Ministry of Housing and Urban Affairs
Government of India

Ministry of Housing and Urban Affairs
Government of India

National Urban Digital Mission

Building cities that work for people

SMS Gateway

Outline

1. Using the generic GET & POST Gateway interface
2. Developing custom Interface

Ministry of Housing and Urban Affairs
Government of India

Using Get/Post sms gateway

Notification SMS Service - Create a common point to manage all the SMS notifications being sent out of the platform. Notification SMS service consumes SMS from the Kafka notification topic and processes them to send it to a third party service. Modules like PT, TL, PGR etc make use of this service to send messages through the Kafka Queue.

Configuration Details :

This service is a consumer, which means it reads from the Kafka queue and does not provide the facility to be accessed through API calls. There is no REST layer here. The producers willing to integrate with this consumer post a JSON onto the topic configured at 'kafka.topics.notification.sms'.

The notification-sms service reads from the queue and sends the SMS to the mentioned phone number using one of the SMS providers configured. The implementation of the consumer is present in the directory src/main/java/org/egov/web/notification/sms/service/impl

These are current providers available

- Generic
- Console
- MSDG

Ministry of Housing and Urban Affairs
Government of India

Using Get/Post sms gateway

The implementation to be used can be configured by setting sms.provider.class.

Console implementation

The Console implementation just prints the message mobile number and message to the console.

Generic **implementation**

To use the generic GET/POST SMS gateway, first, configure the service application properties

```
sms.provider.class=Generic
```

This will set the generic interface to be used. This is the default implementation, which can work with most of the SMS Provider. The generic implementation supports below

- GET or POST based API
- Supports query params, form data, JSON Body

To configure the URL of the SMS provider use `sms.provider.url` property.

To configure the http method used configure the `sms.provider.requestType` property to either `GET` or `POST`.

To configure form data or json api set `sms.provider.contentType=application/x-www-form-urlencoded` or `sms.provider.contentType=application/json` respectively

Ministry of Housing and Urban Affairs
Government of India

Configuring Variables

To configure which data needs to be sent to the API below property can be configured:

- `sms.config.map`={'uname':'$username', 'pwd': '$password', 'sid':'$senderid', 'mobileno':'$mobileno', 'content':'$message', 'smsservicetype':'unicodemsg', 'myParam': '$extraParam' , 'messageType': '$mtype'}
- `sms.category.map`={'mtype': {'*': 'abc', 'OTP': 'def'}}
- `sms.extra.config.map`={'extraParam': 'abc'}

`sms.extra.config.map` is not used currently and is only kept for custom implementation which requires data that doesn't need to be directly passed to the REST API call

`sms.config.map` is a map of parameters and their values

Ministry of Housing and Urban Affairs
Government of India

Special variable needs to be mapped

- `$username` maps to `sms.provider.username`
- `$password` maps to `sms.provider.password`
- `$senderid` maps to `sms.senderid`
- `$mobileno` maps to `mobileNumber` from kafka fetched message
- `$message` maps to the `message` from the kafka fetched message
- `$<name>` any variable that is not from above list, is first checked in `sms.category.map` and then in `application.properties` and then in environment variable with full upper case and _ replacing -, space or .

So if you use `sms.config.map={'u':'$username', 'p':'password'}`. Then the API call will be passed

```
<url>?u=<$username>&p=password
```

Ministry of Housing and Urban Affairs
Government of India

Message Success or Failure

Message success delivery can be controlled using below properties

- `sms.verify.response` (default: false)
- `sms.print.response` (default: false)
- `sms.verify.responseContains`
- `sms.success.codes` (default: 200,201,202)
- `sms.error.codes`

If you want to verify some text in the API call response set `sms.verify.response=true` and `sms.verify.responseContains` to the text that should be contained in the response

Ministry of Housing and Urban Affairs
Government of India

Black Listing or whitelisting numbers

It is possible to whitelist or blacklist phone numbers to which the messages should be sent. This can be controlled using the below properties:

- `sms.blacklist.numbers`
- `sms.whitelist.numbers`

Both of them can be given a `,` separated list of numbers or number patterns. To use patterns use `X` for any UPYOG match and `*` for any number of UPYOGs match.

`sms.blacklist.numbers=5*,9999999999,88888888XX` will blacklist any phone number starting with `5`, or the exact number `9999999999` and all numbers starting from `8888888800` to `8888888899`

Steps to Integration

To integrate, create the SMS request body given in the example below.Provide the correct mobile number and message in the request body and send it to the kafka topic:- egov.core.notification.sms

```
{
  "mobileNumber": "9999999993",
  "message": "Hey, how you doing?"
}
```

The notification-sms service reads from the queue and sends the sms to the mentioned phone number using one of the SMS providers configured.

Ministry of Housing and Urban Affairs
Government of India

SMS formats and Template

https://docs.google.com/spreadsheets/d/1dUwDeCmKImwSI_SW1JQL-SijU7rSpUgM/edit#gid=510937816

Using the above template, the sms provider will provide templateid for each sms format. We are using these templateid along with localization keys to send SMS.

Thank You

“UPYOGal Transformation is more about humans than UPYOGal”