

IM v3 PLATFORM API

Persona/Subscriber

INTRODUCTION

This document gives an outline of the interfaces for persona and subscriber integration with IM v2 Platform.

API REQUEST

Data is passed in JSON format for requests. The response will not return any data but will show success or error message. Interactions with the API are done via HTTP POST requests. Post request is use to update or add data in the system. There is only one POST value, \$_POST["data"], which has JSON data.

The basic form of the URL is:

`http:// im.nmsloop.com/api/users/<function>`

Functions

- Add
- Update

RESPONSE

Response is also in JSON for both error and success messages. If the authentication fails, transaction will not be processed and authentication error is displayed.

SUCCESS:

Success code is 0 for all requests.

```
{ "authentication":{"code":0, "message":"ok"}, "transaction":{"code":0, "message":"Success"}}
```

ERROR:

If there is an authentication error, the transaction will not proceed but will return a code and error message.

```
{"authentication":{"code":"000","message":"Error"}}
```

or

```
{"authentication":{"code":"0","message":"Ok"}, "transaction":{"code":"000", "message":"Error"}}
```

CODE	MESSAGE
500	API Limit Reached
501	Authentication error / IP not whitelisted
502	Invalid format. Check the API documentation
503	Data sent is not a valid json string
504	One or more of the required parameters have not been specified
505	Service does not exist
506	Persona/Subscriber not found in service
507	Persona/Susbscriber already exists in service
508	Subscriber does not exist
509	Persona does not exist
510	Invalid group
511	Invalid status
512	Invalid link
513	Invalid avatar link

Table 1. Error Messages

AUTHENTICATION

Authentication requires username and password. The system will only allow requests from sites which IP are listed on whitelist. The IP addresses that are included on the list are the ones specified under Loop IMv3 Services page.

ADD

This will add new profile. Name should be unique. Thus, adding name that already exists will generate error. The parameters are:

- username – aggregator's username
- password – aggregator's password
- service_code – the unique identifier of the services to be given to the aggregators
- name – unique persona/subscriber name
- group - classification of the user to be added (subscriber or persona)
- profile – an optional parameter for user information. Link and avatar will be placed here.
- membership – an optional parameter for subscribers only that will set the subscriber's status to free or billed. Default value is free.
- additional_info – an optional field with a set of additional parameters.

URL: <http://im.nmsloop.com/api/users/add>

POST: data => {username:"<username>",
password:"<password>",
service_code:"<service_code>",
name:"< name>",
group: "<group>",
profile:[{link: <link>, avatar: <avatar>, <key1>:"<value1>",<key2>:"<value2>"}]},
membership: "free" or "billed",
additional_info: {membership: "button", <parameter>: <value>}}

RESPONSE

SUCCESS:

```
{ "authentication":{ "code":0, "message":"ok"}, "transaction":{"code":0, "message":"Success"}}
```

ERROR:

Please refer to Table 1

UPDATE

This will update the persona's/subscriber's profile.

The parameters are:

- username – aggregator's username
- password – aggregator's password
- service_code – the unique identifier of the services to be given to the aggregators
- name – unique persona/subscriber name
- group - classification of the user to be added (subscriber or persona)
- status – persona/ subscriber status (active, inactive, delete)
- profile – an optional parameter for user information. Link and avatar will be placed here.

- membership – an optional parameter for subscribers only that will set the subscriber's status to free or billed. Default value is free.
- additional_info – an optional field with a set of additional parameters. This will override the additional info that is already existing. So to add more values, send the existing values and append the new ones.

URL: <http://im.nmsloop.com/api/users/update>

POST: data => {username:"<username>",
 password:"<password>",
 service_code:"<service_code>",
 name:"< name>",
 group: <group>,
 status: <status>,
 profile:{[link: <link>, avatar: <avatar>,
 <key1>:"<value1>",<key2>:"<value2>">]}, membership: "free" or "billed",
 additional_info: {membership: "button", <parameter>: <value>, <new_parameter>:
 <new_value> }}

RESPONSE

SUCCESS:

```
{ "authentication":{"code":0, "message":"ok"}, "transaction":{"code":0, "message":"Success"}}
```

ERROR:

Please refer to Table 1