Contacts and Contact Groups API endpoints

Nov 04, 2025

API provided by the HappyFox helpdesk is a RESTful web service. It supports operations like creating a ticket, adding updates to a ticket, listing tickets and users of the Helpdesk. It supports JSON, Form Url encoded and Multipart Form Data formats as payload.

Requirements:

The API requires following skills in any programming language.

- Making HTTP requests (using GET and POST HTTP methods as a minimum requirement).
- Doing HTTP Basic Authentication.
- Generating and reading data in the JSON format.
- Optionally making HTTP POST requests using content type of "multipart/form-data" (needed for ticket attachments)

Documentation Conventions:

The documentation indicates parameters that need to be replaced with actual values using the format . The entire string including the enclosing < and > should be replaced.
For example, if the parameter *email* is it should be replaced with the required email address

Endpoint url format: /api/1.1/<module>/
The above tail url has to be suffixed with your HappyFox account url as shown below
Eg. https://customersuppport.happyfox.com/ (https://customersuppport.happyfox.com/)api/1.1/

All HTTP POST API calls are to be explicitly marked as **https**. Else, it would be considered as GET only

> *Note: If you are using a custom domain, please use the custom domain url*

> *API key and auth code are to be passed in Basic HTTP authentication format only.*

<module> - refers the module which is being accessed viz., /tickets/, /users/ etc A full constructed example url is shown below.
https://customersuppport.happyfox.com/ (https://customersuppport.happyfox.com/)api/1.1/json/users/

This knowledge base article consists of all operations that can be done via the users endpoint.

For HTTP POST methods, please ensure the IDs of the custom fields are as per /api/1.1/json/user_custom_fields/

List of topics covered in this article

> Note: All timestamps in API responses are in UTC

1. Get a list of contacts

API Endpoint: /api/1.1/json/users/

Method: GET

Example response:

```
{
   "page_info": {
      "count": 10,
      "last_index": 62,
      "page_count": 7,
      "start_index": 1,
      "end_index": 10
   },
   "data": [
      {
         "name": "John Doe",
         "primary_phone": {
            "type": "m",
            "number": "98765432",
            "id": 1
         },
         "phones": [
            {
               "type": "m",
               "number": "98765432",
               "id": 1
            }
         ],
         "created_at": null,
         "updated_at": null,
         "pending_tickets_count": 2,
         "contact_groups": [
            {
               "tagged_domains": "example.com",
               "access_all_tickets_in_group": false,
               "description": "Example contact group",
```

```
                "name": "sample 21",
                "id": 2
            }
        ],
        "tickets_count": 2,
        "id": 4,
        "email": "john@example.com",
        "custom_fields": [
            {
                "name": "A",
                "value": null,
                "value_id": null,
                "type": "text",
                "id": 13,
                "visible_to_staff_only": false
            }
        ]
    }
  ]
}
```


The default number of results is 10 and can be set to 50. Use the size and the page parameter as shown below, to get a paginated list of users.

/api/1.1/json/users/?size=50&page=1

The **page_count** field in the response contains the total number of pages when returning 50 results per page.

Back to top

2. Search contacts

API Endpoint: /api/1.1/json/users/?q=<search text>

Search fields:

- name
- email
- phone
- updated_since
- created_since

Notes for search fields:

- If multiple search query params are given, then contacts matching all the filters are returned.
- 2 search params are to be separated by space.
- Search field and value are to be in the format search field:search value
- For phone field an example search API query would be https://<acccount_name>.happyfox.com/api/1.1/json/users/?q=phone:11231231234 (https://newcompany.happyfox.com/api/1.1/json/users/?q=phone:11231231234), the user's actual phone number would be +11231231234. The '+' item should not be included in the search url.

Example search url:

```
https://customerservice.happyfox.com/api/1.1/users/?q=name:adam email:adam@example.com
```

Back to top

3. Contact detail page

API Endpoint:

Get contact by id: /api/1.1/json/user/<id of contact>/

Get contact by email address: /api/1.1/json/user/<email address>/

Method: GET

Examples:

https://acmewidgetsco.happyfox.com/api/1.1/json/user/33/

https://acmewidgetsco.happyfox.com/api/1.1/json/user/james@example.com/

Example Response:

```
{
    "name": "James may",
    "primary_phone": null,
    "phones": [],
    "created_at": null,
    "updated_at": null,
    "pending_tickets_count": 1,
    "contact_groups": [
        {
            "tagged_domains": "example.com,acme.com",
            "access_all_tickets_in_group": false,
            "description": "yet another sample contact group for testing.",
            "name": "sample 3",
            "id": 3
        }
    ],
    "tickets_count": 1,
    "id": 33,
    "email": "james@example.com",
    "custom_fields": [
        {
            "name": "A",
            "value": null,
            "value_id": null,
            "type": "text",
            "id": 13,
            "visible_to_staff_only": false
        }
    ]
}
```


Back to top

4. Add new contact/Edit existing contact

API Endpoint: /api/1.1/json/users/

Method: POST

Parameters:

Field	**Description**	**Required for new contact**
name	• name of contact. • string.	yes
email	• email address of contact. • Is validated as per our overall app email validation	yes, required even if phone is mentioned but can take null values

phones	• it is a list of dict having the following info
	  • id (in case of edit phone number)
	  • type - type of phone
	  • mo - mobile
	  • w - work
	  • m - main
	  • h - home
	  • o - other (default)
	  • number - the number itself as a string
	  • is_primary - indicate whether the number is to be set as the primary phone
		yes if email address is not given

c-cf-<id of custom field>	Contact custom fields Get the id of the custom fields from list contact custom fields api response. Custom fields that are passed in the payload will be updated and all other fields' values will be reset Value format: • text field: string • number: integer / float with 2 decimal places • dropdown: the id of the choice • multiple choice: list of ids of options • date: yyyy-mm-dd	yes for compulsory fields

Sample Requests:

Example 1 (create contact with phone numbers, custom fields)

```
{
    "email": "jamesmay@example1.com",
    "name": "dasdasd",
    "phones": [
        {
            "type": "mo",
            "number": "987654321",
            "is_primary": true
        },
        {
            "type": "mo",
            "number": "876543219",
            "is_primary": false
        }
    ],
    "c-cf-1": "example text field value"
}
```

Example 2 (edit phone number of a contact)

```
{
    "email": "jamesmay@example1.com",
    "phones": [
        {
            "type": "mo",
            "number": "987654321",
            "is_primary": true,
            "id": 20
        }
    ]
}
```

In this example, 20 is ID of the phone number attribute that is to be updated. Refer contact details page response for a HTTP 200 response code.

Failure Response Examples ()

```
{
    "error": [
        {
            "field": "name",
            "errors": [
                "This field is required."
            ]
        }
    ]
}
```

```
{
    "error": [
        {
            "field": "c-cf-13",
            "errors": [
                "This field is required"
            ]
        }
    ]
}
```

To edit one particular contact, please use the below shown method.

API Endpoint: /api/1.1/json/user/<ID_of_the_contact>/

HTTP Method: POST

Example Payload:

```
{
"name" : "John Smith",
"email" : "john.smith.new@example.com"
}
```

Custom fields also can be passed in the payload as ***c-cf-<id of the field>***. The response of the above API call would be same as a GET call to /api/json/1.1/user/<ID_of_the_contact>/

Note: Please use /api/1.1/json/users/ to update phone numbers.

Back to top

5. Add/Edit multiple contacts

API Endpoint: /api/1.1/json/users/

Method: POST

Example:

```
[
    {
    "email": "johnsmith@example.com",
    "name": "John Smith",
    "c-cf-4": "XHR1234"
    },

    {
    "email": "nathen@example.com",
    "name" : "Nathan",
    "c-cf-4": "LDA5000"
    }
]
```

Refer list of contact fields for the other fields that can be passed in the payload.

In the response received for the above API call, ***success - true*** indicates that the contact was added / edited and ***success - false*** indicates that there are some validation errors when trying to create / update contact for one of the contacts info in the payload. Maximum number of contacts that can be created / updated in one request is 100. Example response has been shown below

Example Response:

```
[
{
"email": "johnsmith@example.com",
"success": true,
"id": 14
},
{
"email": "nathen@example.com",
"success": true,
"id": 15
}
]
```

Back to top

6. Remove contact(s) from Contact Groups

API Endpoint: /api/1.1/json/contact_group/<contact_group_ID>/delete_contacts/

Method: POST

Example:

```
{
"contacts": [1, 3, 100]
}
```

Here 1, 3 and 100 are the contact ID of the contacts that are to be removed from the given contact group.

Example Response:

```
[
    {
        "data": {
            "message": "Successfully removed contact from group",
            "contact": 1
        },
        "success": true
    },
    {
        "data": {
            "message": "Contact not part of the contact group",
            "contact": 3
        },
        "success": false
    },
    {
        "data": {
            "message": "Contact does not exist",
            "contact": 100
        },
        "success": false
    }
]
```


Back to top

7. Enable/Disable login permission

This section explains how to enable or disable login access to your account's contact/support portal, restricting it to authorized users only.

API Endpoint:

/api/1.1/json/user/<ID of the contact>/

/api/1.1/json/user/email_address_of_the_contact/

Method: POST

Payload #1: Enable login permissions

```
{
"is_login_enabled" : "TRUE"
}
```

Payload #2: Disable login permissions

```
{
"is_login_enabled" : "FALSE"
}
```

Note: This parameter is supported for contact creation and contact update endpoints as well. Whenever a new contact is created, its defaults to **TRUE**

Back to top

8. Get a list of all contact groups

API Endpoint: /api/1.1/json/contact_groups/

Method: GET

Example response:

```
[
    {
        "tagged_domains": "example.com",
        "id": 1,
        "name": "test group",
        "description": "example description"
    },
    {
        "tagged_domains": "",
        "id": 2,
        "name": "test1 group",
        "description": ""
    }
]
```


Back to top

9. Get contact group details

API Endpoint:/api/1.1/json/contact_group/<id of the group>/

The id of the group can be obtained from the response of the previous API call where all contact groups are listed.

Method: GET

Example response:

```
{
    "tagged_domains": "example.com",
    "id": 1,
    "description": "example description",
    "name": "test group",
    "contacts": []
}
```

Back to top

10. Create a contact group

API Endpoint: /api/1.1/json/contact_groups/

Method: POST

Example request payload:

```
{
"name": "test group",
"description": "example description",
"tagged_domains": "example.com"
}
```

Payload fields:

Field	Description	Required
name	Name of the contact group. Name is unique	yes
description	Description about the contact group and the type of contacts	no

tagged_domains	list of domains to tag to the contact group so that when contacts with email addresses matching the domains are added into the account, they are auto added to the contact group	no

The response for this API call would be similar to the contact details endpoint.

Back to top

11. Edit contact group

API Endpoint: /api/1.1/json/contact_group/<id_of_the_contact_group>/

Fetch <id_of_the_contact_group> from /api/1.1/json/contact_groups/

Method: POST

Payload fields:

Field	Description	Required
description	Description about the contact group	no
tagged_domains	list of domains to tag to the contact group so that when contacts with email addresses matching the domains are added into the account, they are auto added to the contact group	no

Back to top

12. Add/edit contacts of a contact group

- Add multiple contacts to a contact groups.
- Maximum number of contacts that can be added in one request is 100.

API Endpoint: /api/1.1/json/contact_group/<id_of_the_group>/update_contacts/

Method: POST

Payload fields:

Field	Description	Required
contact	ID of the contact to add to the group	yes
access_tickets	true / false - Indicates whether the contact can access tickets of other contacts belonging to the group	no. defaults to false

Example payload:

```
[{
"contact": 41,
"access_tickets": true
},
{
"contact": 200
}
]
```

Example response:

```
[
    {
        "data": {
            "access_tickets": true,
            "contact": 1
        },
        "success": true
    },
    {
        "errors": [
            {
                "field": "contact",
                "errors": [
                    "Select a valid choice. That choice is not one of the available choices."
                ]
            }
        ],
        "success": false
    }
]
```


Back to top

13. List all contact custom fields

API Endpoint: /api/1.1/json/user_custom_fields/

Method: GET

Example Response:

```
[
{
"name": "Account Number",
"depends_on_choice": null,
"required": true,
"id": 4,
"choices": null,
"type": "text",
"order": 1,
"visible_to_staff_only": false
},
{
"name": "Region ID",
"depends_on_choice": null,
"required": false,
"id": 5,
"choices": [
{
"text": "APAC",
"id": 3,
"dependant_fields": []
},
{
"text": "Australia",
"id": 4,
"dependant_fields": []
},
{
"text": "Europe",
"id": 2,
"dependant_fields": []
```

},
{
"text": "North America",
"id": 1,
"dependant_fields": []
}
],
"type": "choice",
"order": 2,
"visible_to_staff_only": false
}
]

14. Update primary phone number of a contact

To update the primary phone number for an existing contact via the HappyFox API, follow these steps:

1. **Fetch the contact information** using the appropriate API endpoint (e.g., /api/1.1/json/user/<id of contact>/). Note the `id` of the primary phone number you wish to update.
2. **Prepare the JSON payload** with the existing phone `id` and the new phone `number`:

```
json
```

Payload:

```
{ "email": "", "phones": [ { "type": "w", "id": , "number": "" } ] }
```

Example Payload:

```
{ "email": "georgen@happyfox-test", "phones": [ { "type": "w", "id": 31, "number": "55
```

3. **Send the payload** to the appropriate API endpoint (e.g.,

`/api/1.1/json/user/<id_of_the_contact>/`) using the HTTP POST.

After above steps, the primary phone number for the contact should be updated with the new value. The same method can be used to update any component of the phone number field.