

## Setup

Fetch postgres data dump from [SQL dump](#)

Above data contains two tables:

- a) account (id, auth\_id, username)
- b) phone\_number (id, number, account\_id (FK to account) )

Install Postgresql/DB of your choice.

Restore the dump into your DB server.

Install redis/in memory cache of your choice.

Implement the requirements to the best of your coding skills in languages/framework of your choice. Add unit tests and integration tests.

Deploy your code to AWS/Heroku/service provider your choice.

Create a github repository and share the link with us. Commit all your code in the master branch in this github repository when done.

Write a README.md with all instructions to setup, install and test the code.

## Build the API service

Write a micro service API server that exposes the following 2 APIs that accept JSON data as input to POST requests. If any other HTTP method is sent to the API, return HTTP 405.

### Authentication

For all API requests, Basic Authentication is required based on username and auth\_id as password.

If authentication is failing, return HTTP 403.

### API /inbound/sms/

Input parameters

**parameter required**

from (string min length 6 max length 16) true

#### **example**

91983435345

to (string min length 6 max length 16) true 14152243533

text (string min length 1 max length 120) true STOP, Hello World, ...

#### Expected API Behavior

- The API should do input validation
- If the 'to' parameter is not present in the phone\_number table for this specific account you used for the basic authentication, return an error (see Output JSON response below).
- When text is STOP or STOP\n or STOP\r or STOP\r\n The 'from' and 'to' pair must be stored in cache as a unique entry and should expire after 4 hours.

#### Output JSON response

If required parameter is missing:

```
{"message": "", "error": "<parameter_name> is missing"}
```

If parameter is invalid:

```
{"message": "", "error": "<parameter_name> is invalid"}
```

If 'to' is not found in the phone\_number table for this account: {"message": "", "error": "to parameter not found"}

Any unexpected error:

```
{"message": "", "error": "unknown failure"}
```

If all parameters are valid:

```
{"message": "inbound sms ok", "error": ""}
```

## **API /outbound/sms/**

Input parameters

### **parameter required**

from (string min length 6 max length 16) true to (string min length 6 max length 16) true

### **example**

919823243432

919343542749

text (string min length 1 max length 120) true hello from India

Expected API Behavior

- The API should do input validation
- If the pair 'to', 'from' matches any entry in cache (STOP), return an error (see Output JSON response below).
- Using cache, do not allow more than 50 API requests using the same 'from' number in 24 hours from the first use of the 'from' number and reset counter after 24 hours. Return an error in case of limit reached (see Output JSON response below).
- If the 'from' parameter is not present in the phone\_number table for this specific account you used for the basic authentication, return an error (see Output JSON response below).

Output JSON response

If required parameter is missing:

```
{"message": "", "error": "<parameter_name> is missing"}
```

If parameter is invalid:

```
{"message": "", "error": "<parameter_name> is invalid"}
```

If the pair 'to', 'from' matches any entry in cache (STOP):

```
{"message": "", "error": "sms from <from> to <to> blocked by STOP request"}
```

If 'from' is not found in the phone\_number table for this account: {"message": "", "error": "from parameter not found"}

If 50 requests in last 24 hours with same 'from' parameter: {"message": "", "error": "limit reached for from <from>"}

Any unexpected error:

```
{"message": "", "error": "unknown failure"}
```

If all parameters are valid:

```
{"message": "outbound sms ok", "error": ""}
```