TokApp API

Documentación

Versión 1.3 – 29/02/2016

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

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

Table of contents

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

Service description

This Web service will allow you doing this [1]:

- Check if one or more users are registered in TokApp from their phone numbers or emails, and get their user-names.
- Send text messages, attaching image files in each delivery (even sending to many users).
- Check delivery status.
- Check and retrieve incoming messages.

All petitions will be done to the same URL, assigned by your provider, and all their parameters will be sent by POST petition in JSON format with UTF-8 encoding. All responses will be received in JSON format too with same encoding.

The server will process either HTTP or HTTPS petitions, leaving to developer concern using one or the other security level, but it is recommended that allways use HTTPS protocol in production environments, using no encrypted communications in developing or testing phases or when sender device does not allow SSL security.

All petitions require sending the API Key provided, that will identify message sender and license.

(1) May vary due your license limitations

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

Petition syntax

All petitions will receive data in JSON format with this structure:

```
{
        key: '<API Key>',
        command: '<Command>',
        data: {<Parameters>}
}
```

key field must provide the API Key that service provider will send to you.

command field will contain the command name to execute, in accordance with command list that will be detailed following in this document.

data field will be another JSON structure with input parameters specific to each command, as detailed in their description. This field may be omitted if the command does not need it.

Response syntax

All responses will have this JSON structure:

```
{
        result: <Error code>,
        message: '<Descriptive message>',
        data: {<Returned data>}
}
```

result field will return a numeric code that will identify the error in case of failure or 0 if operation was completed without errors. The possible error codes are described later in this document.

In case of failure, the *message* field returns a problem description in English.

In case of success, *data* field will contain in JSON format the data returned by the petition, as explained in the executed command description. If command is not returning additional information, this field will be omitted in response.

Command descriptions

Command: status

Description: Gets license status and actual limitations.

Input: Without *data* field.

Output:

- maxContacts: Maximun number of different contacts that you can send messages within the actual natural month.
- maxMessages: Maximun number of messages that you may send in the actual natural day. A zero value indicates that there are no limit.
- contactsSent: Number of different contacts at which was sent messages in the actual natural month.
- messagesSent: Number of messages sent today.
- multiSend: Shows by 0 (false) or 1 (true) if it is possible to send more than one message per petition by *send* command.
- getContacts: Shows by 0 (false) or 1 (true) if *getcontacts* command is available.
- available: Shows by 0 (false) or 1 (true) if service is available.

Example:

Petition:

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

Response:

```
{
    result: 0,
    message: '',
    data: {
        maxContacts: 500,
        maxMessages: 0,
        contactsSent: 18,
        messagesSent: 216,
```

```
            multiSend: 1,
            available: 1
        }
    }
```

Command: getcontacts

Description: Gets TokApp user-names of one or more contacts from their phone number or email. It's recommended making this call periodically to discover TokApp users because each day more users may be registered.

Input:

- phones: Array with phone numbers to check.
- emails: Array with emails to check.

Output: It will return an array with this fields in each element:

- phone: Phone number as sent in petition.
- email: Email address as sent in petition.
- username: Corresponding user-name.

Example:

Petition:

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

Response:

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

Phone numbers or emails not corresponding TokApp users will be no returned in response. If no data match a TokApp user an empty array will be returned but no error will be generated.

Command: send

Description: Sends one or more messages to one or more contacts. This petition allows sending an image, with a text to one or more contacts. The possibility to sending multiple messages in same petition will vary depending on your license limits and may be checked with parameter *multiSend* in *status* command. If an image file is attached, only one message may be sent in same petition even sending to more than one contact.

Input: Array with messages to send (if an image is attached only one element will be accepted in this array). Each element will contain this fields:

- id: Message ID. This ID must be generated by client and must identify this message in a unique and global way. This field may not be empty and must be a numeric value.
- text: Message text as it must be arrived to contact. May include emoticons including the corresponding unicode character with emoji table, implemented in many fonts. This field may not be empty.
- response: If goes with a value of 1 a response will be solicited to contact, if 0 is specified, no response will be solicited. Responses may be queried with *getmessages* command.
- contacts: Array with contact's user-names. User-names may be obtained from contact's phone numbers or emails with *getcontacts* command.

If an image is attached, the file contents must be included in POST petition with filename field assigned to “uploadedfile”. In this case the messages array may contain only one element.

Output:

- id: Global delivery identifier. This ID is generated by the system when delivery is sent and will allow to client identifying it to query it's status later with the *getdeliverystatus* command. The message delivery will be performed in background to avoid client waiting for process to end. This field allows querying later it's status.

Example:

Petition:

```
{
    key: '123456789',
    command: 'send'
    data: [
        {
            id: 1,
            text: 'This is a message',
            response: 1,
            contacts: ['usuario.tokapp.es', 'otrousuario.tokapp.es']
        },
        {
            id: 2,
            text: 'Another message',
            response: 0,
            contacts: ['xxllmm.tokapp.es']
        }
    ]
}
```

Response:

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

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

Command: getdeliverystatus

Description: Queries a delivery status.

Input:

- id: Delivery identifier returned previously by *send* command.

Output:

- completed: If 1 is returned delivery was copleted. If 0 is returned sending is in course.
- problems: Array with possible problems detected. Each element will contain this fields:
  - id: Message ID provided by client in *send* command.
  - contact: Contact's user-name associated with problem or empty if it's irrelevant.
  - error: Error code registered as detailed on error codes table.
  - message: Problem description in English.
- delivered: Delivered times list for every contact. If one contact is not present in this list, it means that message was not delivered yet. Then, you must call this command later. Each item has following fields:
  - id: Message ID provided by client in *send* command.
  - username: Contact's username.
  - delivered: Date/time in which the message was delivered.

Example:

Petition:

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

Output:

```
{
        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’
                        }
                ]
        }
}
```

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

Command: getmessages

Description: Queries incoming messages received.

Input: Without *data* field.

Salida: Array with received messages. Each element will contain this fields:

- time: UTC date/time in which message was received by system. It comes in standar format: yyyy-mm-dd hh:mm:ss
- sender: Sender's TokApp user-name.
- text: Message text.
- responseId: If this message is a response to another one previously sent to this contact in which response was solicited, this is the message ID provided in *send* command.

Example:

Petition:

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

Response:

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

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

Command: setlogo

Description: Sets a new logo to sender. This is the logo that users will see in their devices.

Input: without *data* field. An image file must be attached as uploaded file. Any image format and size are accepted.

Output: Nothing.

Example:

Petition:

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

Response:

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

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

Command: setname

Description: Sets a new name to sender. This is the name that users will see in their devices.

Input: New name to set.

Output: Nothing.

Example:

Petition:

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

Response:

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

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

Error codes

Código	Significado
0	Process completed successfully.
1	Invalid command. Check this documentation to see which commands are accepted.
2	Invalid API key. Ensure that you specify correct API key provided in *key* parameter.
3	Invalid function. Your license does not allow you perform this operation. Use *status* command to know your license limitations. Contact your service provider if you need extend your license features.
4	Invalid user-name. No TokApp user was found corresponding to specified user-name. Ensure the user-name is valid.
5	Contact's limit reached. The number of different contacts at which you may send messages in the actual natural month was reached. You may query this limit and the actual different contacts sent actually by *status* command.
6	Messages limit reached. The number of messages that may be sent today was reached. You may query this limit and actual number of messages sent by *status* command.
7	More than one message with image attached. An image was attached but more than one message was included in this petition.
8	Missed message ID. At least one of messages goes with no ID specified.
9	Missed message text. At least one of messages goes with no text.
10	No contacts. At least one of messages goes with no contacts.

Código	Significado
11	Without messages. There's no messages included in petition.
12	Delivery not found. Specified delivery ID was not found.
13	Empty contacts not allowed. Some message has a contact that is empty. See *send* command syntax for more information.
14	This command requires a file. It tells that there is not attached file in the request and the command requires it. See command documentation for more information.
15	The name parameter is required. See command documentation for more information.
999	Unexpected error. An eventual problem was detected. Tray it again. If problem persists contact your support service for assistance.
1000	Not implemented function.