Sandes Gateway Specification

Version 2.8

GIMS Division, National Informatics Centre

Revision History

Version	Date	Author	Reviewer	Approver	Comments
0.1	07/07/2018	Arun K Varghese	Manoj P A	T Mohana Dhas SIO	Initial Draft Version
0.2	29/07/2018	Arun K Varghese	Manoj P A	T Mohana Dhas SIO	Multicast and Customcast APIs finalized
1.0	13/08/2019	Arun K Varghese	Manoj P A	T Mohana Dhas SIO	Added HMAC security
1.1	27/08/2019	Arun K Varghese	Manoj P A	T Mohana Dhas SIO	Corrected definition of hmac parameter
1.2	17/10/2019	Arun K Varghese	Manoj P A	T Mohana Dhas SIO	Added Organization broadcast API Message payload limited to 1KB
1.3	11/12/2019	Arun K Varghese	Manoj P A	T Mohana Dhas SIO	Rate Limiter implemented Message Prioritization Implemented Content Type header made mandatory in requests
1.4	03/01/2020	Arun K Varghese	Manoj P A	T Mohana Dhas SIO	Group/List broadcast API
1.5	09/01/2020	Arun K Varghese	Manoj P A	T Mohana Dhas SIO	GIMS Registration check API
1.5.1	19/02/2020	Arun K Varghese	Manoj P A	T Mohana Dhas SIO	Corrections in sample data format of group /organization broadcast
2.0	15/06/2020	Arun K Varghese	Manoj P A	T Mohana Dhas SIO	Send messages using mobile number Added OU broadcast API Customast API replaced by Personal messaging API API to get dispatch, delivery and read status Dispatch statistics API to get the number of messages dispatched between given dates GIMS Gateway client service Beta for quick integrations
2.1	06/07/2020	Arun K Varghese	Manoj P A	T Mohana Dhas SIO	End-to-end encrypted confidential messaging
2.2	07/01/2021	Arun K Varghese	Manoj P A	T Mohana Dhas SIO	Correction in field name process_date
2.3	07/04/2021	Arun K Varghese	Manoj P A	T Mohana Dhas SIO	Renamed to Sandes Added new gateway client service APIs
2.4	03/06/2021	Arun K Varghese	Manoj P A	Manoj P A	Confidential File sharing API details added.
2.5	02/07/2021	Arun K Varghese	Manoj P A	Manoj P A	Removing deprecated APIs
2.6	19/07/2021	Arun K Varghese	Manoj P A	Manoj P A	Gateway Client Service Ver 1.4 released Added multi tenancy support Version management and update checking Confidential message flag is deprecated. All messages are now confidential only. Revised normal and fast batch plans
2.7	03/08/2021	Arun K Varghese	Manoj P A	Manoj P A	Gateway Client Service Ver 1.6 released Windows service installer/uninstaller Linux service installer/uninstaller Client service logs are written to a file
2.8	30/12/2021	Arun K Varghese	Manoj P A	Manoj P A	Released Gateway Client Ver.1.7 Added provision for deployment on container

Table of Contents

1. Overview

Sandes gateway provides instant push messaging from government apps to Sandes. An app registered with the Sandes platform can send push messages to users registered in Sandes. After registration and activation of a gateway account, an app can push messages through the Sandes gateway.

1.1. API Integrations

1. Personalized message to a user by email ID or mobile number
2. Confidential message to a user by email ID or mobile number
3. Upload and share files along with the message
4. View the dispatch, delivery and read status along with timestamps for a message request
5. Get statistics on total messages dispatched between two given dates from an account

1.2. Other Features

- API access restricted to registered and activated apps
- HMAC based security to ensure authentication and message integrity
- Messaging prioritization to manage delivery QoS
- Realtime activation, deactivation and live reload of configuration changes to reduce API downtime
- Request IP whitelisting for app accounts
- Rate limiting plans to suit wide range of message load requirements

2. App Registration Process

An app is registered by the OU Admin of any organization unit by logging in to the Sandes Admin Portal. An app is uniquely identified by a client id. The following fields are required during the app registration process:

Description	**Remarks**
Client Id	Unique client Id by which the app is identified. Random GUID string is auto generated during registration
Client Secret	Client secret is a random alphanumeric string (32 chars) auto generated during registration. The client secret will be sent to the technical contacts after the registration.
HMAC Key	HMAC key is a HMAC-SHA256 key randomly generated during registration process. HMAC key is used to compute the HMAC of the request body. The HMAC key will be sent to the technical contacts after the registration.

App Name	Unique name for the app. The app name may be provided by the OU Admin. App name can have only alphabets and hyphen character
App Title	User-friendly display title for the app account. This will be shown as the sender ID for messages received in Sandes Mobile App.
IP Whitelist	Comma separated list of IPs that must be whitelisted by the gateway to allow incoming API requests using the account
App Logo	App logo to be displayed by the Sandes client
Home Page URL	Home page URL of the app
Privacy Policy Link	Link to the privacy policy of the app
Rate Limiting Plan	See 3. Rate Limiting

The client secret and HMAC key must be saved secretly immediately after registration. Client secret and HMAC key are not stored or maintained in Sandes platform. However, the client secret and HMAC key may be reset any time if required.

3. Rate Limiting

Rate limiter is used to limit the number of requests generated by the integrating apps to prevent DOS attacks and to ensure fair usage of gateway services. The gateway integrator must choose a usage plan that suits the app message workload. Hard throttling is configured in all plans and requests will be rejected on exceeding the plan limit by returning **HTTP 429 "Too many requests"** error. The following usage plans are defined for the different workloads:

Sl. No.	Usage Plan	Message Rate	Burst Rate	Remarks
1.	Basic	120 messages/minute	2	For small web applications
2.	Advanced	600 messages/minute	60	For medium size state level applications
3.	Enterprise	6000 messages/minute	100	For large national level applications
4.	Normal Batch	6000 messages/hour	5	Optimized for batch messaging
5.	Fast Batch	12000 messages/hour	5	Optimized for fast batch messaging
6.	Super Burst	10000 messages/hour	1000	Optimized for sporadic messaging

Example-1: The basic plan allows up to a maximum of 120 messages to be sent every minute where up to 2 messages can be dispatched concurrently (the burst rate).

Example-2: The super burst batch plan allows up to a maximum of 10000 messages to be sent every hour where up to 1000 messages can be dispatched (the burst rate).

Response Headers

The following rate limiting headers will be sent in the response from the gateway:

Header	Description	Mandatory
X-Ratelimit-Limit	Allowed rate limit value	Yes
X-Ratelimit-Remaining	Maximum number of requests that are permitted instantaneously	Yes
X-RateLimit-Retry	Time in seconds until the next request will be permitted	Only for HTTP 429
X-RateLimit-Reset	Time in seconds until the rate limiter returns to its initial state	Only for HTTP 429

4. Message Prioritization

Message prioritization is used to control the delivery QoS for messages. Message dispatch is prioritized based on the priority attribute. The following priorities are implemented by the gateway:

1. **high-volatile** – These messages are delivered instantaneously as soon as the request is received. OTP messages may be sent in this priority.
2. **high –** Time sensitive transaction alerts
3. **medium –** Other transactional alerts
4. **low –** Scheduled batch of general informational messages / notifications

5. Gateway Client Service

The gateway client service is a helper client service that enables quick integration between an application and Sandes gateway with minimal coding. Gateway client service is a portable and configurable service that can be deployed locally alongside your application server. It can be configured to receive a simple HTTP request for sending a personal message to a registered user in Sandes.

The following are the advantages in using the gateway client service:

- Inbuilt HMAC generation for requests by configuring the account credentials
- Verify Sandes registration of user before sending the message
- End-to-end encrypted secure messaging for sending confidential messages
- Confidential file sharing service providing E2EE secured file sharing
- Auto rate limiting based on the account policy
- Automatic version check
- Multi-tenancy support

Gateway client service is packaged for deployment to support both Linux and Windows environments. The following deployment packages are available:

- Win64 Executable (win64/gims-gateway-cli-srv.exe)
- Linux Executable (linux/gims-gateway-cli-srv)
- MaxOS Executable (macos/gims-gateway-cli-srv)

The service is configured using the JSON configuration file (cmd.json). The configuration parameters are given below:

Parameter	Description
ClientServicePort	The port on which the service is configured within the application/web server. This port must not be exposed outside the local server.
GatewayBaseURL	The base URL of Sandes gateway API. https://dwar1.gims.gov.in/v2/api
ClientID	The client ID assigned to the account
ClientSecret	The client secret for the account
HMACKey	The HMAC key for the account

5.1. Quick Setup

1. Update client id, client secret and HMAC key of your app account in **cmd.json.** Make sure the configuration file cmd.json is in the same folder as that of the gateway client service
2. Set permissions and execute the service.

   Linux

   - Set executable permission in Linux environment

     > chmod +x gims-gateway-cli-srv
   - Run the service in foreground (to see the console logging)

     > ./gims-gateway-cli-srv
   - Run the service in background

     > nohup ./gims-gateway-cli-srv &

   Windows

   Run gims-gateway-cli-srv.exe to start the service
3. Sending a confidential message

To send a confidential message, call the service with a GET request as shown in the below:

http://localhost:8021/send?receiverid={email or mobile}&msg={message}

The message must be URL encoded before setting the query string parameter.

An example is given below:

http://localhost:8021/send?receiverid=9999999999&msg=A%20normal%20message

A success response (HTTP 202 Accepted) will be received if the message was submitted successfully. A request ID will be received in the response body for tracing the status of dispatch, delivery and read.

Confidential messaging allows an application to push end-to-end encrypted confidential messages to Sandes users. End-to-end encryption is a core implementation in Sandes client app, and the same feature is extended to the Messaging Gateway service client.

E2EE messages are encrypted directly at the source (your application server) and sent to the Sandes gateway for delivery. The message can only be decrypted by the Sandes app of the user. E2EE prevents a man-in-the-middle attack on confidentiality between the sending application and receiving user.

Note: - For broadcasting non-confidential messages to large number of users, confidential messaging may be disabled to improve performance as given below:

http://localhost:8021/send?receiverid={email or mobile}&msg={message}&**confidential=N**

Common Issues

- If the receiver does not have a Sandes account, the error code GEN016 is returned.
- If the machine/server IP is not whitelisted in Sandes gateway, the service will exit with error. Make sure the IP of the machine/server is whitelisted for your app account.
- If the gateway service is not reachable from your machine, a timeout error will be shown, and the service will exist. Please make sure firewall rules are configured to allow the machine/server access to https://dwar1.gims.gov.in

5.2. Upgrading from previous versions

1. Stop the old gateway client service
2. Replace the cmd.json from the new verssion
3. Update client id, client secret and HMAC key of your app acount in the new cmd.json
4. Replace the gateway client service executable (gims-gateway-cli-srv)
5. Start the new gateway client service

5.3. Message Prioritization

The delivery of messages like OTP may be prioritized over other message requests by setting a high-volatile priority. The following example shows a request for a high priority confidential OTP message:

http://localhost:8021/send?receiverid=xyz@nic.in&msg=Your%20OTP%20is%20123456&priority=high-volatile

See section 4 for other message priorities.

5.4. Check Message Status

The status of a message dispatch, delivery and read can be checked using the following API:

http://localhost:8021/status?reqid=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

The reqid is the message request ID received in the response of a message request.

The message status are made available instantly after dispatch, delivery or read activity.

5.5. Get Messaging Statistics

To check the total number of messages dispatched between any two given dates, the following API may be used:

http://localhost:8021/stats?from_date=yyyy-mm-dd&to_date=yyyy-mm-dd

5.6. Confidential File Sharing

A file can be confidentially shared to any registered Sandes user. The gateway client shall be end-to-end encrypted the file at your server side before transferring to the Sandes app of the registered Sandes user.

To share a file confidentially to a single receiver, the file may be submitted either as the body of the request or in multipart/form-data parameter named **file**:

POST
http://localhost:8021/send/file?receiverid=9999999999&msg=Confidential%20File&filename=confidential.pdf

- msg – The message that accompanies the file shared.
- filename – The file name with extension

The above request will upload the file and share it to the receiver.

5.6.1. Sharing a single file to multiple receivers

To share a confidential file to multiple receivers, the file must be uploaded only once to get a file id. The file may be submitted either as the body of the request or in multipart/form-data parameter named **file**:

POST http://localhost:8021/upload?filename=confidential.mp4

Note:-The message or the receivers need not be specified while uploading the file. The file must be set as the body of the POST request.

Response:

```
{
   "file_id": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}
```

Now the file id may be used to send the same file to any number of recipients by using the following file sharing request:

POST
http://localhost:8021/send/file?receiverid=9999999999&msg=Confidential%20File&fileid=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

5.7. Deploying as a Windows Service

1. Make sure that the gims-gateway-cli-srv.exe is running in command line mode and able to send messages.
2. Stop the gateway client service running in command line.
3. Run win64/install-service.bat to install the windows service. The service would be installed as "Sandes Gateway Client Service"
4. To stop and uninstall the service run win64/uninstall-service.bat
5. The log file (gims-gateway-cli-srv.log) for the service would be created in the app folder.

5.8. Deploying as a Linux Service

1. Make sure that the gims-gateway-cli-srv.exe is running in terminal mode and able to send messages.
2. Stop the gateway client service running in terminal.
3. Run linux/install-service.sh to install the systemd service. The service would be installed as "sandes-gateway-cli-srv"

   sudo sh install-service.sh

4. To stop and uninstall the systemd service run win64/uninstall-service.bat:

   sudo sh uninstall-service.sh

5. Installed paths are given below:
   - Logs: /var/log/gims-gateway-cli-srv/gims-gateway-cli-srv.log
   - Configuration: /etc/gims-gateway-cli-srv/cmd.json
   - Binary: /usr/local/bin/gims-gateway-cli-srv

5.9. Deploying on Container

1. To build the Sandes gateway client service container image, run the following in bin/docker

   docker build -t gims-gateway-cli-srv:1.7 .

   (Note: - 1.7 is the version of Gateway Client Service)

2. Update your client ID, client secret and HMAC key in .env file in bin/docker
3. Create a log directory (/var/log/gims-gateway-cli-srv) in the host machine with write permission.
4. Run the following from bin/docker to start Sandes gateway client service in a container:

   docker run --publish 8021:8021 -v /var/log/gims-gateway-cli-srv:/app/log --env-file .env gims-gateway-cli-srv:1.7

5.10. Enabling Multitenancy Support

The gateway client service can run in multi-tenancy mode to allow the gateway account configuration to be passed in the request header.

Run the gateway client service in multitenancy mode by setting the **MultiTenantClient** to **true** in **cmd.json**. In this mode, all API requests from tenants would require the following parameters in the request header:

clientid: xxxxxxxxxxxxxx

clientsecret: xxxxxxxxxxxxx

hmackey: xxxxxxxxxxxxxxx

Note: - The ClientID, ClientSecret and HMACKey configuration in the cmd.json is mandatory and must be set to the parent account credentials of the multi-tenant application.