



# **TokApp API**

## *Documentación*

Versión 1.3 – 29/02/2016



TokApp  
hola@TokApp.com  
(+34) 91 737 29 01  
Rúa Santander 1 Bajo  
Vigo, España



## Índice de contenido

<a href="#">Descripción del servicio.....</a>	<a href="#">3</a>
<a href="#">Sintaxis de las peticiones.....</a>	<a href="#">4</a>
<a href="#">Sintaxis de las respuestas.....</a>	<a href="#">4</a>
<a href="#">Descripción de los comandos.....</a>	<a href="#">5</a>
Command: status.....	5
Command: getcontacts.....	6
Command: send.....	8
Command: getdeliverystatus.....	10
Command: getmessages.....	12
<a href="#">Códigos de error.....</a>	<a href="#">13</a>



## **Descripción del servicio**

Este servicio Web le permitirá realizar lo siguiente <sup>(1)</sup>:

- Comprobar si uno o varios usuarios están registrados en TokApp a partir de su email o teléfono móvil y obtener sus nombres de usuario.
- Enviar mensajes de texto, pudiendo adjuntar un archivo de imagen en cada envío (aunque se envíe a varios destinatarios).
- Comprobar el estado de un envío.
- Comprobar si hay mensajes entrantes.

Todas las peticiones se realizarán a la misma URL asignada por el proveedor y todos sus parámetros se enviarán mediante petición POST en formato JSON con codificación UTF-8. Todas las respuestas se recibirán también en formato JSON con la misma codificación.

El servidor admitirá tanto peticiones HTTP como HTTPS, dejando a criterio del desarrollador usar uno u otro nivel de seguridad, aunque se recomienda usar siempre protocolo HTTPS para entornos de producción, usando las conexiones sin cifrado durante el desarrollo o testeo o si se pretende enviar mensajes desde algún dispositivo que no admita seguridad SSL.

Todas las peticiones requieren el envío de la clave API suministrada por el proveedor y que identifica al remitente de los mensajes y a la licencia de uso

(1) Según las características de su licencia de uso.



## **Sintaxis de las peticiones**

Todas las peticiones recibirán los datos en formato JSON con la siguiente estructura:

```
{  
  key: '<Clave API>',  
  command: '<Comando>',  
  data: {<Parámetros>}  
}
```

El campo *key* deberá proporcionar la clave API suministrada por el proveedor del servicio.

El campo *command* será el nombre del comando a ejecutar en la petición, de acuerdo con la relación de comandos que se indica más adelante.

El campo *data* será otra estructura JSON con los parámetros de entrada específicos de cada comando, como se explica en la descripción de cada uno. Este campo puede omitirse si el comando no lo necesita.

## **Sintaxis de las respuestas**

Todas las respuestas tendrán la siguiente estructura JSON:

```
{  
  result: <Código de resultado>,  
  message: '<Mensaje descriptivo>',  
  data: {<Datos devueltos>}  
}
```

El campo *result* devolverá un código numérico que identifica el error en caso de haberse producido alguno o 0 si la operación fue correcta. Los códigos de error posibles se describen más adelante.

En caso de error, el campo *message* devuelve una descripción en inglés del problema.

En caso de éxito, el campo *data* contendrá en formato JSON los datos devueltos por la petición, tal y como se indica en la descripción del comando ejecutado. Si el comando no devolviese información adicional este campo no vendrá en la respuesta.



## ***Descripción de los comandos***

Command: `status`

Descripción: Obtiene el estado actual de la licencia y las limitaciones actuales.

Entrada: Sin campo *data*.

Salida:

- `maxContacts`: N° máximo de contactos de destino diferentes a los que se pueden enviar mensajes en un mes natural.
- `maxMessages`: N° máximo de mensajes que se pueden enviar en un día natural. Un valor de 0 indica que no hay límite.
- `contactsSent`: N° de contactos diferentes a los que se han enviado mensajes en el mes natural actual.
- `messagesSent`: N° de mensajes enviados en el día de hoy.
- `multiSend`: Indica mediante 0 (falso) o 1 (verdadero) si es posible enviar más de un mensaje por petición en el comando `send`.
- `getContacts`: Indica mediante 0 (falso) o 1 (verdadero) si el comando `getcontacts` está disponible.
- `available`: Indica mediante 0 (falso) o 1 (verdadero) si el servicio de envío de mensajes está disponible en este momento.



### Ejemplo:

Petición:

```
{
  key: '123456789',
  command: 'status'
}
```

Respuesta:

```
{
  result: 0,
  message: "",
  data: {
    maxContacts: 500,
    maxMessages: 0,
    contactsSent: 18,
    messagesSent: 216,
    multiSend: 1,
    available: 1
  }
}
```

Command: getcontacts

Descripción: Obtiene los nombres de usuario en TokApp de uno o varios contactos a partir de su email o nº de teléfono. Es conveniente hacer esta llamada periódicamente para descubrir a los usuarios de TokApp dado que cada día se pueden ir registrando más usuarios.

Entrada:

- phones: Array con los números de teléfono móvil a comprobar.
- emails: Array con los emails a comprobar.

Salida: Devolverá un array con los siguientes campos en cada elemento:

- phone: Nº de teléfono tal y como se envió en la petición.
- email: Dirección de email tal y como se envió en la petición.
- username: Nombre de usuario correspondiente.



Ejemplo:

Petición:

```
{  
  key: '123456789',  
  command: 'getcontacts'  
  data: {  
    phones: ['611223344', '622334455'],  
    emails: ['juanlopez@atth.com', 'emr@etth.com']  
  }  
}
```

Respuesta:





```
{
  result: 0,
  message: "",
  data: [
    {
      phone: '622334455',
      email: "",
      username: 'usuario.tokapp.es'
    },
    {
      phone: "",
      email: 'juanlopez@atth.com',
      username: 'juanlopez.tokapp.es'
    }
  ]
}
```

Los teléfonos o emails que no correspondan con un usuario de TokApp no serán devueltos en la respuesta. Si ninguno de los datos corresponde con un usuario de TokApp se devolverá una lista vacía, pero no generará error.



### Command: send

Descripción: Envía uno o varios mensajes a uno o varios destinatarios. Esta petición también permite el envío de una imagen, acompañada de texto a uno o más destinatarios. La posibilidad de enviar múltiples mensajes en la misma petición dependerá de las características de su licencia y puede ser comprobada mediante el parámetro *multiSend* del comando *status*. Si se adjunta una imagen, sólo se podrá enviar un mensaje en la petición aunque sean varios los destinatarios.

Entrada: Array con los mensajes a enviar (si se adjunta una imagen sólo podrá contener un elemento). Cada elemento tendrá estos campos:

- **id:** Identificador del mensaje. Este identificador debe ser generado por el cliente y debe identificar a este mensaje de forma única y global. Este campo no puede ir vacío y debe ser numérico.
- **text:** Texto del mensaje tal y como le debe llegar al destinatario. Puede incluir emoticonos indicando directamente el carácter unicode correspondiente con la tabla emoji, adoptada en muchas fuentes. Este campo no puede ir vacío.
- **response:** Si se indica un 1 se solicitará respuesta al destinatario, si se indica un 0 no se solicitará respuesta. Las respuestas pueden ser consultadas mediante el comando *getmessages*.
- **contacts:** Array con los nombres de usuario de los destinatarios de este mensaje. Los nombres de usuario pueden ser obtenidos mediante el comando *getcontacts* a partir del teléfono móvil o email de cada destinatario.

Si se envía una imagen deberá adjuntarse el contenido de la misma en la petición POST con filename asignado a "uploadedfile". En este caso el array de mensajes sólo podrá tener un elemento.

### Salida:

- **id:** Identificador global del envío. Este identificador es generado por el sistema al realizar el envío y permitirá al cliente identificarlo para poder consultar su estado posteriormente, mediante el comando *getdeliverystatus*. La entrega de mensajes se realizará en segundo plano para evitar que el cliente tenga que esperar a la finalización del mismo. Este dato permite posteriormente consultar el estado del mismo.



## Ejemplo

Petición:

```
{
  key: '123456789',
  command: 'send'
  data: [
    {
      id: 1,
      text: 'Este es un mensaje',
      response: 1,
      contacts: ['usuario.tokapp.es', 'otrousuario.tokapp.es']
    },
    {
      id: 2,
      text: 'Otro mensaje',
      response: 0,
      contacts: ['xxllmm.tokapp.es']
    }
  ]
}
```

Respuesta:

```
{
  result: 0,
  message: "",
  data: {
    id: '3463nneu73474367'
  }
}
```



Command: `getdeliverystatus`

Descripción: Consulta el estado de un envío realizado con el comando `send`.

Entrada:

- `id`: Identificador del envío a consultar devuelto previamente por el comando `send`.

Salida:

- `completed`: Si devuelve un 1 significa que el envío ha sido completado. En caso de devolver un 0 indica que el envío está en curso.
- `problems`: Array con los posibles problemas detectados durante el envío. Cada elemento tendrá estos campos:
  - `id`: Identificador del mensaje proporcionado por el cliente al ejecutar el comando `send`.
  - `contact`: nombre de usuario del destinatario relacionado con el problema o vacío si es irrelevante.
  - `error`: Código de error registrado según la tabla de errores.
  - `message`: Descripción en inglés del problema detectado.
- `delivered`: Lista con las horas de entrega para cada destinatario. Si un destinatario no está presente en esta lista significa que aún no se le entregó el mensaje. En ese caso se deberá llamar a este comando más tarde. Cada elemento contiene estos campos:
  - `id`: Identificador del mensaje proporcionado por el cliente al ejecutar el comando `send`.
  - `username`: Nombre de usuario del destinatario del mensaje a quien ha sido entregado.
  - `delivered`: Fecha/hora de entrega del mensaje.



Ejemplo:

Petición:

```
{  
  key: '123456789',  
  command: 'getdeliverystatus',  
  data: {  
    id: '3463nneu73474367'  
  }  
}
```



Salida:

```
{
  result: 0,
  message: "",
  data: {
    completed: 1,
    problems: [
      {
        id: 2,
        contact: 'xxllmm.tokapp.es',
        error: 4,
        message: 'Invalid username'
      }
    ],
    delivered: [
      {
        id: 1,
        username: 'usuario.tokapp.es',
        delivered: '2016-02-29T11:32:00+00:00'
      }
    ]
  }
}
```



Command: `getmessages`

Descripción: Consulta los mensajes recibidos en la cuenta del remitente.

Entrada: sin campo *data*.

Salida: Array con los mensajes recibidos. Cada elemento tendrá estos campos:

- `time`: Fecha/hora UTC de recepción del mensaje por el sistema. Vendrá en formato estándar: `yyyy-mm-dd hh:mm:ss`
- `sender`: nombre de usuario en TokApp del remitente del mensaje.
- `text`: Texto del mensaje.
- `responseld`: Si este mensaje es en respuesta a uno enviado previamente este campo contendrá el identificador del mensaje que se corresponde con el campo *id* del mensaje en el comando *send*.



Ejemplo:

Petición:

```
{  
  key: '123456789',  
  command: 'getmessages'  
}
```

Respuesta:

```
{  
  result: 0,  
  message: "",  
  data: [  
    {  
      time: '2014-01-25 17:04:36',  
      sender: 'usuario.tokapp.es',  
      text: 'Mensaje de prueba',  
      responseld: ""  
    },  
    {  
      time: '2014-01-25 17:05:14',  
      sender: 'usuario.tokapp.es',  
      text: 'Mensaje de prueba',  
      responseld: '1'  
    }  
  ]  
}
```





Command: setlogo

Descripción: Asigna un nuevo logotipo al remitente. Este será el logotipo que verán los usuarios en sus dispositivos.

Entrada: sin campo *data*. Se adjuntará un archivo de imagen. Puede tener cualquier tamaño y formato.

Salida: Nada.

Ejemplo:

Petición:

```
{  
    key: '123456789',  
    command: 'setlogo'  
}
```

Respuesta:

```
{  
    result: 0,  
    message: ""  
}
```



Command: setname

Descripción: Asigna un nuevo nombre al remitente. Este será el nombre que verán los usuarios en sus dispositivos.

Entrada: Nuevo nombre a asignar.

Salida: Nada.

Ejemplo:

Petición:

```
{
  key: '123456789',
  command: 'setlogo',
  data: {
    name: 'Nuevo Nombre'
  }
}
```

Respuesta:

```
{
  result: 0,
  message: ""
}
```



## Códigos de error

Código	Significado
0	Operación correcta.
1	Comando inválido. Revise esta documentación para saber cuáles son los comandos permitidos.
2	Clave API inválida. Asegúrese de indicar la clave correcta suministrada por su proveedor en el parámetro "key".
3	Función no permitida. Su licencia no le permite usar esta función. Utilice el comando <i>status</i> para saber las limitaciones de su licencia. Consulte con su proveedor para ampliar sus limitaciones si es necesario.
4	Nombre de usuario no válido. Indica que no existe ningún usuario de TokApp con el nombre de usuario especificado. Compruebe que el nombre de usuario es correcto.
5	Límite de contactos superado. Se ha superado el límite de contactos distintos a los que se puede enviar mensajes en el mes natural según su licencia. Puede consultar cuál es ese límite y el número de contactos diferentes a quienes se envió mensajes mediante el comando <i>status</i> .
6	Límite de mensajes diarios superado. Se ha superado el límite de mensajes que se pueden enviar en un día natural según su licencia. Puede consultar cuál es ese límite y el número de mensajes enviados hoy con el comando <i>status</i> .
7	Más de un mensaje con imagen adjunta. Se ha adjuntado un archivo de imagen pero se indica más de un mensaje a enviar y esto no es posible. Envíe un único mensaje (aunque sea a varios destinatarios) si adjunta una imagen.



Código	Significado
8	Falta identificador de mensaje. Alguno de los mensajes a enviar no tiene asignado el identificador. Revise la documentación del comando <i>send</i> .
9	Mensaje sin texto. Debe indicar obligatoriamente el texto de cada mensaje.
10	Sin destinatarios. Uno o varios de los mensajes que va a enviar no tienen ningún destinatario asignado.
11	Sin mensajes. No se ha incluido ningún mensaje en la petición de envío.
12	Envío no encontrado. El identificador de envío especificado no existe. Consulte la documentación de los comandos <i>send</i> y <i>getdeliverystatus</i> para más información.
13	No se permiten contactos vacíos. Alguno de los mensajes contiene un contacto vacío. Consulte la documentación del comando <i>send</i> .
14	Este comando requiere un fichero. Indica que no se ha adjuntado un fichero a la petición y el comando lo requiere. Véase la documentación del comando para mayor información.
15	El parámetro <i>name</i> es obligatorio. Véase la documentación del comando para mayor información.
999	Error inesperado. Se trata de un problema eventual o que no está previsto. Inténtelo de nuevo más tarde. Si el problema persiste, póngase en contacto con su servicio de soporte técnico.
1000	Función no implementada.