

Text Request APIs

Last updated 11-6-2017

Overview

Text Request offers two separate APIs: A REST API, and Realtime WebHooks.

The REST API allows you to:

- Automate sending SMS messages through the Text Request system.
- Create new contacts.

The REST API is implemented as RESTful web service that can be called from any language that can make HTTP requests.

WebHooks are callbacks that Text Request makes to a HTTP endpoint whenever an event that you're interested in occurs. Text Request currently has WebHooks for the following events:

- When a message is received.
- When a message is sent.
- When a new contact is created.

The Text Request WebHooks and REST API is not available for trial accounts.

REST API

To work with the Text Request REST API, you will need your account's API key. Your API key is a 32-character string that must be included with all web service calls. Administrators for your account can retrieve the API key by doing the following:

1. Log into <https://www.textrequest.com> as an administrator.
2. Click on the Integrations menu item.
3. Click on the API tab. Your account's API key will be listed under the REST API section (See Figure 1 below).

Figure 1: Retrieve your account’s API key by clicking on the Integrations menu item, and the API tab.

Sending a Message

The sendmessage endpoint allows you to send a SMS message through Text Request. This is an excellent way to send automatic notifications from your line of business system, and then allow your team to respond to any replies using the Text Request web application.

All outbound messages sent through the REST API will appear in recipients’ message threads, but will not appear in the main Queue. This is because you likely don’t want your queue cluttered with automatic messages such as appointment reminders and payment requests.

Endpoint: <https://www.textrequest.com/api/v1/sendmessage>

HTTP Verb: POST

Request Parameters

Variable	Required	Description
apikey	Yes	This is the API key that uniquely identifies your account. Please keep this safe – if someone discovers your API key, they will be able to send messages on your behalf.

message	Yes	This is the message that should be sent. There is no limit on the number of characters. Please URL encode the message before sending to prevent issues with spaces and other punctuation.
recipientphonenumber	Yes	The recipient to receive the message. This should be a long format phone number preceded by the country code. For US numbers, this would be a 1 (country code) followed by a three-digit area code, followed by a seven-digit phone number. Example: 14236679106. Note: please strip all punctuation from the RecipientPhoneNumber before sending a request to the API.
sourcephonenumber	Yes	The phone number that the message will be sent from. This must be a phone number that belongs to your account. Like the RecipientPhoneNumber, please precede the SourcePhoneNumber with a country code, and remove all non-numeric punctuation before sending a request to the API.

Responses

HTTP Response Code	Description
200 Ok	The message was sent successfully.
400 Bad Request	There was an error with one or more of the parameters sent. Please check the ErrorMessage property for a detailed description of errors.
401 Unauthorized	A 401 Unauthorized response can occur for several reasons. These include if the APIKey provided is invalid, if your account is delinquent, or if your account has been administratively locked due to violating the terms of service. If you receive a 401 Unauthorized response, please check the ErrorMessage property of the response for details.

Example

The following example shows a screenshot of an API call using Fiddler4:

The screenshot shows the Fiddler4 interface with the following details:

- Method:** POST
- URL:** <https://www.textrequest.com/api/v1/sendmessage>
- Protocol:** HTTP/1.1
- Request Headers:**
 - User-Agent: Fiddler
 - Content-Type: application/x-www-form-urlencoded
 - Host: www.textrequest.com
 - Content-Length: 116
- Request Body:**

```
APIKey = 12345678901234567890123456789012&Message = Hello&RecipientPhoneNumber = 14236679106&SourcePhoneNumber = 14238059990
```

*Note that the API key in this example is fictitious – you will need to substitute your own API key that you receive from Text Request.

All Text Request RESTful API endpoints are case-insensitive for both web service URLs and parameters.

Creating a New Contact

The Create a New Contact API endpoint allows you to create a new contact for one of the text numbers in your Text Request account. Note that a contact is defined as a recipient phone number and one of your text numbers. For example, assume you had 423-867-5309 and 423-839-8464 as text numbers in your Text Request account. You are texting with a recipient whose phone number is 212-777-7777. If you wanted this recipient to be listed as a contact for both phone numbers, you would need to create two separate recipients – one for each of your text numbers.

Using the Create a New Contact API is a great way to keep prospects in your CRM software and contacts in Text Request in synch.

Endpoint: <https://www.textrequest.com/api/v1/contacts>

HTTP Verb: PUT

Request Parameters

Variable	Required	Description
apikey	Yes	This is the API key that uniquely identifies your account. Please keep this safe – if someone discovers your API key, they will be able to send messages on your behalf.
textrequestphonenumber	Yes	This is the eleven-digit long format phone number in your Text Request account that the contact should be listed under.
contactphonenumber	Yes	The eleven-digit long format phone number for the contact that you are adding.
friendlyname	No	This is an alias that is displayed throughout Text Request for this contact. For example, instead of viewing 1-423-218-0111 in your Address Book and Queue, you could supply a friendly name of “Text Request”. If a contact already exists but has a different friendlyname, the friendly name will be updated if it is specified.

Responses

HTTP Response Code	Description
200 Ok	The contact was created successfully. This web service also returns HTTP 200 if the contact already exists.
400 Bad Request	If any required fields were omitted, this web service endpoint will return HTTP 400 Bad Request. This method will also return HTTP 400 Bad Request if the textrequestphonenumber or contactphonenumber are not valid eleven-digit long format phone numbers.
401 Unauthorized	A 401 Unauthorized response will be returned if any of the following conditions are met: <ul style="list-style-type: none">• The apikey is invalid.

	<ul style="list-style-type: none"> • The apikey is valid, but the account is delinquent, is a trial account, or has been administratively locked for violating the Text Request terms of service. • The apikey is valid, but the textrequestphonenumber specified doesn't belong to your account. <p>Please check the ErrorMessage property in the response for details if you receive a HTTP 401 Unauthorized response.</p>
--	--

Note: Creating a contact via this API endpoint will NOT trigger a Contact Created WebHook response.

WebHooks

WebHooks allows you to receive notification whenever certain events occur in your Text Request account. When you've defined one or more WebHooks, Text Request will call your WebHooks when events occur. WebHooks are text number-specific. Each Text Request text number in your Text Request account has its own set of WebHook definitions.

Currently, the events that are available for WebHook notifications are when a message is received, and when a message is sent.

Available WebHooks

The following WebHooks are currently available in Text Request.

Message Received WebHook

The Message Received WebHook is called whenever a Text Request text number receives an inbound message. This is useful if you'd like to record all inbound text messages, or want to perform custom processing when a message is received.

All Message Received WebHooks will be called via HTTP POST.

The fields below will be sent as part of the HTTP POST body. The fields will be in JSON format.

Variable	Description
From	This is the eleven-digit phone number that sent your Text Request phone number a message. Phone numbers are long-format phone number prepended with the country code (example: 14238499205).
To	This is your Text Request eleven-digit phone number that received the SMS message. This is a long format phone number prepended with the country code (example: 14236679107).
Message	this is a string containing the message that was received.
MessageDirection	This will always be set to "R" for Received.
MessageTimestamp	This is the timestamp that your message was received. Timestamps are in UTC time and ISO 8601 format. Specifically, this format is yyyy-MM-ddTHH:mm:ss Example: January 15, 1979 6:15:01AM would be represented as 1979-01-15T06:15:01.
SentByUser	This field will always be blank since the message was received.

CustomerName	This contains the friendly name entered for the customer who sent the message. If there is no friendly name entered, this field will be blank.
--------------	--

Message Sent WebHook

The Message Sent WebHook will be called whenever a SMS message is sent from a Text Request text number.

All Message Sent WebHooks will be called via HTTP POST.

The fields below will be sent as part of the HTTP POST body. The fields will be in JSON format.

Variable	Description
From	This is your eleven-digit Text Request phone number that sent the message. Phone numbers are long-format phone number prepended with the country code (example: 14238499205).
To	This is the eleven-digit phone number that your Text Request account sent the message to. This is a long format phone number prepended with the country code (example: 14236679107).
Message	This is a string containing the message that was sent.
MessageDirection	This will always be "S" for sent messages.
MessageTimestamp	This is the timestamp that your message was sent. Timestamps are in UTC time and ISO 8601 format. Specifically, this format is yyyy-MM-ddTHH:mm:ss Example: January 15, 1979 6:15:01AM would be represented as 1979-01-15T06:15:01.
SentByUser	This is the name of the user who sent the message. If a message was sent by the outbound API, this field will read "API".
CustomerName	This contains the friendly name entered for the customer who received the message. If there is no friendly name entered, this field will be blank.

Contact Created Webhook

The Contact Created WebHook fires each time a new contact is created in the Text Request system for a given Text Request phone number. This WebHook will fire when a new contact is created by:

- Sending a message to a phone number that you've never interacted with before.
- Receiving a message from a phone number that you've never interacted with before.
- When a new contact is manually created in the address book.
- When new contacts are bulk uploaded.

The Contact Created WebHook does NOT fire when a contact is added via the Create Contact RESTful API endpoint.

All Message Sent WebHooks will be called via HTTP POST.

The fields below will be sent as part of the HTTP POST body. The fields will be in JSON format.

Variable	Description
----------	-------------

ContactId	The ID of the newly created contact record. This is an integer, and is unique across all Text Request contacts.
ContactPhoneNumber	This is the eleven-digit long format phone number for the contact. It's a string value. Example: "14232180111".
LocationPhoneNumber	This is the eleven-digit long format phone number for your Text Request phone number that the contact belongs to. Example: "14235887471".
ContactFriendlyName	The friendly name of the newly created contact. If a friendly name was not entered, this field will be null.

Creating a WebHook

There are two steps to creating a WebHook. First, you must write the HTTP endpoint that will be called when a specific event occurs in Text Request. Second, you must register your new HTTP endpoint with Text Request.

Step 1: Create a WebHook Web Service

The first step is to create your RESTful web service endpoint that will receive a call when your WebHooks are triggered. This can be any valid URL, and you can use any programming language that will allow you to define a HTTP endpoint. When a WebHook that you define is triggered, Text Request will send a HTTP POST to your declared endpoint. The body of the post will be in JSON format and will contain the fields for the particular WebHook.

Step 2: Register Your WebHook with Text Request

After creating your inbound messaging webhook, you'll need to register the endpoint with Text Request.

To do so, you'll need to:

1. Log in to your Text Request account as an administrator.
2. Click on the Integrations menu option, then click on the API tab.
3. On the API tab, scroll down to the bottom of the page to the WebHooks section (See Figure 2 below).

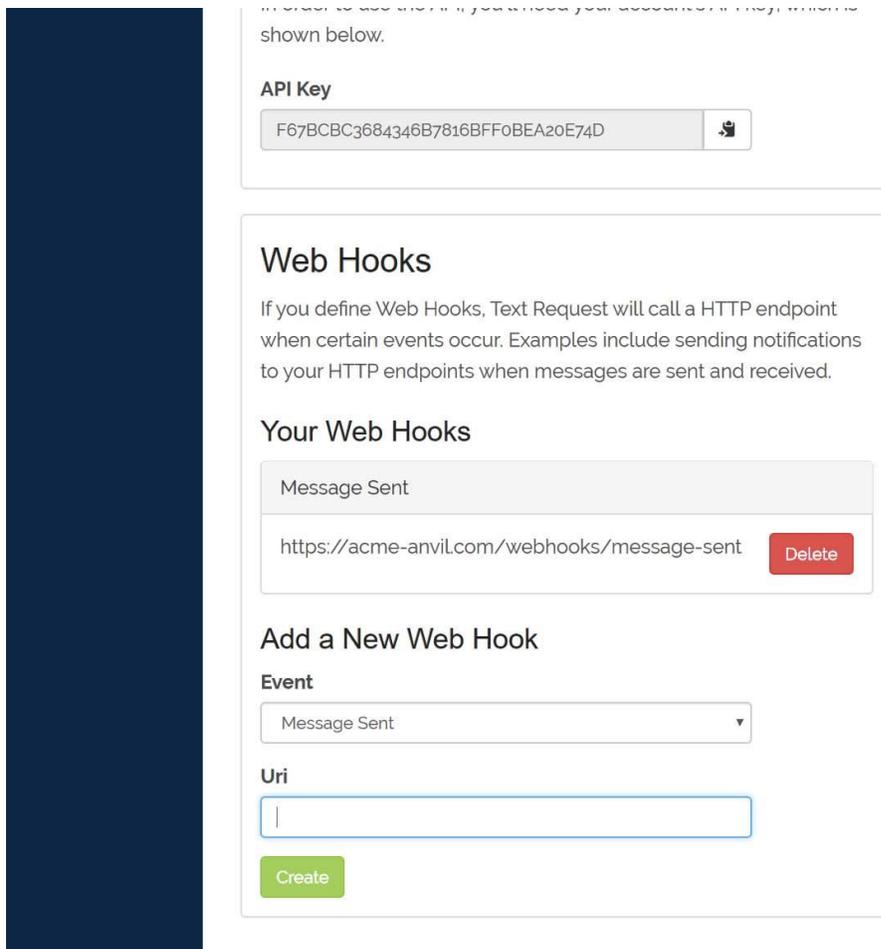


Figure 2: The Web Hooks section on the API screen.

To add a new WebHook, enter the URL of your RESTful web service in the Uri field. You'll also need to select the event type for when the WebHook will be called. You can choose between "Message Sent" and "Message Received". Finally, click the Create button to add your new WebHook.

To delete an existing WebHook, click the Delete button next to the WebHook that you'd like to remove.

You can have up to ten WebHooks defined per text number.

Realtime WebHook Example

Here's an example of a Message Received WebHook in action for the fictitious ACME Anvil Corporation.

- ACME Anvil Corporation has a Text Request text number of 1-423-610-9504.
- They have defined a web service at the URL of <https://www.acme-anvil.com/webservices/textrequest/realtime-sms>.
- They have registered the endpoint on the Text Request API screen as a new WebHook for the Message Received event.
- On the 1-423-610-9504 text number, there is currently a contact with the phone number 1-555-987-2837 and a friendly name of "Wile E. Coyote".

When a new text message of “I need help catching a Roadrunner” from the sender 1-556-987-2837 arrives on December 25, 2016 at 3:30PM at their Text Request phone number, Text Request will call the <https://www.acme-anvil.com/webservices/textrequest/realtime-sms> via HTTP POST. The POST body will be the following:

```
{ "From": "15569872837",  
  "To": "14236109504",  
  "Message": " I need help catching a Roadrunner",  
  "MessageTimestamp": "2016-12-25T153000",  
  "MessageDirection": "R",  
  "CustomerName": "Wile E. Coyote" }
```

Once your web service has successfully received a callback, please respond with a HTTP 200 status.

Error handling and retry

If your WebHook web service does not respond with a HTTP 200, Text Request will wait five minutes and contact your specified WebHook again. Text Request will make up to ten attempts to call your WebHook.

Testing

To test your WebHook, send a SMS message to the number that you registered a WebHook for. You should receive a callback at the specified URL within several seconds.