

## Consumer banking via Twitter

### Objective:

Integrating Smart Message with Twitter to know your account balance.

### Implementation:

Prerequisite before using APIs:

1. Sign in to gupshup.io to get an API Key.
2. Get your Twitter consumerkey, consumerSecret, accessToken and accessTokenSecret from <https://apps.twitter.com/>
3. Save Twitter Account with your API key using Add a Twitter App account details

### URL Endpoint and HTTP Methods: Add a Twitter App account details

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

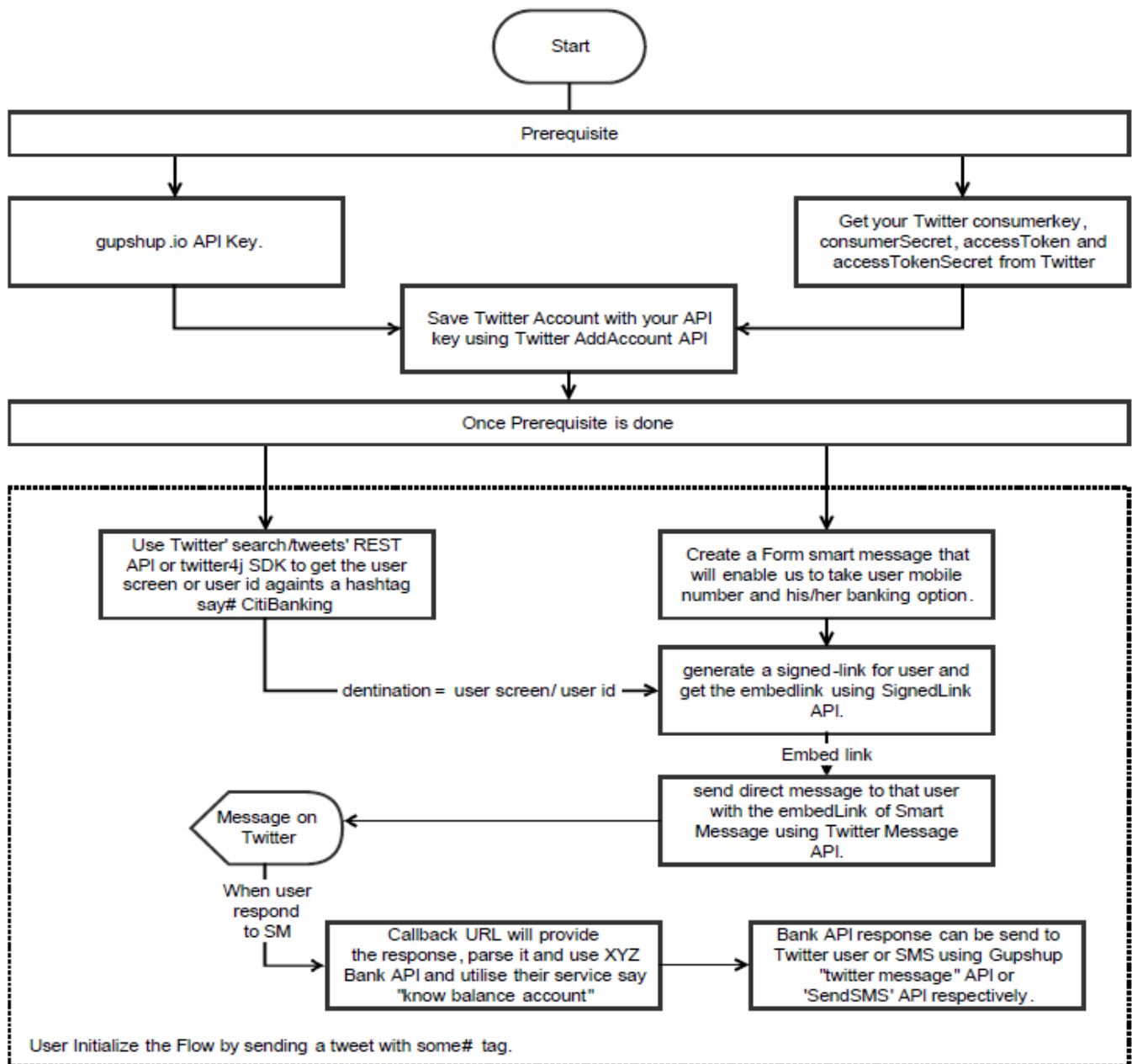
### HTTP Request using Curl to Add a Twitter App account details :

```
curl -X PUT http://api.gupshup.io/sm/api/twitter/account \  
-H "apikey: YOUR API KEY" \  
-H "Content-Type: application/x-www-form-urlencoded" \  
-d 'consumerkey=YOUR CONSUMER KEY \  
    &consumerSecret=YOUR CONSUMER SECRET \  
    &accessToken=YOUR ACCESS TOKEN \  
    &accessTokenSecret=YOUR ACCESS TOKEN SECRET'
```

### Response Code:

```
204
```

### Workflow diagram:



Actual implementation:

- Use Twitter 'search/tweets' REST API or twitter4j SDK to get the user screen or user id againts a hashtag say #CitiBanking and send a direct message using Gupshup "Send a twitter direct message" API
 

```

twitter sdk ConfigurationBuilder cb = new ConfigurationBuilder();
cb.setDebugEnabled(true).setOAuthConsumerKey("Your-consumer-key")
.setOAuthConsumerSecret("Your-Consumer-Secret")
.setOAuthAccessToken("Your-Access-Token")
.setOAuthAccessTokenSecret("Your-Access-Token-Secret");
Twitter twitter = new TwitterFactory(cb.build()).getInstance();
Query query = new Query(hashtag);
QueryResult result = twitter.search(query);
long userId = b.getUser().getId();// user Id
String userName = b.getUser().getName(); // user name

```

2. Create a form smart message that will enable us to take user mobile number and his/her banking option.

```
**URL Endpoint and HTTP Methods: Creating a data capture form:**  
```http  
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=Provide your mobile number and your choice of banking \  
    &callbackUrl=http://requestb.in/se2qvrse \  
    &fields=input:Mobile Number;select:Banking Service>Last Transaction:Desposit  
cheque status:Account Balance'
```

### Response body:

```
{  
  "id": "22",  
  "meta": {  
    "callbackurl": "http://requestb.in/se2qvrse",  
    "deleted": false,  
    "expired": false  
  },  
  "payload": {  
    "content": "Provide your mobile number and your choice of banking",  
    "fields": [  
      {  
        "name": "Mobile Number",  
        "type": "input"  
      },  
      {  
        "name": "Banking Service",  
        "options": [  
          "Last Transaction",  
          "Desposit cheque status",  
          "Account Balance"  
        ],  
        "type": "select"  
      }  
    ],  
    "type": "datacapture"  
  }  
}
```

3. 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:

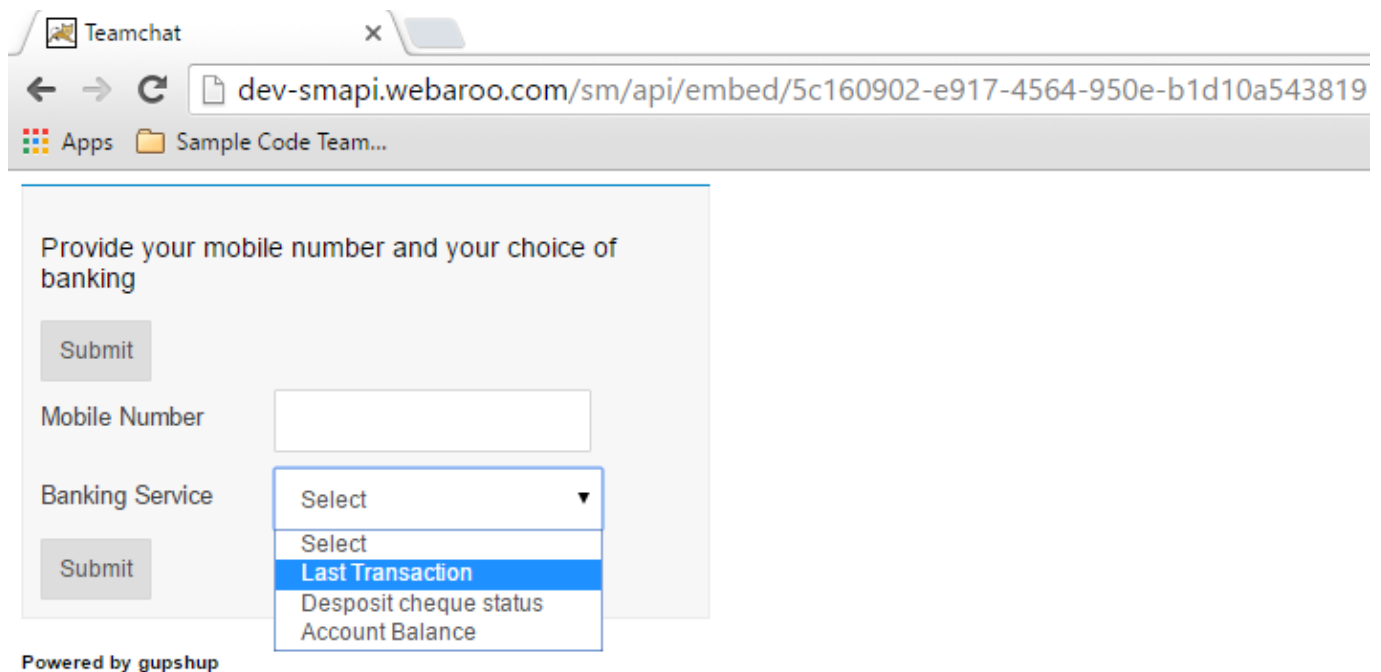
```
POST : http://api.gupshup.io/sm/api/smartmsg/msg/{id}/signedlink
```

## HTTP Request to generate a signed-link for user:

```
curl -X POST http://api.gupshup.io/sm/api/smartmsg/msg/22/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/08a7a0dc-0f98-45c3-ae63-6809f4636435",
    "expired": false,
    "id": "08a7a0dc-0f98-45c3-ae63-6809f4636435",
    "signed-for": {
      "display": "Target user",
      "subdisplay": "Target user"
    },
    "smid": "22"
  }
]
```



The screenshot shows a web browser window with the URL `dev-smapi.webaroo.com/sm/api/embed/5c160902-e917-4564-950e-b1d10a543819`. The page displays a form titled "Provide your mobile number and your choice of banking". The form has two sections. The first section has a "Submit" button, a "Mobile Number" input field, and a "Banking Service" dropdown menu. The dropdown menu is open, showing options: "Select", "Last Transaction", "Desposit cheque status", and "Account Balance". The second section has another "Submit" button. At the bottom left, it says "Powered by gupshup".

4. Once you get the screen name use it to send a direct message to that user with a smart message (Data capture form) embedded.

User must be your account follower in order to send him a direct message

URL Endpoint and HTTP Methods: Sending a twitter direct message:



8. Response gain from Bank API can be send to Twitter user on their phone using Gupshup SendSMS API or on their Twitter account using Gupshup "Sending a twitter direct message" API