsender=60600&idmessage=0&service=chat1&name=minombre&origin=6177&destiny=60600>

Los distintos parámetros que podemos indicar en dicha invocación son los siguientes:

Nombre	Tipo	Obligatorio	Descripción
<u>ucid</u>	String	Si	Identificador del chat
<u>idsender</u>	String	Si	Identificar numérico del usuario que solicita la lectura del mensaje (cliente o agente)
<u>idmessage</u>	String	Si	Identificador del último mensaje recibido
<u>service</u>	String	No	Servicio de chat a utilizar
<u>name</u>	String	No	Nombre del usuario que genera la petición de chat.
<u>origin</u>	String	No	Identificador numérico del usuario que genera la petición de chat.(Número telefónico del cliente)
<u>destiny</u>	String	No	Identificador de la cola de destino del chat

<b>Document type:</b> Acuerdo de interfaz		<b>Title:</b> Acuerdo de Interfaz Servicio de Chat	
<b>Date:</b> 03/05/2016	<b>State:</b> UNDER REVISION	<b>Author:</b> Antonio M. García Ruiz	<b>Version:</b> 0.3

### 2.5.2 Resultado y códigos de error

Si el webservice se ejecuta correctamente enviará el código de respuesta http **SC\_OK 200** acompañado de un JSON con los siguientes datos (si hay algún mensaje pendiente, recibimos el mensaje, junto con su identificador, y el identificador del usuario que lo escribió):

```
{
  "message": "texto a insertar en el chat",
  "idMessage": "144524524",
  "result": "OK",
  "diag": "Up",
  "idUser": "Cliente",
  "nomAgente": "",
  "tipoContenido": "10"}

```

El parámetro idUser devolverá la cadena “Cliente”, “Agente” o “Supervisor”, en el caso de que el idUser sea “Agente” el parámetro nomAgente contendrá el nombre del agente, en caso contrario irá vacío.

Si por el contrario, no existen mensajes pendientes de recibir, el JSON recibido tendrá el siguiente contenido (el identificador de mensajes estará a cero):

```
{
  "message": "NO MORE MESSAGES",
  "idMessage": "0",
  "result": "OK",
  "diag": "Up",
  "idUser": "Cliente",
  "nomAgente": "",
  "tipoContenido": "30"}

```

Si el webservice no se ejecuta de forma correcta, puede devolver los siguientes códigos de error:

Error	Código	Descripción
<i>SC_NOT_FOUND</i>	404	
<i>SC_BAD_REQUEST</i>	400	Not enough parameters
<i>SC_CONFLICT</i>	409	Failed to read the message SessionID no valid
<i>SC_INTERNAL_SERVER_ERROR</i>	500	Database error SessionID is not valid Not session available e.getMessage()
<i>SC_GATEWAY_TIMEOUT</i>	504	Error TimeOut(NNNN) superado