

## Errors

As much as possible, the API will attempt to use the appropriate HTTP status codes to indicate the general class of problem.

400 Bad Request	Any case where a parameter is invalid or a required parameter is missing.
401 Unauthorized	When an authorized request is sent to the server.
404 Not Found	Endpoint does not exist.
405 Method Not Allowed	Attempting to use POST with a GET-only endpoint or vice-versa.
500 Internal Server Error	An error occurred on the server while processing the request. The request is probably valid but needs to be retried later.

Errors will usually include an error detail with technical information about what went wrong. This field is only intended to be use by the developer. In most cases, the server will include an "message", with is intended for the client to display back to the user directly.

## API Call success

200 Success	Any API call from server that will be successful will return a 200 as status Code.
----------------	--

## 1 -Associate Tag with Contacts

### Request URL:

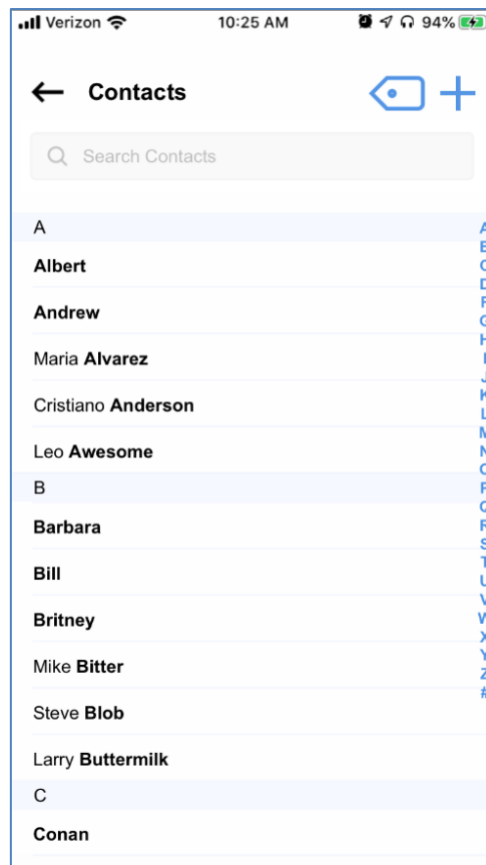
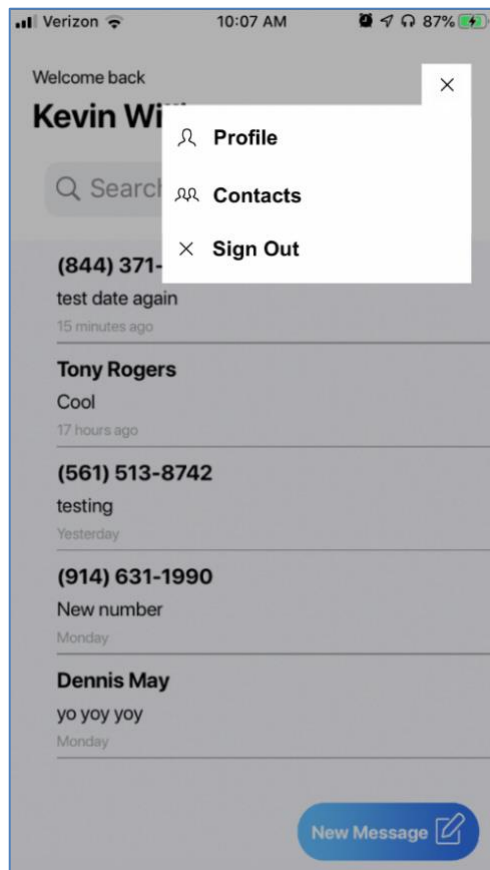
<b>URL</b>	/api/v1/contacts/tags
<b>Description</b>	Get the association of Contacts with Tags
<b>HTTP Method</b>	GET
<b>Note</b>	<p>Contacts API is already implemented (<i>api/v1/contacts?onlyWithChat=true</i>).</p> <p><b>Task 1:</b> Add Contacts into right menu option (graphics below and attached with email)</p> <p><b>Task 1.1:</b> Add Tags with contacts using above API and associate tags with contacts in DB.</p> <p><b>Task 2:</b> Create Contacts screen as below and add options into it.</p> <p><b>Task 2.1:</b> Contact Screen Search based on Contact details i.e. Contact name.</p> <p><b>Task 2.2:</b> Contact Screen Filter based on selected filter.</p>

### Sample Request:

```
{ }
```

### Sample Response:

```
[  
  {  
    "contactId": Integer,  
    "tagId": Integer  
  }  
]
```



## 2 -Tag Contact

### Request URL:

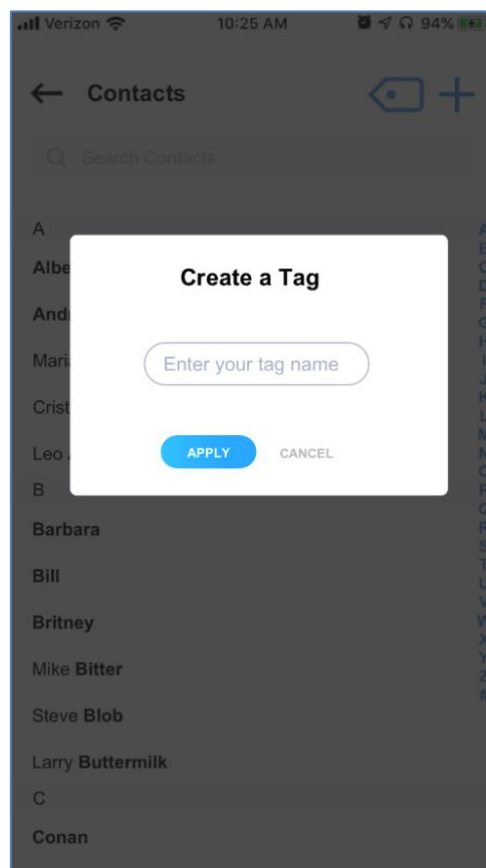
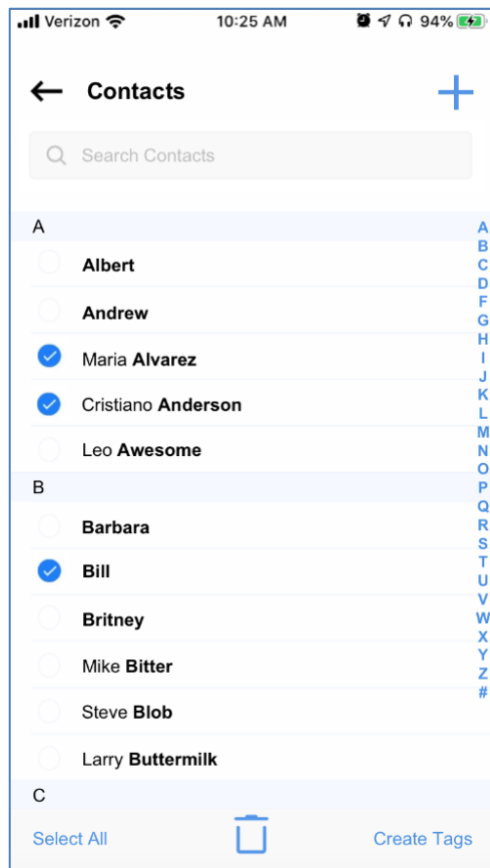
<b>URL</b>	/api/v1/tags/3643/contacts/CONTACTID,CONTACTID
<b>Description</b>	Add Tag to single/multiple Contacts
<b>HTTP Method</b>	POST
<b>Note</b>	<p>This API will be called once user select contact(s) from contact screen and then click on Create Tag button. First Create Tag Service will be called and then Tag will be associated with contacts using this webservice.</p> <p><b>Task 3:</b> If users long press on any contact and then below screen will appear with three options select (one by one) or select all from footer, Delete and Create Tag.</p> <p><b>Task 4:</b> If user will select contact(s) then two options Delete Contacts and Create Tag.</p> <p><b>Task 5:</b> Delete contacts will delete the contacts.</p> <p><b>Task 6:</b> Create Tag will create and also associate it with Contacts. This web service describe this item.</p>

### Sample Request:

```
{  
  This request will be called from URL.  
}
```

### Sample Response:

```
{  
}
```



### 3 -Delete Contacts

**Request URL:**

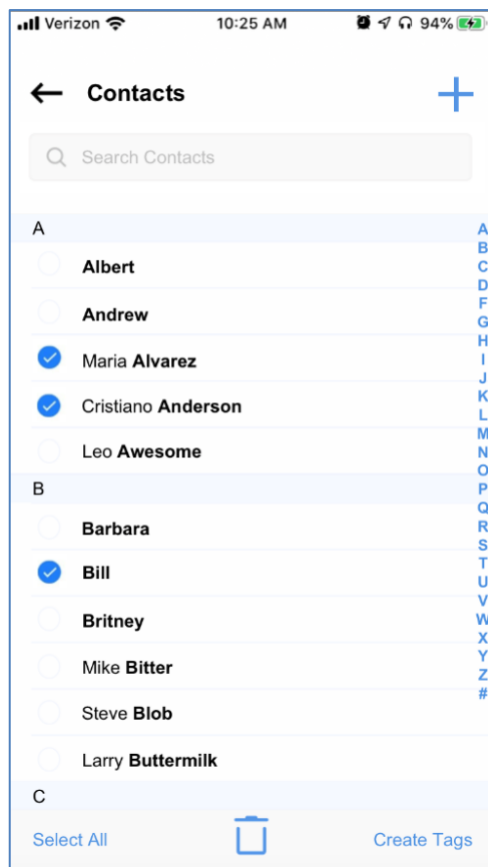
<b>URL</b>	/api/v1/contacts/CONTACTID,CONTACTID
<b>Description</b>	This API will delete the contact(s)
<b>HTTP Method</b>	DELETE
<b>Note</b>	<b>Task 7:</b> Delete API will remove the selected contacts from server and local DB also, need to refresh the Conversation List as the conversation related to deleted contacts will also be removed from server as well as local DB.

**Sample Request:**

```
{  
  This request will be called from URL.  
}
```

**Sample Response:**

```
{ }
```



## 4 -Upload Contact

### Request URL:

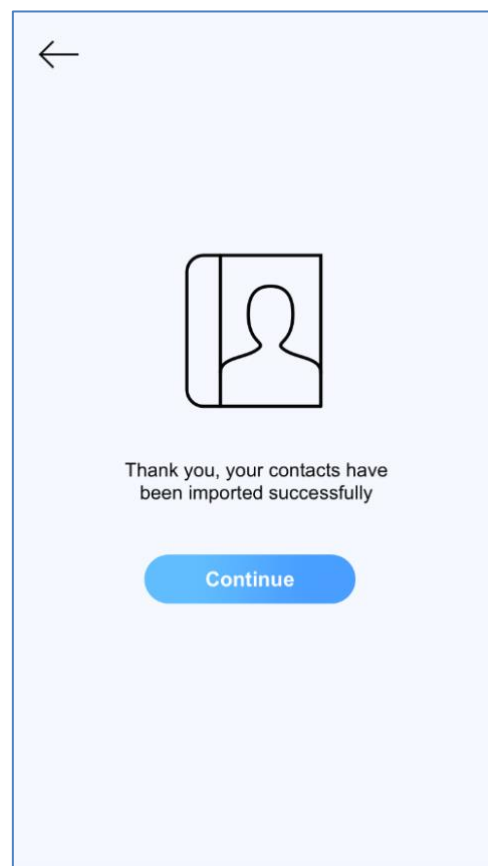
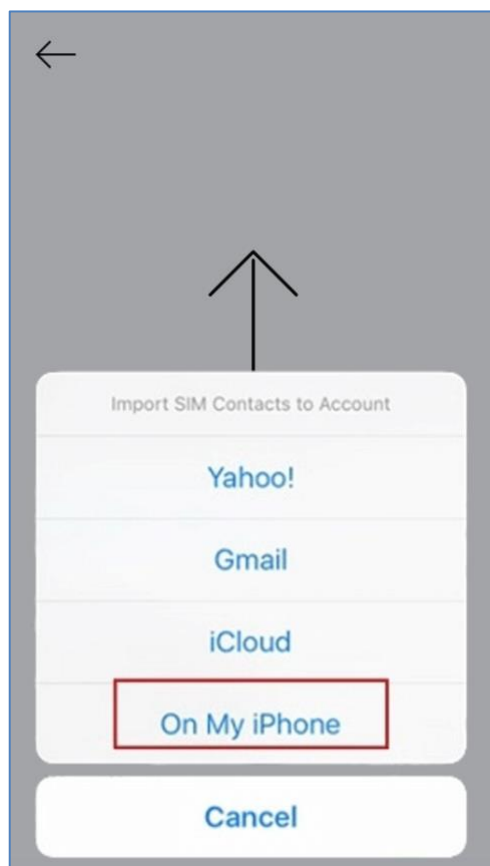
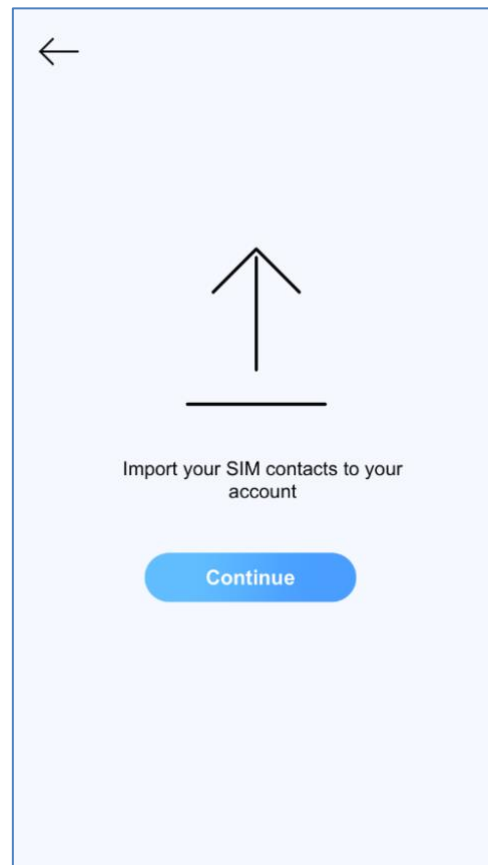
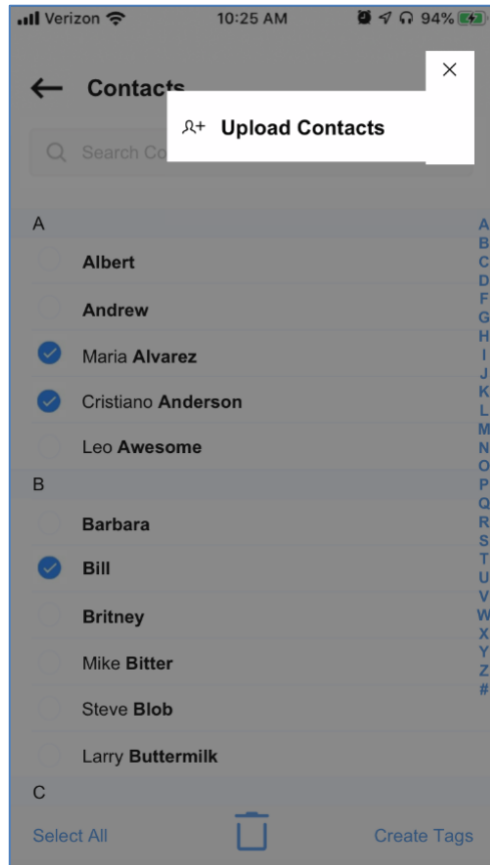
<b>URL</b>	/api/v1/contacts/upload
<b>Description</b>	This API will upload contacts in CSV file, template will be provided within email.
<b>HTTP Method</b>	POST
<b>Note</b>	<b>Task 8:</b> upload contact API will upload the contacts in CSV file. ONLY device contacts will be uploaded not Google etc.

### Sample Request:

```
{  
  CSV file upload to the stream, (formData)  
}
```

### Sample Response:

```
{  
  "totalNumbers": 0,  
  "totalValidNumbers": 0,  
  "duplicatedNumbers": 0,  
  "alreadyInList": 0,  
  "invalidNumbers": {},  
  "message": "string"  
}
```





## 5 -Create Tag API

### Request URL:

<b>URL</b>	/api/v1/tags
<b>Description</b>	This API will create Tag
<b>HTTP Method</b>	POST
<b>Note</b>	<b>Task 9:</b> User will click on contact and it will show contact details and in contact details user will be able to see contact details (same screen will be shown in Message menu option) and user will be able to create tags from here as well.

### Sample Request:

```
{
  "name": "String",
  "autoResponse": "String" Optional no need to send at this stage
}
```

### Sample Response:

```
{
  "name": "String",
  "autoResponse": "String",
  "id": Integer
}
```

Verizon 10:25 AM 94%

← Contact Details (561) 815-4881

First Name  
Steven

Last Name  
Rogers

Email  
srogers@test.com

Date of Birth  
04/23/1989

Gender  
M

Address  
Endgame St  
State  
FL  
Zip Code  
33406

Tags  
Add Thanos Technology Shield App Endgame

Save Details

Verizon 10:25 AM 94%

← Contacts

Search Contacts

Create a Tag

Enter your tag name

APPLY CANCEL

A  
Albe  
And  
Mari  
Crist  
Leo  
B  
Barbara  
Bill  
Britney  
Mike Bitter  
Steve Blob  
Larry Buttermilk  
C  
Conan

## 6 -Delete Tag API

### Request URL:

URL	/api/v1/tags/TAGID
Description	This API will delete Tag
HTTP Method	DELETE
Note	<b>Task 10:</b> User will click on cross on tag and the tag will be removed.

### Sample Request:

```
{
}
```

### Sample Response:

```
{
}
```

Verizon 10:25 AM 94%

← Contact Details (561) 815-4881

First Name  
Steven

Last Name  
Rogers

Email  
srogers@test.com

Date of Birth  
04/23/1989

Gender  
M

Address  
Endgame St

State  
FL

Zip Code  
33406

Tags

Add Thanos Technology

Shield App Endgame

Save Details