

connectus

API v.1.3

Contexto	4
Llave de API	4
Webhooks	6
Actualizaciones de estado de SMS	6
<i>Respuesta:</i>	6
<i>Ejemplo</i>	6
Reenvío de SMS recibidos	7
<i>Respuesta:</i>	7
<i>Ejemplo</i>	7
Enviar SMS individual	8
Parámetros de entrada:	8
Respuesta:	8
Ejemplo de uso:.....	8
Hacer envío masivo de SMS	9
Parámetros de entrada:	9
Respuesta:	9
Ejemplo de uso:.....	9
Revisar estado de SMS	11
Parámetros de entrada:	11
Respuesta:	11
Ejemplo de uso:.....	11
Revisar estado de envío masivo de SMS	12
Parámetros de entrada:	12
Respuesta:	12
Ejemplo de uso:.....	12
Revisar saldo de SMS	13
Respuesta:	13
Ejemplo de uso:.....	13
Enviar E-Mail	14
Parámetros de entrada:	14
Respuesta:	14
Ejemplo de uso:.....	14
Enviar E-Mail Bulk	15
Parámetros de entrada:	15
Respuesta:	15
Ejemplo de uso:.....	15
Revisar estado de E-Mail	17
Parámetros de entrada:	17
Respuesta:	17
Ejemplo de uso:.....	17
Revisar saldo de E-Mail	18
Respuesta:	18

Ejemplo de uso:.....18

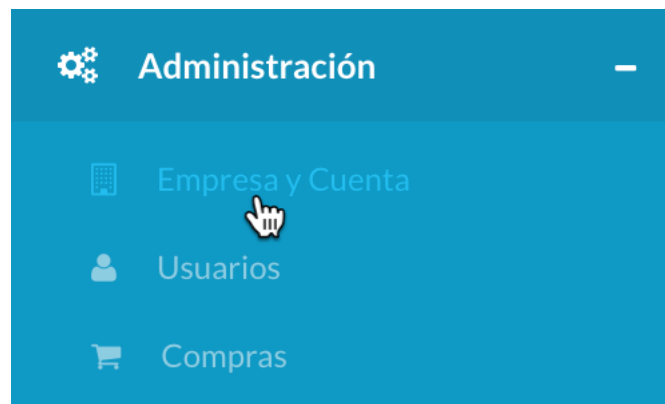
Contexto

El presente documento es una especificación técnica de la API de los servicios disponibles de la plataforma de Connectus.

Llave de API


La plataforma de Connectus cuenta con una API de integración bajo el protocolo *HTTP* y puerto 80, utilizando formato *JSON*. Cada servicio de la API deberá ser consumido indicando en el encabezado de autenticación del llamado una llave única para la cuenta y usuario, que se genera en la plataforma de Connectus.


Para obtener su llave, ingrese a la plataforma de Connectus, en <https://web.connectus.cl>. Cuando haya ingresado, navegue a la sección Administración - Empresa y Cuenta.



Luego, haga *click* en “Generar Llave”

API

ID CUENTA: 



Requiere una llave para utilizar la API de Connectus. Esta es única para su usuario y cuenta y debe integrarse en cada llamado.

Se desplegará por primera y única vez la llave generada para utilizar la API. Cópiala y guárdela, sino deberá repetir el proceso.

Su nueva llave de API es:

16c329a233a047ef879eca2d66c70768

Copiar
llave



Copie y guarde la llave, esta será la única vez que se mostrará completa. En caso de perderla deberá generar una nueva.

Para cada llamado a la API deberá generar el encabezado estándar de Basic Auth sobre HTTPS para sus llamados a la API. El usuario será su identificador de cuenta (que se encuentra en la misma sección) y la contraseña será la llave recién generada.

Webhooks

Para ciertas funcionalidades donde resulta más eficiente realizar una actualización en vivo que constantes requerimientos a la API, Connectus le provee la posibilidad de configurar *webhooks*. Estos se pueden configurar en la plataforma de Connectus, en <https://web.connectus.cl>. Cuando haya ingresado, navegue a la sección Administración - Empresa y Cuenta.




Actualizaciones de estado de SMS

Para la entrega de actualizaciones de estado de SMS enviados configure la URL correspondiente.

URL de envío de actualizaciones de estado de SMS

Sin url de envío de actualizaciones

Editar


Al quedar configurada, toda actualización de parte de los proveedores de telefonía móvil (Entel, etc.) será redirigida a través de HTTP a la URL provista.

Respuesta:

id_sms (string)	Identificador hexadecimal único del mensaje.
status (string)	Estado del mensaje, pudiendo este ser: PROGRAMADO, EN_COLA, NUMERO_INVALIDO, ENVIADO_A_CARRIER, ESPERANDO_RESPUESTA, NO_ENTREGABLE, ENTREGADO

Ejemplo

```
1. {"id_sms": "f04789942fe84541b26b8707207e4193", "status": "ENTREGADO"}
```

Reenvío de SMS recibidos

Para el reenvío de SMS recibidos configure la URL correspondiente.

URL de reenvío de SMS recibidos

Sin url de reenvío

Editar



Al quedar configurada, todo SMS recibido en sus números exclusivos será redirigido a través de HTTP a la URL provista.

Respuesta:

id_sms (string)	Identificador hexadecimal único del mensaje.
dst_number (long)	Número de destino en Connectus en formato internacional.
src_number (long)	Número de origen en formato internacional.
src_provider (string)	Proveedor de telefonía móvil de origen.
sms_content (string)	Texto de contenido del mensaje

Ejemplo

```
1. {"id_sms": "ec3d38a076ec4206a0e4e3fe039ddcc8", "dst_number": "56442330001", "src_number": "56990324203", "src_provider": "Wom", "sms_content": "Hola Connectus!"}
```

Enviar SMS individual

Servicio para mandar un único SMS a un número de destino.

URL: 'https://api.connectus.cl/api_v1/send_sms'

Tipo de operación HTTP: POST

Parámetros de entrada:

dst_number (long)	Número de destino en formato internacional de 11 dígitos. (569.....)
sms_content (string)	Texto de contenido del mensaje.

Respuesta:

id_sms (string)	Identificador hexadecimal único del mensaje.
-----------------	--

Ejemplo de uso:

Código (Lenguaje Python):

```
1. import requests
2. url = 'https://api.connectus.cl/api_v1/send_sms'
3. params = dict()
4. params['dst_number'] = 56990324203
5. params['sms_content'] = "The ships hung in the sky in much the same way that bricks don
   't."
6. response = requests.post(url, params=params, auth=('84f71863b7f64738aabc97afed7cd54d',
   16c329a233a047ef879eca2d66c70768))
7. print response.text
```

Respuesta:

```
2. {"id_sms": "f04789942fe84541b26b8707207e4193"}
```


Hacer envío masivo de SMS

Servicio para mandar una serie de SMS (hasta 1.000 por consulta)

URL: 'https://api.connectus.cl/api_v1/send_sms_bulk'

Tipo de operación HTTP: POST

Parámetros de entrada:

messages (JSON)	Arreglo de diccionarios que especifican cada mensaje con los parámetros <i>dst_number</i> y <i>sms_content</i> . Estos son idénticos a los especificados en la descripción del servicio anterior.
delivery_name (string)	Nombre del envío.
deliver_at (datetime string)	Fecha y hora programada de entrega del envío en formato datetime. ('2018-03-25 10:30:00')

Respuesta:

id_delivery (string)	Identificador hexadecimal único del envío.
messages (array)	Arreglo de respuestas a cada mensaje individual. Esto incluye mensajes de error en caso de formateo erróneo o <i>id_sms</i> , identificadores únicos de cada mensaje tal como descrito en el servicio anterior.

Ejemplo de uso:

Código (Lenguaje Python):

```
1. import requests
2. import json
3. url = 'https://api.connectus.cl/api_v1/send_sms_bulk'
4. params = dict()
5. params['messages'] = json.dumps([
6.     {'dst_number': 56944444444,
7.      'sms_content': "The ships hung in the sky in much the same way that bricks don't."
8.     },
9.     {'dst_number': 56990324203,
10.     'sms_content': "So long, and thanks for all the fish."}
11. ])
12. params['delivery_name'] = "Douglas Adams quotes"
13. params['deliver_at'] = '2018-03-25 10:30:00'
14. response = requests.post(url, params=params, auth=('84f71863b7f64738aabc97afed7cd54d',
15.     16c329a233a047ef879eca2d66c70768))
16. print response.text
```

Respuesta:

```
1. {
2.   "id_delivery": "0a45704eca394c3fbf6eb8eaf7950b5e",
3.   "messages": [
4.     {
5.       "id_sms": "6bfb1d9d4c6640bab93f23cfdbd5750f"
6.     },
7.     {
8.       "id_sms": "6cf199208eb440cc8801b486f4baa8e5"
9.     }
10.  ]
11. }
```

Revisar estado de SMS

Responde con el estado de entrega de un SMS dado.

URL: 'https://api.connectus.cl/api_v1/check_sms_status'

Tipo de operación HTTP: GET

Parámetros de entrada:

id_sms (string) Identificador hexadecimal único del mensaje.

Respuesta:

status (string) Estado del mensaje, pudiendo este ser:

PROGRAMADO, EN_COLA, NUMERO_INVALIDO, ENVIADO_A_CARRIER,
ESPERANDO_RESPUESTA, NO_ENTREGABLE, ENTREGADO

Ejemplo de uso:

Código (Lenguaje Python):

```
1. import requests
2. url = 'https://api.connectus.cl/api_v1/check_sms_status'
3. params = dict()
4. params['id_sms'] = '6bfb1d9d4c6640bab93f23cfdbd5750f'
5. response = requests.get(url, params=params, auth=('84f71863b7f64738aabc97afed7cd54d',
16c329a233a047ef879eca2d66c70768))
6. print response.text
```

Respuesta:

```
1. {"status": "ENTREGADO"}
```

Revisar estado de envío masivo de SMS

Responde con los estados individuales de cada SMS de cierto envío.

URL: 'https://api.connectus.cl/api_v1/check_delivery_status'

Tipo de operación HTTP: GET

Parámetros de entrada:

id_delivery (string) Identificador hexadecimal único del envío.

Respuesta:

id_delivery (string)	Identificador hexadecimal único del envío.
delivery_name (string)	Nombre del envío.
messages (array)	Arreglo del detalle para cada mensaje individual, que incluye los campos <i>status</i> , que toma los mismos valores de la función anterior e <i>id_sms</i> , el identificador único del mensaje.

Ejemplo de uso:

Código (Lenguaje Python):

```
1. import requests
2. url = 'https://api.connectus.cl/api_v1/check_delivery_status'
3. params = dict()
4. params['id_delivery'] = '0a45704eca394c3fbf6eb8eaf7950b5e'
5. response = requests.get(url, params=params, auth=(
6.     '84f71863b7f64738aabc97afed7cd54d', '16c329a233a047ef879eca2d66c70768'))
7. print response.text
```

Respuesta:

```
1. {
2.     "delivery_status": {
3.         "id_delivery": "0a45704eca394c3fbf6eb8eaf7950b5e",
4.         "delivery_name": "Douglas Adams quotes",
5.         "messages": [
6.             {
7.                 "status": "PROGRAMADO",
8.                 "id_sms": "6bfb1d9d4c6640bab93f23cfdbd5750f"
9.             },
10.            {
11.                "status": "PROGRAMADO",
12.                "id_sms": "6cf199208eb440ccc8801b486f4baa8e5"
13.            }
14.        ],
15.    }
16. }
```

Revisar saldo de SMS

Responde con el saldo de SMS disponible para la cuenta.

URL: 'https://api.connectus.cl/api_v1/check_balance'

Tipo de operación HTTP: GET

Respuesta:

balance (integer) Saldo de SMS disponible para la cuenta.

Ejemplo de uso:

Código (Lenguaje Python):

```
1. import requests
2. url = 'https://api.connectus.cl/api_v1/check_balance'
3. response = requests.get(url, auth=('84f71863b7f64738aabc97afed7cd54d',
   16c329a233a047ef879eca2d66c70768))
4. print response.text
```

Respuesta:

```
1. {"balance": "1432"}
```

Enviar E-Mail

Servicio para mandar un único E-Mail a un correo de destino.

URL: 'https://api.connectus.cl/api_v1/send_email'

Tipo de operación HTTP: POST

Parámetros de entrada:

dst_email (string)	Correo de destino
subject (string)	Asunto del E-Mail.
email_content (string)	Texto de contenido del E-Mail.
sender (string) <i>opcional</i>	Remitente del E-Mail
src_email (string) <i>opcional</i>	Correo de origen. En caso de no ser indicado se utilizará el correo por defecto de la cuenta. Cualquier correo debe ser validado desde la plataforma en https://web.connectus.cl/#/app/empresa&cuenta

Respuesta:

id_email (string) Identificador hexadecimal único del E-Mail.

Ejemplo de uso:

Código (Lenguaje Python):

```
8. import requests
9. url = 'https://api.connectus.cl/api_v1/send_email'
10. params = dict()
11. params['dst_email'] = 'unemail@uncliente.com'
12. params['email_content'] = "The ships hung in the sky in much the same way that bricks d
    on't."
13. response = requests.post(url, params=params, auth=('84f71863b7f64738aabc97afed7cd54d',
    16c329a233a047ef879eca2d66c70768))
14. print response.text
```

Respuesta:

```
3. {"id_email": "f04789942fe84541b26b8707207e4193"}
```

Enviar E-Mail Bulk

Servicio para mandar una serie de E-Mails.

URL: 'https://api.connectus.cl/api_v1/send_email_bulk'

Tipo de operación HTTP: POST

Parámetros de entrada:

emails (JSON)	Arreglo de diccionarios que especifican cada email con parametros idénticos a los especificados en la descripción del servicio anterior. (Excluyendo src_email)
delivery_name (string)	Nombre del envío.
deliver_at (datetime string)	Fecha y hora programada de entrega del envío en formato datetime. ('2019-03-25 10:30:00')
src_email (string)	Correo de origen. En caso de no ser indicado se utilizará el correo por defecto de la cuenta. Cualquier correo debe ser validado desde la plataforma en https://web.connectus.cl/#/app/empresa&cuenta

Respuesta:

id_delivery (string)	Identificador hexadecimal único del envío.
emails (array)	Arreglo de respuestas a cada E-Mail individual. Esto incluye mensajes de error en caso de formateo erróneo o <i>id_email</i> , identificadores únicos de cada mensaje tal como descrito en el servicio anterior.

Ejemplo de uso:

Código (Lenguaje Python):

```
1. import requests
2. import json
3. url = 'http://api.connectus.cl/api_v1/send_email_bulk'
4. params = dict()
5. params['delivery_name'] = 'jancss@gmail.com'
6. params['emails'] = json.dumps([
7.     {'dst_email': 'unemail@uncliente.com',
8.      'email_content': "The ships hung in the sky in much the same way that bricks don't
9.      'subject': 'Demolitions'
10.    },
11.    {'dst_email': 'otroemail@otrocliente.com',
12.     'email_content': "So long, and thanks for all the fish.",
13.     'subject': 'Dolphins'
14.    }
15. ])
16. params['delivery_name'] = "Douglas Adams quotes"
17. params['deliver_at'] = '2019-03-25 10:30:00'
18. params['src_email'] = 'juan@empresa.com'
19. response = requests.post(url, params=params, auth=('84f71863b7f64738aabc97afed7cd54d',
16c329a233a047ef879eca2d66c70768))
```

```
20. print response.text
```

Respuesta:

```
12. {  
13.   "id_delivery": "0a45704eca394c3fbf6eb8eaf7950b5e",  
14.   "messages": [  
15.     {  
16.       "id_email": "6bfb1d9d4c6640bab93f23cfdbd5750f"  
17.     },  
18.     {  
19.       "id_email": "6cf199208eb440cc8801b486f4baa8e5"  
20.     }  
21.   ]  
22. }
```


Revisar estado de E-Mail

Responde con el estado de entrega de un E-Mail dado.

URL: 'https://api.connectus.cl/api_v1/check_email_status_events'

Tipo de operación HTTP: GET

Parámetros de entrada:

id_email (string) Identificador hexadecimal único del E-Mail.

Respuesta:

status (string) Estado del E-Mail, pudiendo este ser:

PROGRAMADO, EN_COLA, NUMERO_ENVIADO, NO_ENTREGABLE, ENTREGADO

events (array) Arreglo de eventos del E-Mail *email_event*, pudiendo estos ser:

CLICKEADO, ABIERTO

y *event_timestamp*, fecha y hora del evento en format datetime ('2019-03-25 10:30:00')

Ejemplo de uso:

Código (Lenguaje Python):

```
1. import requests
2. url = 'https://api.connectus.cl/api_v1/check_email_status_events'
3. params = dict()
4. params['id_email'] = '6bfb1d9d4c6640bab93f23cfdbd5750f'
5. response = requests.get(url, params=params, auth=('84f71863b7f64738aabc97afed7cd54d',
16c329a233a047ef879eca2d66c70768))
6. print response.text
```

Respuesta:

```
1. {"status": "ENTREGADO", "events": [{"timestamp": "05-03-19 17:05:25", "event": "ABIERTO"}, {"timestamp": "05-03-19 17:05:28", "event": "CLICKEADO"}]}
```

Revisar saldo de E-Mail

Responde con el saldo de E-Mail disponible para la cuenta.

URL: 'https://api.connectus.cl/api_v1/check_email_balance'

Tipo de operación HTTP: GET

Respuesta:

balance (integer) Saldo de E-Mail disponible para la cuenta.

Ejemplo de uso:

Código (Lenguaje Python):

```
5. import requests
6. url = 'https://api.connectus.cl/api_v1/check_email_balance'
7. response = requests.get(url, auth=('84f71863b7f64738aabc97afed7cd54d',
16c329a233a047ef879eca2d66c70768))
8. print response.text
```

Respuesta:

```
2. {"balance": "35123"}
```