


```
&imageUrl= Thumbnail for Group'
```

Response body: ``JSON String { "creator": { "userId": "33910" }, "developerkey": "", "groupRequest": { "adminOnly": false, "createdDateTime": 1454404312824, "groupId": "56b072d8ae0b18cf2b539e06", "imageUrl": "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQABAAD/2wCEAAkGBxQTEhUSExMW", "memberProfileHidden": false, "name": "In-app messaging.", "type": "n" }, "transactionId": "3391096175215312623ce4e56b0ff42011189cb31" }

```
**Response Code:**  
``Response code  
200
```

2. Creating a form that can take the details - complaint no, caller's name, address, etc.

URL Endpoint and HTTP Methods: GroupForm

```
PUT : http://api.gupshup.io/inapp/api/group/{Group ID}/messages/form
```

HTTP Request using Curl to create a form:

```
curl -X PUT  
http://api.gupshup.io/inapp/api/groups/56b072d8ae0b18cf2b539e06/messages/form \  
-H "apikey: YOUR API KEY" \  
-H "Content-Type: application/x-www-form-urlencoded" \  
-d 'groupId=56b072d8ae0b18cf2b539e06 \  
&callbackUrl= http://requestb.in/1bcn97q1 \  
&text=Fill in details of the complaints received \  
&detailsEnabled=true \  
&fields=  
  [{  
    "type": "input",  
    "label": "Complaint no.:",  
    "key": "complaintNo"  
  }, {  
    "type": "input",  
    "label": "Name:",  
    "key": "customerName"  
  }, {  
    "type": "input",  
    "label": "Address",  
    "key": "address"  
  }, {  
    "type": "input",  
    "label": "Station no.:",  
    "key": "stationNo"  
  }]  
'
```

Response body: ``JSON String { "creator": { "userId": "33910" }, "developerkey": "", "messageRequest": { "callbackUrl": "http://requestb.in/1bcn97q1", "category": "message", "createdDateTime": 1454480433741, "formId": "56b19c31e4b04445822ec9a4", "groupId": "56b072d8ae0b18cf2b539e06", "subType": "form", "type": "form" }, "transactionId": "33910916523486369526708475e9c4a9ae98e7de9f6" }

```
**Response Code:**  
``Response code  
200
```

Form Output: Chatlet and Form.

Posted on: 03 Feb 2016 11:50 AM IST
Fill in details of the complaints received

REPLY | 1 Replies

← Reply ✓

Complaint no.:
AF293

Name:
Shawn Smith

Address
3 Bethesda Metro Center Suite 700, Bethesda
MD 20814

Station no.:
ZONCXMA

q w e r t y u i o p
a s d f g h j k l
↑ z x c v b n m ✕
?123 , . ←

3. Form callback response: Based on response to your form, url in your call back will receive a JSON payload. using that payload you can design further workflow

```

seqId: 03aa10d6-2b7b-404a-89c0-b9b39cc15527
payload: {
  "category": "message",
  "transactionId": "33910916523486369526708475e9c4a9ae98e7de9f6",
  "values": ["AF293", "Shawn Smith", "3 Bethesda Metro Center Suite 700, Bethesda
MD 20814", "ZONCXMA"],
  "subType": "form",
  "roomId": "56b072d8ae0b18cf2b539e06",
  "type": "reply",
  "responder": {
    "email": "responder@domain.com",
    "userId": "33910",
    "name": "Support Member 1"
  },
  "creator": {
    "email": "creator@domain.com",
    "userId": "33910",
    "name": "Support Member 1"
  },
  "formid": "56b19c31e4b04445822ec9a4"
}

```

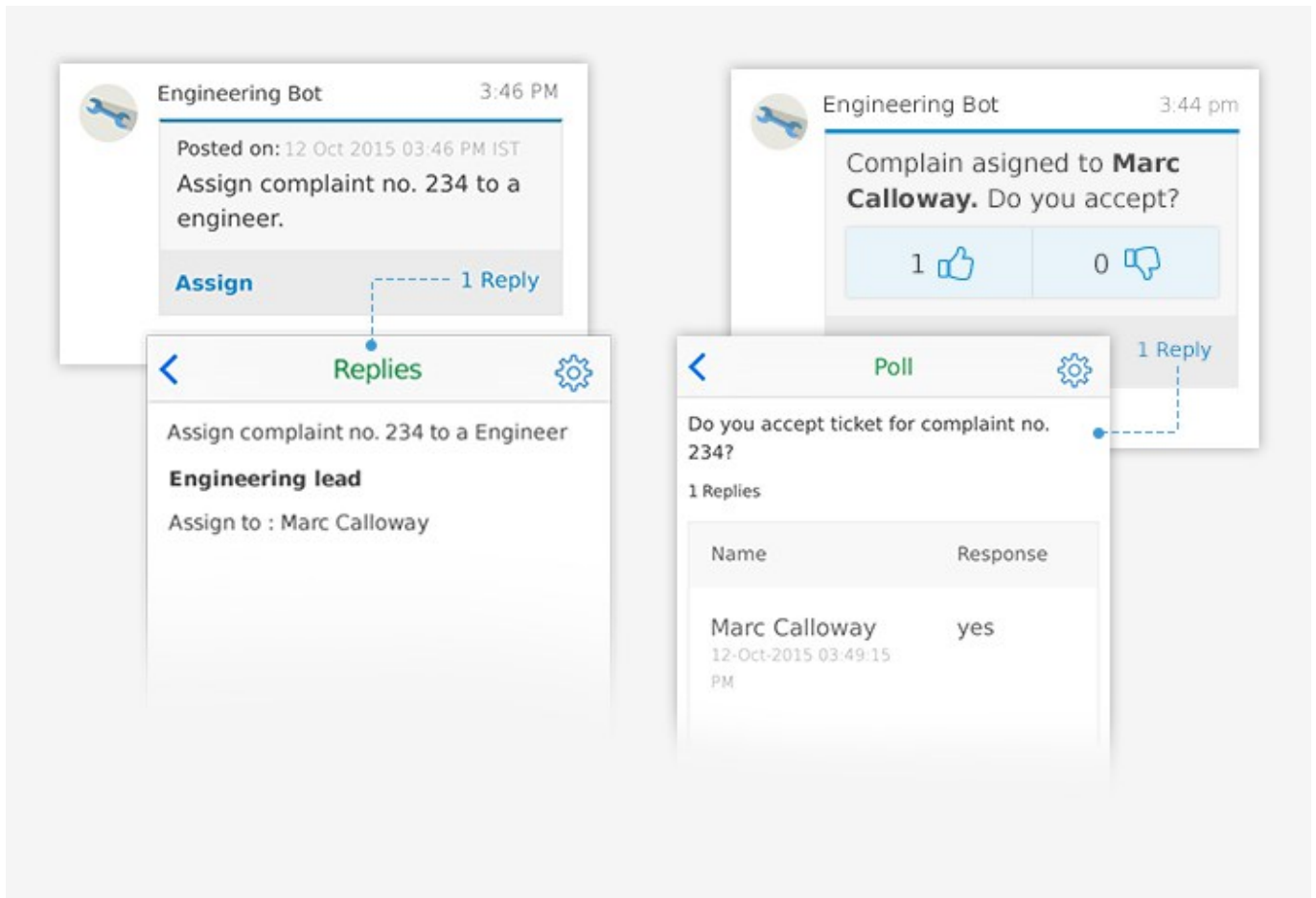
Pseudo code for forwarding the details with the caller's name, address, etc automatically to the free engineer nearest to the customer location.:

```

1. String seqId = getParameter("seqId");
2. String payload = getParameter("payload");
3. payloadJsonObject = new JSONObject(payload);
4. String[] formValues = responseJsonObject.getString("values").split("\\");
5. String complaintNumber = formValues[1].toString();
6. String address = formValues[2].toString();
7. Write a BOT to mine the address and retrieve the free engineer nearest to the
address and send a poll - Do you accept ticket for complaint no. 234?
6. Sending poll to Marc Calloway asking him for his willingness to accept ticket.
String url = http://api.gupshup.io/inapp/api/p2p/messages/poll
HttpResponse<String> data = Unirest
    .put(url)
    .header("apikey", YOUR API KEY)
    .header("Content-Type",
"application/x-www-form-urlencoded")
    .field("toEmail","engineer@domain.com").asString()
    .field("text","Complain assigned to Marc
Calloway. Do you accept?").
field("callbackUrl","http://requestb.in/1bcn97q1");

```

The service engineer accepts the task, shares location upon reaching the customer site and fills in the update once the issue is resolved.



4. **Poll callback response:** Based on response to your poll, URL in your call back will receive a JSON payload.

```
seqId: ec35ae8d-92a0-4c4d-9a30-d22fea64dbbe
payload: {
  "category": "message",
  "transactionId": "339109253452140535602b5497390defdd3c743284e",
  "values": ["yes"],
  "subType": "poll",
  "roomId": "56b2f40ae9a1e402456de57e",
  "type": "reply",
  "responder": {
    "email": "engineer@domain.com",
    "userId": "34967",
    "name": "Marc Calloway"
  },
  "creator": {
    "email": "creator@domain.com",
    "userId": "33910",
    "name": "Engineering lead"
  },
  "formid": "56b2f4c9e4b04445822efbe0"
}
```

Pseudo code for scenarios where the service engineer accepts the task, shares location upon reaching the customer site and fills in the update once the issue is resolved.:

```

1. String seqId = getParameter("seqId");
2. String payload = getParameter("payload");
3. payloadJsonObject = new JSONObject(payload);
4. String[] pollValues = responseJsonObject.getString("values").split("\\");
5. String pollResponse = pollValues[0].toString();
6. Write a BOT which will send a chatlet to Marc Calloway based on his reply, if
he says yes go for point 7 or else go to 8.
7. Send a form to Marc Calloway asking him to Fill in the status of the complaint
String url = http://api.gupshup.io/inapp/api/p2p/messages/form
HttpResponse<String> data = Unirest
    .put(url)
    .header("apikey",YOUR API KEY)
    .header("Content-Type",
"application/x-www-form-urlencoded")

    .field("toEmail","engineer@domain.com").asString()

    .field("callbackUrl","http://requestb.in/1bcn97q1")
        .field("text","Fill in the status of the
complaint").asString()

    .field("fields","formFields.json").asString();
    **Note: extract the JSON using a reader
class.
8. Send a form to Marc Calloway asking him reason for denial.
String url = http://api.gupshup.io/inapp/api/p2p/messages/form
HttpResponse<String> data = Unirest
    .put(url)
    .header("apikey", YOUR API KEY)
    .header("Content-Type",
"application/x-www-form-urlencoded")

    .field("toEmail","engineer@domain.com").asString()

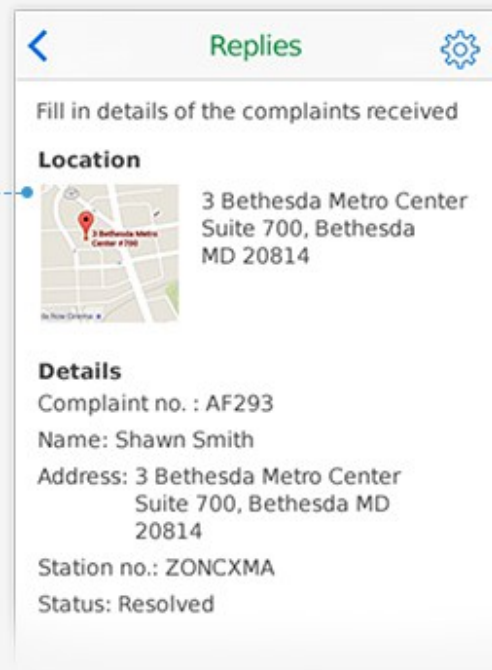
    .field("callbackUrl","http://requestb.in/1bcn97q1")
        .field("text","Please comment your reason
for denial").asString()

        .field("fields","[{\\"type\\":
\\"input\\",\\"label\\": \\"Comment:\\",\\"key\\": \\"comment\\"}]").asString();

```

JSON File - formFields.json Json File [{ "type": "location", "label": "Enter your Location", "key": "customerlocation" }, { "type": "input", "label": "Complaint no.:", "key": "complaintNo" }, { "type": "input", "label": "Name:", "key": "name" }, { "type": "input", "label": "Address:", "key": "address" }, { "type": "input", "label": "Station no.:", "key": "stationNo" }, { "type": "select", "label": "Status", "options": "Resolved,Unsolved,In-process", "key": "country" }]

The customer support team as well as the field service managers received real-time status reports of all activities. An automated leaderboard identifies the top engineers that have performed the most service requests during the day.



Results:

1. The support and servicing process of this organization is now streamlined and efficient.
2. Optimized service scheduling led to lower response and resolution times, increasing customer satisfaction.
3. Network downtime decreased as issues were resolved instantly.
4. Based on an analysis of the detailed, real-time tracking data identified weak spots where the services needed to be improved.
5. Employee morale of the field service engineers improved greatly as they received peer recognition.