**How to use Omni Channel Engine:**

**Omni Channel Engine API Guide**

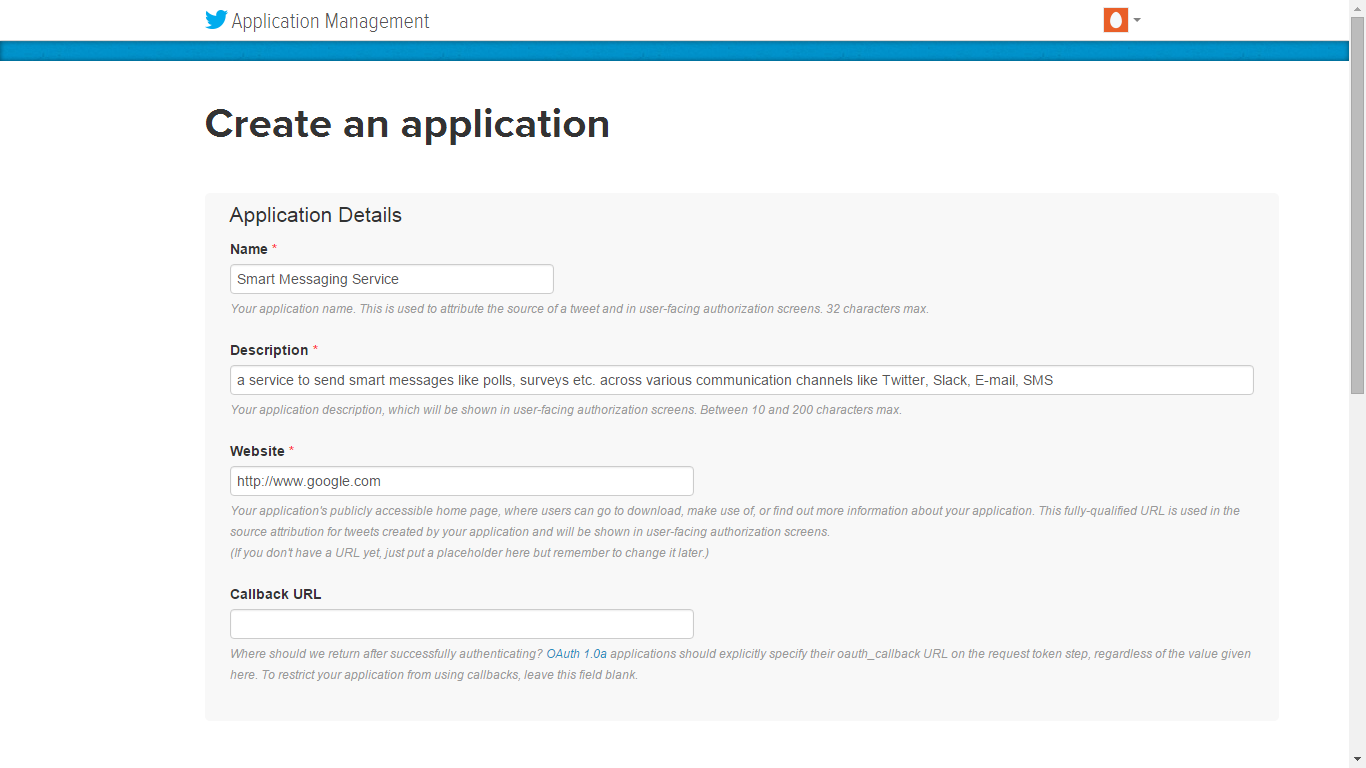
Omni Channel Engine is an API designed to send smart messages like polls, surveys etc. across various communication channels like Twitter, Slack, E-mail, SMS.

In order to use the API , you will need various things like an API Key and credentials for the channels you want to send the message through. This guide will help you to understand where you can get the things required to start using the API.

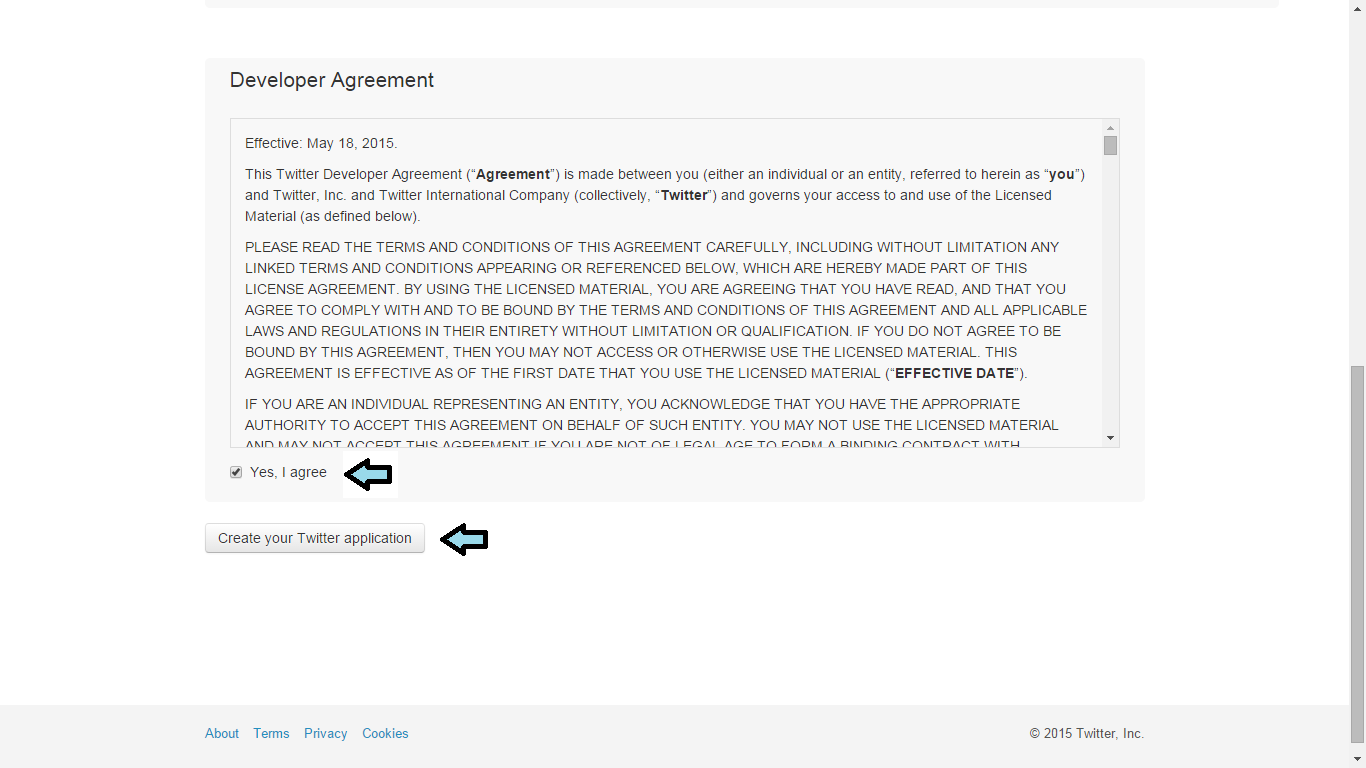
**In Twitter :**

In order to send a smart message through Twitter, you will need 4 parameters. To get these 4 things you will need to create a Twitter app.

1. First go to <https://apps.twitter.com/> and log in if necessary.
2. Now click on **Create New App** button.
3. Enter your Application Name, Description and your website address. You can leave the callback URL empty. (If you do not have a website address , you can use a **temporary** placeholder like “http://www.google.com”)

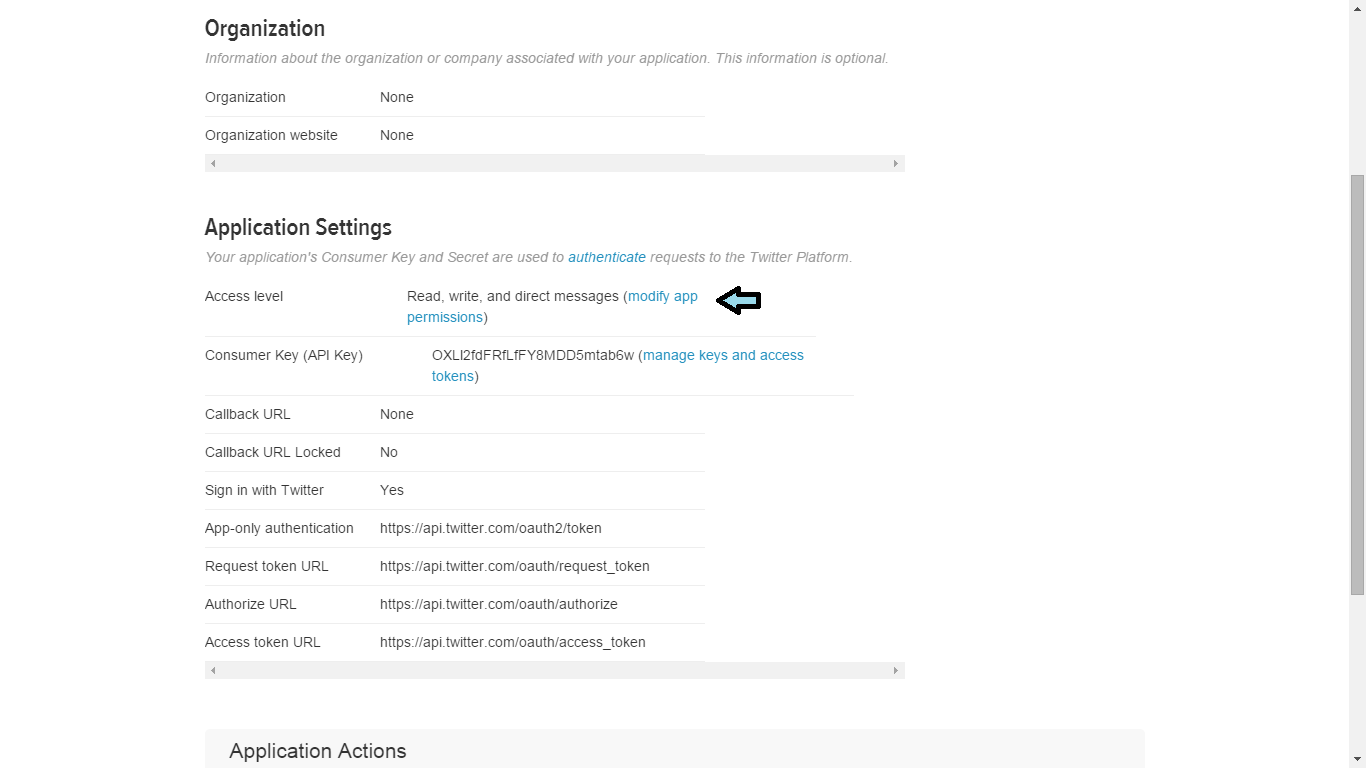


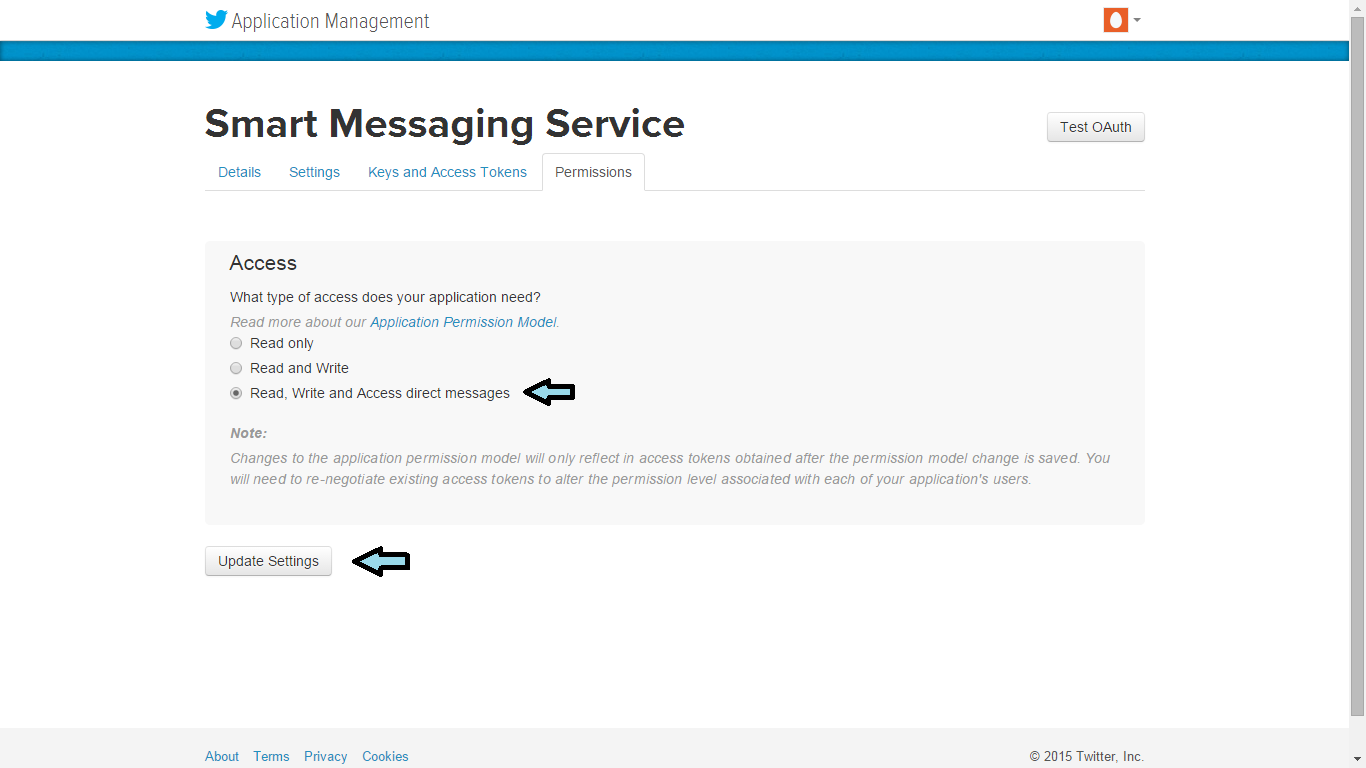
1. Accept the **Developer Agreement** and click on **Create your Twitter application** button.



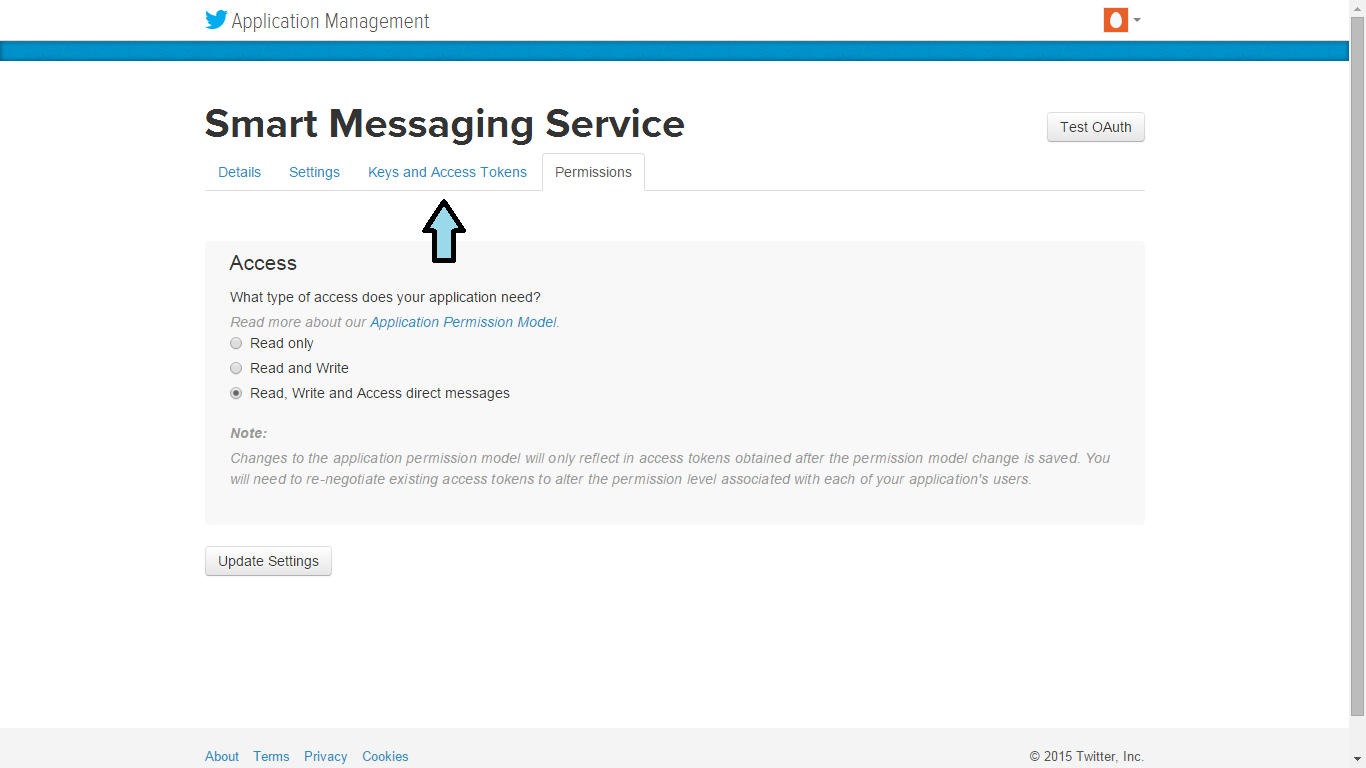
1. Now on app-details page, go to **modify app permissions** and update the permissions to

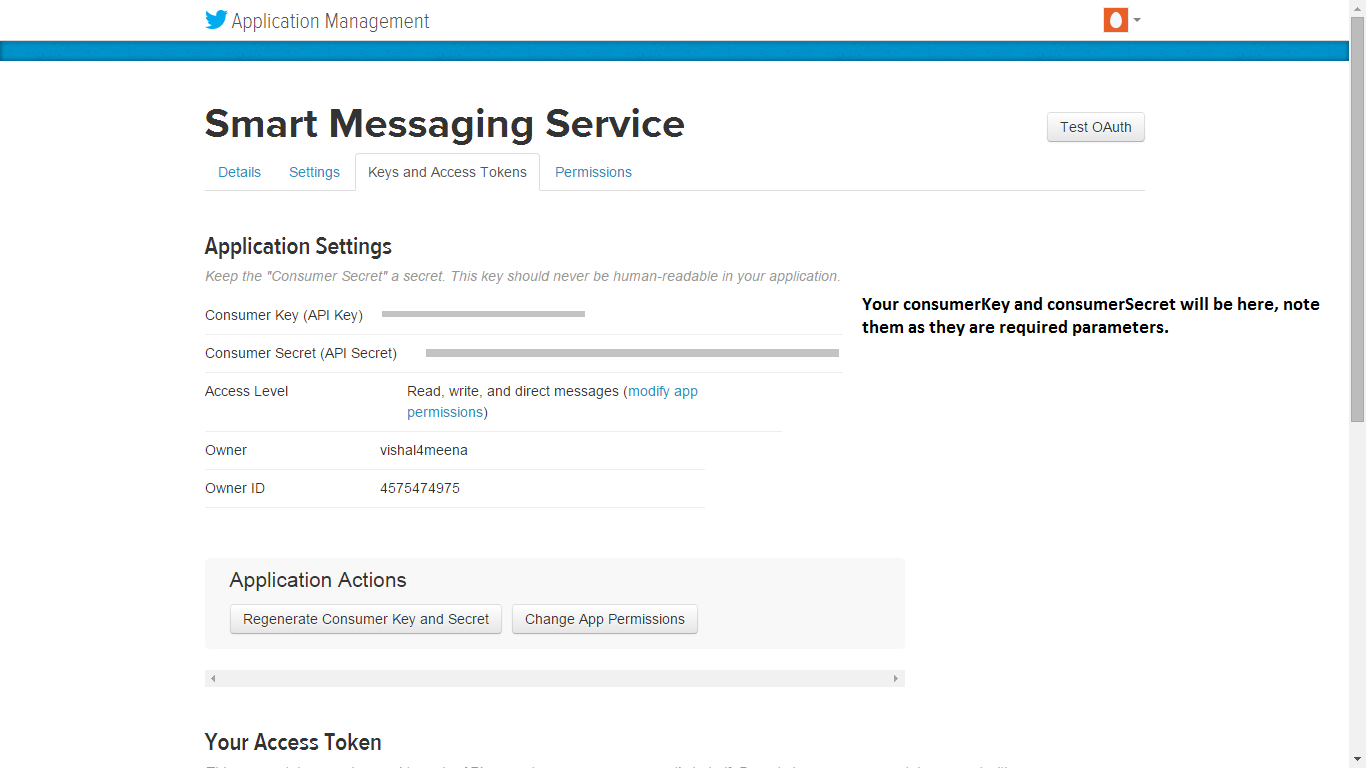
**Read, Write and Access direct messages**

****

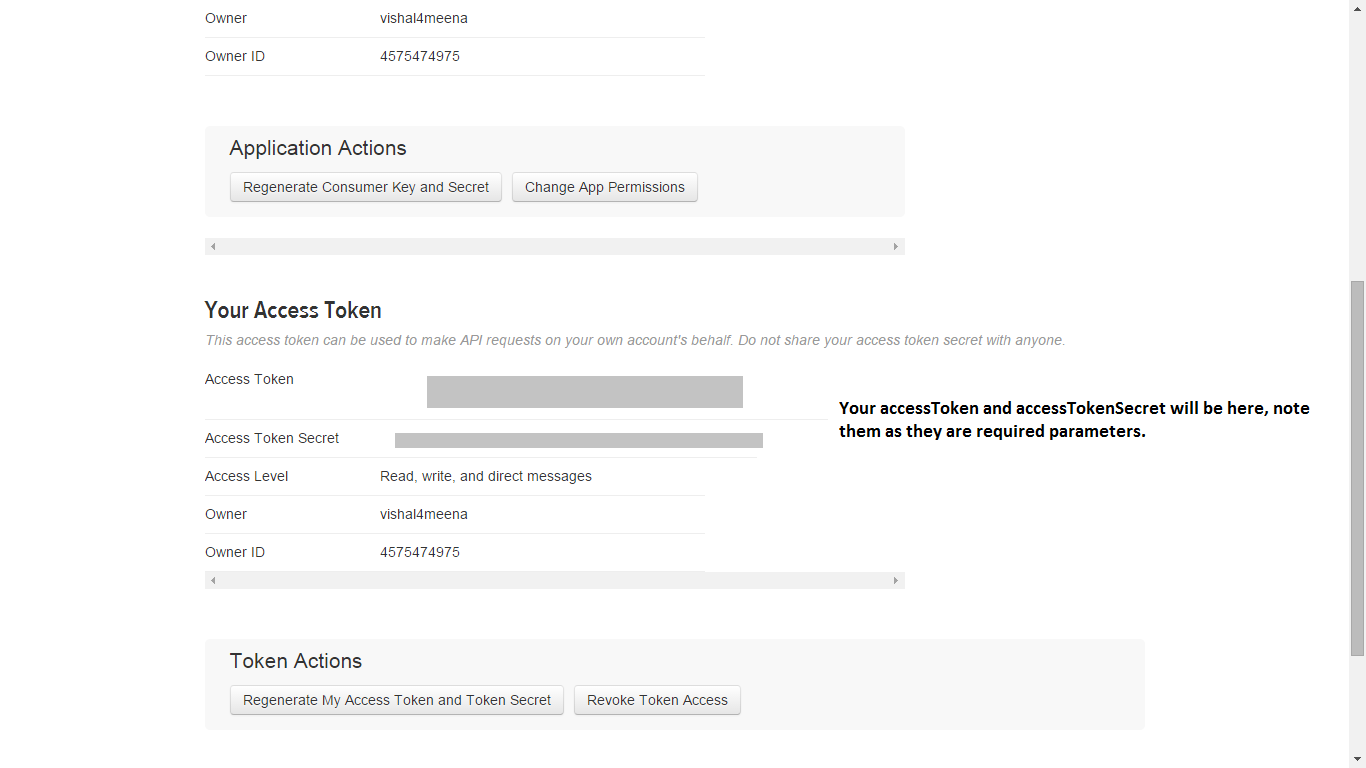
****

1. Now click on **Keys and Access Tokens** to go keys page and note down consumer Key and consumer Secret.





1. Now click on **Create my Access Token** and note your access Token and access Token Secret.



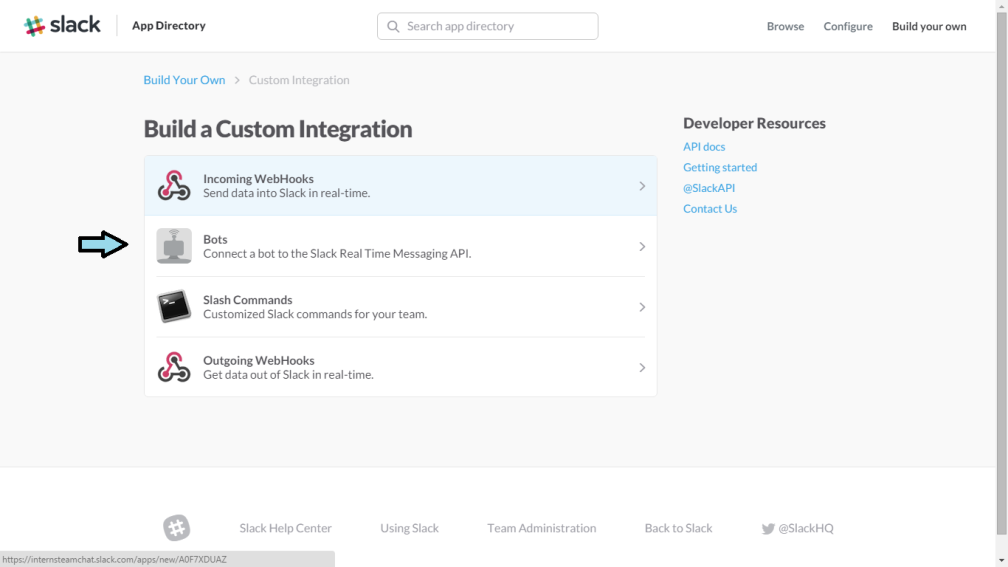
1. Now you have the four authentication parameters (consumerKey, consumerSecret, accessToken, accessTokenSecret) associated with your twitter account (app).

**In Slack :**

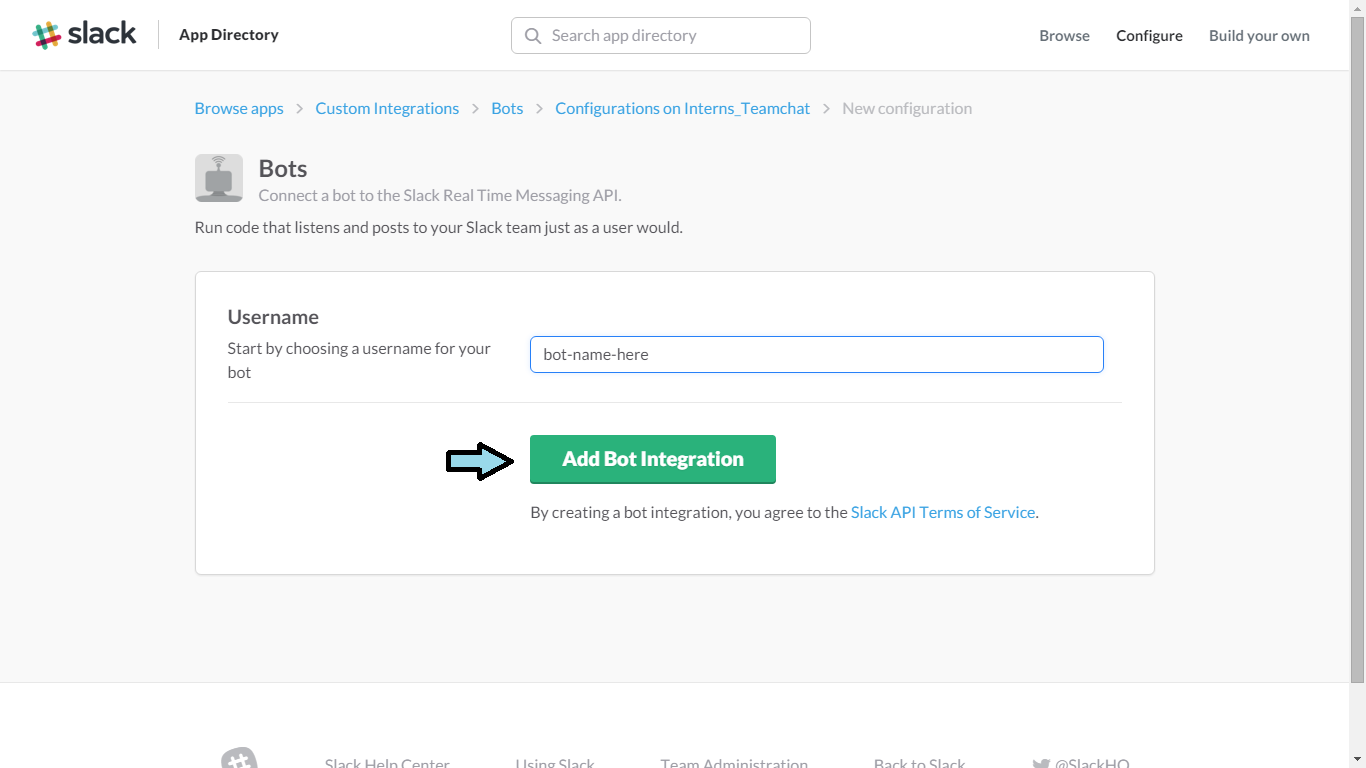
In order to send a smart message through Slack, you will a bot user integration and its corresponding bot Token. You can also view [Slack's bot-user guide](https://api.slack.com/bot-users) for information.

To create your own bot in your team follow the steps below.

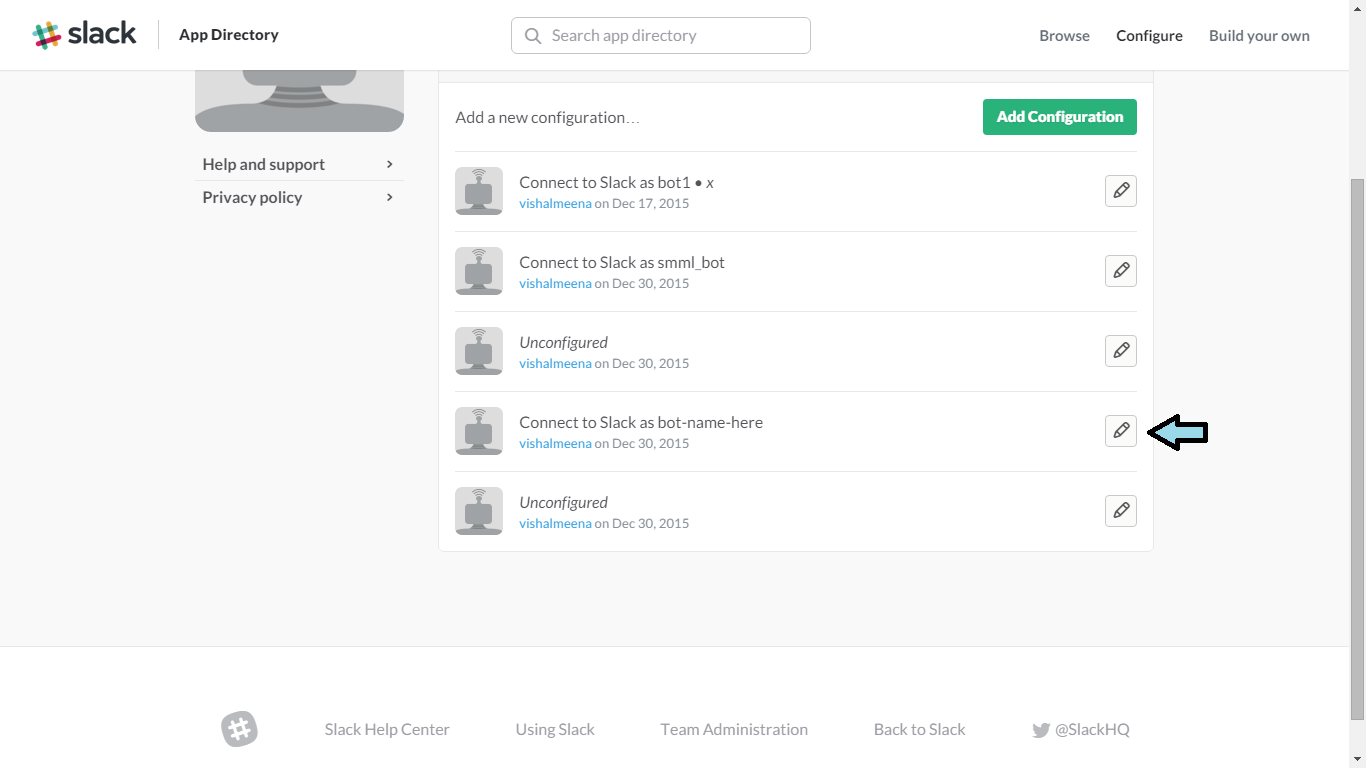
1. Go to <https://slack.com/apps/build/custom-integration> and log in if necessary.



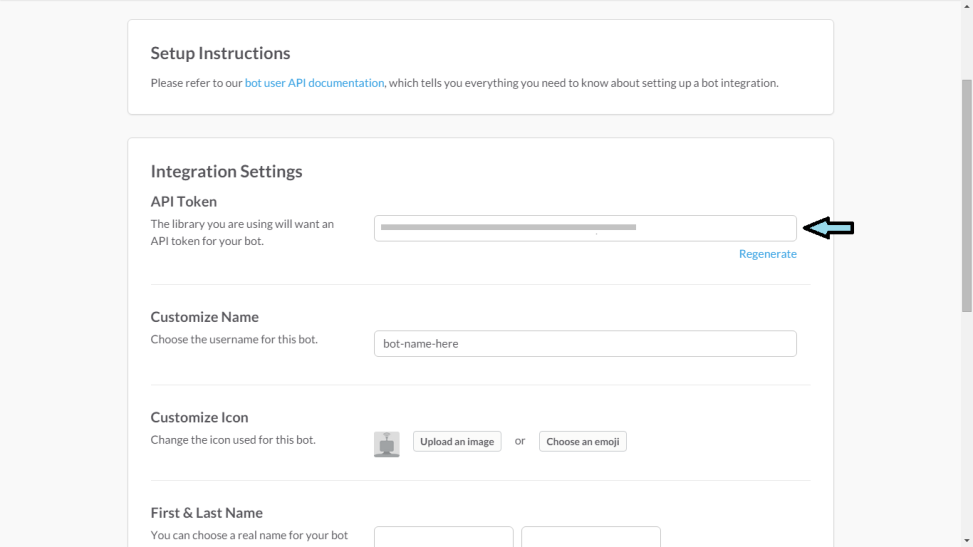
1. Click on Bots and give a name to your bot-user.



1. Now, go to <https://slack.com/apps/manage/custom-integrations> and click on Bots again.
2. Now, click on **edit configuration** button next to the bot you just created.

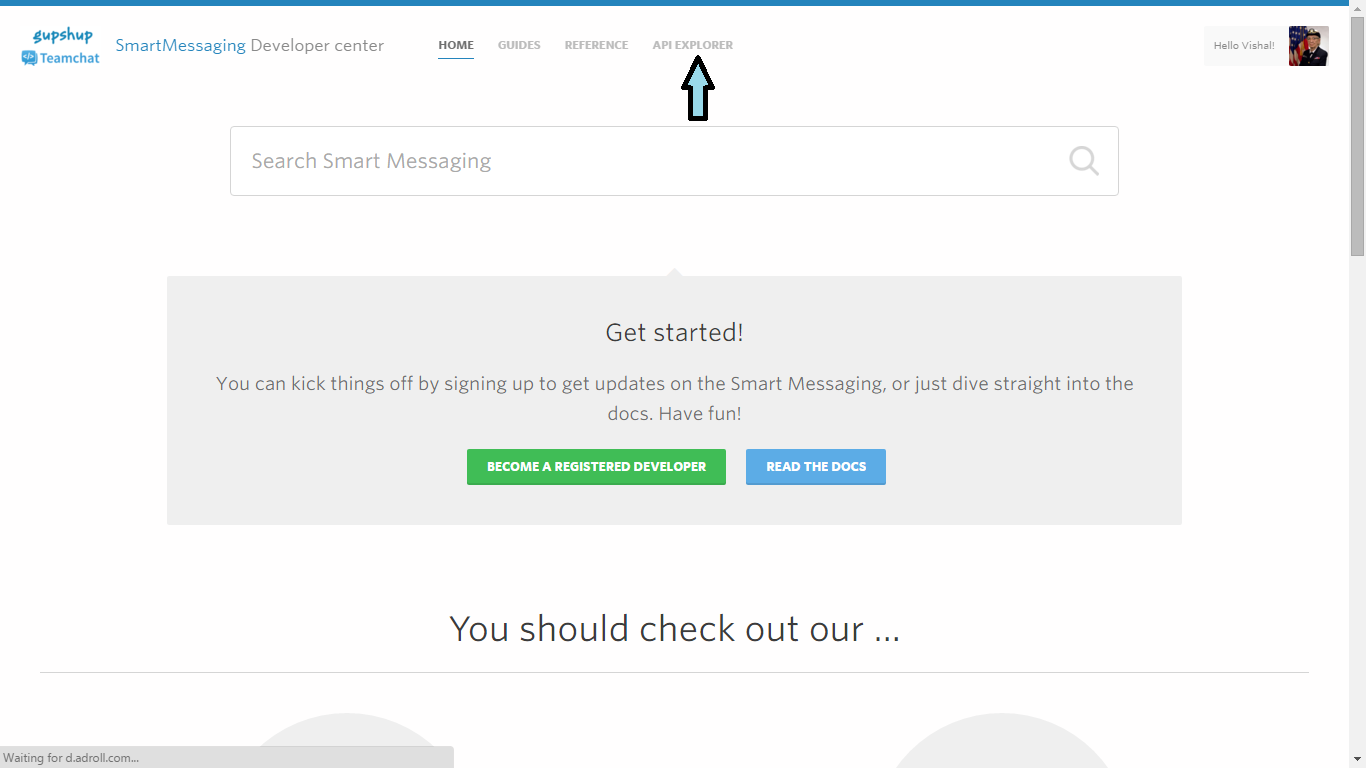


1. Note down the apiToken , as this is the only authentication parameter needed for Slack.

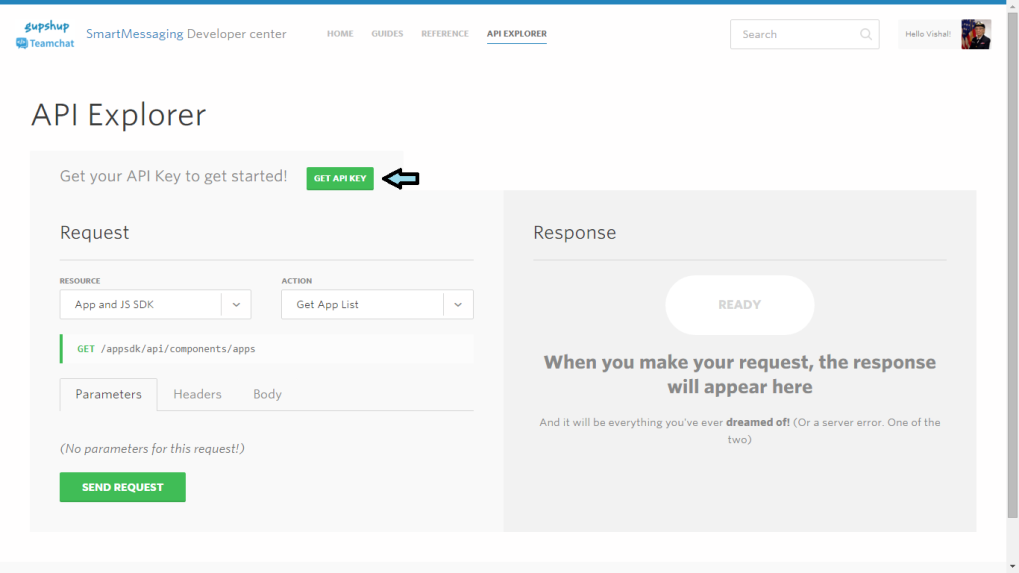


**Getting the API Key :**

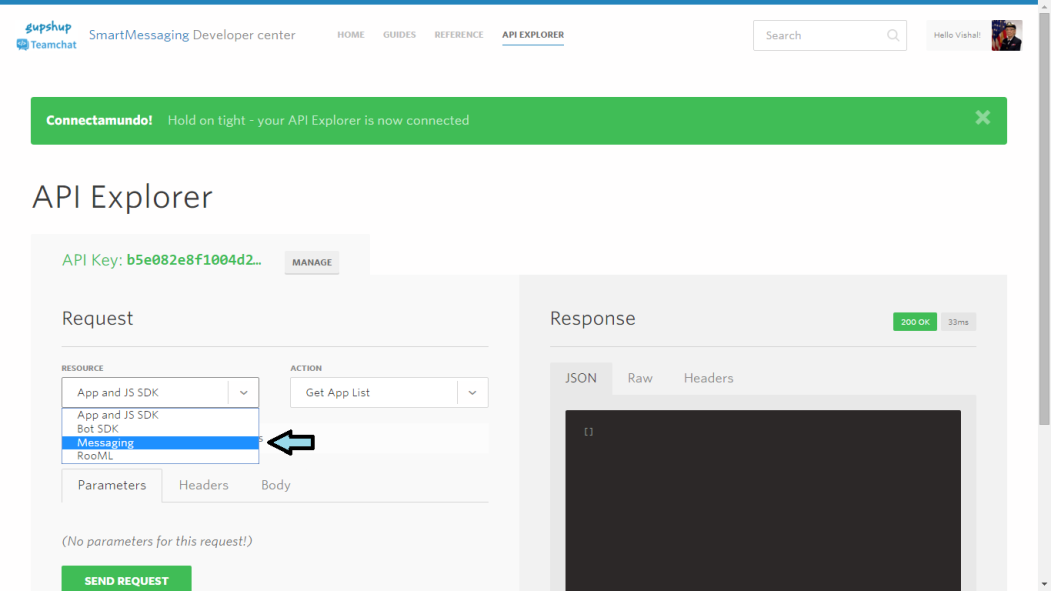
1. Go to <https://www.smartmessaging.io> and login (sign-up if necessary).
2. Now, click on **API Explorer .**

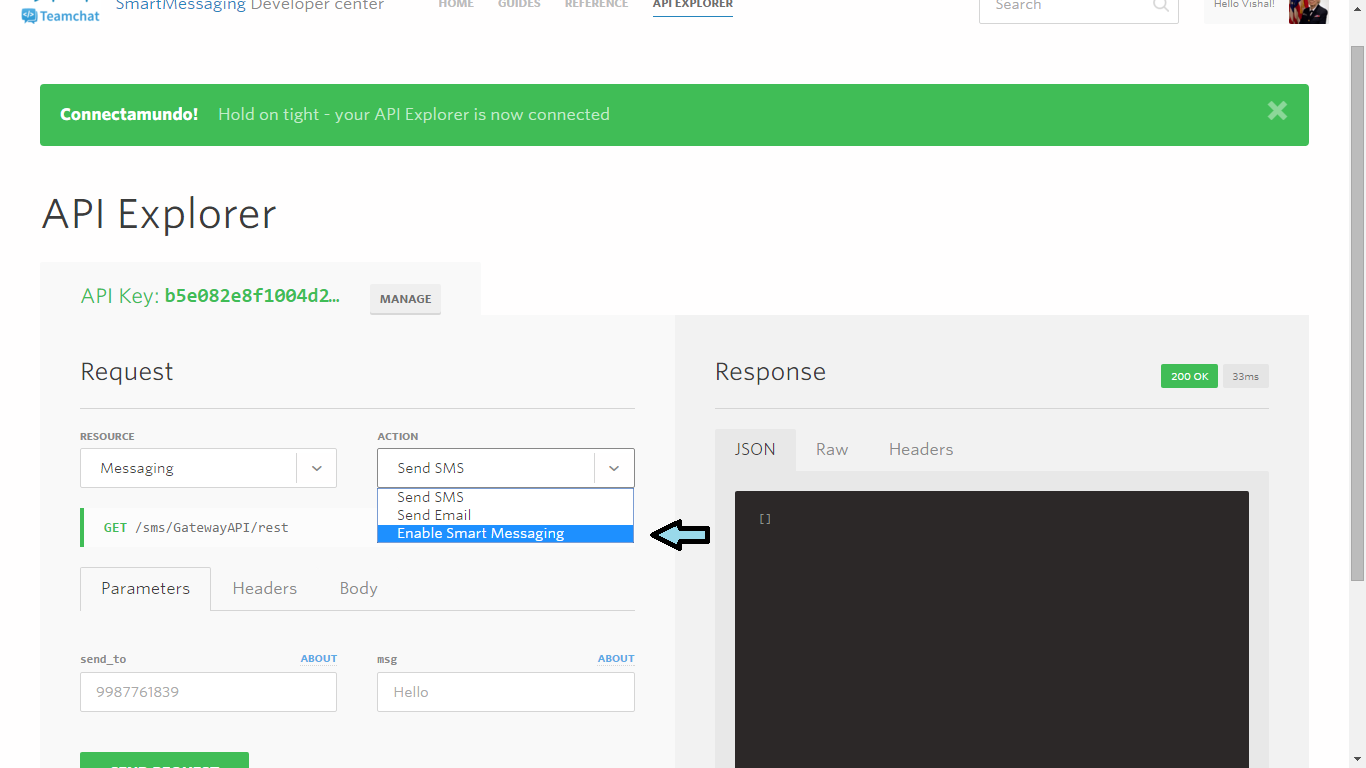


1. Click on **GET API KEY** button then again click on **GET API KEY** button in pop-up. Then you should receive a success message that your API is now connected.

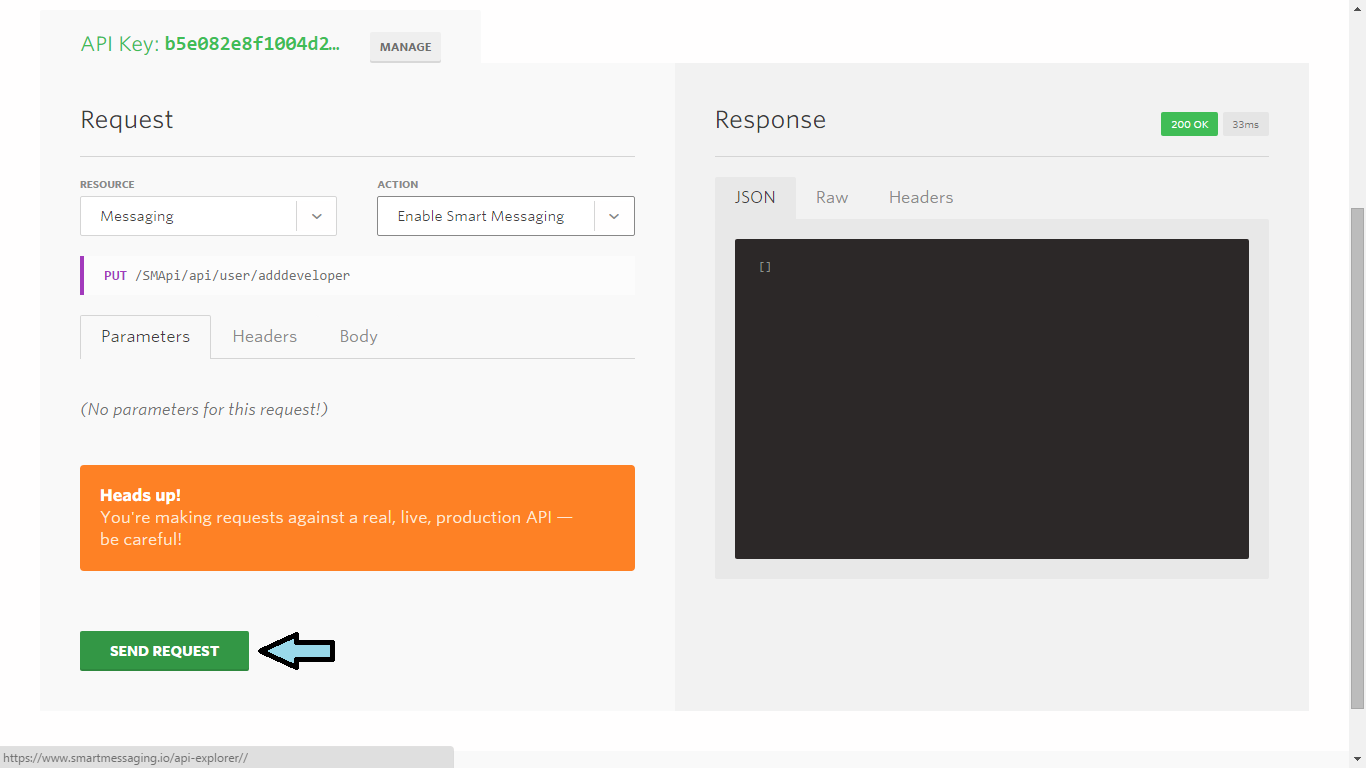


1. Now select **Messaging** from Resources and then choose **Enable Smart Messaging** from actions.





1. Now click on **Send Request** button.



1. Now, you should get a success message as a response. Then click on **MANAGE** button and note down your **SmartMessaging** API Key.
2. This will be the apiKey, which you will use in all your Omni Channel Engine API requests as a Header.

**Using the API :**

Once you have the API key and credentials for the desired channels, you can make a POSTMAN call as shown below to test if you have everything set up in order to use the API.

Along with credentials of each channel, you also need to send the recipients in the request parameters.

URL:<http://stag-solutions.teamchat.com/SmartMessageService/sendMsg>   
METHOD: POST

Content-type: application/x-www-form-urlencoded

Header : Description  
apiKey : your api key  
Body:

Body Parameters are as follows :

Parameter : Description  
1. Question : question text for the smart message   
2. msgType : describes the message’s type (poll/survey)  
3. msgDesc : provides a context for the smart message

6. twitterCreds : A JSON-TEXT containing the authentication parameters and recipients for twitter  
6. slackCreds : A JSON-TEXT containing the authentication parameters and recipients for slack  
6. smsCreds : A JSON-TEXT containing the authentication parameters and recipients for sms  
7. emailCreds : A JSON-TEXT containing the authentication parameters and recipients for email

Example of parameter values :

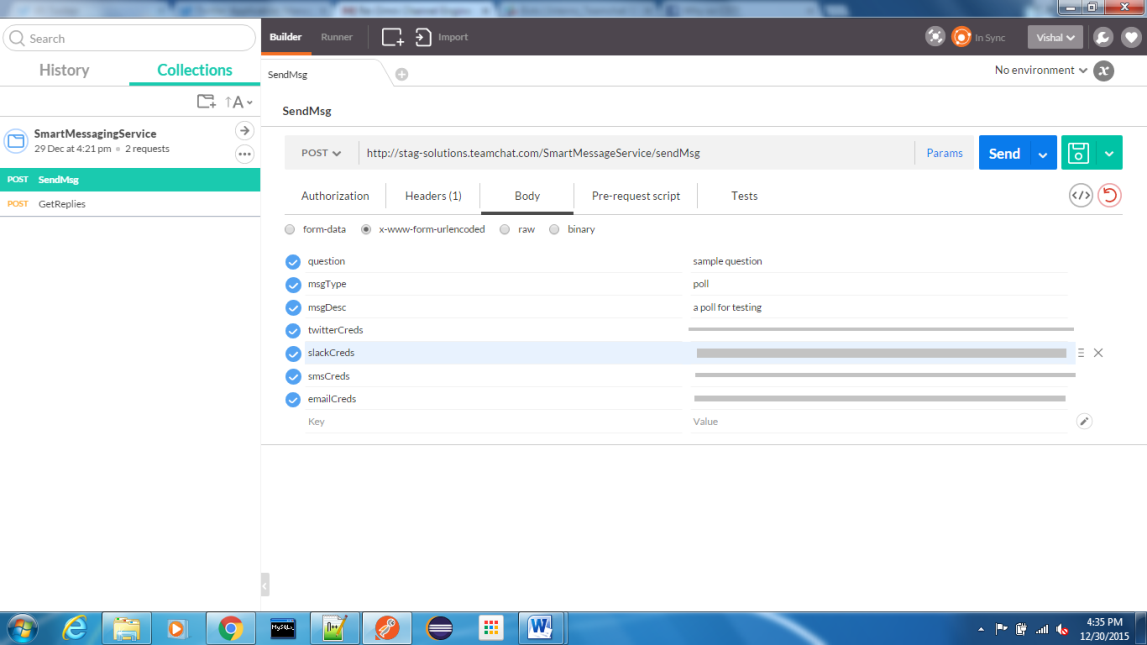
question : Who likes polls??  
msgType : Poll  
msgDesc : A poll about polls

twitterCreds: {"consumerKey":"consumer key here", "consumerSecret":"consumer secret here", "accessToken":"access token here ", "accessTokenSecret":"access token secret here", "recipients":["recipient1 screen name","recipient2 screen name"] }   
slackCreds: {"slackToken":"bot api token here ", "recipients":["recipient1 username","recipient2 username"] }

smsCreds: {"accountId":"account Id here", "password":"password here", " recipients":["recipient phone number"], "source":"source here" }

emailCreds: {"accountId":" account Id here ", "password":" password here ", "recipients":["recipient1 email id","recipient2 email id"],"name":"name here", "subject":"subject here" }

Replace the dummy value of channel credentials and recipients by real ones in your request.



On Success, this request will give a smart message id (smId) corresponding to your message back in response.

Then you can check replies to the smart message by doing another API request :

URL:<http://stag-solutions.teamchat.com/SmartMessageService/getReplies>   
METHOD: POST

Content-type: application/x-www-form-urlencoded

Header : Description  
apiKey : your api key  
Body:

smId : smId of your message here

This will give you a response showing the replies in HTML format.

