

API Hey Now  
Documento técnico



## Contenido

API Hey Now.....	1
Documento técnico.....	1
1. Versiones.....	3
2. Introducción.....	3
3. Configuración de canales.....	3
4. Acceso.....	3
5. Caso de uso.....	4
1 Autenticación.....	4
6 Envío de mensaje.....	5
6. Envíos masivos.....	8
7. Consulta de estado de mensaje.....	8
8. Consulta de agenda.....	9

## 1. Versiones

1.0	Abril 2018	Primer versión.
1.1	Mayo 2018	Se agregan métodos GET para consultar estado de mensaje enviado, obtener agenda y consultar por un contacto particular.
1.2	Agosto 2018	Se agrega capítulo de configuración de canales. Se agrega parámetro contactValidation en método message. Se agrega método massiveMessage.

## 2. Introducción

El presente documento contiene información técnica sobre la interfaz de programación (API) de la plataforma Hey Now.

Dicha API está diseñada para que sistemas de terceros puedan interactuar con Hey Now para distintos fines, por ejemplo enviar mensajes a través de los canales contratados, obtener estadísticas, obtener los contactos de un canal, configurar aspectos de un bot, etc.

A nivel técnico, se trata de una API REST y se utiliza el framework Swagger (<https://swagger.io>) como herramienta de diseño, documentación y testing, el cual implementa el standard OpenAPI Specification (OAS).

## 3. Configuración de canales

Los diferentes canales que los clientes pueden contratar en Hey Now (por ejemplo números de WhatsApp) se configuran según los acuerdos comerciales de cada caso. Algunas de esas configuraciones incluyen la posibilidad de definir si un determinado canal **debe validar** que un contacto pertenece a su agenda, es decir que ése contacto le ha escrito previamente; otra configuración posible es definir un canal alternativo asociado al primero. Esto último típicamente se utiliza cuando se define un canal entrante y otro saliente.

## 4. Acceso

El punto de entrada a la API es

<https://web.heynow.com.uy:8310/api>

y puede encontrarse la especificación completa en el documento JSON que se encuentra en

<https://web.heynow.com.uy:8310/swagger.json>

con el cual se pueden utilizar herramientas de generación de código automático como por ejemplo la que dispone Swagger (<https://swagger.io/swagger-codegen>).

Como página de testing, se dispone de

<https://web.heynow.com.uy:8310>

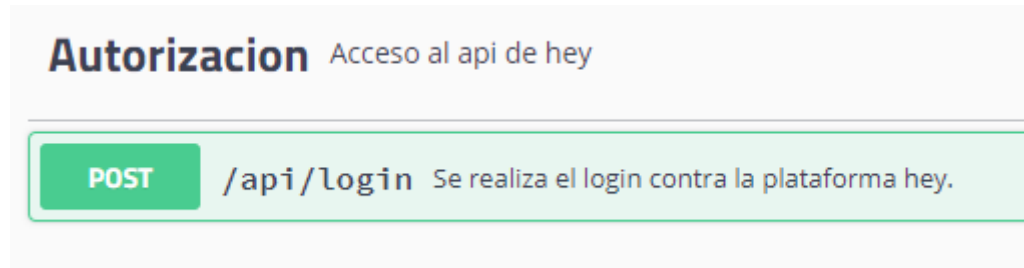
en donde pueden utilizarse los diferentes métodos y encontrar descripción de los mismos.

## 5. Caso de uso

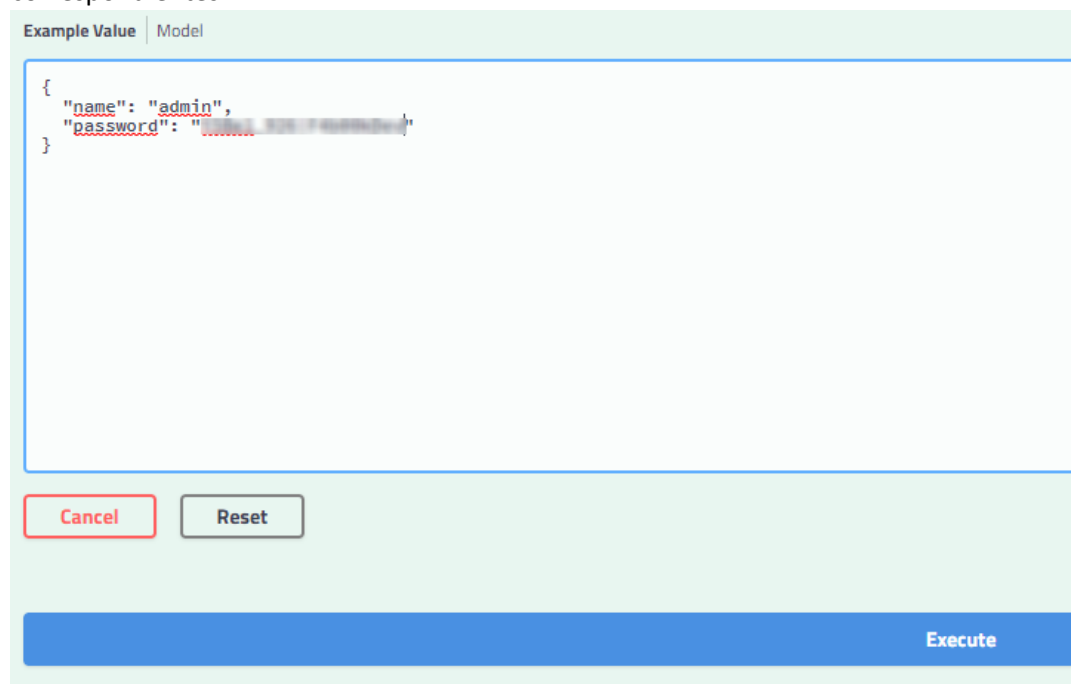
A continuación se detalla el proceso completo para el envío de un mensaje de WhatsApp utilizando la página de Swagger. Esta secuencia de pasos es la misma que debe ejecutarse de un sistema informático para lograr el envío de un mensaje.

### 1 Autenticación

La API es autenticada y por tanto el primer paso será el login utilizando las credenciales de un usuario de Hey Now asociado a la cuenta desde la que se realizará el envío. Para esto se debe ejecutar el método 'login' dentro de la sección 'Autorización'

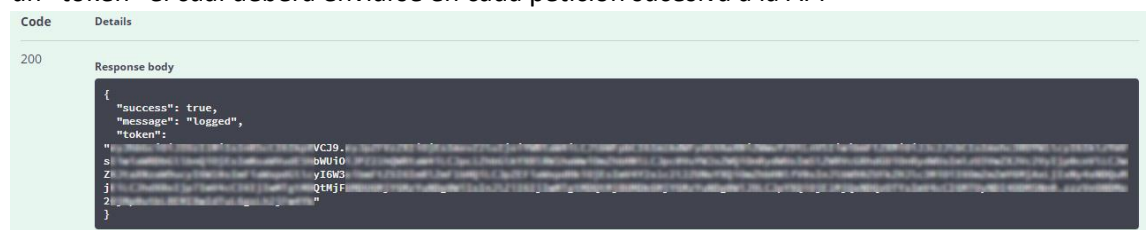


- Haciendo click en el botón 'Try it out' se podrán editar los valores de "name" y "password" dentro de 'Example Value' para escribir las credenciales correspondientes



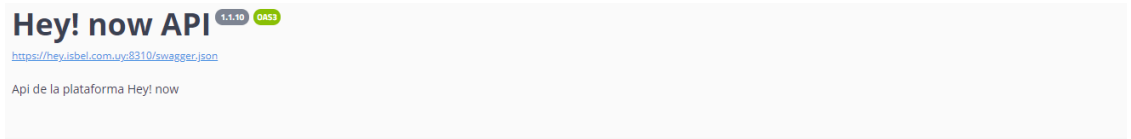
y luego click en 'Execute'.

- Si la autenticación es exitosa, se obtendrá código de respuesta 200 junto con un "token" el cual deberá enviarse en cada petición sucesiva a la API



Es importante notar que el token tiene una validez de 24Hs y pasado este lapso se deberá ejecutar el login nuevamente.

- Una vez copiado el token se debe hacer click en 'Authorize' en la parte superior de la página



- Pegar el token en 'Value' y click en 'Authorize'. El ícono del candado quedará cerrado, indicando que el usuario está autenticado.

### Available authorizations

#### hey\_auth (http, Bearer)

Value:

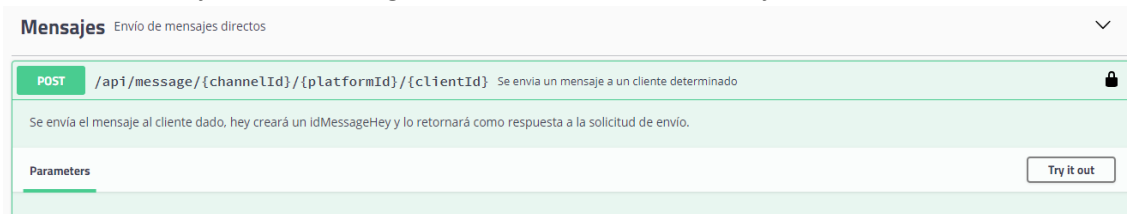
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTUxNiJ9

Authorize

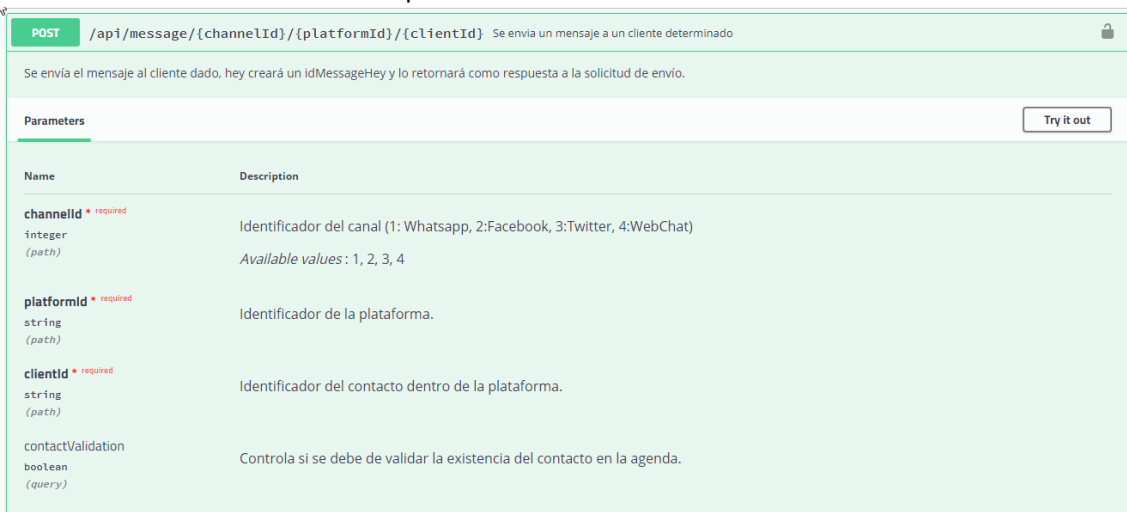
Close

- Envío de mensaje

En este punto, ya se podrá utilizar el método para el envío de mensajes, para esto se ejecutará 'message' dentro de la sección 'Mensajes'



Haciendo click en 'Try it out' se accede a la sección 'Parameters' para ingresar la información requerida



**channelId** es un valor numérico fijo que depende del canal por el que se desea realizar el envío, los valores permitidos son:

- **1** para WhatsApp
- **2** para Messenger
- **3** para Twitter

**platformId** es el identificador del canal del envío:

- Número de celular para WhatsApp
- Id de Facebook para Messenger
- Id de cuenta para Twitter

**clientId** es el identificador del cliente, quien recibirá el mensaje

- Número de celular del cliente para WhatsApp
- Id de Facebook del cliente para Messenger
- Id de cuenta del cliente para Twitter

**contactValidation** indica si debe verificarse el destino del mensaje antes de enviarlo. El comportamiento es el siguiente

- Si el canal está configurado **con** validación de contactos
  - **Sin** bot alternativo
    - **(contactValidation == true) OR (contactValidation == false)**
      - Si **no** existe contacto en la agenda **NO** se envía mensaje
      - Si existe contacto en la agenda **SI** se envía mensaje
    - **Con** bot alternativo
      - **(contactValidation == true)**
        - Si **no** existe contacto en la agenda **NO** se envía mensaje
        - Si existe contacto en la agenda **SI** se envía mensaje
      - **(contactValidation == false)**
        - Si **no** existe contacto en la agenda, se intenta enviar el mensaje por el bot alternativo
        - Si existe contacto en la agenda **SI** se envía mensaje
  - Si el canal está configurado **sin** validación de contactos
    - **Con o sin** bot alternativo
      - **(contactValidation == true)**
        - Si **no** existe contacto en la agenda **NO** se envía mensaje
        - Si existe contacto en la agenda **SI** se envía mensaje
      - **(contactValidation == false)**
        - **SI** se envía mensaje sin importar si el contacto está en la agenda

Por último, se escribe el mensaje a enviar en el parámetro “text” dentro de ‘Example Value’ en ‘Request body’

Name	Description
<b>channelId</b> * required integer (path)	<input type="text" value="1"/>
<b>platformId</b> * required string (path)	<input type="text" value="59891827058"/>
<b>clientId</b> * required string (path)	<input type="text" value="59893323778"/>

**Request body** required

Example Value | Model

```
{
  "text": "prueba desde API de Hey Now!"
}
```

y luego click en ‘Execute’.

Como respuesta, se pueden obtener los siguientes valores:

Código	Descripción	Causa probable
200	OK	Mensaje enviado a la plataforma correspondiente
401	Token no valido o autenticación no realizada	No se ejecutó el login previamente o el token utilizado expiró
403	Permisos insuficientes	El platformId proporcionado no corresponde a las credenciales
404	Recurso no encontrado	El clientId no pertenece a la agenda del platformId, es decir, el cliente nunca se contactó con ese canal (bot).

## 6. Envíos masivos

Únicamente para aquellos canales configurados para **NO** verificar contacto antes de enviar mensajes, es posible enviar mensajes a cualquier destino.

Para este fin se ejecuta 'masiveMessage' de manera similar al método 'message':



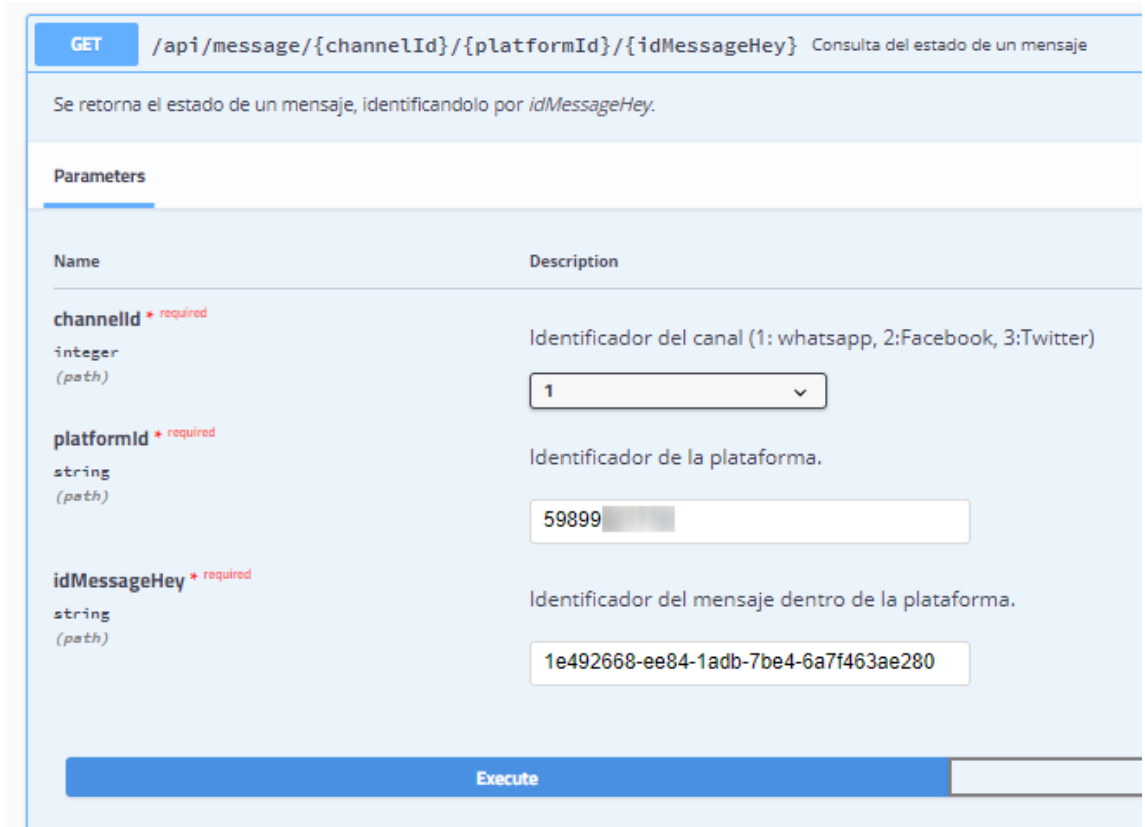
The screenshot shows an API endpoint for sending a mass message. The endpoint is `POST /api/masiveMessage/{channelId}/{platformId}/{clientId}`. The description states: "Se envía un mensaje a un cliente determinado sin importar que se encuentre en la agenda de la plataforma dada." Below the endpoint, there is a "Parameters" section with a "Try it out" button. The parameters are:

Name	Description
<b>channelId</b> * required integer (path)	Identificador del canal (1: Whatsapp, 2:Facebook, 3:Twitter, 4:WebChat) Available values: 1, 2, 3, 4
<b>platformId</b> * required string (path)	Identificador de la plataforma.
<b>clientId</b> * required string (path)	Identificador del contacto dentro de la plataforma.

## 7. Consulta de estado de mensaje

Una vez realizado el envío, es posible consultar el estado del mensaje.

Para esto se ejecuta el método de Consulta dentro de 'Mensajes' utilizando el *idMessageHey* devuelto como respuesta del método de envío



The screenshot shows an API endpoint for checking the status of a message. The endpoint is `GET /api/message/{channelId}/{platformId}/{idMessageHey}`. The description states: "Se retorna el estado de un mensaje, identificandolo por idMessageHey." Below the endpoint, there is a "Parameters" section with an "Execute" button. The parameters are:

Name	Description
<b>channelId</b> * required integer (path)	Identificador del canal (1: whatsapp, 2:Facebook, 3:Twitter) 1
<b>platformId</b> * required string (path)	Identificador de la plataforma. 59899
<b>idMessageHey</b> * required string (path)	Identificador del mensaje dentro de la plataforma. 1e492668-ee84-1adb-7be4-6a7f463ae280

para obtener su estado



Code	Details
200	Response body "LU"

Como respuesta, se pueden obtener los siguientes valores:

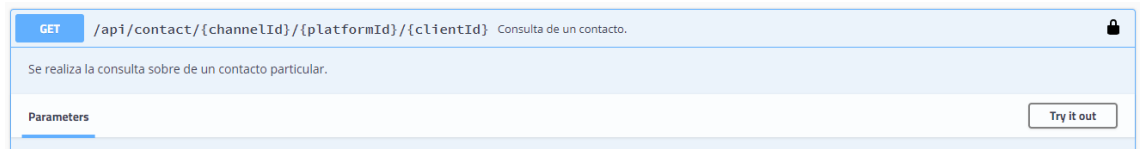
Código	Descripción	Causa probable
200	OK	Resultado del envío: <ul style="list-style-type: none"> <li>• ? Estado desconocido</li> <li>• RS Recibido por servidor de plataforma (típicamente un tick)</li> <li>• RU Recibido por usuario (dos ticks)</li> <li>• LU Leído por usuario (tick azul en caso que el usuario lo habilite)</li> </ul>
401	Token no valido o autenticación no realizada	No se ejecutó el login previamente o el token utilizado expiró
403	Permisos insuficientes	El platformId proporcionado no corresponde a las credenciales
404	Mensaje no encontrado	No se encuentra el mensaje proporcionado
500	Error interno	Error a nivel de servidor

## 8. Consulta de agenda

A menos que se explicite lo contrario, la API de envío de mensajes verifica que el destino del mismo pertenezca a la agenda de la plataforma que envía, es decir, que el destinatario se haya contactado al menos una vez con ese canal.

Es posible realizar esta verificación antes de invocar el método de envío y para ello se debe ejecutar alguno de los métodos de la sección 'Contactos'.

Para consultar por un contacto particular



Click en 'Try it out'

**GET** /api/contact/{channelId}/{platformId}/{clientId} Consulta de un contacto.

Se realiza la consulta sobre de un contacto particular.

**Parameters**

Name	Description
<b>channelId</b> * required integer (path)	Identificador del canal (1: whatsapp, 2:Facebook, 3:Twitter)
<b>platformId</b> * required string (path)	Identificador de la plataforma.
<b>clientId</b> * required string (path)	Identificador del contacto dentro de la plataforma.

**Execute**

Si el contacto se encuentra en la agenda del canal

Code	Details
200	<p><b>Response body</b></p> <pre> {   "_id": "59ced3c032fb0814d5618eed",   "phones": [     "59895 ██████████"   ],   "first_name": "59895 ██████████",   "last_name": null,   "profile_pic": null,   "locale": null,   "timezone": null,   "gender": null,   "channels": [     {       "channel": 1,       "platformId": "59899 ██████████",       "clientId": "59895 ██████████",       "metaData": {}     }   ] } </pre>

Como respuesta, se pueden obtener los siguientes valores:

Código	Descripción	Causa probable
200	OK	Contacto encontrado en agenda del canal
401	Token no valido o autenticación no realizada	No se ejecutó el login previamente o el token utilizado expiró
403	Permisos insuficientes	El platformId proporcionado no corresponde a las credenciales
404	Contacto no encontrado	El clientId no pertenece a la agenda del platformId, es decir, el cliente nunca se contactó con ese canal (bot).
500	Error interno	Error a nivel de servidor

De manera similar, es posible descargar toda la agenda del canal hasta ese momento, para esto

**Contactos** Agenda de contactos

---

**GET** `/api/contact/{channelId}/{platformId}` Agenda de la plataforma

Se retornan todos los usuarios que pertenecen a una plataforma dada.

**Parameters** Try it out

se envían los parámetros correspondientes

**GET** /api/contact/{channelId}/{platformId} Agenda de la plataforma

Se retornan todos los usuarios que pertenecen a una plataforma dada.

**Parameters**

Name	Description
<b>channelId</b> * required integer (path)	Identificador del canal (1: whatsapp, 2:Facebook, 3:Twitter) <input type="text" value="1"/>
<b>platformId</b> * required string (path)	Identificador de la plataforma. <input type="text" value="59899"/>

**Execute**

y se obtienen todos los clientes que han contactado con el canal

```

"gender": null,
"channels": [
  {
    "channel": 1,
    "platformId": "598[REDACTED]",
    "clientId": "59891[REDACTED]",
    "metaData": {}
  }
]
},
{
  "_id": "5acd53304dbff26358e3206",
  "phones": [
    "59891[REDACTED]"
  ],
  "first_name": "59891[REDACTED]",
  "last_name": null,
  "profile_pic": null,
  "locale": null,
  "timezone": null,
  "gender": null,
  "channels": [
    {
      "channel": 1,
      "platformId": "598[REDACTED]",
      "clientId": "59891[REDACTED]",
      "metaData": {}
    }
  ]
},
{
  "_id": "59e0fa558d803f08968ba35e",
  "phones": [
    "59891238805"
  ],
  "first_name": "59891[REDACTED]",
  "last_name": null,
  "profile_pic": null,
  "locale": null,
  "timezone": null,
  "gender": null,
  "channels": [
    {
      "channel": 1,
      "platformId": "598[REDACTED]",
      "clientId": "59891[REDACTED]",
      "metaData": {}
    }
  ]
},
{
  "_id": "5a9c30b92633c948958cd2f5",
  "phones": [
    "59891[REDACTED]"
  ],
  "first_name": "59891[REDACTED]",
  "last_name": null,
  "profile_pic": null,
  "locale": null,
  "timezone": null,
  "gender": null,
}

```

Como respuesta, se pueden obtener los siguientes valores:

Código	Descripción	Causa probable
200	OK	Se obtiene la agenda completa del canal
401	Token no valido o autenticación no realizada	No se ejecutó el login previamente o el token utilizado expiró
403	Permisos insuficientes	El platformId proporcionado no corresponde a las credenciales
500	Error interno	Error a nivel de servidor