
How to build your own Call Centre

Learn how to build a call centre using CallHub's open APIs.



1st Edition

Introduction to CallHub	4
Why build your own Call Centre?.....	4
Overview	5
Getting Started with the Conference APIs	5
Steps to building a Call Centre	6
Setting up a campaign	7
<i>Guide to manager APIs</i>	<i>7</i>
Create an account.....	7
API key	7
Register WebHooks to get Call Centre Notes	8
Import contacts	8
Create Agents and Teams	8
Create your first Campaign	9
Building the Agent interface	10
<i>Guide to Agent APIs</i>	<i>10</i>
Login	10
Agent status	11
Agent details.....	12
Get Agent Campaigns.....	13
Get Campaign info	14
Create a Conference	15
Join a campaign	16
Get campaign script	17
Add the first and subsequent callers	18
Save the survey.....	19
Hangup the call	20
Leave a voicemail	20
More features	22

Add a supervisor22

SIP endpoints.....22

Call Centre Application23

Javascript call centre app23

Introduction to CallHub

Before the surge of internet telephony, call centres were typically a room full of phone lines connected to a physical PBX with agents using a phone line to make calls. The PBX was expensive to own and maintain. With the advent of modern technology and telephony there are now cloud based call centre available, like CallHub's Call centre software. It removes the need for physical phone lines or even the physical presence of agents at the call centre. Agents can now call remotely from their homes with just an internet connection.

As easy as it is to use and setup, a cloud based call centre has a fixed user interface without the ability to customise screens to meet your specific requirements.

Why build your own Call Centre?

The ability to create your own call centre using API's gives you the freedom to design according to your needs while still taking advantage of all the features we have to offer.

- Design your own workflows
- Completely brand the service as your own.
- Create custom landing pages for your volunteers.
- Build the call centre on your own domain to maintain brand and reduce confusion for volunteers or agents.
- Create your own Volunteer or Agents signup pages.

With the launch of CallHub's conference APIs, you can now create your own call centre. Leverage features of CallHub like CRM integration, Agent and Campaign Analytics, Contacts and List Management to name a few.

Chapter 1

Overview

There are a couple of prerequisites you need to complete before using the APIs. You need to first create an account on CallHub and then create a Call Centre campaign through our interface.

Getting Started with the Conference APIs

CallHub APIs are neatly stacked into different sections in accordance with the major aspects of any call centre.

Create agents

Use our [Agent APIs](#) to create agents in your account when a person submits a form. This form can be hosted on any site. Here is an example.

Make Calls

Use our [Media APIs](#) to make calls. These APIs give an agent the ability to join a campaign and then make calls to the contacts on their call list.

View contact information

[Subscriber APIs](#) allow you to view details about the contact being called. Using them you can see all the information and history of the contact being called. This information is useful for the agent when talking to the contact.

Script and Questionnaire

Our [Script APIs](#) give you the ability to display the calling script and save replies to the questions asked.

Steps to building a Call Centre

To build a Call Centre using Callhub APIs, you need to implement the APIs in the following order.

Set up a campaign

1. Create an account on CallHub if you haven't already.
2. Create teams of agents - [API](#)
3. Create your agents - [API](#)
4. Create a Call Centre campaign in CallHub using the account you just created.

Build agent interface

Steps	API https://api.callhub.io
Login with agent account	POST /v2/agent-key/
Get agent status	GET /v2/agent-status/
Get agent details	GET /v2/user-details/
Display the list of campaigns assigned to the agent.	GET /v2/agent-campaigns/ GET /v2/campaign-info/:id
Create a conference	POST /v2/conference/
Get agent to join the conference	PUT /v2/conference/:id/agent/
Get the script/survey for the campaign	GET /v2/script-sections/:id
Make a call to the contact	PUT /v2/conference/:id/ subscriber/
Get subscriber information	GET /v2/current-subscriber/:id
Submit the answers to the survey	POST /v2/script-answers/:id
Hang up the call	DELETE /v2/conference/:id/ subscriber/
Leave the campaign	DELETE /v2/conference/:id/agent/

Setting up a campaign

Guide to manager APIs

Create an account

Go to CallHub.IO and create an account by choosing an email ID, username and password.

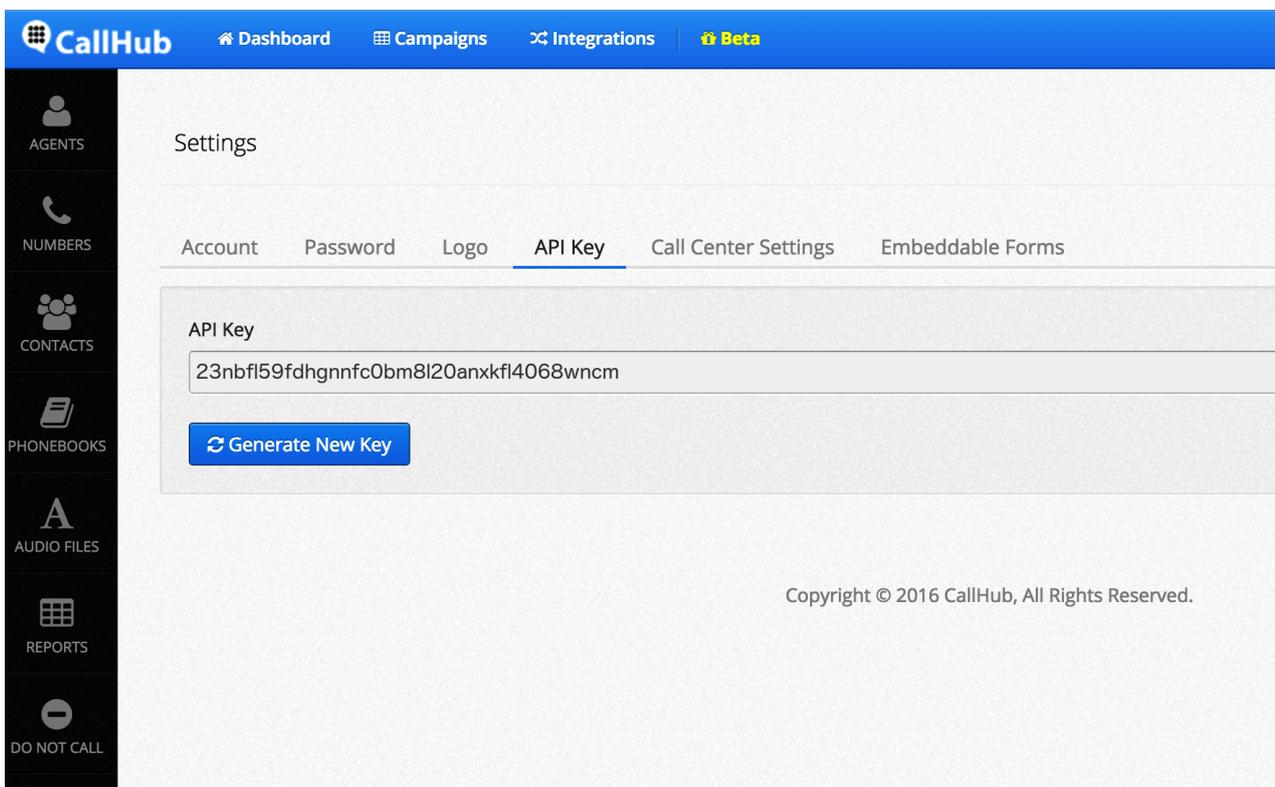
[Create an account](#)

Verify account and phone number

You will receive a verification mail from CallHub to verify your email ID. You will be asked to verify a phone number. This number will be used in the campaigns.

API key

After your account is created you are given a unique key code that you will use to talk to our APIs. You will find your API key under the settings tab. Copy the API key and keep it safe.



The screenshot shows the CallHub user interface. At the top is a blue navigation bar with the CallHub logo and menu items: Dashboard, Campaigns, Integrations, and Beta. On the left is a dark sidebar with icons for AGENTS, NUMBERS, CONTACTS, PHONEBOOKS, AUDIO FILES, REPORTS, and DO NOT CALL. The main content area is titled 'Settings' and has tabs for Account, Password, Logo, API Key (which is selected), Call Center Settings, and Embeddable Forms. Under the 'API Key' tab, there is a text box containing the key '23nbf159fdhggnfc0bm8l20anxkfl4068wncm' and a blue button labeled 'Generate New Key'. At the bottom right of the page, there is a copyright notice: 'Copyright © 2016 CallHub, All Rights Reserved.'

Register WebHooks to get Call Centre Notes

You can [register WebHooks for events](#) that you want to be notified of. One important event is Call Centre notes. When an agent saves a survey, the event **CC.Notes** occurs. This is a summary of the information you receive when the event occurs.

1. Survey responses
2. Call disposition
3. NationBuilder Tags
4. Notes saved by the agent
5. Contact address
6. Contact information
7. Contact job information
8. Link to the recorded conversation.

[WebHook documentation](#)

Import contacts

Contacts can be imported into CallHub in 3 ways:

- Through integration with NationBuilder, Salesforce, Google, Zapier etc
- Manually import contacts through a CSV
- Use [APIs](#) to create contacts and PhoneBooks in CallHub.

Your contacts will be grouped into PhoneBooks. A PhoneBook is a collection of contacts. You may have multiple PhoneBooks. These PhoneBooks will be assigned to campaigns.

Create Agents and Teams

CallHub lets you create an unlimited number of agents. To create an agent you can use the CallHub interface to manually create accounts for agents or you can implement our APIs that allow agents to register themselves through a web form. An agent will need a username, email ID and password. The email ID does not need to be unique across agents. This is especially useful if you as the campaign manager would like to manage agent login accesses.

Teams are used to logically group agents together. You can assign only 1 team to a campaign.

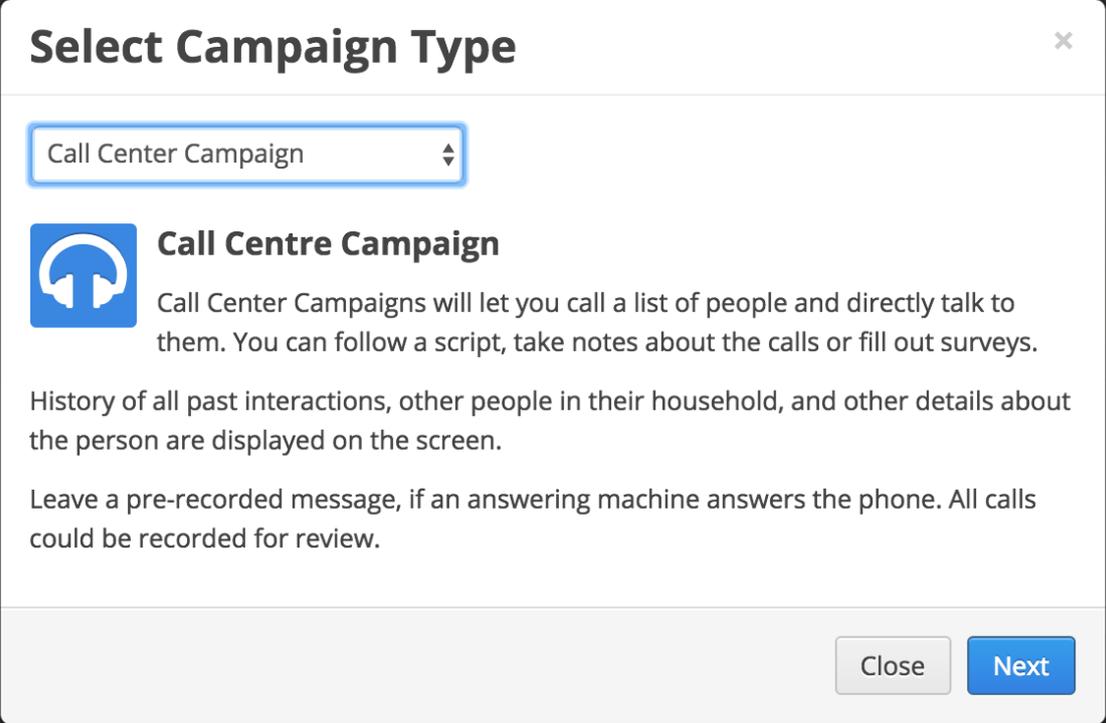
[Agent and Team APIs](#)

Create your first Campaign

Using CallHub's user interface, create your Call Centre campaign.

To create a call centre campaign you will need to:

1. Enter your Caller ID, Voicemail message, Script, Disposition and Survey Questions.
2. Select your phonebook/call list.
3. Assign Agents or Teams to the campaign.
4. Enter other settings relevant to your campaign such as Start and Stop time.



Select Campaign Type ×

Call Center Campaign ▾

 **Call Centre Campaign**

Call Center Campaigns will let you call a list of people and directly talk to them. You can follow a script, take notes about the calls or fill out surveys.

History of all past interactions, other people in their household, and other details about the person are displayed on the screen.

Leave a pre-recorded message, if an answering machine answers the phone. All calls could be recorded for review.

Close Next

Copyright © 2016 CallHub, All Rights Reserved.

Chapter 3

Building the Agent interface

Guide to Agent APIs

You have now built the foundation of the Call Centre. Next, you need to build an interface for your agents to make calls from.

For the purpose of this demo, all code samples are given in Javascript.

Login

To login an agent into the system, take the username and password provided and use the Agent-Key API to get an agent token in return. This TOKEN is what you will use to authenticate all future queries in this session.

POST /v2/agent-key/

```
$.ajax({
  url: 'https://api.callhub.io/v2/agent-key/',
  data: {
    username: 'agentusername',
    password: 'agentpassword'
  },
  method: 'POST'
}).done(function(data) {
  console.log('Token is: ', data.token);
});
```

Response <200 OK>

```
{"token": "dew2rr4wedewddfwerwec43asdf"}
```

Agent status

Check the status of the login before you proceed. Use the `/agent-status/` api to fetch their status information.

GET `/v2/agent-status/`

```
$.ajax({
  url: 'https://api.callhub.io/v2/agent-status/',
  headers: {
    // Use the token obtained from /v2/agent-key api
    'Authorization': 'Token ' + token
  },
  method: 'GET'
}).done(function(data) {
  console.log('Current Agent Status: ', data);
  // ID of the campaign the agent has joined. If null, then agent has
  // not joined a campaign
  // If this is not null, yet call_uuid is null, then agent is being
  // called
  console.log('Agent is connecting to the campaign: ',
data.connected_campaign);

  // If call_uuid is not null, agent is connected to the campaign
  console.log('Agent is connected to the campaign: ',
data.call_uuid);

  // If current_subscriber is not null, agent is calling a subscriber
  console.log('Agent is calling a subscriber: ',
data.current_subscriber);

  // If subscriber_uuid is not null, agent is currently on call with
  // a subscriber
  console.log('Agent is on call with a subscriber: ',
data.subscriber_uuid);

  // If supervisor_uuid is not null, a supervisor is connected to the
  // call
  console.log('Agent is on call with a supervisor: ',
data.supervisor_uuid);
});
```

Response <200 OK>

```
{
  "call_uuid": null,
  "connected_campaign": null,
  "current_subscriber": null,
  "subscriber_uuid": null,
  "supervisor_uuid": null
}
```

Agent details

Retrieve information about the agent such as their first name, last name, connection preference, phone number, logo and other settings related to the agent. Use the [GET API](#) to retrieve information and the [POST API](#) to save agent details.

GET /v2/user-details/

```
$.ajax({
  url: 'https://api.callhub.io/v2/user-details/',
  headers: {
    'Authorization': 'Token ' + token
  },
  method: 'GET'
}).done(function(data) {
  console.log('Agent Details: ', data);
});
```

Response <200 OK>

```
{
  "user": {
    "first_name": "Agent",
    "last_name": "Name",
    "email": "agent@email.com"
  },
  "agent": {
    "country": "US",
    "phone_no": "+1 55673411",
    "language": "en"
  }
}
```

Get Agent Campaigns

Get the IDs of all the campaigns the agent has been assigned to. Use this campaign **ID** for all future subscriber requests.

GET /agent-campaigns/

```
$.ajax({
  url: 'https://api.callhub.io/v2/agent-campaigns/',
  headers: {
    // Use the token obtained from /v2/agent-key api
    'Authorization': 'Token ' + token
  },
  method: 'GET'
}).done(function(data) {
  console.log('Agent Assigned Campaigns: ', data);
});
```

Response <200 OK>

```
{
  campaigns: [456, 457, 458]
}
```

Get Campaign info

Using the campaign IDs received from the previous API to get the campaign information of each campaign.

GET /campaigns-info/

```
// Array of Campaign IDs. These can be fetched using the /v2/
agent-campaigns API
var campaigns = [6, 7];
$.ajax({
  url: 'https://api.callhub.io/v2/campaign-info/?id=' +
  JSON.stringify(campaigns),
  headers: {
    'Authorization': 'Token ' + token
  },
  method: 'GET'
}).done(function(data) {
  console.log('Agent Campaigns Info: ', data);
});
```

Response <200 OK>

```
{
  "campaign_info": [
    {
      "status": 4,
      "assign_agents": true,
      "code": null,
      "name": "CallCentre Campaign, Aug 04, 22:29",
      "script": "Something",
      "invite_person": false,
      "incomingid": null,
      "totalcontact": 5,
      "team": null,
      "completed": 1,
      "voicemail_audiofile": null,
      "recording": true,
      "phonebook": [
        3
      ],
      "pending_notes": false,
      "created_date": "2016-08-05T05:29:50.855Z",
      "notes_required": false,
      "voicemail_detection": false,
      "id": 6
    }
  ]
}
```

Create a Conference

Use this API to create a conference instance that the agent and callers will join until the agent signs out.

POST /conference/:id

```
$.ajax({
  url: 'https://api.callhub.io/v2/conference/',
  data: {
    // Campaign ID of the campaign the agent is joining
    campaign_id: 6
  },
  headers: {
    // Use the token obtained from /v2/agent-key api
    'Authorization': 'Token ' + token
  },
  method: 'POST'
}).done(function(data) {
  console.log('Conference Details: ', data);
});
```

Response <200 OK>

```
{
  "recording": False,
  "current_subscriber": None,
  "supervisor": None,
  "member_count": 0,
  "voicemail": False,
  "campaign_id": 1463,
  "agent": None,
  "conference_uuid": "bcd6db96-ed5c-4581-b26e-d49d089fa5fd",
  "subscriber": [],
  "manager": None,
  "active": False,
  "last_updated": "2016-07-13T20:58:26.284659Z",
  "created_at": "2016-07-13T20:58:26.284653Z",
  "notes_required": True,
  "id": "9786ab72f8c1fd62d4368fae"
}
```

Join a campaign

Use CallHub conference APIs to add an agent to a conference. This conference is used throughout the agents session to make calls to contacts.

PUT /v2/conference/:id/agent/

```
// This value is the conference_uuid value returned when
creating a conference using /v2/conference
var conference_uuid = "aabbccdd";
$.ajax({
  url: 'https://api.callhub.io/v2/conference/' +
conference_uuid + '/agent/',
  data: {
    'endpoint': 'PHONE'
  },
  headers: {
    // Use the token obtained from /v2/agent-key api
    'Authorization': 'Token ' + token
  },
  method: 'PUT'
}).done(function(data) {
  console.log('Agent joined Successfully: ', data);
});
```

Response <200 OK>

```
{
  "success_msg": "Your phone number will be called in a few
seconds"
}
```

Get campaign script

Get the campaign survey to be displayed to the agent. Script will be shared with campaign information. Use the [results](#) to create a form.

GET /script-sections/:id

```
// This is the campaign the agent is getting the script for
var campaign_id = 6;
$.ajax({
  url: 'https://api.callhub.io/v2/script-sections/' +
  campaign_id + '/',
  headers: {
    // Use the token obtained from /v2/agent-key api
    'Authorization': 'Token ' + token
  },
  method: 'GET'
}).done(function(data) {
  console.log('Campaign Survey Sections: ', data);
});
```

Response <200 OK>

(Truncated: Go to [documentation](#) for complete response)

```
{
  "script_sections": [
    {
      "is_bound": false,
      "non_field_errors": [],
      "errors": {},
      "title": "MultiChoiceSurvey",
      "fields": {
        "question": {
          "widget": {
            "needs_multipart_form": false,
            "title": "RadioSelect",
            "input_type": "radio",
            "choices": [
              {
                "name": "question",
                "value": 12817,
                "display": "NO_ANSWER"
              } ...
            ]
          }
        }
      }
    }
  ]
}
```

Add the first and subsequent callers

Once the agent has joined the conference, you can start calling the contacts assigned to the campaign. Use this API to call the next contacts in the phonebook.

PUT /v2/conference/:id/subscriber/

```
# You'll need to install requests
import requests
# api_key can be found by using /v2/agent-key API
headers = {'Authorization': 'Token %s' % api_key}
data = {'campaign_id': '%d'%campaign_id}
conf = requests.post('https://api.callhub.io/v2/conference/',
data=data, headers=headers)
conf_uuid = conf.json()['conference_uuid']

# join this conference
#p_connection is connection preference of agent.
join_data = {'endpoint':p_connection}
r=requests.put('https://api.callhub.io/v2/conference/%s/
agent/'%conf_uuid,data=join_data,headers=headers)

#Calling the next available subscriber in the list
s=requests.put('https://api.callhub.io/v2/conference/%s/
subscriber/'%conf_uuid,headers=headers)
print 'Subscriber called' , s.json()
```

Response <200 OK>

```
{
  "success_msg": "Call Connected"
}
```

Save the survey

Save the filled in survey into the system.

POST /script-answers/:id

```
// This is the campaign the agent is currently connected to
var campaign_id = 6;
// The field name is a combination of the prefix for a field
obtained from the /v2/script-sections/ api and the string '-
question'
var script_answers = {
  '1234-question': 'Answer',
  '1235-question': 1
};
$.ajax({
  url: 'https://api.callhub.io/v2/script-sections/' +
campaign_id + '/',
  headers: {
    // Use the token obtained from /v2/agent-key api
    'Authorization': 'Token ' + token
  },
  method: 'GET'
}).done(function(data) {
  console.log('Campaign Survey Answers Submit: ', data);
});
```

Response <200 OK>

```
{
  "success_msg": "Survey submitted succesfully"
}
```

Hangup the call

Use the DELETE method on `/conference/:id/subscriber/` to end the call with a contact.

DELETE `/v2/conference/:id/subscriber/`

```
// This value is the conference_uuid value returned when
creating a conference using /v2/conference
var conference_uuid = "aabbccdd";
$.ajax({
  url: 'https://api.callhub.io/v2/conference/' +
conference_uuid + '/subscriber/',
  headers: {
    // Use the token obtained from /v2/agent-key api
    'Authorization': 'Token ' + token
  },
  method: 'DELETE'
}).done(function(data) {
  console.log('Hanging Up Subscriber: ', data);
});
```

Leave a voicemail

If you want to leave a voicemail instead, use this API to do so. After the message is played the connection with the contact is closed. You can now call the next contact in the list.

PUT `/conference/:id/subscriber/voicemail`

```
// This value is the conference_uuid value returned when
creating a conference using /v2/conference
var conference_uuid = "aabbccdd";
$.ajax({
  url: 'https://api.callhub.io/v2/conference/' +
conference_uuid + '/subscriber/voicemail/',
  headers: {
    'Authorization': 'Token ' + token
  },
  method: 'PUT'
}).done(function(data) {
  console.log('Dropping Voicemail: ', data);
});
```

Response <200 OK>

```
{  
  "success_msg": "Voicemail Dropped"  
}
```

More features

We've only taken you through the basic set of APIs that you need to call contacts in your phonebook. You can do a lot more with CallHub. Please go through our APIs for complete documentation of all available APIs.

Add a supervisor

Use the supervisor APIs to add **Warm Transfer** functionality to your Call Centre. At any time during a call with a contact, the agent can add a supervisor to the call and then transfer the call to the supervisor. This is useful for sales teams when a lead becomes warm and the agent would like the supervisor to take over the conversation and help close the lead.

PUT /conference/:id/supervisor

Use our conference APIs add a supervisor to the conference and even transfer the call to them.

SIP endpoints

The SIP endpoint APIs add functionality of letting an agent join a campaign using a SIP client instead of calling from a browser or phone.

GET /sip-endpoint/

Use the SIP endpoint details in a SIP client to receive calls on the client.

POST /sip-endpoint/

Connect agents to the conference using SIP endpoints.

Chapter 4

Call Centre Application

Javascript call centre app

We've built a complete mini Call Centre application in Javascript. This app should help you get started with creating your own Call Centre app.

[Get Code](#)