gupshup

Enterprise Email API Document

Published: January 2021

Contents

1. Getting Started

Our app resides at http://enterprise.webaroo.com/GatewayAPI/rest
Please use this URL for all API methods.

1.1 Pre-start Checklist

Here's what you need to know prior to using any API
User name & password. If you don't have an account, you can create one at http://enterprise.webaroo.com
URL encoding of your message, password etc.

Our support is available for you at 022-42006799 or you can email us at enterprise-support@gupshup.io ©Gupshup Technology India Pvt. Ltd. 2020

2. Post a Campaign

Below is the API that can be integrated with for posting a campaign or uploading information for executing a camping

2.1 Email Campaign with comma separated recipient list

API Endpoint:

http://enterprise.webaroo.com/GatewayAPI/rest
Supported methods: POST (for email attachments) and Get (if no email attachments)

Sample request:

```
curl --location --request POST 'http://enterprise.webaroo.com/GatewayAPI/rest' \
--form 'method="EMS_POST_CAMPAIGN"' \
--form 'userid="xxxxxxxxxx"' \
--form 'password="xxxxx"' \
--form 'v="1.1"' \
--form 'content_type=" text/html"' \
--form 'name="test1"' \
--form 'fromEmailId="email@example.com"' \
--form 'subject="Verify EMS_POST 4"' \
--form 'recipients="test@example1.com"' \
--form 'ccEmailId="test1@example2.com"' \
--form 'bccEmailId="test2@example3.com"' \
--form 'content="Hi,
This is a EMS_POST_CAMPAIGN"' \
--form 'replyToEmailID ="test@example1.com"' \
--form 'attachment1=@"/C:/Users/TEST/Downloads/EnterpriseEmailAPIDocument.pdf"' \
--form   'attachment2=@"/C:/Users/TEST/Downloads/Gupshup   Enterprise   Email   API
Document.pdf"' \
--form 'attachment3=@"/C:/Users/TEST/Downloads/EnterpriseEmailAPIDocument.pdf"' \
--form 'attachment4=@"/C:/Users/TEST/Downloads/EnterpriseEmailAPIDocument.pdf"' \
--form   'attachment5=@"/C:/Users/TEST/Downloads/Gupshup   Enterprise   Email   API
Document.pdf"'
```

Parameters

Key	**Description**	**Example**
userid **Required** | string	The userid of your Gupshup account. The number must be in pure numeric format with no special characters.	2000155005

password **Required** | string	The password of your Gupshup account for authentication of the userid	sh1gw4e
method **Required** | string	The API method to perform a specific action i.e. post a an email campaign *Must be*: EMS_POST_CAMPAIGN	EMS_POST_CAMPAIGN
v **Required** | string	The API version *Must be*:1.1	1.1
name Optional | string	Name of the sender as it will appear to the receiver	X Company / name of individual
content_type **Required** | string	The data type for the email body that is being executed *Must be*: text/html	text/html
subject **Required** | string	The Subject of the email that is being sent via the campaign	Email Notification for Subscription renewal
recipients **Required** | string	List of recipients email ids separated by commas	u1@example.com,u2@example.com
scheduled_at Optional | string	Date and Time when email should be sent out to the recipient. URL encoded string in the format YYYY-MM-DD%20HH:mm:ss.	2020-04-16 12:53:00
ccEmailId Optional | string	List of cc email ids separated by commas Default number of email addresses permitted is 10	c1@example.com,c2@example.com
bccEmailId Optional | string	List of Bcc email ids separated by commas Default number of email addresses permitted is 10	c1@example.com,c2@example.com
content Optional | string	Content of the mail	Hi, This is a EMS_POST_CAMPAIGN
replyToEmailId Optional | string	Email ID on which replies to the email will be received. This must be configured for the account otherwise will result in error	example@example1.com
attachment1 Optional | string	Attachments to be added along with the email, maximum allowed attachments is 5 of 5MB each Path indicating location of the audio file (publicly accessible path / location of the file on the source system triggering the API)	@"/C:/Users/TEST/Downloads/EnterpriseEmailAPIDocument.pdf

Response:
Success | 1234567890 | Campaign posted

2.2 Email Campaign by Uploading HTML content & comma separated recipient list

API Endpoint:

http://enterprise.webaroo.com/GatewayAPI/rest

Supported methods: POST
CC and Bcc is not permitted for this API

Sample request:

```
curl --location --request POST 'http://enterprise.webaroo.com/GatewayAPI/rest' \
--form 'method="EMS_UPLOAD_CAMPAIGN"' \
--form 'userid="xxxxxxxxxx"' \
--form 'password="xxxxxx"' \
--form 'v="1.1"' \
--form 'name="QA 033"' \
--form 'subject="Verify Email 033"' \
--form
'newsletter_file=@"/C:/Users/TEST/Downloads/recreatingonleadsfromemailsbodytoemail
dl.zip"' \
--form 'content_type="text/html"' \
--form 'replyToEmailId="test@example1.com"' \
--form 'recipients="test@example1.com"' \
--form 'attachment1=@"/home/TEST/Desktop/Testing/Testing_100KB.pdf"'
```

***Parameters*:**

Key	**Description**	**Example**
userid **Required** | string	The userid of your Gupshup account. The number must be in pure numeric format with no special characters.	2000155005
password **Required** | string	The password of your Gupshup account for authentication of the userid	sh1gw4e
method **Required** | string	The API method to perform a specific action i.e. post a an email campaign *Must be*: EMS_UPLOAD_CAMPAIGN	EMS_UPLOAD_CAMPAIGN
v **Required** | string	The API version *Must be*:1.1	1.1
name Optional | string	Name of the sender as it will appear to the receiver	X Company / name of individual
content_type **Required** | string	The data type for the email body that is being executed *Must be*: text/html	text/html
subject **Required** | string	The Subject of the email that is being sent via the campaign	Email Notification for Subscription renewal

recipients **Required** | string	List of recipients email ids separated by commas	u1@example.com,u2@example.com
newsletter_file **Required** | File	File containing mail body (in ZIP format) Path indicating location of the newsletter file	@"/C:/Users/TEST/Downloads/recreatingonleadsfromemailsbodytoemaildl.zip
scheduled_at Optional | string	Date and Time when email should be sent out to the recipient. URL encoded string in the format YYYY-MM-DD%20HH:mm:ss.	2020-04-16 12:53:00
content Optional | string	Content of the mail	Hi, This is a EMS_POST_CAMPAIGN
replyToEmailId Optional | string	Email ID on which replies to the email will be received. This must be configured for the account otherwise will result in error	example@example1.io
attachment1 Optional | string	Attachments to be added along with the email, maximum allowed attachments is 5. Path indicating location of the audio file (publicly accessible path / location of the file on the source system triggering the API)	@"/C:/Users/TEST/Downloads/EnterpriseEmailAPIDocument.pdf

2.3 Email Campaign by Uploading a Recipient List

You can upload a file containing all the email addresses and mail them your campaign when using this method. This API will not work if the TO recipient visibility is on.

CC and Bcc is not permitted for this API

API Endpoint:

http://enterprise.webaroo.com/GatewayAPI/rest

Supported methods: POST

Sample request

```
curl --location --request POST 'http://enterprise.webaroo.com/GatewayAPI/rest' \
--form 'method="EMS_POST_CAMPAIGN_TO_UPLOADED_ADDRESSES"' \
--form 'userid="xxxxxxxxxx"' \
--form 'password="xxxxx"' \
--form 'v="1.1"' \
--form 'name="API 2  "' \
--form
'subject="%E0%A4%86%E0%A4%AA%E0%A4%95%E0%A4%BE%20%E0%A4%A6%E0%A4%BF%E0%A4%A8%20%E0
%A4%B6%E0%A5%81%E0%A4%AD%20%E0%A4%B9%E0%A5%8B%20!%40%24%25%5E*()%2C.%3F%2F%3A%3B%2
7%22%7B%7D%5B%5D%5C%7C%23%26_%2B"' \
--form 'recipients_file=@"/C:/Users/TEST/Desktop/list.csv"' \
--form 'content="Hi ${name}, Your OTP is ${mobile} Thank you
%E0%A4%86%E0%A4%AA%E0%A4%95%E0%A4%BE%20%E0%A4%A6%E0%A4%BF%E0%A4%A8%20%E0%A4%B6%E0%
A5%81%E0%A4%AD%20%E0%A4%B9%E0%A5%8B%20!%40%24%25%5E*()%2C.%3F%2F%3A%3B%27%22%7B%7D
%5B%5D%5C%7C%23%26_%2B"' \
--form 'content_type="text/html"' \
--form 'replyToEmailId="testemail@example1.com"' \
--form 'attachment=@"/C:/Users/TEST/Downloads/EnterpriseEmailAPIDocument.pdf"'
```

Parameters

Key	**Description**	**Example**
userid **Required** | string	The userid of your Gupshup account. The number must be in pure numeric format with no special characters.	2000155005
password **Required** | string	The password of your Gupshup account for authentication of the userid	sh1gw4e
method **Required** | string	The API method to perform a specific action i.e. post a an email campaign *Must be*: EMS_POST_CAMPAIGN_TO_UPLOADED_ADDRESSES	EMS_POST_CAMPAIGN_TO_UPLOADED_ADDRESSES
v **Required** | string	The API version *Must be*:1.1	1.1
name Optional | string	Name of the sender as it will appear to the receiver	X Company / name of individual
content_type **Required** | string	The data type for the email body that is being executed *Must be*: text/html	text/html
subject **Required** | string	The Subject of the email that is being sent via the campaign	Email Notification for Subscription renewal
recipients_file **Required** | File	List of recipients email ids Path indicating location of the recipient file	@"/C:/Users/TEST/Desktop/list.csv
scheduled_at Optional | string	Date and Time when email should be sent out to the recipient. URL encoded string in the format YYYY-MM-DD%20HH:mm:ss.	2020-04-16 12:53:00
Content Optional | string	Content of the mail	Hi, This is a CAMPAIGN
replyToEmailId Optional | string	Email ID on which replies to the email will be received. This must be configured for the account otherwise will result in error	example@example1.com
attachment1 Optional | string	Attachments to be added along with the email, maximum allowed attachments is 5. Path indicating location of the audio file (publicly accessible path / location of the file on the source system triggering the API)	@"/C:/Users/TEST/Downloads/EnterpriseEmailAPIDocument.pdf

2.4 Email Upload Campaign using Uploaded recipient list

This method is similar to Upload Campaigns. Here, instead of typing email addresses you can upload a file containing the list of recipients email ids.

API end-point:
http://enterprise.webaroo.com/GatewayAPI/rest
Supported methods: POST
CC and Bcc is not supported for this API

Sample Request:

```
curl --location --request POST 'http://enterprise.webaroo.com/GatewayAPI/rest' \
--form 'method="EMS_UPLOAD_CAMPAIGN_TO_UPLOADED_ADDRESSES"' \
--form 'userid="xxxxxxxxxx"' \
--form 'password="xxxxx"' \
--form 'v="1.1"' \
--form 'name="QA API 011"' \
--form 'subject="QA API 011 - Recipient List and Newsletter"' \
--form 'content_type="text/html"' \
--form 'recipients_file=@"/home /Desktop/Testing/Recipient_list.csv"' \
--form 'newsletter_file=@"/home /Desktop/Testing/Test_2-link.zip"' \
--form 'replyToEmailId="email@example.com"' \
--form 'attachment1=@"/C:/Users/TEST/Downloads/EnterpriseEmailAPIDocument.pdf"'
```

Parameters

Key	**Description**	**Example**
userid **Required** | string	The userid of your Gupshup account. The number must be in pure numeric format with no special characters.	2000155005
password **Required** | string	The password of your Gupshup account for authentication of the userid	sh1gw4e
method **Required** | string	The API method to perform a specific action i.e. post a an email campaign *Must be*: EMS_UPLOAD_CAMPAIGN_TO_UPLOADED_ADDRESSES	EMS_UPLOAD_CAMPAIGN_TO_UPLOADED_ADDRESSES
v **Required** | string	The API version *Must be*:1.1	1.1
name Optional | string	Name of the sender as it will appear to the receiver	X Company / name of individual
content_type **Required** | string	The data type for the email body that is being executed *Must be*: text/html	text/html
subject **Required** | string	The Subject of the email that is being sent via the campaign	Email Notification for Subscription renewal
recipients_file **Required** | File	List of recipients email ids Path indicating location of the recipient file	@"/C:/Users/TEST/Desktop/list.csv
newsletter_file **Required** | File	File containing mail body (in ZIP format) Path indicating location of the newsletter file (publicly accessible path /	@"/C:/Users/TEST/Downloads/recreatingonleadsfromemailsbodytoemaildl.zip

	location of the file on the source system triggering the API	
scheduled_at Optional | string	Date and Time when email should be sent out to the recipient. URL encoded string in the format YYYY-MM-DD%20HH:mm:ss.	2020-04-16 12:53:00
replyToEmailId Optional | string	Email ID on which replies to the email will be received. This must be configured for the account otherwise will result in error	example@example1.com
attachment1 Optional | string	Attachments to be added along with the email, maximum allowed attachments is 5. Path indicating location of the audio file (publicly accessible path / location of the file on the source system triggering the API)	@"/C:/Users/TEST/Downloads/EnterpriseEmailAPIDocument.pdf

2.5 Email text campaign with TO recipient list:

This API can be used to post an email text campaign with the recipients in TO will be visible to everyone.
The recipient list will be a comma separated email list only and maximum recipients are 50 per request.
BCC and CC are not supported due to the TO recipient list visibility.
This is an account level feature and has to be enabled as per required on an Enterprise account level.

API end-point:

http://enterprise.webaroo.com/GatewayAPI/rest

***Supported methods**: GET and POST (for up-to 5 email attachments)*

Sample Request (GET)

```
curl --location --request GET
'http://enterprise.webaroo.com/GatewayAPI/rest?method=EMS_POST_CAMPAIGN&userid=xxxxxxxxxx&password=xxxxx&v=1.1&name=QA%201&content_type=text/html&fromEmailId=email@example.com&subject=Verify%20EMS_POST%204&content=Hi,%0AThis%20is%20a%20EMS_POST_CAMPAIGNsubject=Verify%20EMS_POST%204&recipients=test@example1.com,test@example1.io&content=Hi,%0AThis%20is%20a%20EMS_POST_CAMPAIGN'
```

Parameters:

Key	**Description**	**Example**
userid **Required** | string	The userid of your Gupshup account. The number must be in pure numeric format with no special characters.	2000155005

<table>
<tr><td>password
<b>Required</b> | string</td><td>The password of your Gupshup account for authentication of the userid</td><td>sh1gw4e</td></tr>
<tr><td>method
<b>Required</b> | string</td><td>The API method to perform a specific action i.e. post a an email campaign
<i>Must be:</i> EMS_POST_CAMPAIGN</td><td>EMS_POST_CAMPAIGN</td></tr>
<tr><td>v
<b>Required</b> | string</td><td>The API version
<i>Must be:</i>1.1</td><td>1.1</td></tr>
<tr><td>name
Optional | string</td><td>Name of the sender as it will appear to the receiver</td><td>X Company / name of individual</td></tr>
<tr><td>content_type
<b>Required</b> | string</td><td>The data type for the email body that is being executed
<i>Must be:</i> text/html</td><td>text/html</td></tr>
<tr><td>subject
<b>Required</b> | string</td><td>The Subject of the email that is being sent via the campaign</td><td>Email Notification for Subscription renewal</td></tr>
<tr><td>scheduled_at
Optional | string</td><td>Date and Time when email should be sent out to the recipient. URL encoded string in the format YYYY-MM-DD%20HH:mm:ss.</td><td>2020-04-16 12:53:00</td></tr>
<tr><td>replyToEmailId
Optional | string</td><td>Email ID on which replies to the email will be received. This must be configured for the account otherwise will result in error</td><td>example@example1.com</td></tr>
<tr><td>recipients
<b>Required</b> | string</td><td>List of recipients email ids separated by commas</td><td>u1@example.com,u2@example.com</td></tr>
</table>

2.6 Email upload campaign with TO recipient list:

This API can be used to post an email campaign with uploaded HTML content (.zip file) in which the recipients in TO will be visible to everyone.

The recipient list will be a comma separated email list only and maximum recipients are 50 per request.

BCC and CC are not supported due to the TO recipient list visibility.

This is an account level feature and has to be enabled as per required on an Enterprise account level.

API end-point:

http://enterprise.webaroo.com/GatewayAPI/rest

***Supported methods**: POST (for up-to 5 email attachments)*

Sample API:

```
curl --location --request POST 'http://enterprise.webaroo.com/GatewayAPI/rest' \
--form 'method="EMS_UPLOAD_CAMPAIGN"' \
--form 'userid="xxxxxxxxxx"' \
--form 'password="xxxxxx"' \
--form 'v="1.1"' \
--form 'name="QA%201"' \
--form 'content_type="text/html"' \
--form 'fromEmailId="email@example.com"' \
--form 'subject="Verify%20EMS_POST%204"' \
--form 'newsletter_file=@"/C:/Users/test/Downloads/example_Testing.zip"' \
--form 'recipients="test@example1.com,test@example1.com"' \
--form 'replyToEmailId="email@example.com"' \
--form 'attachment1=@"/C:/Users/TEST/Downloads/EnterpriseEmailAPIDocument.pdf"'
```

Parameters:

Key	**Description**	**Example**
userid **Required** | string	The userid of your Gupshup account. The number must be in pure numeric format with no special characters.	2000155005
password **Required** | string	The password of your Gupshup account for authentication of the userid	sh1gw4e
method **Required** | string	The API method to perform a specific action i.e. post a an email campaign *Must be*: EMS_UPLOAD_CAMPAIGN	EMS_UPLOAD_CAMPAIGN
v **Required** | string	The API version *Must be*:1.1	1.1
name Optional | string	Name of the sender as it will appear to the receiver	X Company / name of individual
content_type **Required** | string	The data type for the email body that is being executed *Must be*: text/html	text/html
subject **Required** | string	The Subject of the email that is being sent via the campaign	Email Notification for Subscription renewal
recipients **Required** | string	List of recipients email ids separated by commas	u1@example.com,u2@example.com
newsletter_file **Required** | File	File containing mail body (in ZIP format) Path indicating location of the newsletter file	@"/C:/Users/cathie/Downloads/recreatingonleadsfromemailsbodytoemaildl.zip
scheduled_at Optional | string	Date and Time when email should be sent out to the recipient. URL encoded string in the format YYYY-MM-DD%20HH:mm:ss.	2020-04-16 12:53:00
replyToEmailId Optional | string	Email ID on which replies to the email will be received. This must be configured for the account otherwise will result in error	example@example1.com
attachment1 Optional | string	Attachments to be added along with the email, maximum allowed attachments is 5. Path indicating location of the attachment file (publicly accessible path / location of the file on the source system)	@"/C:/Users/cathie/Downloads/EnterpriseEmailAPIDocument.pdf

2.7 Custom Email campaigns:

It is possible to execute email campaigns with variable email body content.

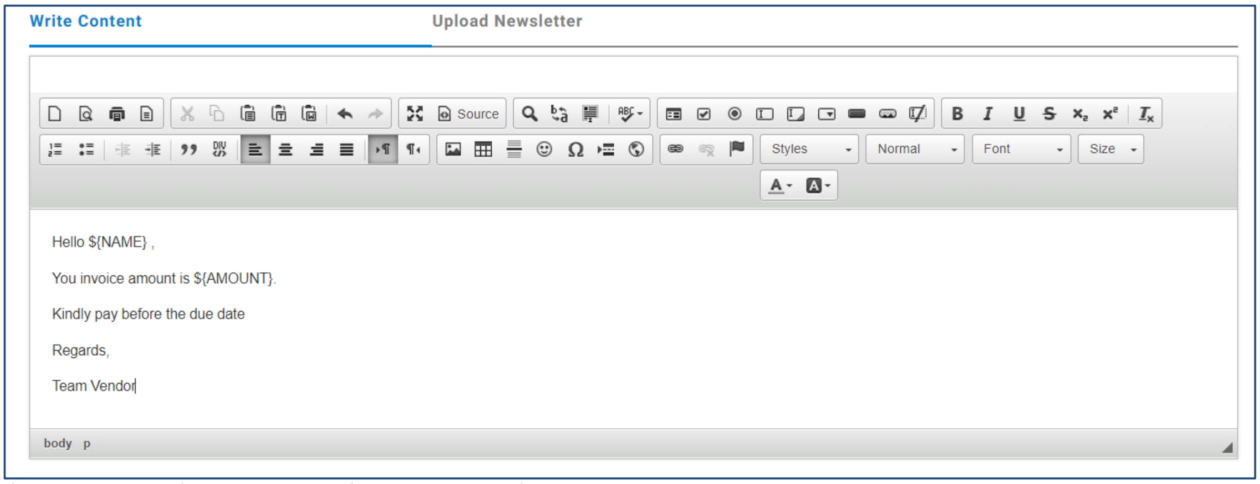
To send customized messages to a list of email addresses:

a) Excel sheet format Stated below is a list of the headers to be used to send customized emails.

  i) EMAIL IDs: List of email addresses to which emails need to be sent.

  ii) Email body content – Message which needs to be sent out to the customer. This message should clearly indicate the variable fields which are denoted by actual variable name as specified on the panel.

For example: consider the email body content and the upload list of email ids as below:



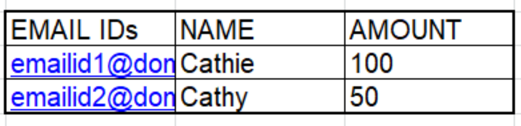
Hello ${NAME} ,

You invoice amount is ${AMOUNT}.

Kindly pay before the due date

Regards,

Team Vendor



EMAIL IDs	NAME	AMOUNT
emailid1@don	Cathie	100
emailid2@don	Cathy	50

2.8 Using SMTP

This method of sending emails is relevant for integrations with repository systems like a CRM where system outbound email SMTP settings can be easily configured in order to send outbound emails, workflow alerts, marketing campaigns, and system notifications.

Host/Domain: Enter the hostname: i.e. http://enterpriseemail.webaroo.com/

Port: enter the port number as 587

Other details to be set as below:

- **Username**: enter the Gupshup userid
- **Password**: enter the corresponding password for Guphup’s enterprise account
- **Security**: has to be none
- **From email address**: Configure the email address that is also set at Gupshup Enterprise account level. This will be the ‘from’ email address visible to recipient
- **To email address** : Add the recipient’s email address

};

3. JAVA class code

```
import java.io.File;

import java.net.URLEncoder;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.Date;

import java.util.TimeZone;

import java.util.concurrent.atomic.AtomicInteger;

import org.apache.commons.httpclient.HttpClient;

import org.apache.commons.httpclient.methods.PostMethod;

import org.apache.commons.httpclient.methods.multipart.FilePart;

import org.apache.commons.httpclient.methods.multipart.MultipartRequestEntity;

import org.apache.commons.httpclient.methods.multipart.Part;

import org.apache.commons.httpclient.methods.multipart.StringPart;

public class NewBlastSubmission
{
private void sendFileToEnterprise(String url, String filename)
{
try
{
HttpClient client = new HttpClient();

PostMethod method = new PostMethod(url);

File f = new File(filename);

/*
http://enterprise.webaroo.com/GatewayAPI/rest?method=EMS_POST_CAMPAIGN&userid=xxxxxxxXXX&password=*****
*&v=1.1&name=TEST TECH &recipients= reply1@example.com&subject=Test estatement&content=Testing e-
statementfrom API
Gupshup&content_type=text/html&attachment1=C:\Users\Administrator\Downloads\1001005064119.pdf
*/
Part[] parts = {
new StringPart("method", "EMS_POST_CAMPAIGN"), new StringPart("userid", "xxxxxxxXXX"),new StringPart("password",
"******"), new StringPart("v","1.1"), new StringPart("recipients", URLEncoder.encode("reply1@example.com", "UTF-8")),
new StringPart("subject", URLEncoder.encode("Test e-statement", "UTF-8")), new StringPart("content", "Testing e-
statement from API Gupshup"), new StringPart("content_type", "text/html"), new FilePart("attachment0", f),new
StringPart("name", "GUPSHUP TECH")
};

/*
new StringPart("method", "EMS_POST_CAMPAIGN"), new StringPart("userid", "xxxxxxxXXX"), new StringPart("password",
"******"), new StringPart("recipients", " reply1@example.com"),
new StringPart("subject", URLEncoder.encode("Test estatement", "UTF-8")), new StringPart("content", ""), new
StringPart("format", "json"),
new StringPart("filetype", "zip"), new StringPart("v", "1.1"), new StringPart("mask", mask), new StringPart("dnd", "true"),
new FilePart(f.getName(), f),
new StringPart("timestamp", dateFormat.format(new Date(scheduleTime))),
*/
method.setRequestEntity(new MultipartRequestEntity(parts, method.getParams()));

int statusCode = client.executeMethod(method);

System.out.println( filename + "," +method.getResponseBodyAsString());

System.out.println(statusCode);
}
catch (Exception e)
{
e.printStackTrace();
}
}
public static void main(String[] args)
{
String url = "http://enterprise.webaroo.com/GatewayAPI/rest";

String fileDir = "/home/Desktop/SimpleEMailTesthtml.zip";

New NewBlastSubmission().sendFileToEnterprise(url, fileDir);
}
}
```

4. Delivery reports

Detailed reports are available on panel http://enterprise.webaroo.com for download after 2 hrs of campaign execution. Real time delivery events will be pushed over a call-back URL of the client by Gupshup's platform. Following events will be sent out:

- DELIVERED:

After successfully delivering an email, no action (response) has been taken by user.

- OPENED:

The response generated by a recipient opening an email.

- CLICKED:

Recipient clicked one of the Click Tracked links in your email.

- UNSUBSCRIBE:

Recipient has unsubscribed for your emails.

- SPAM REPORTS:

Whenever a recipient marks your email as spam and their mail server tells us about it.

- NOT SENT:

In certain cases, Gupshup will not send a message to a specific email address in order to protect your sender reputation. Gupshup keeps Email Lists to track bounces, spam reports, and unsubscribes for each of our users. If a user sends a message to an email address that exists on one of these lists within their account, Gupshup will automatically drop the message (i.e. not send to the address).

• NOT DELIVERED: BOUNCE & BLOCKED

Bounce is an e-mail that is returned to the sender because it was rejected by the receiving mail server due to various reasons. There are many reasons for the mail to get rejected and we've divided them into two categories:

Soft Bounce:

This is a temporary issue where the recipient's mail server rejects the email due to reasons such as

- Recipient's mailbox is likely to be full
- Mail server is temporarily unavailable
- Recipient has closed the email account with the particular provider while the email was being sent

Note

When an email address gets a soft bounce and three successive email campaigns get undelivered, the email address is marked as a hard bounce.

Hard Bounce:

This is a permanent bounce when the receiving server rejects the email due to the following reasons:

- Invalid email address (typos, domain does not exist)

Email address is not in existence

- Recipient's mail server has blocked the sending server

• DEFERRED:

A deferred event, or deferral, is simply an event Gupshup has received back from the receiving server that tells us that the receiving server has temporarily limited access to their system or a busy signal.

It does not mean that your message will not be delivered. Rather, it is a signal that your message will not be delivered immediately. This can happen for a variety of reasons. Some of the most common reasons a message might be deferred at an inbox provider are that the inbox provider is seeing too many spam complaints for your email that has already been delivered or that the receiving server is having technical issue at that time.

When Gupshup sees your mail being deferred at a particular inbox provider, we will continue to try to deliver that email for you for up to 72 hours. If a message to a recipient is deferred for more than 72 hours, that deferral will turn into a block. Otherwise, it will result in a delivery event once took the message has been accepted by the receiving server.

• RESPONSE_AWAITED:

It means that Gupshup has successfully processed the mail request but have not received the delivery event and hence awaiting for the Response from Recipient server.

4.1 How to use

It the delivery report setting is enabled at the account level, Webhook URL given by client will receive a HTTP POST request.

"Content-type" is "application/x-www-form-urlencoded;charset=utf-8" The key parameter is "response"

The value corresponding to the above key is in json and the format is mentioned below:

```
**Example** [{
"timestamp": "1493577000", "email": "email@example.com", "event": "DELIVERED",
"campaignid": "3319237025400370211”
}]
Note - timestamp is in seconds [{
"timestamp": "1493577000", "email": "email@example.com", "event": "DELIVERED",
"campaignid": "3319237025400370211”
},
{
"timestamp": "1493577001", "email": "email2@example.com", "event": "SOFT_BOUNCE",
"campaignid": "3319237025400370211”
},
{
"timestamp": "1493577001", "email": "email3@example.com", "event": "CLICK", "link":"http://www.example.com",
"campaignid": "4019237025400370222”
},
{
"timestamp": "1493577004", "email": "email4@example.com", "event": "DELIVERED",
"campaignid": "4319237025400370233”
}
]
```

4.2 Code snippet in Java to handle multiple events in single call

Sample code to accept multiple events in a single call so that you don't have to modify your code later:

```
String response = req.getParameter("response");

JSONArray jsonArray = new JSONArray(response);

int length = jsonArray.length();

for (int i = 0; i < length; i++)
{
    JSONObject jsonObject = jsonArray.getJSONObject(i);

    String campaignid = jsonObject.getString(("campaignid"));

    callBusinessMethod(jsonObject);
}
```

5. Guidelines for Email Campaigns

5.1 Warm up plan

The goal here for a sender is: to be considered a legitimate email sender and to build a positive reputation.

Whether you are a client moving to a new Email Service Provider (ESP) i.e. Gupshup or simply adding new domains to their current email program, ‘warming’ is one of the most important steps to establishing a good sender reputation with Internet Service Providers (ISPs) and maintaining strong email delivery performance.

Warming a new domain is the process of gradually increasing volume from each over a certain amount of time, based on total volume you plan to send daily – usually following a customized schedule developed by your ESP. The overall goal is to be considered a legitimate email sender by the ISPs, and to build a positive reputation.

When you get a new domain, it is known as ‘cold,’ meaning it has been dormant for some period of time, and no longer maintains a sender reputation. When campaigns are sent from new domains, the ISPs evaluate them more carefully in an effort to protect users. If they didn’t take these precautions, the “true” spammers would be able to get away with a lot more. During the evaluation process, ISPs pay close attention to volume, complaints, engagement, relevancy and list hygiene. With volume being one of the main items monitored, it is best to begin sending low volumes with gradual increases toward the maximum daily volume.

Overall, warming new domains to establish a strong sender reputation with ISPs takes patience and time. Start with your most engaged recipients first, then slowly increase send volume overtime.

5.2 Unsubscribe Link

Ideally, every email should have Unsubscribe link. It is in your favour to include an 'unsubscribe link' in your emails as if someone wants to stop receiving your emails they can easily opt out. We recommend that, your emails should already include an option to allow customers to unsubscribe. Not only is it a basic part of unsubscribe best practices, but it's required by law under the CAN-SPAM Act.

If you don't have an unsubscribe link the recipient is likely to mark your email as spam as they don't know how else to unsubscribe. If you get too many spam complaints then domain and IP reputation will get negatively impacted and it will hamper the campaign performance.

5.3 Non-delivery of Emails:

Key points to note for non-delivery of emails

1) If the email is not delivered to a recipient email address, due to reasons such as Dropped, Deferred, Bounce, Spam Report or Unsubscribe- it is **blocked**. The blocking is applied to those recipient email addresses since sending email on such email addresses will impact your domain reputation.

2) If the recipient has marked the email as SPAM, he/she needs to manually mark the same email as "Not Spam".

3) If the recipient has unsubscribed from receiving email from a particular sender, we cannot unblock the recipient unless we receive an email from the business POC, stating that the email address needs will require receiving the emails again.