

Sales tracking via Slack

Objective:

To capture daily sales figures in a structured way from a large sales team who uses Slack for communication.

Implementation:

Prerequisite before using APIs:

1. Sign in to gupshup.io to get an API Key.
2. Get your slack token from <https://api.slack.com/web>
3. Save Slack Account with your API key using Attach a slack account.

URL Endpoint and HTTP Methods: Save Slack Account

```
PUT : http://api.gupshup.io/sm/api/slack/account
```

HTTP Request using Curl to save slack account :

```
curl -X PUT http://api.gupshup.io/sm/api/slack/account \  
-H "apikey: YOUR API KEY" \  
-H "Content-Type: application/x-www-form-urlencoded" \  
-d 'slackToken=YOUR SLACK TOKEN '
```

Response Code:

```
204 No Content
```

Actual implementation:

1. Create a form smart message that will enable us to take daily sales figures.

URL Endpoint and HTTP Methods: Creating a data capture form:

```
PUT : http://api.gupshup.io/sm/api/smartmsg/datacapture
```

HTTP Request using Curl to create a data capture form:

```
curl -X PUT http://api.gupshup.io/sm/api/smartmsg/datacapture \  

```

```
-H "apikey: YOUR API KEY" \  
-H "Content-Type: application/x-www-form-urlencoded" \  
-d 'question=What is your sales figure for today? \  
    &callbackUrl=http://requestb.in/se2qvrse&fields=input:Total'
```

Response body:

```
{  
  "id": "21",  
  "meta": {  
    "callbackurl": "",  
    "deleted": false,  
    "expired": false  
  },  
  "payload": {  
    "content": "What is your sales figure for today?",  
    "fields": [  
      {  
        "name": "Total",  
        "type": "input"  
      }  
    ],  
    "type": "datacapture"  
  }  
}
```

2. Once the form is created, generate a signed-link for user and get the embedlink using SignedLink API.

URL Endpoint and HTTP Methods: Generate a signed link for a smart message:

```
POST : http://api.gupshup.io/sm/api/smartmsg/msg/{id}/signedlink
```

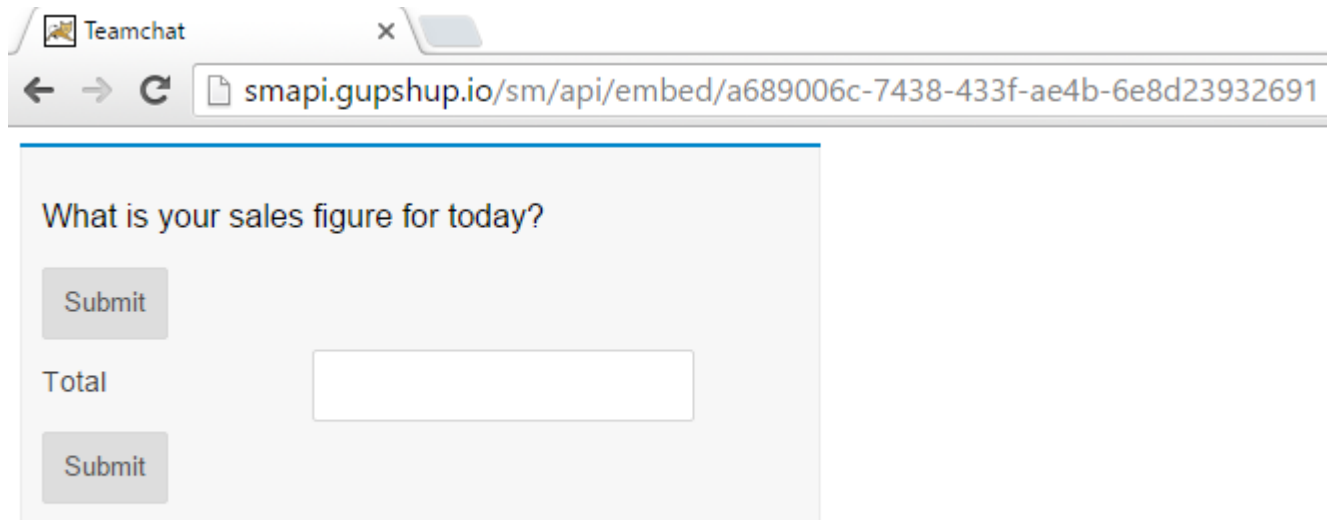
HTTP Request to generate a signed-link for user:

```
curl -X POST http://api.gupshup.io/sm/api/smartmsg/msg/21/signedlink \  
-H "apikey: YOUR API KEY" \  
-H "Content-Type: application/x-www-form-urlencoded"\  
-d 'destination = Target user'
```

Response body:

```
[  
  {  
    "embedlink":  
    "http://smapi.gupshup.io/sm/api/embed/a689006c-7438-433f-ae4b-6e8d23932691",  
    "expired": false,  
    "id": "a689006c-7438-433f-ae4b-6e8d23932691",  
    "signed-for": {
```

```
    "display": "Target user",
    "subdisplay": "Target user"
  },
  "smid": "21"
}
]
```



The screenshot shows a web browser window with the URL `smapi.gupshup.io/sm/api/embed/a689006c-7438-433f-ae4b-6e8d23932691`. The page content is a form with the question "What is your sales figure for today?". It features a "Submit" button, a "Total" label, an empty input field, and another "Submit" button.

Powered by gupshup

3. Now to send a message to slack use Send a simple text message on slack API

URL Endpoint and HTTP Methods: Send a simple text message on slack:

```
PUT : http://api.gupshup.io/sm/api/slack/msg
```

HTTP Request using Curl to send a simple text message on slack:

```
curl -X PUT http://api.gupshup.io/sm/api/slack/msg \
-H "apikey: YOUR API KEY" \
-d 'destination=Target slack user name \
    &text=To share your sales for today click here
http://smapi.gupshup.io/sm/api/embed/a689006c-7438-433f-ae4b-6e8d23932691 \
    &name=Sender Name' \
-H "Content-Type: application/x-www-form-urlencoded"
```

Message OUTPUT on slack.



Nik BOT 4:28 PM

To share your sales for today click here <http://smapi.gupshup.io/sm/api/embed/a689006c-7438-433f-ae4b-6e8d23932691>

Response Code: `` `response code 204 on success

```
4. &nbsp;&nbsp;  **Callback:** when you fillup the data capture form following parameters are pass to your callback URL in the form:
```

```
`` `http
```

```
GET :
```

```
http://requestb.in/se2qvrse?&response={"Total":"32423"}&user=Target+User&linkId=a689006c-7438-433f-ae4b-6e8d23932691
```

Pseudo code for capturing data on submitting the form:

```
1. getParameter("user");
2. getParameter("response");
3. JSONObject responseJsonObject = new
4. JSONObject(response);
5. String Total = (String) responseJsonObject.get("Customer Name");
6. You can store this data at backend and manipulate as you like.
```