

[SurveyMonkey API](#)

[Overview](#)

[Getting Started](#)

[Public Apps](#)

[Private Apps](#)

[Registering an App](#)

[Deploying an App](#)

[Scopes](#)

[Request and Response Limits](#)

[Authentication](#)

[OAuth 2.0 Flow](#)

[Token expiration and revocation](#)

[Access URL](#)

[Unauthorizing an app](#)

[Quick Start Guide: Two Common Use Cases](#)

[Export the Results of a Survey](#)

[Send an Email Invitation Message](#)

[Pagination](#)

[Headers](#)

[Data Types](#)

[Error Codes](#)

[Help and Resources](#)

[API Endpoints](#)

[Default](#)

[Users and Teams](#)

[/users/me](#)

[/users/{user_id}/workgroups](#)

[/users/{user_id}/shared](#)

[/groups](#)

[/groups/{group_id}](#)

[/groups/{group_id}/activities](#)

[/groups/{group_id}/activities/{action_type}](#)

[/groups/{group_id}/members](#)

[/groups/{group_id}/members/{member_id}](#)

[/workgroups](#)

[/workgroups/{workgroup_id}](#)

[/workgroups/{workgroup_id}/shares](#)

[/workgroups/{workgroup_id}/shares/{share_id}](#)

[/workgroups/{workgroup_id}/shares/bulk](#)

[Survey Pages and Questions](#)

[/surveys](#)

[/surveys/{id}](#)

[/surveys/{id}/details](#)

[/surveys/{survey_id}/share](#)

[/surveys/{survey_id}/share_list](#)

[/survey_categories](#)

[/survey_templates](#)

[/team_survey_templates](#)

[/survey_languages](#)

[/surveys/{survey_id}/pages](#)

[/surveys/{survey_id}/pages/{page_id}](#)

[/surveys/{survey_id}/pages/{page_id}/questions](#)

[/surveys/{survey_id}/pages/{page_id}/questions/{question_id}](#)

[Formatting Question Types](#)

[Question Bank](#)

/question_bank/questions
Survey Folders
/survey_folders
Translations for Multilingual Surveys
/surveys/{survey_id}/languages
/surveys/{survey_id}/languages/{language_code}
Contacts and Contact Lists
/contact_lists
/contact_lists/{contact_list_id}
/contact_lists/{contact_list_id}/copy
/contact_lists/{contact_list_id}/merge
/contact_lists/{contact_list_id}/contacts
/contact_lists/{contact_list_id}/contacts/bulk
/contacts
/contacts/bulk
/contacts/{contact_id}
/contact_fields
/contact_fields/{contact_field_id}
Collectors and Invite Messages
/surveys/{survey_id}/collectors
/collectors/{collector_id}
/collectors/{collector_id}/messages
/collectors/{collector_id}/messages/{message_id}
/collectors/{collector_id}/messages/{message_id}/send
/collectors/{collector_id}/messages/{message_id}/recipients
/collectors/{collector_id}/messages/{message_id}/recipients/bulk
/collectors/{collector_id}/recipients
/surveys/{survey_id}/collectors/{collector_id}/recipients/{recipient_id}
/collectors/{collector_id}/stats
/surveys/{survey_id}/collectors/{collector_id}/messages/{message_id}/stats
Survey Responses
/surveys/{survey_id}/responses
/collectors/{collector_id}/responses
/surveys/{survey_id}/responses/bulk
/collectors/{collector_id}/responses/bulk
/surveys/{survey_id}/responses/{response_id}
/collectors/{collector_id}/responses/{response_id}
/surveys/{survey_id}/responses/{response_id}/details
/collectors/{collector_id}/responses/{response_id}/details
Response Counts and Trends
/surveys/{survey_id}/rollups
/surveys/{survey_id}/pages/{page_id}/rollups
/surveys/{survey_id}/pages/{page_id}/questions/{question_id}/rollups
/surveys/{survey_id}/trends
/surveys/{survey_id}/pages/{page_id}/trends
/surveys/{survey_id}/pages/{page_id}/questions/{question_id}/trends
Webhooks
/webhooks
/webhooks/{webhook_id}
Webhook Callbacks
Benchmarks
/benchmark_bundles
/benchmark_bundles/{bundle_id}
/benchmark_bundles/{id}/analyze
/surveys/{survey_id}/pages/{page_id}/questions/{question_id}/benchmark
Organizations
/workgroups/{id}/members
/workgroups/{id}/members/bulk
/workgroups/{workgroup_id}/members/{member_id}
/organizations

[/organization/{organization_id}](#)
[/roles](#)
[Errors](#)
[/errors](#)
[/errors/{error_id}](#)
[SCIM API](#)
[Users](#)
[/Users](#)
[/Users/{id}](#)
[Schemas](#)
[/Schemas](#)
[/Schemas/{id}](#)
[ResourceTypes](#)
[/ResourceTypes](#)
[/ResourceTypes/{id}](#)
[ServiceProviderConfig](#)
[/ServiceProviderConfig](#)
[API Libraries & SDKs](#)
[Feedback SDKs for iOS and Android](#)

SurveyMonkey API

Scroll down for code samples, example requests and responses. Select a language for code samples from the tabs above or the mobile navigation menu.

Overview

Getting Started

The SurveyMonkey API is REST-based, employs OAuth 2.0, and returns responses in JSON. Our documentation is organized by endpoint, has code examples in Python and cURL, and provides a Postman collection with calls for each available method. We also have a quick start guide outlining flows for some common use cases.

To use our API, you'll need to register a draft app to your SurveyMonkey account. You have a 90-day window to develop in a draft app, after which you'll need to deploy publicly or privately. Public apps are available to anyone with a SurveyMonkey account and published in our App Directory. If you're building an app for yourself or your organization, you can deploy a Private app.

Public Apps

Public apps extend features to SurveyMonkey users. All apps must be reviewed and approved by SurveyMonkey and adhere to our terms of use before they can be published in our App Directory.

Public apps use scopes to request permissions from app users during OAuth. Some scopes require your app's users to have a paid SurveyMonkey plan.

Public apps published in our App Directory can make up to 500,000 requests per day to our API. When a Public app is in draft (during development), it's subject to draft request limits.

Private Apps

Private apps don't need to be reviewed by SurveyMonkey. Private apps are subject to our terms of use. Private apps have API request limits and higher limits are available for purchase.

All users of a Private app must belong to the same SurveyMonkey team and have a paid SurveyMonkey plan that offers direct API access.

Registering an App

Anyone with a SurveyMonkey account can register an app. Registering creates a draft app with an access token you can use to query against your account for 90 days. No other SurveyMonkey accounts can authenticate a draft app. Before the 90-day period ends, you must deploy your app as either Public or Private and upgrade your account as needed.

Deploying an App

Before the 90-day draft period ends, you must deploy your app or it'll be disabled. If your app is disabled, you must deploy it or contact us at our help center to request an extension.

Scopes

Scopes allow apps to access particular resources on behalf of a user. For example, the **Create/Modify surveys** scope allows your app to create a survey in a user's account. During the OAuth process, a user is asked to grant permission to scopes your app is requesting access to.

Some scopes require your app's users to have SurveyMonkey paid plans. If your Public app uses scopes tied to paid plans, any accounts attempting to authenticate without the necessary plan will be asked to upgrade to proceed. Our request headers include the scopes available to users' plan, as well as those they've granted your app permission to.

You can set scopes to be required, optional, or not required in your app's settings. All required scopes must be approved by and available to the user for the OAuth process to succeed.

Two scopes, **Create/Modify Responses** and **Create/Modify Surveys**, require SurveyMonkey's approval to use in a Public app. If you've deployed a Public app and want to change your scope requirements to include these, you'll need to contact us at our help center to tell us more about your app and use case.

Create/Modify Collectors requires a paid SurveyMonkey plan to create any collector that is not a weblink collector.

Scope Label	Scope Description (text shown during OAuth, “you” refers to owner of authenticating account)	Scope Name	Requires Paid SurveyMonkey Account?
View Surveys	View your surveys and those shared with you	surveys_read	No
Create/Modify Surveys	Create or edit surveys in your account	surveys_write	No
View Collectors	View collectors for your surveys and those shared with you	collectors_read	No
Create/Modify Collectors	Create or edit collectors for surveys in your account	collectors_write	Yes (Except for weblink collectors)
View Contacts	View your contacts and contact lists	contacts_read	No
Create/Modify Contacts	Create or edit contacts in your account	contacts_write	No
View Responses	View if surveys in your account have responses and their metadata	responses_read	No

Scope Label	Scope Description (text shown during OAuth, “you” refers to owner of authenticating account)	Scope Name	Requires Paid SurveyMonkey Account?
View Response Details	View answers along with responses and answer counts and trends	responses_read_detail	Yes
Create/Modify Responses	Create or edit survey responses in your account	responses_write	Yes
View Webhooks	View webhooks to receive notifications when there are changes in your account	webhooks_read	No
Create/Modify Webhooks	Create and edit webhooks to receive notifications when there are changes in your account	webhooks_write	No
View Users	View your user information	users_read	No
View Teams	View teams you belong to	groups_read	Yes
View Library Assets	View your library of survey themes and templates	library_read	No
View Workgroups	View the workgroups in your team	workgroups_read	Yes
Create/Modify Workgroups	Create or edit workgroups in your team	workgroups_write	Yes
View Workgroup Members	View the members in a workgroup	workgroups_members_read	Yes
Create/Modify Workgroup Members	Add or remove members from a workgroup	workgroups_members_write	Yes
View Roles	View the roles available to assign to members in your team	roles_read	Yes
Create/Modify Roles	Create or edit roles for your team	roles_write	Yes
View Workgroups Shared Resources	View resources shared within a workgroup	workgroups_shares_read	Yes
Create/Modify Workgroups Shared Resources	Add or remove resources shared within a workgroup	workgroups_shares_write	Yes

Request and Response Limits

Draft and Private apps are subject to rate limits:

Max Requests Per Minute	Requests per day* (starting at)
120	500 *Sign in to view limits

The API limit is reset at 12:00 AM Greenwich Mean Time (GMT) every day. We permit three violations of up to 150% within a 30-day window. We begin to enforce limits at 100% once your app exceeds its limits three times within 30 days.

We return your app's rate limits, requests remaining, and seconds to reset in our request headers.

In addition, requests made to the API to create contacts or send invite messages are subject to our sending and contact limits.

INCREASING LIMITS

All Draft and Private apps will initially be subject to request limits of 120 calls/minute and starting at 500 calls/day (contact sales). If you think you may be in danger of exceeding your request-limit threshold and only require a temporary increase contact our help center otherwise, click here and then the “Talk with us” button to contact our Enterprise Sales Team. SurveyMonkey will review all requests within 5 business days. Additional fees may apply for increased rate limits.

RESPONSE LIMITS

We also have global response limits for database utilization fair use.

Property	Limit
Max Page Size	1000 resources, unless otherwise specified
Max Survey Size	1000 questions, surveys over limit will return a 413

Authentication

If you have a Private app and will only access your own SurveyMonkey account, you can use the access token, generated when you registered your app, as part of your app's configuration. Obtain this token in the **Settings** of your app in the **MY APPS** tab.

If your app will access many SurveyMonkey accounts, implement the OAuth 2.0 three-step flow outlined below to allow users to authorize your app to access their accounts. This flow generates a long-lived access token your app can use with every API call to the associated SurveyMonkey account. It's important to note the access token only grants access when used in combination with your API credentials (client ID) and only to the SurveyMonkey account which was authorized. Your app will need to obtain additional access tokens for each SurveyMonkey account you wish to access.

If your app has required scopes, users will need to approve all of them and may need a paid SurveyMonkey plan to successfully OAuth into your app.

OAuth 2.0 Flow

STEP 1: DIRECT USER TO SURVEYMONKEY'S OAUTH AUTHORIZATION PAGE

Your app should send the user whose SurveyMonkey account you wish to access to a specially crafted OAuth link at <https://api.surveymonkey.com>. The page presented to the user will identify your app, ask them to log into SurveyMonkey if they aren't already, and ask them to authorize any required scopes.

The OAuth link should be <https://api.surveymonkey.com/oauth/authorize> with urlencoded parameters: `redirect_uri`, `client_id`, `response_type` and `state`.

`response_type` will always be set to the value `code`
`client_id` the unique SurveyMonkey client id you got when registering your app
`redirect_uri` URL encoded OAuth redirect URI you registered for your app (can be found and edited here)
`state` (recommended) A value included in the request that will also be returned in the token response. It can be a string of any content that you wish. A randomly generated unique value is typically used for preventing cross-site request forgery attacks.

Example OAuth Link

```
SM_API_BASE = "https://api.surveymonkey.com"
AUTH_CODE_ENDPOINT = "/oauth/authorize"

def oauth_dialog(client_id, redirect_uri):
    url_params = urllib.urlencode({
        "redirect_uri": redirect_uri,
        "client_id": client_id,
        "response_type": "code"
    })

    auth_dialog_uri = SM_API_BASE + AUTH_CODE_ENDPOINT + "?" + url_params
    print "\nThe OAuth dialog url was " + auth_dialog_uri + "\n"

    # Insert code here that redirects user to OAuth Dialog url
```

STEP 2: USER AUTHORIZATION GENERATES SHORT-LIVED CODE

Once the user makes their choice whether to authorize access or not, SurveyMonkey will generate a 302 redirect sending their browser to your redirect URI along with a short-lived code included as a query parameter (as well as the state parameter if it was specified originally). Your app needs to use that code to make another API request before it expires (5 minutes). In that request, you'll send us the code you received, along with your client secret, client ID, and redirect URI. We'll verify all that information. If it's good, we'll return a long-lived access token in exchange.

Generate Short Code or Deny Access

```

def handle_redirect(redirect_uri):
    # Parse authorization code out of url
    query_string = urlparse.urlsplit(redirect_uri).query
    authorization_code = urlparse.parse_qs(query_string).get("code", [])

    # parse_qs returns a list for every query param, just get the first one
    if not authorization_code:
        return

    return authorization_code[0]

```

STEP 3: EXCHANGING FOR A LONG-LIVED ACCESS TOKEN

Create a form-encoded HTTP POST request to <https://api.surveymonkey.com/oauth/token> with the following encoded form fields: `client_id`, `client_secret`, `code`, `redirect_uri` and `grant_type`. The grant type must be set to “authorization_code”. The `client_secret` can be found here.

If successful, the access token will be returned encoded as JSON in the response body of your POST request. The key will be `access_token` and the value can be passed to our API as an HTTP header in the format `Authorization: bearer YOUR_ACCESS_TOKEN`. The value of the header must be “bearer” followed by a single space and then your access token.

Exchange for long-lived token

```

SM_API_BASE = "https://api.surveymonkey.com"
ACCESS_TOKEN_ENDPOINT = "/oauth/token"

def exchange_code_for_token(auth_code, client_secret, client_id, redirect_uri):
    data = {
        "client_secret": client_secret,
        "code": auth_code,
        "redirect_uri": redirect_uri,
        "client_id": client_id,
        "grant_type": "authorization_code"
    }

    access_token_uri = SM_API_BASE + ACCESS_TOKEN_ENDPOINT
    access_token_response = requests.post(access_token_uri, data=data)
    access_json = access_token_response.json()

    if "access_token" in access_json:
        return access_json["access_token"]
    else:
        print access_json
        return None

```

Token expiration and revocation

Our access tokens don't currently expire but may in the future. We'll warn all developers before making changes.

Access tokens can be revoked by the user. If this happens, you'll get a JSON-encoded response body including a key `status` with a value of `1` and a key `errmsg` with the value of `Client revoked access grant` when making an API request. If you get this response, you'll need to complete OAuth again.

Access URL

Depending on the originating datacenter of the SurveyMonkey account, the API access URL may be different than <https://api.surveymonkey.com>.

The API for the EU datacenter is <https://api.eu.surveymonkey.com> and the API for the Canadian datacenter is <https://api.surveymonkey.ca>.

The correct API access URL for each SurveyMonkey account is returned in the response body of the code for token exchange under the `access_url` key.

Unauthorizing an app

To unauthorize an app:

Log into the linked SurveyMonkey account
Select **My Account** from the username dropdown in the upper right
Scroll to **Linked Accounts** and click **Remove** next to the app you want to unauthorize

Quick Start Guide: Two Common Use Cases

Export the Results of a Survey

While a call to `/surveys/{id}/responses/bulk` returns responses with ids for all answers chosen, your use case likely involves associating this with the values chosen. The following example takes you through exporting the results of a survey and associating responses with the corresponding answer values.

Fetch the first 1,000 surveys in a SurveyMonkey account with a GET `/surveys`. This call returns a list resource containing survey ids. Using a survey id from the previous call, make a GET to `/surveys/{id}/details`. This call returns the survey's design with all question ids and answer option ids, as well as the values associated with them. Cache these values on your end to limit future calls and economize your request and response limits. Using the same survey id, fetch the first 100 responses to your survey with a GET to `/surveys/{id}/responses/bulk?per_page=100`. This endpoint returns question ids and the id of the selected answer or choice for each response. You can use these to associate the selected answer id to those returned from your GET to `/surveys/{id}/details` and match question values to the answers selected. Fetch the next page of 100 responses using the resource url returned in the `links.next` field.

To export the results of all surveys in an account, iterate through all surveys ids returned in step 1, completing steps 2 through 4.

To export the results of a single collector, a GET to `/surveys/{id}/collectors` returns a list collector ids associated with a given survey. A GET to `/surveys/{id}/details` returns responses from a single collector.

Send an Email Invitation Message

The following example takes you through creating an Email Invitation Collector and sending it to a list of recipients.

Fetch the first 1,000 surveys in a SurveyMonkey account with a GET `/surveys`. This call returns a list resource containing survey ids. Using a survey id from the previous call, create a collector of type `email` by making a POST to `/surveys/{id}/collectors`. Create and format an email message with a POST to `/collectors/{id}/messages`. Upload recipients to receive the message with a POST to `/collectors/{id}/messages/{id}/recipients/bulk`. Send your message with a POST to `/collectors/{id}/messages/{id}/send`.

Pagination

When requesting list resources, you can set the size of a page by using `per_page=#` and indicate which page to return with `page=#`. So a request to https://api.surveymonkey.com/v3/surveys?page=2&per_page=5 will return the second page of five surveys.

Any request to a list resource returns the following pagination fields, if available:

Name	Description	Type
per_page	Number of resources per page	Integer
total	Total number of resources available	Integer
page	Indicates which page is being returned	Integer
links.self	Resource URL for the current page	String
links.prev	Resource URL for the previous page of results	String

Name	Description	Type
links.next	Resource URL for the subsequent page of results	String
links.first	Resource URL for the first page of results	String
links.last	Resource URL for the last page of results	String

Headers

Our API returns the following custom headers:

Header	Description
X-OAuth-Scopes-Available	Which scopes are available to the user using an app
X-OAuth-Scopes-Granted	Which scopes the user has granted permission to, to the app
X-Ratelimit-App-Global-Day-Limit	Per day request limit the app has
X-Ratelimit-App-Global-Day-Remaining	Number of remaining requests app has before hitting daily limit
X-Ratelimit-App-Global-Day-Reset	Number of seconds until the rate limit remaining resets
X-Ratelimit-App-Global-Minute-Limit	Per minute request limit the app has
X-Ratelimit-App-Global-Minute-Remaining	Number of remaining requests app has before hitting per minute limit
X-Ratelimit-App-Global-Minute-Reset	Number of seconds until the rate limit remaining resets

Data Types

Our API uses the following data types:

Data Type	Description
Integer	An integer number with a maximum value of 2147483647. Negatives are disallowed unless otherwise specified.
String	A string of text.
String-ENUM	Predefined string values. Values are defined per field throughout our documentation.
Boolean	A boolean value: true or false. In JSON it will be represented using the native boolean type.
Date string	Dates are usually in the format YYYY-MM-DDTHH:MM:SS+HH:MM. Any deviations from this are shown in the documentation. All date strings are implicitly in UTC.
Phone number string	Phone numbers must be in the format +1XXXXYYYZZZZ
Hex string	Must be in the format #HHH or #HHHHHH where each H represents hexadecimal format
Array	A simple list of values. In JSON this will be an array.
Object	A collection of name/value pairs. In JSON this will be an object.
Null	A null value. In JSON this is represented as the native null type.

Error Codes

Example Error Response

```
{  
  "error": {  
    "docs": "https://developer.surveymonkey.com/api/v3/#error-codes",  
    "message": "Oh bananas! We couldn't process your request.",  
    "id": "1050",  
    "name": "Internal Server Error",  
    "http_status_code": 500  
  }  
}
```

Error Code	HTTP Status Code	Message
1000	400 Bad Request	Unable to process the request with the provided input.
1001	400 Bad Request	The body provided was not a proper JSON string.
1002	400 Bad Request	Invalid schema in the body provided.
1003	400 Bad Request	Invalid URL parameters.
1004	400 Bad Request	Invalid request headers.
1005	402 Payment Required	Account must be upgraded to access this feature.
1010	401 Authorization Error	The authorization token was not provided.
1011	401 Authorization Error	The authorization token provided was invalid.
1012	401 Authorization Error	The authorization token provided has expired.
1013	401 Authorization Error	Client revoked access to the authorization token provided.
1014	403 Permission Error	Permission has not been granted by the user to make this request.
1015	403 Permission Error	The user does not have the required plan to make this request.
1016	403 Permission Error	The user does not have permission to access the resource.
1017	403 Permission Error	The user has hit a quota limit on this resource.
1018	403 Permission Error	The user does not have permission to access the host in this region. See Access URL
1020	404 Resource Not Found	There was an error retrieving the requested resource.
1025	409 Resource Conflict	Unable to complete the request due to a conflict. Check the settings for the resource.
1026	409 Resource Conflict	The requested resource already exists.
1030	413 Request Entity Too Large	The requested entity is too large, it can not be returned.
1040	429 Rate Limit Reached	Too many requests were made, try again later.
1050	500 Internal Server Error	Oh bananas! We couldn't process your request.
1051	503 Internal Server Error	Service unreachable. Please try again later.
1052	404 User Soft Deleted	The user you are making this request for has been soft deleted.
1053	410 User Deleted	The user you are making this request for has been deleted.
1054	502 Bad Gateway Error	Network was unable to connect to server.
1055	504 Gateway Timeout	Service connection has timed out. Please try again later.

Error Code	HTTP Status Code	Message
1056	400 Bad Request	Too many recipients supplied for this message.
1057	401 Authorization Error	Authorization token has no grants.
1058	403 Permission Error	The user's email must be verified before performing this request.
1059	400 Bad Request	Trying to send a Message but no recipient was provided
1060	400 Bad Request	Embedded Body not saved
1061	405 Method Not Allowed	The specified HTTP method is not allowed.
1062	400 Bad Request	Request limit exceeded for this resource
1063	401 Bad Unauthorized	User is not authorized to perform this action. Please review the request details and try again.

Help and Resources

SurveyMonkey lists code examples on Github. If you have an SDK or example you would like added, let us know. We also offer support through our help center.

API Endpoints

Scroll down for code samples, example requests and responses. Select a language for code samples from the tabs above or the mobile navigation menu.

Base URLs:

<https://api.surveymonkey.com/v3>

Default

Users and Teams

Use these endpoints to get user's account information, any teams a user belongs to, the account information for the team's owner, and other members including their roles and whether or not they have accepted an invite into the team (are `active`).

 While our API uses the term group, SurveyMonkey's UI uses teams. These two terms refer to the same concept. Learn more about teams in our Help Center.

GET /users/me

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/users/me", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

[GET /users/me](#)

AVAILABLE METHODS

[HEAD](#): Checks if resource is available

[OPTIONS](#): Returns available methods and options

[GET](#): Returns the current user's account details including their plan. Public App users need access to the **View Users** scope parameters:

Example Response for GET /users/me

```
{
    "id": "1",
    "username": "username",
    "first_name": "firstName",
    "last_name": "lastName",
    "language": "en",
    "email": "firstname.lastname@surveymonkey.com",
    "email_verified": false,
    "account_type": "enterprise",
    "date_created": "2021-05-27T16:29:00+00:00",
    "date_last_login": "2021-06-21T13:57:15.657000+00:00",
    "question_types": {
        "comment_box_question_type": true,
        "file_upload_question_type": true,
        "matrix_question_type": true,
        "ranking_question_type": true,
        "rating_question_type": false,
        "slider_question_type": true,
        "star_rating_question_type": true,
        "textbox_multiple_question_type": true,
        "video_question_type": false
    },
    "scopes": {
        "granted": [
            "surveys_write",
            "surveys_read"
        ],
        "available": [
            "users_read",
            "surveys_read",
            "collectors_read",
            "collectors_write",
            "contacts_read",
            "contacts_write",
            "surveys_write",
            "responses_read",
            "responses_read_detail",
            "responses_write",
            "groups_read",
            "groups_write",
            "webhooks_read",
            "webhooks_write",
            "library_read",
            "workgroups_read",
            "workgroups_write",
            "workgroups_members_read",
            "workgroups_members_write",
            "roles_read",
            "roles_write",
            "workgroups_shares_read",
            "workgroups_shares_write"
        ]
    }
}
```

```

        ],
    },
    "sso_connections": [
        "enterprise_saml"
    ],
    "features": {
        "collector_create_limit": "unlimited",
        "collector_email_enabled": true,
        "collector_thank_you_enabled": true,
        "create_question_limit": "unlimited",
        "num_free_responses": "unlimited"
    },
    "href": "https://api.surveymonkey.com/v3/users/me"
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	User id
username	string	Username
first_name	string	User's first name
last_name	string	User's last name
language	string	ISO 639-1 code for the language set for the user's account
email	string	Email address for user's account
email_verified	boolean	Email address for user's account has been verified
account_type	string	The user's SurveyMonkey plan
date_created	string	Date user's account was created
date_last_login	string	Date user last logged in
question_types	object	Question types available to the user, identified by boolean values
scopes	object	Contains the scopes the user has available to them, and the scopes that are granted
granted	[string]	Lists the scopes the user has approved during OAuth
available	[string]	Lists the scopes available to the user
sso_connections	[string]	Third-party connections
features	object	Features available to the user. They are identified by string or boolean values.
href	string	The link associated with this endpoint

GET /users/{user_id}/workgroups

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/users/1234/workgroups", headers=headers)

```

```

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /users/{user_id}/workgroups`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns the workgroups that a specific user is in. Public App users need access to the **View Workgroups** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
per_page	string	false	Number of resources to return per page. Defaults to 50
page	string	false	Which page of resources to return. Defaults to 1

Example Response for `GET /users/{user_id}/workgroups`

```
{
  "data": [
    {
      "id": "1",
      "name": "name",
      "description": "some description",
      "is_visible": true,
      "metadata": {},
      "created_at": "2019-05-22T17:38:58",
      "updated_at": "2020-07-29T20:32:33",
      "members": [
        {
          "user_id": "2",
          "is_owner": true
        },
        {
          "user_id": "1",
          "is_owner": false
        }
      ],
      "membership": {
        "status": "active",
        "is_owner": true
      },
      "shares_count": 3,
      "members_count": 2,
      "default_role": {
        "id": "2",
        "name": "Viewer",
        "description": "",
        "is_enabled": true,
        "metadata": {}
      },
      "organization_id": "1"
    }
  ],
  "per_page": 50,
  "page": 1,
  "total": 1,
  "links": {
    "self": "https://api.surveymonkey.com/v3/users/2/workgroups?page=1&per_page=50"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[object]	Contains the information associated with the workgroups

Name	Type	Description
id	string	ID of the user
name	string	Name of the workgroup
description	string	Description of the workgroup
is_visible	boolean	Whether the workgroup is publicly visible or only visible to its members and administrators
metadata	object	Extra data related to the workgroup
created_at	string	Datetime when the workgroup was created
updated_at	string	Datetime when the workgroup was last updated
members	[object]	The active members of the workgroup
user_id	string	The user ID of the active member
is_owner	boolean	The active member is the owner of the workgroup
membership	object	Membership information for the requested user
is_owner	boolean	The user is the owner of the workgroup
status	string	The status of the user. One of [active, pending]
shares_count	integer	Number of resources shared with the workgroup
members_count	integer	Number of members in the workgroup (includes non-active)
default_role	object	The default role for the workgroup
id	string	ID of the workgroup
name	string	The type of the default role. One of Analyzer, Collector, Full Access, Viewer, Editor
description	string	Extra information about the default role
is_enabled	boolean	The default role is enabled
metadata	object	Extra metadata of the default role
organization_id	string	The ID of the organization this workgroup belongs to.
per_page	integer	Number of resources returned per page
page	integer	Page of the resources returned
total	integer	Number of workgroups returned
links	object	Contains this endpoint's links
self	string	The link associated with this endpoint

GET /users/{user_id}/shared

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"}
```

```

}

conn.request("GET", "/v3/users/1234/shared", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /users/{user_id}/shared`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns the resources shared with the user across all workgroups. Public App users need access to the **View Workgroup Shares** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
per_page	string	false	Number of resources to return per page. Defaults to 50
page	string	false	Which page of resources to return. Defaults to 1
resource_type	string	false	The type of the shared resource, one of: <code>survey</code>
resource_id	string	false	Comma-separated IDs of shared resources (must be used with <code>resource_type</code>)
include	string	false	Specify additional fields to return for each resource. One of <code>permissions</code> , <code>resource_details</code> . <code>permissions</code> shows collector and survey action permissions of each workgroup and <code>resource_details</code> shows more information about each shared resource

Example Response for `GET /users/{user_id}/shared`

```
{
  "data": [
    {
      "share_id": "0ca749c2893245c5b96a128eee4c2d42",
      "workgroup_id": "71d9d408d1914c9ca85ffcd8a8330d675",
      "resource_type": "survey",
      "resource_id": "101101101",
      "privileges": [
        "design.read_only",
        "collect.read_only",
        "analyze.read_only"
      ]
    },
    {
      "per_page": 50,
      "page": 1,
      "total": 1,
      "links": {
        "self": "https://api.surveymonkey.com/v3/users/61003234/shared?page=1&per_page=50"
      }
    }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[object]	Contains the shared resources with their information
share_id	string	ID of the share record
workgroup_id	string	ID of the workgroup that was shared with
resource_type	string	The type of the shared resource (e.g. <code>survey</code>)
resource_id	string	The ID of the shared resource (e.g. the ID of a survey)

Name	Type	Description
privileges	[string]	An array of scoped privileges granted to the user by this share record
per_page	integer	Number of resources returned per page
page	integer	Page of the resource returned
total	integer	Number of shared resources returned
links	object	Contains this endpoint's links
self	string	The link associated with this endpoint

GET /groups

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/groups", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /groups](#)

AVAILABLE METHODS

[HEAD](#): Checks if resource is available

[OPTIONS](#): Returns available methods and options

[GET](#): Returns a list of groups. Public App users need access to the **View Teams** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
per_page	integer	false	Number of resources to return per page. Defaults to 50
page	integer	false	Which page of resources to return. Defaults to 1

Example Response for GET /groups

```
{
  "data": [
    {
      "href": "https://api.surveymonkey.com/v3/groups/1231511",
      "id": "1231511",
      "name": "Your Group"
    }
  ],
  "links": {
    "self": "https://api.surveymonkey.com/v3/groups?per_page=50&page=1"
  },
  "page": 1,
  "per_page": 50,
  "total": 1
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
per_page	integer	
total	integer	
page	integer	
data	[object]	
id	string	Group id
href	string	Resource API URL
name	string	Group name
links	object	
self	string	

GET /groups/{group_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/groups/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /groups/{group_id}](#)

AVAILABLE METHODS

[HEAD](#): Checks if resource is available

[OPTIONS](#): Returns available methods and options

[GET](#): Returns group information. Public App users need access to the **View Teams** scope

Example Response for GET /groups/{group_id}

```
{
  "date_created": "2333-01-07T21:27:32+00:00",
  "href": "https://api.surveymonkey.com/v3/groups/123",
  "id": "123",
  "max_invites": 0,
  "member_count": 10,
  "name": "Group Name"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
date_created	string	

Name	Type	Description
href	string	
id	string	
max_invites	integer	
member_count	integer	
name	string	

GET /groups/{group_id}/activities

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/groups/1234/activities", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /groups/{group_id}/activities](#)

AVAILABLE METHODS

[GET](#): Returns a list of activity related data for the specified group. Public App users must be admin.

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
limit	integer	false	amount of activites returned
offset	integer	false	offset of the returned list
start_date	string	false	the start date of activites
end_date	string	false	list all activites up to the end date

Example Response for GET /groups/{group_id}/activities

```
{
  "limit": 2,
  "offset": 0,
  "results": [
    {
      "activity_msg": "<span>Installed app <span class=\"notranslate\"><b>mpt testing</b></span></span>",
      "activity_type": "grant_info_created",
      "city": null,
      "country": null,
      "date_created": "2023-12-05 20:49:49",
      "division_name": null,
      "email": "test@surveymonkey.com",
      "group_id": 2093412,
      "ip_address": null,
      "member_type": "<span>(Primary Admin)</span>",
      "user_id": 12312,
      "user_name": "Testing"
    }
  ]
}
```

```

        }
    ],
    "sl_translate": "activity_msg,member_type",
    "total": 1932
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
limit	integer	
offset	integer	
sl_translate	string	
total	integer	
results	[object]	
activity_msg	string	
activity_type	string	
city	string	
country	string	
date_created	string	
division_name	string	
email	string	
group_id	integer	
ip_address	string	
member_type	string	
user_id	integer	
user_name	string	

GET /groups/{group_id}/activities/{action_type}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/groups/1234/activities/member_joined?interval=daily", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

[GET /groups/{group_id}/activities/{action_type}](#)

AVAILABLE METHODS

GET: Returns a count of a specific activity by interval for the specified group given an action type enum value. Public App users must be admin

ACTION TYPE ENUM VALUES

authentication_succeeded
authentication_failed
authentication_signout
group_info_updated_group_name
invite_created
invite_resent
member_deleted
member_joined
survey_info_create
survey_info_delete
survey_info_copy
survey_info_update
survey_info_transfer
collector_info_created
collector_info_deleted
collector_info_updated
member_updated_group_member_type
permission_created
permission_updated
shared_view_created
shared_view_updated
export_export_create
export_downloaded
respondent_updated
respondent_deleted
grant_info_created
grant_info_deleted

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
interval	string	true	The selected timing interval of the chosen action type
start_date	string	false	the start date of activites
end_date	string	false	list all activites up to the end date

ENUMERATED VALUES

Parameter	Value
interval	daily
interval	weekly
interval	monthly
interval	yearly

Example Response for GET /groups/{group_id}/activities/{action_type}

```
{
  "interval": "yearly",
  "series": [
    2,
    7
  ],
  "times": [
    "2023",
    "2022"
  ]
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
interval	string	
series	[integer]	
times	[string]	

GET /groups/{group_id}/members

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/groups/1234/members", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`GET /groups/{group_id}/members`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns a list of users who have been added as members of the specified group. Public App users need access to the **View Teams** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
per_page	integer	false	Number of resources to return per page. Defaults to 50
page	integer	false	Which page of resources to return. Defaults to 1
division_ids	string	false	optional list of member ids
include	string	false	whether or not to include properties options are first_name, last_name, user_id, email, type, language, status

DETAILED DESCRIPTIONS

`include`: whether or not to include properties options are first_name, last_name, user_id, email, type, language, status

Example Response for `GET /groups/{group_id}/members`

```
{
  "data": [
    {
      "id": "1",
      "username": "user1",
      "href": "https://api.surveymonkey.com/v3/groups/1/members/1"
    },
    {
      "id": "2",
      "username": "user2",
      "href": "https://api.surveymonkey.com/v3/groups/1/members/2"
    }
  ],
  "per_page": 50,
  "page": 1,
```

```

    "total": 2,
    "links": {
      "self": "https://api.surveymonkey.com/v3/groups/1/members?page=1&per_page=50"
    }
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
per_page	integer	
total	integer	
data	[object]	
id	string	Member id
href	string	Resource API URL
username	string	Member username
links	object	
self	string	
page	integer	

GET /groups/{group_id}/members/{member_id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/groups/1234/members/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

[GET /groups/{group_id}/members/{member_id}](#)

AVAILABLE METHODS

[HEAD](#): Checks if resource is available

[OPTIONS](#): Returns available methods and options

[GET](#): Returns a group member's details including their [role](#) and [status](#). Public App users need access to the **View Teams** scope

Example Response for GET /groups/{group_id}/members/{member_id}

```
{
  "id": "1",
  "username": "username",
  "email": "username@surveymonkey.com",
  "type": "regular",
  "status": "active",
  "user_id": "1",
  "language": "en",
  "first_name": "firstname",
  "last_name": "lastname"
}
```

```

    "last_name": "lastname",
    "date_created": "2021-06-24T18:27:33+00:00",
    "href": "https://api.surveymonkey.com/v3/groups/1/members/1"
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
user_id	string	User ID of the group member
date_created	string	Date and time when member was created
last_name	string	
language	string	
id	string	Member id
href	string	
type	string	Which type of role the member has: <code>regular</code> , <code>account_owner</code> , or <code>admin</code>
first_name	string	
email	string	User's email address
username	string	Member username
status	string	A members status (if they have accepted an invitation into the group): <code>active</code> , <code>pending</code>

PATCH /groups/{group_id}/members/{member_id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"email\":\"string\"}"

headers = {
    'Content-Type': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/v3/groups/1234/members/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`PATCH /groups/{group_id}/members/{member_id}`

`PATCH`: Update a group members email address

GET /workgroups

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/workgroups", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /workgroups`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns a list of all workgroups that are available. Public App users need access to the **View Workgroup Shares** scope

`POST`: Add a new shared resource to the workgroup. Public App users need access to the **Create/Modify Workgroup Shares** scope

200 Response

```
{
  "data": [
    {
      "created_at": "string",
      "default_role": {
        "description": "string",
        "id": "string",
        "is_enabled": true,
        "metadata": {},
        "name": "string"
      },
      "id": "string",
      "description": "string",
      "is_visible": true,
      "members": [
        {
          "is_owner": true,
          "user_id": "string"
        }
      ],
      "members_count": 0,
      "membership": {
        "is_owner": true,
        "status": "string"
      },
      "name": "string",
      "organization_id": "string",
      "share_count": "string",
      "updated_at": "string"
    }
  ],
  "per_page": 0,
  "page": 0,
  "total": 0,
  "links": {
    "self": "string"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[object]	Contains the list of workgroups
created_at	string	date and time when the workgroup group was created

Name	Type	Description
default_role	object	The default role for the workgroup
description	string	
id	string	
is_enabled	boolean	
metadata	object	
name	string	
id	string	ID of the workgroup that was shared with
description	string	Description of the workgroup
is_visible	boolean	Visability status of the workgroup
members	[object]	
is_owner	boolean	
user_id	string	
members_count	integer	amount of memebers in the workgroup
membership	object	The current user membership status in the workgroup
is_owner	boolean	
status	string	
name	string	The name of the workgroup
organization_id	string	The ID of the organization
share_count	string	The amount of shared resources with this group
updated_at	string	The last time the workgroup was updated
per_page	integer	Number of resources returned per page
page	integer	Page of the resource returned
total	integer	Number of shared resources returned
links	object	Contains this endpoint's links
self	string	The link associated with this endpoint

POST /workgroups

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"name\":\"Brand Awareness\",\"description\":\"Spreading the company brand\",\"is_visible\":true,\"metadata\":{}}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/workgroups", payload, headers)

res = conn.getresponse()
data = res.read()

print(data)
```

```

conn.request("POST", "/v3/workgroups", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

POST /workgroups

Body parameter

```
{
  "name": "Brand Awareness",
  "description": "Spreading the company brand",
  "is_visible": true,
  "metadata": {}
}
```

REQUEST BODY FOR POST /WORKGROUP

Name	Type	Required	Description
name	string	true	Name of the workgroup
is_visible	boolean	true	visibility of the workgroup to other org members
metadata	object	undefined	Key-pair data that can be attached to the workgroup. This is a required field.

Example Response for POST /workgroups

```
{
  "created_at": "2022-12-01T18:03:45",
  "default_role": {
    "description": "",
    "id": "40825440989a41cfaf8c50704",
    "is_enabled": true,
    "metadata": {},
    "name": "Full Access"
  },
  "description": "Spreading the company brand",
  "id": "9bccda1faa0d45363edb7fc22",
  "is_visible": true,
  "members": [
    {
      "is_owner": true,
      "user_id": "101592"
    }
  ],
  "members_count": 1,
  "membership": {
    "is_owner": true,
    "status": "active"
  },
  "metadata": {
    "something": "ok"
  },
  "name": "Creating workgroup via api",
  "organization_id": "2112",
  "shares_count": 0,
  "updated_at": "2022-12-05T18:03:45"
}
```

RESPONSE SCHEMA

Status Code 201

Name	Type	Description
data	[object]	Contains the list of workgroups
created_at	string	date and time when the workgroup group was created

Name	Type	Description
default_role	object	The default role for the workgroup
description	string	
id	string	
is_enabled	boolean	
metadata	object	
name	string	
id	string	ID of the workgroup that was shared with
description	string	Description of the workgroup
is_visible	boolean	Visability status of the workgroup
members	[object]	
is_owner	boolean	
user_id	string	
members_count	integer	amount of memebers in the workgroup
membership	object	The current user membership status in the workgroup
is_owner	boolean	
status	string	
name	string	The name of the workgroup
organization_id	string	The ID of the organization
share_count	string	The amount of shared resources with this group
updated_at	string	The last time the workgroup was updated

GET /workgroups/{workgroup_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/workgroups/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /workgroups/{workgroup_id}](#)

AVAILABLE METHODS

[HEAD](#): Checks if resource is available

[OPTIONS](#): Returns available methods and options

GET: Returns information on the selected workgroup **View Workgroup Shares** scope
PATCH: modify information on the workgroup. Public App users need access to the **Create/Modify Workgroup Shares** scope
DELETE: Remove the workgroup. Public App users need access to the **Create/Modify Workgroup Shares** scope

Example Response for GET /workgroups/{workgroup_id}/shares

```
{  
    "created_at": "2022-02-24T19:36:28",  
    "default_role": {  
        "description": "",  
        "id": "40825449a41cfaf8c50df160bf704",  
        "is_enabled": true,  
        "metadata": {},  
        "name": "Full Access"  
    },  
    "description": "Spreading the company brand",  
    "id": "8bd869a4ac84d945fa7d16e",  
    "is_visible": false,  
    "members": [  
        {  
            "is_owner": true,  
            "user_id": "91111084"  
        }  
    ],  
    "members_count": 1,  
    "membership": {  
        "is_owner": false,  
        "status": null  
    },  
    "metadata": {},  
    "name": "sample test workgroup3",  
    "organization_id": "211112",  
    "shares_count": 0,  
    "updated_at": "2022-03-24T19:36:28"  
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
created_at	string	date and time when the workgroup group was created
default_role	object	The default role for the workgroup
description	string	
id	string	
is_enabled	boolean	
metadata	object	
name	string	
id	string	ID of the workgroup that was shared with
description	string	Description of the workgroup
is_visible	boolean	Visability status of the workgroup
members	[object]	
is_owner	boolean	
user_id	string	
members_count	integer	amount of memebers in the workgroup
membership	object	The current user membership status in the workgroup
is_owner	boolean	

Name	Type	Description
status	string	
name	string	The name of the workgroup
organization_id	string	The ID of the organization
share_count	string	The amount of shared resources with this group
updated_at	string	The last time the workgroup was updated

PATCH /workgroups/{workgroup_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"name\":\"Patching the company brand\",\"description\":\"string\",\"is_visible\":false,\"metadata\":{}}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/v3/workgroups/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[PATCH /workgroups/{workgroup_id}](#)

Body parameter

```
{
  "name": "Patching the company brand",
  "description": "string",
  "is_visible": false,
  "metadata": {}
}
```

REQUEST BODY FOR POST /WORKGROUP

Name	Type	Required	Description
name	string	false	Name of the workgroup
is_visible	boolean	false	visibility of the workgroup to other org members
metadata	object	undefined	Key-pair data that can be attached to the workgroup

Example Response for PATCH /workgroups/{workgroup_id}/shares

```
{
  "created_at": "2022-02-24T19:36:28",
  "default_role": {
    "description": "",
    "id": "40825449a41cfaf8c50df160bf704",
    "is_enabled": true,
    "metadata": {},
    "name": "Full Access"
  },
}
```

```

"description": "Spreading the company brand",
"id": "8bd869a4ac84d945fa7d16e",
"is_visible": false,
"members": [
  {
    "is_owner": true,
    "user_id": "91111084"
  }
],
"members_count": 1,
"membership": {
  "is_owner": false,
  "status": null
},
"metadata": {},
"name": "sample test workgroup3",
"organization_id": "211112",
"shares_count": 0,
"updated_at": "2022-03-24T19:36:28"
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
created_at	string	date and time when the workgroup group was created
default_role	object	The default role for the workgroup
description	string	
id	string	
is_enabled	boolean	
metadata	object	
name	string	
id	string	ID of the workgroup that was shared with
description	string	Description of the workgroup
is_visible	boolean	Visability status of the workgroup
members	[object]	
is_owner	boolean	
user_id	string	
members_count	integer	amount of memebers in the workgroup
membership	object	The current user membership status in the workgroup
is_owner	boolean	
status	string	
name	string	The name of the workgroup
organization_id	string	The ID of the organization
share_count	string	The amount of shared resources with this group
updated_at	string	The last time the workgroup was updated

GET /workgroups/{workgroup_id}/shares

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/v3/workgroups/1234/shares", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /v3/workgroups/{workgroup_id}/shares

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns the resources shared with the user across all workgroups. Public App users need access to the **View Workgroup Shares** scope

POST: Add a new shared resource to the workgroup. Public App users need access to the **Create/Modify Workgroup Shares** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
per_page	string	false	Number of resources to return per page. Defaults to 50
page	string	false	Which page of resources to return. Defaults to 1

Example Response for GET /workgroups/{workgroup_id}/shares

```
{
  "per_page": 50,
  "total": 1,
  "data": [
    {
      "workgroup_id": "71d9d408d19ffcd8330d675",
      "organization_id": "54321",
      "owner_user_id": "1234567",
      "resource_id": "101101101",
      "created_at": "2018-04-21T14:36:38",
      "id": "0ca749c2893245c5b96a128eee4c2d42",
      "resource_type": "survey"
    }
  ],
  "page": 1,
  "links": {
    "self": "https://api.surveymonkey.com/v3/workgroups/71d9d408d19ffcd8330d675/shares?page=1&per_page=50"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[object]	Contains the shared resources with their information
workgroup_id	string	ID of the workgroup that was shared with
organization_id	string	ID of the organization

Name	Type	Description
owner_user_id	string	User of ID of the resource owner
resource_type	string	The type of the shared resource (e.g. <code>survey</code>)
created_at	string	The type of the shared resource (e.g. <code>survey</code>)
resource_id	string	The ID of the shared resource (e.g. the ID of a survey)
per_page	integer	Number of resources returned per page
page	integer	Page of the resource returned
total	integer	Number of shared resources returned
links	object	Contains this endpoint's links
self	string	The link associated with this endpoint

POST /workgroups/{workgroup_id}/shares

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"resource_type\":\"survey\", \"resource_id\":\"string\"}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/v3/workgroups/1232141/shares", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`POST /v3/workgroups/{workgroup_id}/shares`

Body parameter

```
{
  "resource_type": "survey",
  "resource_id": "string"
}
```

REQUEST BODY FOR POST /WORKGROUPS/{WORKGROUP_ID}/SHARES

Name	Type	Required	Description
resource_type	string	true	The type of resource that is supported, currently only sharing surveys is possible
resource_id	string	true	The id of the resource to share

Example Response for POST /workgroups/{workgroup_id}/shares

```
{
  "created_at": "2020-10-37T22:30:00",
  "id": "87782957243442b315097477094369",
  "organization_id": "93112",
```

```

    "owner_user_id": "13392",
    "resource_id": "120335",
    "resource_type": "survey",
    "workgroup_id": "8b9a4ac84492f9c5d8d947d16e",
    "workgroup_name": "sample test workgroup"
}

```

RESPONSE SCHEMA

Status Code 201

Name	Type	Description
created_at	string	The date and time the share was created
id	string	The ID of the share that was created
organization_id	string	The ID of the organization the new shared resource belongs too
owner_user_id	string	The ID of the User that owns the resource
resource_id	string	The ID of the User that owns the resource
resource_type	string	The type of resource currently, only supports surveys
workgroup_id	string	The ID of the workgroup that the resource is shared with
workgroup_name	string	The name of the workgroup the resources is shared too.

GET /workgroups/{workgroup_id}/shares/{share_id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/v3/workgroups/string/shares/string", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

[GET /v3/workgroups/{workgroup_id}/shares/{share_id}](#)

AVAILABLE METHODS

[HEAD](#): Checks if resource is available

[OPTIONS](#): Returns available methods and options

[GET](#): Returns a resource shared with the user across all workgroups. Public App users need access to the **View Workgroup Shares** scope

[DELETE](#): Remove a resource from being shared within a workgroup. Public App users need access to the **Create/Modify Workgroup Shares** scope

Example Response for GET /workgroups/{workgroup_id}/shares/{share_id}

```
{
    "workgroup_id": "71d9d408d19ffcd8330d675",
    "organization_id": "54321",
    "owner_user_id": "1234567",
    "resource_id": "101101101",
    "created_at": "2018-04-21T14:36:38",
    "id": "0ca749c2893245c5b96a128eee4c2d42",
}
```

```
        "resource_type": "survey"
    }
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
workgroup_id	string	ID of the workgroup that was shared with
organization_id	string	ID of the organization
owner_user_id	string	User or ID of the resource owner
resource_type	string	The type of the shared resource (e.g. <code>survey</code>)
created_at	date time string of when the shared resource was created	The type of the shared resource (e.g. <code>survey</code>)
resource_id	string	The ID of the shared resource (e.g. the ID of a survey)
id	string	The ID of the share itself

POST /workgroups/{workgroup_id}/shares/bulk

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"shares\": [{\"resource_type\": \"survey\", \"resource_id\": \"string\"}]}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/workgroups/1232141/shares/bulk", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`POST /v3/workgroups/{workgroup_id}/shares/bulk`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`POST`: Add bulk amount of shared resource to the workgroup. Public App users need access to the **Create/Modify Workgroup Shares** scope

Body parameter

```
{
  "shares": [
    {
      "resource_type": "survey",
      "resource_id": "string"
    }
  ]
}
```

REQUEST BODY FOR POST /WORKGROUPS/{WORKGROUP_ID}/SHARES/BULK

Name	Type	Required	Description
resource_type	string	true	The type of resource that is supported, currently only sharing surveys is possible
resource_id	string	true	The id of the resource to share

Example Response for POST /workgroups/{workgroup_id}/shares/bulk

```
{
  "data": [
    {
      "created_at": "2023-11-27T22:30:32",
      "id": "87786e957e454442b315097477094369",
      "organization_id": "2093412",
      "owner_user_id": "101549492",
      "resource_id": "120472435",
      "resource_type": "survey",
      "workgroup_id": "8bd869a4ac84492f9c5d8d945fa7d16e",
      "workgroup_name": "sample test workgroup3"
    },
    {
      "created_at": "2023-11-28T19:03:05",
      "id": "8c01cc349f8d69404b43f68543793",
      "organization_id": "2093412",
      "owner_user_id": "10492",
      "resource_id": "12445",
      "resource_type": "survey",
      "workgroup_id": "869a4ac84492f9c5d945fa7d16e",
      "workgroup_name": "sample test workgroup3"
    }
  ],
  "links": {
    "self": "https://api.surveymonkey.com/v3/workgroups/8bd869a4ac84492f9c5d8d945fa7d16e/shares/bulk?per_page=50&page=1"
  },
  "page": 1,
  "per_page": 50,
  "total": 2
}
```

RESPONSE SCHEMA

Status Code 201

Name	Type	Description
page	integer	
per_page	integer	
total	integer	
links	object	
self	string	
data	[object]	
created_at	string	The date and time the share was created
id	string	The ID of the share that was created
organization_id	string	The ID of the organization the new shared resource belongs too
owner_user_id	string	The ID of the User that owns the resource
resource_id	string	The ID of the User that owns the resource
resource_type	string	The type of resource currently, only supports surveys
workgroup_id	string	The ID of the workgroup that the resource is shared with
workgroup_name	string	The name of the workgroup the resources is shared too.

Survey Pages and Questions

Folders are used to organize surveys. The following endpoints let you retrieve and create folders. Add surveys to a certain folder with a POST, PUT, or PATCH to /surveys including it's `folder_id`. You can also filter surveys by folders using `folder_id` as a query parameter on GET /surveys.

GET /surveys

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /surveys

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns a list of surveys owned or shared with the authenticated user. Public App users need access to the **View Surveys** scope

`POST`: Creates a new empty survey or, if a template id or an existing survey id is specified, a survey prepopulated with pages and questions. Public App users need access to the **Create/Modify Surveys** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
per_page	string	false	Number of resources to return per page
sort_by	string-Enum	false	Field used to sort returned survey list: <code>title</code> , <code>date_modified</code> , or <code>num_responses</code>
sort_order	string-Enum	false	Sort order: <code>ASC</code> or <code>DESC</code>
include	comma separated string-Enum	false	Use to filter survey list: <code>shared_with</code> , <code>shared_by</code> , or <code>owned</code> (useful for teams) or use to specify additional fields to return per survey: <code>response_count</code> , <code>date_created</code> , <code>date_modified</code> , <code>language</code> , <code>question_count</code> , <code>analyze_url</code> , <code>preview</code> , <code>collect_stats</code>
title	string	false	Search survey list by survey title
start_modified_at	string	false	Surveys must be last modified after this date. Date string in format YYYY-MM-DDTHH:MM:SS (no offset)
end_modified_at	string	false	Surveys must be last modified before this date. Date string in format YYYY-MM-DDTHH:MM:SS (no offset)
folder_id	string	false	Specify the id of a folder to only return surveys in it.

Example response for GET /surveys

```
{  
  "data": [  
    {  
      "id": "56789012345678901234567890123456",  
      "title": "My Survey",  
      "date_modified": "2023-01-01T12:00:00Z",  
      "date_created": "2023-01-01T12:00:00Z",  
      "language": "en-US",  
      "response_count": 0,  
      "question_count": 5,  
      "analyze_url": "https://analyze.surveymonkey.com/56789012345678901234567890123456",  
      "preview": "https://preview.surveymonkey.com/56789012345678901234567890123456",  
      "collect_stats": false  
    }  
  ]  
}
```

```
{
  "id": "1234",
  "title": "My Survey",
  "nickname": "",
  "href": "https://api.surveymonkey.com/v3/surveys/1234"
}
],
"per_page": 50,
"page": 1,
"total": 1,
"links": {
  "self": "https://api.surveymonkey.com/v3/surveys?page=1&per_page=50",
  "next": "https://api.surveymonkey.com/v3/surveys?page=2&per_page=50",
  "last": "https://api.surveymonkey.com/v3/surveys?page=5&per_page=50"
}
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
per_page	integer	
total	integer	
data	[object]	
id	string	Id of survey
title	string	Title of survey
nickname	string	Survey nickname
href	string	Resource URL for survey
links	object	
next	string	
last	string	
self	string	
page	integer	

POST /surveys

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"title\":\"New Survey\",\"from_template_id\":\"\",\"from_survey_id\":\"\",\"from_team_template_id\":\"\",\"nickname\":\"My Survey\",\"language\":\"en\",\"buttons_text\":{},\"next_button\":\"string\",\"prev_button\":\"string\",\"exit_button\":\"string\",\"done_button\":\"string\"},\"custom_variables\":{},\"footer\":true,\"folder_id\":\"1506280\",\"theme_id\":1506280,\"quiz_options\":{\"is_quiz_mode\":true,\"default_question_feedback\":{\"correct_text\":\"string\",\"incorrect_text\":\"string\",\"partial_text\":\"string\"},\"show_results_type\":\"string\"},\"feedback\":{\"ranges_type\":\"string\",\"ranges\":[{\"min\":0,\"max\":0,\"message\":\"string\"}]},\"pages\":[{\"questions\":[\"See formatting question types for more details\"]}]}"
headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/surveys", payload, headers)

res = conn.getresponse()
```

```

data = res.read()

print(data.decode("utf-8"))

```

[POST /surveys](#)

Call Types:

- [Blank](#): Creating a survey from scratch
- [Copy](#): Creating a survey by Copying an existing survey/template
- [Bulk](#): Creating multiple surveys at once

Body parameter

```
{
  "title": "New Survey",
  "from_template_id": "",
  "from_survey_id": "",
  "from_team_template_id": "",
  "nickname": "My Survey",
  "language": "en",
  "buttons_text": {
    "next_button": "string",
    "prev_button": "string",
    "exit_button": "string",
    "done_button": "string"
  },
  "custom_variables": {},
  "footer": true,
  "folder_id": "",
  "theme_id": 1506280,
  "quiz_options": {
    "is_quiz_mode": true,
    "default_question_feedback": {
      "correct_text": "string",
      "incorrect_text": "string",
      "partial_text": "string"
    },
    "show_results_type": "string",
    "feedback": {
      "ranges_type": "string",
      "ranges": [
        {
          "min": 0,
          "max": 0,
          "message": "string"
        }
      ]
    }
  },
  "pages": [
    {
      "questions": [
        "See formatting question types for more details"
      ]
    }
  ]
}
```

REQUEST BODY FOR POST /SURVEYS

Name	Type	Required	Call	Description
title	string	false	Copy Blank Bulk	Survey title (default="New Survey")
from_template_id	string	false	Copy	Survey template to Copy from. See /survey_templates (required if from_survey_id not provided)
from_survey_id	string	false	Copy	Survey id to Copy from (required if from_template_id not provided)
from_team_template_id	string	false	Copy	Team survey template to Copy from. See /team_survey_templates
nickname	string	false	Blank Bulk (default="")	Survey nickname

Name	Type	Required	Call	Description
language	string	false	Blank Bulk (default='en')	Survey language
buttons_text	object	false	Blank Bulk	Survey Buttons text container
next_button	string	false	Blank Bulk	Button text
prev_button	string	false	Blank Bulk	Button text
exit_button	string	false	Blank Bulk	Button text. If set to an empty string, button will be omitted from survey
done_button	string	false	Blank Bulk	Button text
custom_variables	object	false	Blank	Dictionary of survey variables
footer	boolean	false	Blank	If false, SurveyMonkey's footer is not displayed. (default=true)
folder_id	string	false	Blank	If specified, adds the survey to the folder with that id.
theme_id	integer	false	Blank	If specified, creates the survey using the theme provided
quiz_options	object	false	Blank	An object describing the quiz settings, if this survey is a quiz
is_quiz_mode	boolean	true	Blank	On/off toggle for setting this survey as a quiz
default_question_feedback	object	false	Blank	An object containing the default feedback for newly created questions in this survey
correct_text	string	true	Blank	Text to show when answer is correct
incorrect_text	string	true	Blank	Text to show when answer is incorrect
partial_text	string	true	Blank	Text to show when answer is partially correct
show_results_type	string	true	Blank	What to reveal to the user when they complete the quiz: disabled , results_only or results_and_answers
feedback	object	true	Blank	Text to show the user when they complete the quiz
ranges_type	string	true	Blank	Configure whether the following parameters use percentage or points . Note that these ranges are inclusive and may not overlap
ranges	[object]	true	Blank	The ranges at which to show users certain feedback
min	integer	true	Blank	Minimum score for this feedback
max	integer	true	Blank	Maximum score for this feedback
message	string	true	Blank	Feedback message
pages	[object]	true	Bulk	Pages to be created
questions	array	true	Bulk	Questions to be created

Example response for POST /surveys

```
{
  "title": "New Survey",
  "nickname": "",
  "language": "en",
  "folder_id": "0",
  "category": "",
  "question_count": 0,
  "page_count": 1,
  "response_count": 0,
  "date_created": "2021-07-26T14:18:00",
  "date_modified": "2021-07-26T14:18:00",
  "buttons_text": {
    "next": "Next",
    "prev": "Previous",
    "exit": "Exit"
  }
}
```

```

"id": "309683094",
"buttons_text": {
  "next_button": "Next",
  "prev_button": "Prev",
  "done_button": "Done",
  "exit_button": "Exit"
},
"is_owner": true,
"footer": true,
"custom_variables": {},
"href": "https://api.surveymonkey.com/v3/surveys/309683094",
"analyze_url": "https://www.surveymonkey.com/analyze/tdx6osJq3_2BVcoKKHn_2BZbgT0q9Dj4_2BqWejehwAh1_2BfQA_3D",
"edit_url": "https://www.surveymonkey.com/create/?sm=tdx6osJq3_2BVcoKKHn_2BZbgT0q9Dj4_2BqWejehwAh1_2BfQA_3D",
"collect_url": "https://www.surveymonkey.com/collect/list?sm=tdx6osJq3_2BVcoKKHn_2BZbgT0q9Dj4_2BqWejehwAh1_2BfQA_3D",
"summary_url": "https://www.surveymonkey.com/summary/tdx6osJq3_2BVcoKKHn_2BZbgT0q9Dj4_2BqWejehwAh1_2BfQA_3D",
"preview": "https://www.surveymonkey.com/r/Preview/?sm=B20TAhiT8YpGuFWTVyhm1Rpxe5fj_2BBP0hSy_2BKVTkwuN4gN4Rf6YxaaZ5KNey_2BAv"
}

```

RESPONSE SCHEMA

Status Code 201

Name	Type	Description
id	string	Survey id
title	string	Survey title
nickname	string	Survey nickname
custom_variables	object	Dictionary of survey variables
category	string	Survey category chosen when creating the survey
language	string	ISO 639-1 code for survey language
question_count	integer	Number of questions in survey
page_count	integer	Number of pages in survey
date_created	string	Date and time when survey was created. Date string in format YYYY-MM-DDTHH:MM:SS (no offset)
date_modified	string	Date and time when survey was last modified. Date string in format YYYY-MM-DDTHH:MM:SS (no offset)
buttons_text	object	
next_button	string	Button text
prev_button	string	Button text
exit_button	string	Button text
done_button	string	Button text
preview	string	Survey preview URL
folder_id	string	If applicable, the id of the folder the survey is in
edit_url	string	Survey edit URL
collect_url	string	Survey collect URL
analyze_url	string	Survey analyze URL
summary_url	string	Survey summary URL
href	string	Resource API URL
response_count	integer	Number of responses survey has received
footer	boolean	Whether or not SurveyMonkey's footer is not displayed

Name	Type	Description
theme_id	integer	The ID of the theme used to create the survey, if a theme was used.
template_id	integer	The ID of the template used to create the survey, if a template was used.

GET /surveys/{id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /surveys/{id}

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns a survey's details. To get an expanded version showing all pages and questions use /surveys/{survey_id}/details. Requires **View Surveys** scope

PATCH: Modifies a survey's title, nickname or language. Public App users need access to the **Create/Modify Surveys** scope

PUT: Replaces a survey. Request body arguments the same as POST /surveys. Public App users need access to the **Create/Modify Surveys** scope

DELETE: Deletes a survey. Public App users need access to the **Create/Modify Surveys** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
include	comma seperated string-Enum	false	Use to specify additional information. Supported values: <code>collect_stats</code> for detailed collector status.

Example response for calling GET, PUT, or PATCH /surveys

```
{
  "title": "New Survey",
  "nickname": "",
  "language": "en",
  "folder_id": "0",
  "category": "",
  "question_count": 0,
  "page_count": 1,
  "response_count": 0,
  "date_created": "2021-07-26T14:18:00",
  "date_modified": "2021-07-26T14:18:00",
  "id": "309683094",
  "buttons_text": {
    "next_button": "Next",
    "prev_button": "Prev",
    "done_button": "Done",
    "exit_button": "Exit"
  },
  "is_owner": true,
  "footer": true,
  "custom_variables": {},
  "href": "https://api.surveymonkey.com/v3/surveys/309683094",
  "analyze_url": "https://www.surveymonkey.com/analyze/ttx6osJq3_2BVcoKKHn_2BZbgT0q9Dj4_2BqWejehwAh1_2BfQA_3D",
```

```

"edit_url": "https://www.surveymonkey.com/create/?sm=tdx6osJq3_2BVcoKKHn_2BZbgT0q9Dj4_2BqWejehwAh1_2BfQA_3D",
"collect_url": "https://www.surveymonkey.com/collect/list?sm=tdx6osJq3_2BVcoKKHn_2BZbgT0q9Dj4_2BqWejehwAh1_2BfQA_3D",
"summary_url": "https://www.surveymonkey.com/summary/tdx6osJq3_2BVcoKKHn_2BZbgT0q9Dj4_2BqWejehwAh1_2BfQA_3D",
"preview": "https://www.surveymonkey.com/r/Preview/?sm=B20TAhiT8YpGuFWTyhmu1Rpxe5fj_2BBP0hSy_2KVTKwuW4gN4RF6YxaZ5KNEY_2BAv"
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Survey id
title	string	Survey title
nickname	string	Survey nickname
custom_variables	object	Dictionary of survey variables
category	string	Survey category chosen when creating the survey
language	string	ISO 639-1 code for survey language
question_count	integer	Number of questions in survey
page_count	integer	Number of pages in survey
date_created	string	Date and time when survey was created. Date string in format YYYY-MM-DDTHH:MM:SS (no offset)
date_modified	string	Date and time when survey was last modified. Date string in format YYYY-MM-DDTHH:MM:SS (no offset)
buttons_text	object	
next_button	string	Button text
prev_button	string	Button text
exit_button	string	Button text
done_button	string	Button text
preview	string	Survey preview URL
folder_id	string	If applicable, the id of the folder the survey is in
edit_url	string	Survey edit URL
collect_url	string	Survey collect URL
analyze_url	string	Survey analyze URL
summary_url	string	Survey summary URL
href	string	Resource API URL
response_count	integer	Number of responses survey has received
footer	boolean	Whether or not SurveyMonkey's footer is not displayed
theme_id	integer	The ID of the theme used to create the survey, if a theme was used.
template_id	integer	The ID of the template used to create the survey, if a template was used.

PUT /surveys/{id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"title\":\"New Survey\",\"from_template_id\":\"\",\"from_survey_id\":\"\",\"from_team_template_id\":\"\",\"nickname\":\"My Survey\",\"language\":\"en\",\"buttons_text\":[{\"next_button\":\"string\",\"prev_button\":\"string\",\"exit_button\":\"string\",\"done_button\":\"string\"}],\"custom_variables\":{},\"footer\":true,\"folder_id\":\"\",\"theme_id\":1506280,\"quiz_options\":{\"is_quiz_mode\":true,\"default_question_feedback\":{\"correct_text\":\"string\",\"incorrect_text\":\"string\",\"partial_text\":\"string\"}},\"show_results_type\":\"string\",\"feedback\":{\"ranges_type\":\"string\",\"ranges\":[{\"min\":0,\"max\":0,\"message\":\"string\"}]}},\"pages\":[{\"questions\":[\"See formatting question types for more details\"]}]}"}

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PUT", "/v3/surveys/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`PUT /surveys/{id}`

Request body identical to POST /surveys

PATCH /surveys/{id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"title\":\"New Survey\",\"nickname\":\"My Survey\",\"language\":\"en\",\"buttons_text\":[{\"next_button\":\"string\",\"prev_button\":\"string\",\"exit_button\":\"string\"}],\"done_button\":\"string\"},\"custom_variables\":{},\"footer\":true,\"folder_id\":\"\"}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/v3/surveys/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`PATCH /surveys/{id}`

Body parameter

```
{
    "title": "New Survey",
    "nickname": "My Survey",
    "language": "en",
    "buttons_text": [
        "next_button": "string",
        "prev_button": "string",
        "exit_button": "string",
        "done_button": "string"
    ],
    "custom_variables": {},
    "footer": true,
    "folder_id": ""
}
```

REQUEST BODY FOR PATCH /SURVEYS/{ID}

Name	Type	Required	Description
title	string	false	Survey title (PUT default="New Survey")
nickname	string	false	Survey nickname (PUT default "")
language	string	false	Survey language (PUT default="en")
buttons_text	object	false	Survey Buttons text container
next_button	string	false	Button text
prev_button	string	false	Button text
exit_button	string	false	Button text. If set to an empty string, button will be omitted from survey
done_button	string	false	Button text
custom_variables	object	false	Dictionary of survey variables
footer	boolean	false	If false, SurveyMonkey's footer is not displayed (default=true)
folder_id	string	false	If specified, adds the survey to the folder with that id.

GET /surveys/{id}/details

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/details", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /surveys/{id}/details

AVAILABLE METHODS

GET: Returns an expanded survey resource with a `pages` element containing a list of all page objects, each containing a list of questions objects. Public App users need access to the **View Surveys** scope

Example Response for GET /surveys/{id}/details

```
{
    "title": "New Survey",
    "nickname": "",
    "language": "en",
    "folder_id": "0",
    "category": "",
    "question_count": 0,
    "page_count": 1,
    "response_count": 0,
    "date_created": "2021-07-26T18:09:00",
    "date_modified": "2021-07-26T19:32:00",
    "id": "1",
    "buttons_text": {
        "next_button": "Next",
        "prev_button": "Prev",
    }
}
```

```

    "done_button": "Done",
    "exit_button": "Exit"
},
"is_owner": true,
"footer": true,
"custom_variables": {},
"href": "https://api.surveymonkey.com/v3/surveys/1",
"analyze_url": "https://www.surveymonkey.com/analyze/g1_2BAICXZEi4rH4ITcFzAin50QyBg8dHsw877lCBjYlk_3D",
"edit_url": "https://www.surveymonkey.com/create/?sm=g1_2BAICXZEi4rH4ITcFzAAin50QyBg8dHsw877lCBjYlk_3D",
"collect_url": "https://www.surveymonkey.com/collect/list?sm=g1_2BAICXZEi4rH4ITcFzAAin50QyBg8dHsw877lCBjYlk_3D",
"summary_url": "https://www.surveymonkey.com/summary/g1_2BAICCXZEi4rH4ITcFzAin50QyBg8dHsw877lCBjYlk_3D",
"preview": "https://www.surveymonkey.com/r/Preview/?sm=UY_2B1ACesAm789uYe_2B0Zln_2Fs_2F9GndhH015uffhkTaxfBCBn3Gcj_2BTQrIRea7upQwr2",
"pages": [
{
    "title": "",
    "description": "",
    "position": 1,
    "question_count": 0,
    "id": "1",
    "href": "https://api.surveymonkey.com/v3/surveys/1/pages/1",
    "questions": []
}
]
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
preview	string	
response_count	integer	
collect_url	string	
buttons_text	object	
exit_button	string	
prev_button	string	
done_button	string	
next_button	string	
footer	boolean	
date_created	string	
is_owner	boolean	
custom_variables	object	
language	string	
title	string	
question_count	integer	
pages	[object]	
question_count	integer	
questions	[string]	
description	string	
position	integer	
id	string	

Name	Type	Description
href	string	
title	string	
date_modified	string	
nickname	string	
edit_url	string	
id	string	
href	string	
folder_id	string	
category	string	
analyze_url	string	
summary_url	string	
page_count	integer	

POST /surveys/{survey_id}/share

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"title\":\"title\",\"description\":\"share description\",\"modes\":[\"summary\",\"trends\"],\"enabled\":true,\"is_public\":true,\"branded\":true,\"hide_open-ended\":true,\"password\":\"p6nehgv2**3ddd\",\"domain\":\"www.surveymonkey.com\"}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/surveys/111/share", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[POST /surveys/{survey_id}/share](#)

AVAILABLE METHODS

POST: Create a survey results page link. Public App users need access to the **View Responses** scope

PUT: Update an existing survey results page link. Public App users need access to the **View Responses** scope

Body parameter

```
{
    "title": "title",
    "description": "share description",
    "modes": [
        "summary",
        "trends"
    ],
    "enabled": true,
    "is_public": true,
    "branded": true,
```

```

    "hide_open-ended": true,
    "password": "p6nehgv2**3ddd",
    "domain": "www.surveymonkey.com"
}

```

REQUEST BODY FOR POST /SURVEYS/{SURVEY_ID}/SHARE

Name	Type	Required	Description
title	string	true	Title of the survey results page
description	string	false	Description of the survey results page. Defaults to empty string.
modes	[string]	false	Defines the modes available on the survey results pages. Defaults to [summary].
enabled	boolean	false	If the link to be created should be enabled. If false, the link will redirect to /unavailable. Defaults to true
is_public	boolean	false	Determines if the share results page is public. Defaults to true
branded	boolean	false	If the branding is turned on (has the SurveyMonkey logo) on the survey results page. Defaults to true (branded). Custom branding is not supported. User must have a paid plan to hide branding
hide_open-ended	boolean	false	Display text responses on the share results page. Defaults to true
password	string	false	If set, it creates a password for the survey results page for access control. Must be 4-35 characters. User must have a paid plan to set a password
domain	string	false	The domain hosting the shared link. Use this together with the shared view key to generate your shared URL, we will default to using "www.surveymonkey.com"

Example Response for POST /surveys/{survey_id}/share

```
{
  "enabled": true,
  "is_public": true,
  "hide_open-ended": true,
  "modes": [
    "summary"
  ],
  "type": "weblink",
  "title": "Survey Results",
  "description": "Survey Results Description",
  "branding": "none",
  "password": "p6nehgv2**3ddd",
  "id": 89202,
  "date_created": "2023-08-03T16:50:02.877000+00:00",
  "num_of_views": 0,
  "survey_id": "111",
  "key": "SM-2mkV76m9_2F_0BXyxsks99XjICXg_0D_3D",
  "has_password": true
}
```

RESPONSE SCHEMA

Status Code 201

Name	Type	Description
enabled	boolean	If the link is enabled. If false, the results link will default to /unavailable
is_public	boolean	The share results page is public view
modes	[string]	Defines the modes available on the survey results page.
type	string	Defines what the share results type is out of the options: weblink
title	string	Title of the survey results page
description	string	Description of the survey results page

Name	Type	Description
branding	string	The branding is on: <code>default</code> or off: <code>none</code>
password	string	The password for the survey results page
id	number	The id of the survey results
date_created	string	Date the share results were created
num_of_views	number	The number of views the share results has
survey_id	string	Id of the survey the share results are created for
key	string	The key used in the share results url of the form: domain/results/{key}, where the domain is the associated survey domain. Note that the domain is not customizable via this endpoint.
has_password	boolean	If the share results url has a password

GET /surveys/{survey_id}/share_list

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/111/share_list", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`GET /surveys/{survey_id}/share_list`

AVAILABLE METHODS

`GET`: Return a list of survey results page links generated for the survey with this `survey_id`.

Example Response for POST /surveys/{survey_id}/share

```
[
  {
    "enabled": true,
    "is_public": true,
    "hide_open_ended": true,
    "modes": [
      "summary"
    ],
    "type": "weblink",
    "title": "Survey Results",
    "description": "Survey Results Description",
    "branding": "none",
    "password": "p6nehgv2**3dd",
    "id": 89202,
    "date_created": "2023-08-03T16:50:02.877000+00:00",
    "num_of_views": 0,
    "survey_id": "111",
    "key": "SM-2mkV76n9_2F_0BXyxsks99XjICXg_0D_3D",
    "has_password": true
  }
]
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
enabled	boolean	If the link is enabled. If false, the results link will default to <code>/unavailable</code>
is_public	boolean	The share results page is public view
modes	[string]	Defines the modes available on the survey results page.
type	string	Defines what the share results type is out of the options: <code>weblink</code>
title	string	Title of the survey results page
description	string	Description of the survey results page
branding	string	The branding is on: <code>default</code> or off: <code>none</code>
password	string	The password for the survey results page
id	number	The id of the survey results
date_created	string	Date the share results were created
num_of_views	number	The number of views the share results has
survey_id	string	Id of the survey the share results are created for
key	string	The key used in the share results url of the form: domain/results/{key}, where the domain is the associated survey domain. Note that the domain is not customizable via this endpoint.
has_password	booleans	If the share results url has a password

GET /survey_categories

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/survey_categories", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`GET /survey_categories`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns a list of survey categories that can be used to filter survey templates. Public App users need access to the **View Library Assets** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1

Name	Type	Required	Description
per_page	string	false	Number of resources to return per page
language	string	false	ISO 639-1 code for language to filter by (default=en)

Example Response for GET /survey_categories

```
{
  "page": 1,
  "per_page": 1,
  "total": 1,
  "data": [
    {
      "name": "Category Name",
      "id": "community"
    }
  ],
  "links": {
    "self": "https://api.surveymonkey.com/v3/survey_categories?page=1&per_page=1"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
per_page	integer	
page	integer	
total	integer	
data	[object]	
name	string	Resource name
id	string	Resource id
links	object	
self	string	

GET /survey_templates

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
  'Accept': "schema",
  'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/survey_templates", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /survey_templates

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns a list of survey templates. Survey template ids can be used as an argument to **POST** a new survey. Public App users need access to the **View Library Assets** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
per_page	string	false	Number of resources to return per page
language	string	false	ISO 639-1 code for language to filter by (default=en)
category	string	false	Category to filter by, see /survey_categories for a list of available categories if not specified, all categories are returned

Example Response for GET /survey_templates

```
{  
  "page": 1,  
  "per_page": 1,  
  "total": 1,  
  "data": [  
    {  
      "category": "community",  
      "name": "Template Name",  
      "title": "Template Name",  
      "available": true,  
      "id": "49",  
      "num_questions": 10,  
      "description": "Template description",  
      "preview_link": "https://www.surveymonkey.com/r/Preview/?sm=ID"  
    }  
  ],  
  "links": {  
    "self": "https://api.surveymonkey.com/v3/survey_templates?page=1&per_page=1"  
  }  
}
```

RESPONSE SCHEMA

GET /team_survey_templates

Code samples

```
import http.client  
  
conn = http.client.HTTPSConnection("api.surveymonkey.com")  
  
headers = {  
  'Accept': "application/json",  
  'Authorization': "Bearer {access-token}"  
}  
  
conn.request("GET", "/v3/team_survey_templates", headers=headers)  
  
res = conn.getresponse()  
data = res.read()  
  
print(data.decode("utf-8"))
```

GET /team_survey_templates

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns a list of team survey templates corresponding to the team to which the calling user belongs. Team survey template ids (team_template_id) can be used as an argument to **POST** a new survey. Public App users need access to the **View Library Assets** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
per_page	string	false	Number of resources to return per page

Example Response for GET /team_survey_templates

```
{  
    "page": 1,  
    "per_page": 1,  
    "total": 1,  
    "data": [  
        {  
            "name": "Template Name",  
            "team_template_id": "49",  
            "survey_id": "50"  
        }  
    ],  
    "links": {  
        "self": "https://api.surveymonkey.com/v3/survey_templates?page=1&per_page=1"  
    }  
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
per_page	integer	
page	integer	
total	integer	
data	[object]	
name	string	Name of the team template
team_template_id	string	Team template ID (can be used when posting a new survey)
survey_id	string	ID of the survey from which this team template was derived
links	object	
self	string	

GET /survey_languages

Code samples

```
import http.client  
  
conn = http.client.HTTPSConnection("api.surveymonkey.com")  
  
headers = {  
    'Accept': "application/json",  
    'Authorization': "Bearer {access-token}"  
}  
  
conn.request("GET", "/v3/survey_languages", headers=headers)  
  
res = conn.getresponse()
```

```
data = res.read()  
  
print(data.decode("utf-8"))
```

GET /survey_languages

AVAILABLE METHODS

- HEAD**: Checks if resource is available
- OPTIONS**: Returns available methods and options
- GET**: Returns a list of survey languages that can be used to generate translations for multilingual surveys

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
per_page	string	false	Number of resources to return per page

Example Response for GET /survey_languages

```
{  
  "page": 1,  
  "per_page": 1,  
  "total": 1,  
  "data": [  
    {  
      "name": "English",  
      "native_name": "English",  
      "id": "en"  
    }  
  ],  
  "links": {  
    "self": "https://api.surveymonkey.com/v3/survey_languages?page=1&per_page=1"  
  }  
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
per_page	integer	
page	integer	
total	integer	
data	[object]	
name	string	Name of language
id	string	Language code
native_name	string	Name of language in native language
links	object	
self	string	

GET /surveys/{survey_id}/pages

Code samples

```
import http.client
```

```

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/pages", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /surveys/{survey_id}/pages`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns a page's details. Public App users need access to the **View Surveys** scope

`POST`: Creates a new, empty page, see `/surveys/{id}/pages/{id}/questions` to add questions to pages. Public App users need access to the **Create/Modify Surveys** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
per_page	integer	false	Number of resources to return per page. Defaults to 50
page	integer	false	Which page of resources to return. Defaults to 1

Example Response for `GET /surveys/{id}/pages`

```
{
  "data": [
    {
      "title": "",
      "description": "",
      "position": 1,
      "id": "172949600",
      "href": "https://api.surveymonkey.com/v3/surveys/309695619/pages/172949600"
    },
    {
      "title": "My Folder1",
      "description": "",
      "position": 2,
      "id": "173176852",
      "href": "https://api.surveymonkey.com/v3/surveys/309695619/pages/173176852"
    }
  ],
  "per_page": 50,
  "page": 1,
  "total": 3,
  "links": {
    "self": "https://api.surveymonkey.com/v3/surveys/309695619/pages?page=1&per_page=50"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[object]	
title	string	Name of the page
description	string	Description of the page
position	integer	Position of the page in the survey
id	string	ID of the page

Name	Type	Description
href	string	Link to the page
per_page	integer	Number of resources per page
page	integer	Page of the resource returned
total	integer	Number of resources returned
links	object	Contains this endpoint's links
self	string	The link associated with this endpoint

POST /surveys/{survey_id}/pages

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"title\":\"Page title\",\"description\":\"\",\"position\":3}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/surveys/1234/pages", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

POST /surveys/{survey_id}/pages

Body parameter

```
{
  "title": "Page title",
  "description": "",
  "position": 3
}
```

REQUEST BODY FOR POST /SURVEYS/{SURVEY_ID}/PAGES

Name	Type	Required	Description
title	string	false	Page title
description	string	false	Page description
position	integer	false	Page position in survey

GET /surveys/{survey_id}/pages/{page_id}

Code samples

```
import http.client
```

```

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/pages/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

GET /surveys/{survey_id}/pages/{page_id}

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns a page's details. Public App users need access to the **View Surveys** scope

PATCH: Modifies a page (updates any fields accepted as arguments to **POST** /surveys{id}/pages). Public App users need access to the **Create/Modify Surveys** scope

PUT: Replaces a page (same arguments and requirements as **POST** /surveys{id}/pages). Public App users need access to the **Create/Modify Surveys** scope

DELETE: Deletes a page. Public App users need access to the **Create/Modify Surveys** scope

Example Response for GET, PATCH, PUT /surveys/{survey_id}/pages/{page_id}

```
{
    "title": "My Page1",
    "description": "",
    "position": 4,
    "question_count": 0,
    "id": "173177756",
    "href": "https://api.surveymonkey.com/v3/surveys/309695619/pages/173177756"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
title	string	Name of the page
description	string	Description of the page
position	integer	Position of the page in the survey
question_count	integer	Number of questions on the page
id	string	ID of the page
href	string	Link to the page

PATCH /surveys/{survey_id}/pages/{page_id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"title\":\"Page title\",\"description\":\"\",\"position\":3}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/v3/surveys/1234/pages/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

```
}

conn.request("PATCH", "/v3/surveys/1234/pages/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

PATCH /surveys/{survey_id}/pages/{page_id}

PATCH: Modifies a page (updates any fields accepted as arguments to **POST** /surveys{id}/pages). Public App users need access to the **Create/Modify Surveys** scope

PUT /surveys/{survey_id}/pages/{page_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"title\":\"Page title\",\"description\":\"\",\"position\":3}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PUT", "/v3/surveys/1234/pages/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

PUT /surveys/{survey_id}/pages/{page_id}

PUT: Replaces a page (same arguments and requirements as **POST** /surveys{id}/pages). Public App users need access to the **Create/Modify Surveys** scope

GET /surveys/{survey_id}/pages/{page_id}/questions

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/pages/1234/questions", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /surveys/{survey_id}/pages/{page_id}/questions

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns a list of questions on a survey page. Public App users need access to the **View Surveys** scope

POST: Creates a new question on a survey page, see formatting question types. Public App users need access to the **Create/Modify Surveys** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
per_page	integer	false	Number of resources to return per page
page	integer	false	Which page of resources to return. Defaults to 1

Example Response for GET /surveys/{survey_id}/pages/{page_id}/questions

```
{
  "data": [
    {
      "heading": "Question 1",
      "position": 1,
      "id": "1",
      "href": "https://api.surveymonkey.com/v3/surveys/1/pages/1/questions/1"
    },
    {
      "heading": "Question 2",
      "position": 2,
      "id": "2",
      "href": "https://api.surveymonkey.com/v3/surveys/1/pages/1/questions/2"
    }
  ],
  "per_page": 50,
  "page": 1,
  "total": 2,
  "links": {
    "self": "https://api.surveymonkey.com/v3/surveys/1/pages/1/questions?page=1&per_page=50"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
page	integer	
per_page	integer	
total	integer	Number of questions
data	object	
position	integer	Position of the question in the page
id	string	ID of the question
href	string	Link to the question
heading	string	Heading of the question
links	object	
self	string	

POST /surveys/{survey_id}/pages/{page_id}/questions

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"headings\": [{\"heading\": \"Which monkey would you rather have as a pet?\", \"description\": \"\", \"image\": {\"url\": \"string\"}, \"random_assignment\": {\"percent\": 0, \"position\": 0}, \"variable_name\": \"string\"}], \"position\": 0, \"family\": \"single_choice\", \"subtype\": \"vertical\", \"sorting\": {\"type\": \"string\", \"ignore_last\": true}, \"required\": false, \"validation\": {\"type\": \"string\", \"text\": \"string\", \"min\": \"string\", \"max\": \"string\", \"sum\": 0, \"sum_text\": \"string\"}, \"forced_ranking\": false, \"quiz_options\": {\"scoring_enabled\": true, \"feedback\": {}, \"feedback.correct_text\": \"string\", \"feedback.incorrect_text\": \"string\", \"feedback.partial_text\": \"string\"}, \"answers\": {\"choices\": [{\"text\": \"Capuchin\"}, {\"text\": \"Mandrill\"}]}}, \"display_options\": {}}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/surveys/1234/pages/1234/questions", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`POST /surveys/{survey_id}/pages/{page_id}/questions`

Body parameter

```
{
  "headings": [
    {
      "heading": "Which monkey would you rather have as a pet?",
      "description": "",
      "image": {
        "url": "string"
      },
      "random_assignment": {
        "percent": 0,
        "position": 0,
        "variable_name": "string"
      }
    }
  ],
  "position": 0,
  "family": "single_choice",
  "subtype": "vertical",
  "sorting": {
    "type": "string",
    "ignore_last": true
  },
  "required": false,
  "validation": {
    "type": "string",
    "text": "string",
    "min": "string",
    "max": "string",
    "sum": 0,
    "sum_text": "string"
  },
  "forced_ranking": false,
  "quiz_options": {
    "scoring_enabled": true,
    "feedback": {},
    "feedback.correct_text": "string",
    "feedback.incorrect_text": "string",
    "feedback.partial_text": "string"
  },
  "answers": {
    "choices": [
      {
        "text": "Capuchin"
      },
      {
        "text": "Mandrill"
      }
    ]
  }
},
```

```

    "display_options": {}
}

```

REQUEST BODY FOR POST /SURVEYS/{SURVEY_ID}/PAGES/{PAGE_ID}/QUESTIONS

Name	Type	Required	Description
headings	[object]	true	List of question headings objects
heading	string	true	The title of the question, or empty string if <code>random_assignment</code> is defined
description	string	false	If <code>random_assignment</code> is defined, and <code>family</code> is <code>presentation_image</code> this is the title
image	object	false	Image data when question <code>family</code> is <code>presentation_image</code>
url	string	false	URL of image when question <code>family</code> is <code>presentation_image</code>
random_assignment	object	false	Random assignment data
percent	integer	true	Percent chance of this random assignment showing up (must sum to 100)
position	integer	false	Position of the random_assignment in survey creation page
variable_name	string	false	Internal use name for question tracking, can be <code>""</code>
position	integer	false	Position of question on page (default=last)
family	string	true	Question family determines the type of question, see formatting question types
subtype	string	true	Question family's subtype further specifies the type of question, see formatting question types
sorting	object	false	Sorting details of answers
type	string	true	Sort answer choices by: <code>default</code> , <code>textasc</code> , <code>textdesc</code> , <code>resp_count_asc</code> , <code>resp_count_desc</code> , <code>random</code> , <code>flip</code>
ignore_last	boolean	false	If true, does not sort the last answer option (useful for “none of the above”, etc)
required	object	false	Whether an answer is required for this question
text	string	true	Text to display if a required question is not answered
type	string	true	Specifies how much of the question must be answered: <code>all</code> , <code>at_least</code> , <code>at_most</code> , <code>exactly</code> , or <code>range</code> . Required if question is matrix_single, matrix_ranking, and matrix menu
amount	string	true	The amount of answers required to be answered. If the required type is range then this is two numbers separated by a comma, as a string (e.g. “1,3” to represent the range of 1 to 3). Required if type is defined.
validation	object	false	Whether the answer must pass certain validation parameters
type	string	true	Type of validation to use: <code>any</code> , <code>integer</code> , <code>decimal</code> , <code>date_us</code> , <code>date_intl</code> , <code>email</code> , or <code>text_length</code>
text	string	true	Text to display if validation fails
min	string	true	Minimum value an answer can be to pass validation. Date string, integer, or null depending on validation.type
max	string	true	Maximum value an answer can be to pass validation. Date string, integer, or null depending on validation.type
sum	integer	false	Only accepted is family=open-ended and subtype=numerical, the exact integer textboxes must sum to in order to pass validation
sum_text	string	false	Only accepted is family=open-ended and subtype=numerical, the message to display if textboxes do not sum to validation.sum
forced_ranking	boolean	false	Required if type is matrix and subtype is rating or single, whether or not to force ranking
quiz_options	object	false	Object containing the quiz properties of this question, if quiz-mode is enabled

Name	Type	Required	Description
scoring_enabled	boolean	true	Whether this question is quiz-enabled
feedback	object	true	Object containing the definitions for feedback on this quiz question
feedback.correct_text	string	true	Text to show when answer is correct
feedback.incorrect_text	string	true	Text to show when the answer is incorrect
feedback.partial_text	string	true	Text to show when the answer is partially correct
answers	object	true	Answers object, refer to Formatting Question Types. Required for all question types except open_ended_single
display_options	object	false	Display option object, refer to Formatting Question Types. Required for File Upload, Slider, Image Choice, & Emoji/Star Rating question types

200 Response

```
{
  "headings": [
    {
      "heading": "string",
      "description": "string",
      "image": {
        "url": "string"
      },
      "random_assignment": {
        "percent": 0,
        "position": 0,
        "variable_name": "string",
        "assignment_id": "string"
      }
    }
  ],
  "position": 0,
  "visible": true,
  "family": "string",
  "subtype": "string",
  "sorting": {
    "type": "string",
    "ignore_last": true
  },
  "required": {
    "text": "string",
    "type": "string",
    "amount": "string"
  },
  "validation": {
    "type": "string",
    "text": "string",
    "min": "string",
    "max": "string",
    "sum": 0,
    "sum_text": "string"
  },
  "forced_ranking": true,
  "answers": {},
  "id": "string"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
headings	[object]	List of question headings objects
heading	string	The title of the question, or empty string if <code>random_assignment</code> is defined
description	string	If <code>random_assignment</code> is defined, and <code>family</code> is <code>presentation_image</code> this is the title

Name	Type	Description
image	object	Image data when question <code>family</code> is <code>presentation_image</code>
url	string	URL of image when question <code>family</code> is <code>presentation_image</code>
random_assignment	object	Random assignment data
percent	integer	Percent chance of this random assignment showing up (must sum to 100)
position	integer	Position of the random_assignment in survey creation page
variable_name	string	Internal use name for question tracking, can be <code>""</code>
assignment_id	string	Internal use id for question tracking
position	integer	Position of question on page (default=last)
visible	boolean	Whether the question is visible (corresponds with being deleted in the UI)
family	string	Question family determines the type of question, see formatting question types
subtype	string	Question family's subtype further specifies the type of question, see formatting question types
sorting	object	Sorting details of answers
type	string	Sort answer choices by: <code>default</code> , <code>textasc</code> , <code>textdesc</code> , <code>resp_count_asc</code> , <code>resp_count_desc</code> , <code>random</code> , <code>flip</code>
ignore_last	boolean	If true, does not sort the last answer option (useful for "none of the above", etc)
required	object	Whether an answer is required for this question
text	string	Text to display if a required question is not answered
type	string	Specifies how much of the question must be answered: <code>all</code> , <code>at_least</code> , <code>at_most</code> , <code>exactly</code> , or <code>range</code> . Required if question is matrix_single, matrix_ranking, and matrix menu
amount	string	The amount of answers required to be answered. If the required type is range then this is two numbers separated by a comma, as a string (e.g. "1,3" to represent the range of 1 to 3). Required if type is defined
validation	object	Whether the answer must pass certain validation parameters
type	string	Type of validation to use: <code>any</code> , <code>integer</code> , <code>decimal</code> , <code>date_us</code> , <code>date_intl</code> , <code>email</code> , or <code>text_length</code>
text	string	Text to display if validation fails
min	string	Minimum value an answer can be to pass validation. Date string, integer, or null depending on validation.type
max	string	Maximum value an answer can be to pass validation. Date string, integer, or null depending on validation.type
sum	integer	Only accepted is family=open-ended and subtype=numerical, the exact integer textboxes must sum to in order to pass validation
sum_text	string	Only accepted is family=open-ended and subtype=numerical, the message to display if textboxes do not sum to validation.sum
forced_ranking	boolean	Required if type is matrix and subtype is rating or single, whether or not to force ranking
answers	object	Answers object, refer to Formatting Question Types. Required for all question types except open-ended_single
id	string	Question id

GET /surveys/{survey_id}/pages/{page_id}/questions/{question_id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/pages/1234/questions/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /surveys/{survey_id}/pages/{page_id}/questions/{question_id}`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns a question. Requires **View Surveys** scope

`PATCH`: Updates a question (updates any fields accepted as arguments to `POST` `/surveys/{id}/pages/questions`). Public App users need access to the

Create/Modify Surveys scope

`PUT`: Replaces a question (same arguments and requirements as `POST` `/surveys/{id}/pages/questions`). Public App users need access to the

Create/Modify Surveys scope

`DELETE`: Deletes a question. Public App users need access to the **Create/Modify Surveys** scope

200 Response

```
{
  "headings": {
    "heading": "string"
  },
  "position": 0,
  "family": "string",
  "subtype": "string",
  "answers": {
    "choices": [
      {
        "text": "string"
      }
    ],
    "others": [
      {
        "text": "string",
        "num_chars": 0,
        "num_lines": 0
      }
    ]
  }
}
```

Formatting Question Types

All questions have a `family` and `subtype` that define their type and some questions have a `display_type` and `display_subtype` that further define their type. See below for example formatting of the answers object and `display_options` object for different question types. Read more about SurveyMonkey's question types in our help center.

Family	Subtype	Display_Type	Display_Subtype
single_choice	'vertical', 'horiz', 'menu'	'image_choice'	NA
matrix	'single', 'rating', 'ranking', 'menu', 'multi'	'emoji' (with 'ranking')	'star'
open-ended	'single', 'multi', 'numerical', 'essay'	'slider', 'file_upload' (with 'single')	NA
demographic	'international', 'us', 'address', 'phone', 'email', 'name'	NA	NA

Family	Subtype	Display_Type	Display_Subtype
datetime	'both', 'date_only', 'time_only'	NA	NA
multiple_choice	'vertical'	'image_choice'	NA
presentation	'descriptive_text', 'image'	NA	NA

SINGLE CHOICE

Single Choice

```
{
  "headings": [
    {
      "heading": "Which monkey would you rather have as a pet?"
    }
  ],
  "position": 1,
  "family": "single_choice",
  "subtype": "vertical",
  "answers": {
    "choices": [
      {
        "text": "Capuchin"
      },
      {
        "text": "Mandrill"
      }
    ],
    "other": [
      {
        "text": "Other",
        "num_chars": 100,
        "num_lines": 3
      }
    ]
  }
}
```

1. Which monkey would you rather have as a pet?

- Capuchin
- Mandrill

Name	Description	Data Type
choices (required)	List of available choices for the user	Array
choices[].text (required)	Choice for user selection	String
choices[].position (optional)	Position of the current choice	Integer
other (optional)	List of other answer options	Object
other[].text	Text to display next to other option	String
other[].num_chars	Set a character limit to the option	Integer
other[].num_lines	Set a line limit to the option	Integer

MULTIPLE CHOICE

Multiple Choice

```
{
  "headings": [
    {

```

```

        "heading": "Which monkeys would you like as pets?"
    },
    "position": 1,
    "family": "multiple_choice",
    "subtype": "vertical",
    "answers": {
        "choices": [
            {
                "text": "Capuchin"
            },
            {
                "text": "Mandrill"
            }
        ]
    }
}

```

1. Which monkey would you rather have as a pet?

- Capuchin
- Mandrill

Name	Description	Data Type
choices (required)	List of available choices for the user	Array
choices[_.text (required)	Choice for user selection	String
choices[_.position (optional)	Position of the current choice	Integer
other (optional)	List of other answer options	Object
other[_.text	Text to display next to other option	String
other[_.num_chars	Set a character limit to the option	Integer
other[_.num_lines	Set a line limit to the option	Integer

IMAGE CHOICE

Image Choice

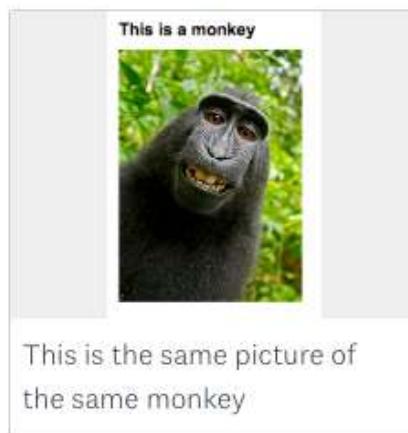
```

{
    "headings": [
        {
            "heading": "Which monkey do you like more? Pick one or more"
        }
    ],
    "position": 1,
    "family": "multiple_choice",
    "subtype": "vertical",
    "display_options": {
        "display_type": "image_choice"
    },
    "answers": {
        "choices": [
            {
                "text": "This is an image of a monkey",
                "image": {
                    "url": "https://path.to.image.jpg"
                }
            },
            {
                "text": "This is the same picture of the same monkey",
                "image": {
                    "url": "https://path.to.image.jpg"
                }
            }
        ]
    }
}

```

```
}
```

1. Which monkey do you like more? Pick one or more



Name	Description	Data Type
choices (required)	List of available choices for the user	Array
choices[_.image] (required)	Image choice	Object
choices[_.image[_.url] (required)	URL of image choice	String
choices[_.text] (optional)	Choice for user selection	String
choices[_.position] (optional)	Position of the current choice	Integer
display_options (required)	Object of display options	Object
display_options[_.display_type]	Type of the display	String ("image_choice")

MATRIX - SINGLE

Matrix - Single

```
{
  "headings": [
    {
      "heading": "What is each of your family members' favorite ice cream?"
    }
  ],
  "position": 1,
  "family": "matrix",
  "subtype": "single",
  "forced_ranking": false,
  "answers": {
    "rows": [
      {
        "text": "Mother"
      },
      {
        "text": "Father"
      }
    ],
    "choices": [
      {
        "text": "Chocolate"
      },
      {
        "text": "Vanilla"
      },
      {
        "text": "Strawberry"
      }
    ]
  }
}
```

```
}
```

1. What is each of your family members' favorite ice cream?

	Chocolate	Vanilla	Strawberry
Mother	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Father	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Name	Description	Data Type
rows (required)	List of rows in the matrix	Array
rows[_].text (required)	Text label for the row	String
rows[_].position (optional)	Position of the row	Integer
choices (required)	List of available choices for the user	Array
choices[_].text (required)	Choice for user selection	String
choices[_].position (optional)	Position of the current choice	Integer
other (optional)	List of other answer options	Object
other[_].text	Text to display next to other option	String
other[_].num_chars	Set a character limit to the option	Integer
other[_].num_lines	Set a line limit to the option	Integer

MATRIX - RATING

Matrix - Rating

```
{
  "headings": [
    {
      "heading": "What is each of your family members' favorite ice cream?"
    }
  ],
  "position": 1,
  "family": "matrix",
  "subtype": "rating",
  "forced_ranking": false,
  "answers": {
    "rows": [
      {
        "text": "Mother"
      },
      {
        "text": "Father"
      }
    ],
    "choices": [
      {
        "text": "Chocolate",
        "weight": 1
      },
      {
        "text": "Vanilla",
        "weight": 1
      },
      {
        "text": "Strawberry",
        "weight": 1
      }
    ]
  }
}
```

```
}
```

1. What is each of your family members favorite ice cream?

	Chocolate	Vanilla	Strawberry
Mother	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Father	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Name	Description	Data Type
rows (required)	List of rows in the matrix	Array
rows[_.text (required)	Text label for the row	String
rows[_.position (optional)	Position of the row	Integer
choices (required)	List of available choices for the user	List
choices[_.text (required)	Choice for user selection	String
choices[_.weight (required)	Weight value of the choice	Integer
choices[_.position (optional)	Position of the row	Integer
other (optional)	List of other answer options	Object
other[_.text	Text to display next to other option	String
other[_.num_chars	Set a character limit to the option	Integer
other[_.num_lines	Set a line limit to the option	Integer

MATRIX - RANKING

Matrix - Ranking

```
{
  "headings": [
    {
      "heading": "Rank these flavors in terms of your preference!"
    }
  ],
  "position": 1,
  "family": "matrix",
  "subtype": "ranking",
  "answers": {
    "rows": [
      {
        "text": "Vanilla"
      },
      {
        "text": "Chocolate"
      },
      {
        "text": "Strawberry"
      }
    ]
  }
}
```

1. Rank these flavors in terms of your preference!

The image shows a user interface for ranking three ice cream flavors. On the left, there are three vertical menus with up and down arrows. The top menu is for Vanilla, the middle for Chocolate, and the bottom for Strawberry. Each menu has three positions labeled 1, 2, and 3. The menu for Vanilla has '1' highlighted in blue.

Name	Description	Data Type
rows (required)	List of rows in the matrix	Array
rows[_.text] (required)	Text label for the row	String
rows[_.position] (optional)	Position of the row	Integer

MATRIX - MENU

Matrix - Menu

```
{
  "headings": [
    {
      "heading": "Which texture and taste do you prefer on your ice cream?"
    }
  ],
  "position": 1,
  "family": "matrix",
  "subtype": "menu",
  "answers": {
    "rows": [
      {
        "text": "Chocolate"
      },
      {
        "text": "Cookies and Cream"
      }
    ],
    "cols": [
      {
        "text": "Texture",
        "choices": [
          {
            "text": "Smooth",
            "visible": true,
            "position": 1
          },
          {
            "text": "Chunky",
            "visible": true,
            "position": 2
          }
        ]
      },
      {
        "text": "Flavor",
        "choices": [
          {
            "text": "Light",
            "visible": true,
            "position": 1
          },
          {
            "text": "Full",
            "visible": true,
            "position": 2
          }
        ]
      }
    ]
  }
}
```

```
}
```

1. Which texture and taste do you prefer on your ice cream?

	Texture	Flavor
Chocolate	<input type="text"/>	<input type="text"/>
Cookies and Cream	<input type="text"/> Smooth Chunky	<input type="text"/>

Name	Description	Data Type
rows (required)	List of rows in the matrix	Array
rows[_.text (required)	Text label for the row	String
rows[_.position (optional)	Position of the row	Integer
cols (required)	List of columns in the matrix	Array
cols[_.text (required)	Text label for column	String
cols[_.choices (required)	List of available choices for the user in dropdown menu	Array
cols[_.choices[_.text (required)	Choice for user selection	String
cols[_.choices[_.position (required)	Position of choice	Integer
other (optional)	List of other answer options	Object
other[_.text	Text to display next to other option	String
other[_.num_chars	Set a character limit to the option	Integer
other[_.num_lines	Set a line limit to the option	Integer

OPEN ENDED - SINGLE OR ESSAY

Open Ended - Single or Essay

```
{
  "headings": [
    {
      "heading": "What's your favorite thing about monkeys?"
    }
  ],
  "position": 1,
  "family": "open_ended",
  "subtype": "single"
}
```

1. What's your favorite thing about monkeys?

Only requires a **heading**.

OPEN ENDED - MULTI

Open Ended - Multi

```
{
  "headings": [
    {
      "heading": "What are your favorite monkey species?"
    }
  ],
  "position": 1,
  "family": "open-ended",
  "subtype": "multi",
  "answers": {
    "rows": [
      {
        "text": "Your favorite:"
      },
      {
        "text": "Second favorite:"
      }
    ]
  }
}
```

1. What are your favorite monkey species?

Your favorite:

Second favorite:

Name	Description	data Type
rows (required)	List of textboxes	Array
rows[_.text (required)	Text label for textbox	String
rows[_.position (optional)	Position of the current row	Integer

OPEN ENDED - NUMERICAL

Open Ended - Numerical

```
{
  "headings": [
    {
      "heading": "If you could split 10 pounds of ice cream, how would you split it?"
    }
  ],
  "position": 1,
  "family": "open-ended",
  "subtype": "numerical",
  "answers": {
    "rows": [
      {
        "text": "Pounds for family:"
      },
      {
        "text": "Pounds for friends:"
      },
      {
        "text": "Pounds for self:"
      }
    ]
  },
  "validation": {
    "text": "Please enter a valid integer.",
    "type": "integer",
    "sum": 10,
    "sum_text": "Your input must add up to 10 pounds!"
  }
}
```

1. If you could split 10 pounds of ice cream, how would you split it?

Pounds for family:

Pounds for friends:

Pounds for self:

Name	Description	Data Type
rows (required)	List of textboxes	Array
rows[_.text (required)	Text label for textbox	String
rows[_.position (optional)	Position of the current row	Integer

DEMOGRAPHIC - INTERNATIONAL/US

Demographic - international/us

```
{
  "headings": [
    {
      "heading": "What's your address?"
    }
  ],
  "position": 1,
  "visible": true,
  "family": "demographic",
  "subtype": "international",
  "answers": {
    "choices": [
      {
        "visible": true,
        "required": true,
        "type": "name",
        "text": "Your Name"
      },
      {
        "visible": true,
        "required": false,
        "type": "company"
      },
      {
        "visible": false,
        "required": false,
        "type": "address"
      },
      {
        "visible": false,
        "required": false,
        "type": "address2"
      },
      {
        "visible": true,
        "required": true,
        "type": "city"
      },
      {
        "visible": true,
        "required": true,
        "type": "state"
      },
      {
        "visible": true,
        "required": true,
        "type": "zip"
      },
      {
        "visible": true,
        "required": true,
        "type": "country"
      }
    ]
  }
}
```

```

},
{
  "visible": true,
  "required": true,
  "type": "email"
},
{
  "visible": true,
  "required": true,
  "type": "phone"
}
]
}
}

```

1. What's your address?

Name	<input type="text"/>
Company	<input type="text"/>
Address	<input type="text"/>
Address 2	<input type="text"/>
City/Town	<input type="text"/>
State/Province	<input type="text"/>
ZIP/Postal Code	<input type="text"/>
Country	<input type="text"/>
Email Address	<input type="text"/>
Phone Number	<input type="text"/>

This corresponds to the **Contact Information** question type in the SurveyMonkey UI.

Each `type` represented in the example object must be included, to disable one, specify `visible` as `False`

Name	Description	Data Type
choices	List of available demographic data	Array
choices[<u>_</u>].visible (required)	Visibility of demographic data type	Boolean
choices[<u>_</u> .required (required)	Whether demographic data type is required	Boolean
choices[<u>_</u> .type (required)	Type of demographic data	String
choices[<u>_</u> .text (optional)	Optional label of demographic data, will default to type	String

ADDRESS

Address

```

{
  "headings": [
    {
      "heading": "Address"
    }
  ],
  "position": 1,
  "visible": true,

```

```

"family": "demographic",
"subtype": "address",
"answers": [
  {
    "choices": [
      {
        "visible": true,
        "required": false,
        "type": "address1_label",
        "text": "Street"
      },
      {
        "visible": true,
        "required": false,
        "type": "address2_label",
        "text": "Street 2"
      },
      {
        "visible": true,
        "required": false,
        "type": "city_label",
        "text": "City"
      },
      {
        "visible": true,
        "required": false,
        "type": "state_label",
        "text": "State"
      },
      {
        "visible": true,
        "required": false,
        "type": "zip_label",
        "text": "Zip Code"
      },
      {
        "visible": true,
        "required": false,
        "type": "country_label",
        "text": "Country"
      }
    ]
  }
]
}

```

This corresponds to the **Address** question type in the SurveyMonkey UI.

Each `type` represented in the example object must be included, to disable one, specify `visible` as `False`

Name	Description	Data Type
choices	List of available address data	Array
choices[<u>_</u>].visible (required)	Visibility of address data type	Boolean
choices[<u>_</u> .required (required)	Whether address data type is required	Boolean
choices[<u>_</u> .type (required)	Type of address data	String
choices[<u>_</u> .text (required)	Optional label of address data, will default to type	String
choices[<u>_</u> .options	Optional address fields	Object
choices[<u>_</u> .options[<u>_</u> .placeholder_text (optional)]	Text that appears before the user types their answer in the fields	String
choices[<u>_</u> .options[<u>_</u> .default_country (optional)]	The default country code for the country_label type only	String

PHONE

Phone

```
{
  "headings": [
    {
      "heading": "Phone number"
    }
}
```

```

],
"position": 1,
"visible": true,
"family": "demographic",
"subtype": "phone",
"answers": {
  "choices": [
    {
      "visible": true,
      "type": "phone_number",
      "text": "Please provide your phone number"
    }
  ]
}
}

```

This corresponds to the **Phone** question type in the SurveyMonkey UI.

Each `type` represented in the example object must be included, to disable one, specify `visible` as `False`

Name	Description	Data Type
choices	List of available phone data	Array
choices[_.visible (required)	Visibility of phone data type	Boolean
choices[_.type (required)	Type of phone data	String
choices[_.text (required)	Optional label of phone data, will default to type	String
choices[_.options	Optional phone fields	Object
choices[_.options[_.placeholder_text (optional)	Text that appears before the user types their answer in the fields	String

EMAIL

Email

```

{
  "headings": [
    {
      "heading": "Email address"
    }
  ],
  "position": 1,
  "visible": true,
  "family": "demographic",
  "subtype": "email",
  "answers": {
    "choices": [
      {
        "visible": true,
        "type": "email_address",
        "text": "Please provide your email address"
      }
    ]
  }
}

```

This corresponds to the **Email** question type in the SurveyMonkey UI.

Each `type` represented in the example object must be included, to disable one, specify `visible` as `False`

Name	Description	Data Type
choices	List of available email data	Array
choices[_.visible (required)	Visibility of email data type	Boolean
choices[_.type (required)	Must be <code>email_address</code>	String
choices[_.text (required)	Optional label for email field	String

Name	Description	Data Type
choices[].options	Optional email fields	Object
choices[].options[].placeholder_text (optional)	Text that appears before the user types their answer in the fields	String

NAME

Name

```
{
  "headings": [
    {
      "heading": "Name"
    }
  ],
  "position": 1,
  "visible": true,
  "family": "demographic",
  "subtype": "name",
  "answers": {
    "choices": [
      {
        "visible": true,
        "required": true,
        "type": "first_name",
        "text": "Please provide your first name"
      },
      {
        "visible": true,
        "type": "last_name",
        "text": "Please provide your last name"
      },
      {
        "visible": true,
        "type": "middle_name",
        "text": "Please provide your middle name"
      },
      {
        "visible": true,
        "required": false,
        "type": "title",
        "text": "Please provide your title"
      },
      {
        "visible": true,
        "type": "suffix",
        "text": "Please provide any name suffixes"
      }
    ]
  }
}
```

This corresponds to the **Name** question type in the SurveyMonkey UI.

Each `type` represented in the example object must be included, to disable one, specify `visible` as `False`

Name	Description	Data Type
choices	List of available name data	Array
choices[].visible (required)	Visibility of name data type	Boolean
choices[].required (optional)	Determines if the field needs have an to be answer. By default all fields are not required.	Boolean
choices[].type (required)	Type of name data, may be <code>first_name</code> , <code>last_name</code> , <code>middle_name</code> , <code>title</code> , or <code>suffix</code> .	String
choices[].text (required)	Optional label of name data, will default to type	String
choices[].options	Optional name fields	Object

Name	Description	Data Type
choices[...].options[...].placeholder_text (optional)	Text that appears before the user types their answer in the fields	String

DATETIME

DateTime

```
{
  "headings": [
    {
      "heading": "When did you last eat ice cream?"
    }
  ],
  "position": 1,
  "visible": true,
  "family": "datetime",
  "subtype": "both",
  "validation": {
    "type": "date_us",
    "text": "Please enter a date."
  },
  "answers": {
    "rows": [
      {
        "text": "At:",
        "position": 1
      }
    ]
  }
}
```

This corresponds to the **DateTime** question type in the SurveyMonkey UI.

`validation` is **required** for DateTime questions and must include both `type` and `text`. `type` can be one of `date_us` or `date_intl`.

1. When did you last eat ice cream?

At:

MM	DD	YYYY	hh	mm	AM/PM
<input type="text"/>	<input type="button" value="-"/> <input type="button" value="▲"/>				

Name	Description	Data Type
rows[...].text (required)	Label for date/time input box	String
rows[...].position (optional)	Position of date/time input box	Integer

PRESENTATION

Presentation

```
{
  "headings": [
    {
      "heading": "This is a monkey",
      "image": {
        "url": "http://surveymonkey.com/monkey.jpg"
      }
    }
  ],
  "position": 1,
  "family": "presentation",
  "subtype": "image"
}
```

This is a monkey



If `image` is included, this corresponds to the **Image** question type in the SurveyMonkey UI. If not included, it corresponds to the **Text** question type.

Name	Description	Data Type
image (optional)	Image to present	Object
image[_.image_url] (required)	URL of image to present	String

FILE UPLOAD

File Upload

```
{
  "headings": [
    {
      "heading": "Upload your favorite monkey picture."
    }
  ],
  "position": 1,
  "family": "open_ended",
  "subtype": "single",
  "display_options": {
    "display_type": "file_upload"
  }
}
```

2. Upload your favorite monkey picture.

Choose File

No file chosen

A open ended single answer with the `display_options` object can become a File Upload question.

Name	Description	Data Type
display_options[_.display_type] (required)	Turns an open ended, single answer question into a file upload question. Always <code>file_upload</code>	String-ENUM

SLIDER

Slider

```
{
  "headings": [
    {
      "heading": "On a scale of 1 to 100, how much do you like monkeys?"
    }
  ],
  "position": 1,
  "family": "open_ended",
```

```

        "subtype": "single",
        "display_options": {
            "display_type": "slider"
        }
    }
}

```

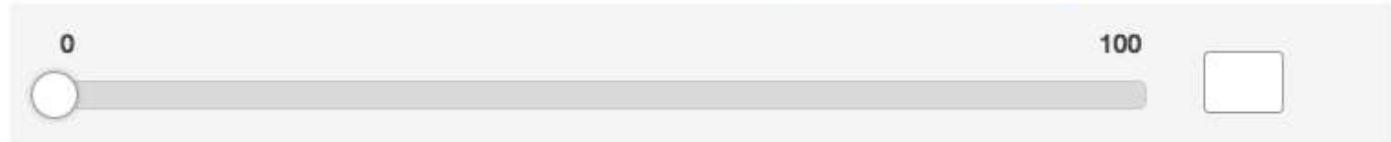
Slider Complex

```

{
    "headings": [
        {
            "heading": "On a scale of 1 to 100, how much do you like monkeys?"
        }
    ],
    "position": 1,
    "family": "open_ended",
    "subtype": "single",
    "display_options": {
        "right_label": "",
        "display_type": "slider",
        "custom_options": {
            "starting_position": 0,
            "step_size": 1
        },
        "left_label": ""
    },
    "validation": {
        "sum_text": "",
        "min": 45,
        "text": "Please enter a whole number between {0} and {1}.",
        "max": 10000,
        "type": "integer"
    }
}

```

3. On a scale of 1 to 100, how much do you like monkeys?



A horizontal slider with a scale from 0 to 100. The slider has a circular handle at position 0. At the right end of the slider is a small input field for a rating.

A open ended single answer with the `display_options` object can become a slider question.

Name	Description	Data Type
<code>display_options[].display_type</code> (required)	Turns an open ended single answer question into a slider question. Always <code>slider</code>	String-ENUM
<code>display_options[].right_label</code> (optional)	Label to place at the right end of the slider	String
<code>display_options[].left_label</code> (optional)	Label to place at the left end of the slider	String
<code>display_options[].custom_options[].starting_position</code> (optional)	Where the slider marker is positioned by default	Integer
<code>display_options[].custom_options[].step_size</code> (optional)	Step size the slider increments when moved	Integer

EMOJI (STAR RATING)

Emoji (Star Rating)

```

{
    "headings": [
        {
            "heading": "How many stars on the walk of fame should monkeys have?"
        }
    ],
    "position": 1,
    "family": "open_ended"
}

```

```

"family": "matrix",
"subtype": "rating",
"display_options": {
    "display_type": "emoji",
    "display_subtype": "star"
},
"forced_ranking": false,
"answers": {
"rows": [
{
    "visible": true,
    "text": "",
    "position": 1
}
],
"choices": [
{
    "weight": 1,
    "text": ""
},
{
    "weight": 2,
    "text": ""
},
{
    "weight": 3,
    "text": ""
},
{
    "weight": 4,
    "text": ""
},
{
    "weight": 5,
    "text": ""
}
]
}
}

```

1. How many stars on the walk of fame should monkeys have?



A rating question with the `display_options` object can become an emoji or Star Rating question.

Name	Description	Data Type
<code>display_options[].display_type</code> (required)	Turns and open ended single answer question into an emoji question. Always <code>emoj</code>	String-ENUM
<code>display_options[].display_subtype</code> (required)	Which emoji is displayed: <code>star</code> , <code>smiley</code> , <code>heart</code> , or <code>thumb</code>	String-ENUM

QUIZ QUESTIONS

Quiz Questions

```

{
"headings": [
{
    "heading": "Which monkeys are named after monks?"
}
],
"position": 1,
"family": "multiple_choice",
"subtype": "vertical",
"answers": {
"choices":[
{

```

```

        "text": "Capuchin",
        "quiz_options": {
          "score": 10
        }
      },
      {
        "text": "Mandrill",
        "quiz_options": {
          "score": 0
        }
      }
    ]
  },
  "quiz_options": {
    "scoring_enabled": true,
    "feedback": {
      "correct_text": "Nice! That's correct",
      "incorrect_text": "Nope, that's wrong"
    }
  }
}

```

Quiz scores can be added to any question type with a `choices` field.

Name	Description	Data Type
<code>choices[_.quiz_options</code>	The configuration object for quiz options	Object
<code>choices[_.quiz_options.score</code>	How many points the respondent gets when answering this choice	Integer

QUESTION BANK QUESTIONS

Question Bank Questions

```
{
  "question_bank": {
    "question_bank_question_id": "669",
    "modifier_options": {
      "36628": null
    }
  }
}
```

Question bank questions can be created with just the question id. Modifier options are either null, or a string, depending on whether the modifier option has user input (e.g. “this company” would be null, “specify a company” would be String).

These question ids can be retrieved via the `/v3/question_bank/questions` endpoint.

Name	Description	Data Type
<code>question_bank</code>	The configuration object for question bank options	Object
<code>question_bank[_.question_bank_question_id</code>	The id of the question bank question to be created	String
<code>question_bank[_.modifier_options</code>	Optional or required depending on the question, used to specify choices that would be in the dropdown menu in the ‘create survey’ page, not applicable for custom question bank. This object is a dictionary of String:String, or String:Null	Object
<code>question_bank[_.is_custom</code>	Whether or not the question being created comes from a Team’s custom question bank, as opposed to the main SurveyMonkey one, defaults to <code>false</code>	Boolean

Question Bank

The following endpoint lets you get a list of questions that exist in the question bank. You can then use these to create a question in a survey (See formatting question types).

GET /question_bank/questions

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/question_bank/questions", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /question_bank/questions

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns a list of question bank questions available to the user

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
per_page	string	false	Number of resources to return per page. Defaults to 50.
locale	string-Enum	false	Return questions for this locale, (see /v3/survey_languages), defaults to en_US
search	string	false	Return only questions containing (or related to) this text
custom	boolean	false	Whether to look through the regular Question Bank, or a Team's Custom Question Bank, defaults to false

Example Response for GET /question_bank/questions

```
{
  "data": [
    {
      "modifiers": [
        {
          "modifier_type": "open_ended",
          "options": [
            {
              "modifier_options_id": "34658",
              "value": "this neighborhood"
            },
            {
              "modifier_options_id": "34659",
              "value": "Specify neighborhood"
            }
          ],
          "modifier_id": "13196"
        }
      ],
      "text": "Overall, how clean is ()?",
      "locales": [
        "en_AU",
        "en_CA",
        "en_US"
      ],
      "question_id": "26073"
    }
  ]
}
```

```

        ],
        "per_page": 1,
        "page": 1,
        "total": 2598,
        "links": {
          "self": "https://api.surveymonkey.com/v3/question_bank/questions?page=1&per_page=1",
          "next": "https://api.surveymonkey.com/v3/question_bank/questions?page=2&per_page=1",
          "last": "https://api.surveymonkey.com/v3/question_bank/questions?page=2598&per_page=1"
        }
      }
    }
  }
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
per_page	integer	
total	integer	
data	[object]	
modifiers	[object]	Modifiers that the question allows. e.g. ("this company", "this brand, "this product") (empty for Custom QB)
modifier_type	string	
options	[object]	
modifier_options_id	string	
value	string	
modifier_id	string	
locales	[string]	A list of locales that the question supports (empty for Custom QB)
text	string	The text in the question
question_id	string	id of the question, to be used in creating it
links	object	
next	string	
last	string	
self	string	
page	integer	

Survey Folders

Folders are used to organize surveys. The following endpoints let you retrieve and create folders. Add surveys to a certain folder with a POST, PUT, or PATCH to /surveys including it's `folder_id`. You can also filter surveys by folders using `folder_id` as a query parameter on GET /surveys.

GET /survey_folders

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

```

```

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/survey_folders", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /survey_folders`

ENDPOINT CURRENTLY NOT AVAILABLE FOR CANADIAN USERS.

AVAILABLE METHODS

- `HEAD`: Checks if resource is available
- `OPTIONS`: Returns available methods and options
- `POST`: Creates
- `GET`: Returns available folders

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
per_page	string	false	Number of resources to return per page

Example Response for `GET /survey_folders`

```
{
  "data": [
    {
      "title": "SomeApp Polls",
      "id": "1845492",
      "num_surveys": 19,
      "href": "https://api.surveymonkey.com/v3/survey_folders/1845492"
    },
    {
      "title": "Teams Polls",
      "id": "1846967",
      "num_surveys": 29,
      "href": "https://api.surveymonkey.com/v3/survey_folders/1846967"
    }
  ],
  "per_page": 50,
  "page": 1,
  "total": 2,
  "links": {
    "self": "https://api.surveymonkey.com/v3/survey_folders?page=1&per_page=50"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
per_page	integer	Number of resources per page
total	integer	
data	[object]	
num_surveys	integer	Number of surveys in the folder
id	string	ID of the folder
href	string	Link to the folder

Name	Type	Description
title	string	Name of the folder
links	object	
self	string	
page	integer	

POST /survey_folders

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"title\":\"NPS Surveys\"}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/survey_folders", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

POST /survey_folders

ENDPOINT CURRENTLY NOT AVAILABLE FOR CANADIAN USERS.

Body parameter

```
{
    "title": "NPS Surveys"
}
```

REQUEST BODY FOR POST /SURVEY_FOLDERS

Name	Type	Required	Description
title	string	false	Title of the survey folder, used to house a collection of surveys

Example Response for POST /survey_folders

```
{
    "title": "My Folder",
    "id": "1",
    "num_surveys": 0,
    "href": "https://api.surveymonkey.com/v3/survey_folders/1"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
title	string	Title of the folder

Name	Type	Description
href	string	Link to the folder
id	string	ID of the folder
num_surveys	integer	Number of surveys in the folder

Translations for Multilingual Surveys

These endpoints let you view and create translations for a multilingual survey. Use GET /survey_languages to see all available languages and their codes, a GET to /surveys/{survey_id}/languages/{language_code} to get all strings for translation, and a POST to /surveys/{survey_id}/languages/{language_code} create a translation.

GET /surveys/{survey_id}/languages

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/languages", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /surveys/{survey_id}/languages](#)

AVAILABLE METHODS

[HEAD](#): Checks if resource is available

[OPTIONS](#): Returns available methods and options

[GET](#): Returns all existing translations

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
per_page	string	false	Number of resources to return per page

Example Response for /surveys/{survey_id}/languages

```
{
  "per_page": 50,
  "total": 1,
  "data": [
    {
      "stale_string_count": 0,
      "translated_string_count": 4,
      "href": "https://api.surveymonkey.com/v3/surveys/1234/languages/fr",
      "inherited_string_count": 0,
      "string_count": 8,
      "name": "French",
      "native_name": "français",
```

```

        "enabled": true,
        "id": "fr"
    },
],
"page": 1,
"links": {
    "self": "https://api.surveymonkey.com/v3/surveys/1234/languages/?page=1&per_page=50"
}
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
per_page	integer	
page	integer	
total	integer	
data	[object]	
stale_string_count	string	Count of strings that have been updated in the source language since the translation was last updated
id	string	Language code for the translation
translated_string_count	integer	Count strings that have <code>translations</code>
href	string	Resource URL
inherited_string_count	integer	Count of strings that have empty <code>translations</code> but have a parent translation with an existing <code>translation</code> they can fall back to
string_count	integer	Count of strings in the array
name	string	The name of the language in English
native_name	string	The name of the language in the language itself
enabled	boolean	If True, translation is available for respondents to select
links	object	
self	string	

GET /surveys/{survey_id}/languages/{language_code}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/languages/", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /surveys/{survey_id}/languages/{language_code}`

AVAILABLE METHODS

- HEAD**: Checks if resource is available
- OPTIONS**: Returns available methods and options
- GET**: Returns existing translations or the translation structure
- POST**: Creates a translation
- PATCH**: Updates a translation
- DELETE**: Deletes a translation

Example Response for GET /surveys/{survey_id}/languages/{language_code}

```
{  
  "href": "https://api.surveymonkey.com/v3/surveys/1234/languages/fr",  
  "translations": [  
    {  
      "default": "Next",  
      "translation": "Suiv.",  
      "context": "survey_next",  
      "resource_id": "1234"  
    },  
    {  
      "default": "Prev",  
      "translation": "Préc.",  
      "context": "survey_prev",  
      "resource_id": "1234"  
    },  
    {  
      "default": "Done",  
      "translation": "Terminé",  
      "context": "survey_done",  
      "resource_id": "1234"  
    },  
    {  
      "default": "Exit",  
      "translation": "Quitter",  
      "context": "survey_exit",  
      "resource_id": "1234"  
    },  
    {  
      "default": "Thank you for completing our survey!",  
      "translation": "Merci d'avoir répondu à notre sondage!",  
      "context": "collector_disqualification",  
      "resource_id": "1234"  
    },  
    {  
      "default": "Are you multilingual?",  
      "translation": "",  
      "context": "question_heading",  
      "resource_id": "5678"  
    },  
    {  
      "default": "Yes",  
      "translation": "Oui",  
      "context": "question_option_text",  
      "resource_id": "8912"  
    },  
    {  
      "default": "No",  
      "translation": "Non",  
      "context": "question_option_text",  
      "resource_id": "3456"  
    },  
    {  
      "default": "My Survey",  
      "translation": "Mon Sondage",  
      "context": "survey_title_html",  
      "resource_id": "1234"  
    }  
  "enabled": true,  
  "id": "fr"  
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
href	string	
enabled	boolean	
id	string	
translations	[object]	
default	string	Untranslated text
translation	string	Translated text shown when a respondent chooses the survey in this language. Translate a survey by filling in all empty translations,
resource_id	string	Survey, question or answer option id related to the string
context	integer	Describes the string's role in the survey

POST /surveys/{survey_id}/languages/{language_code}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"enabled\":true,\"translations\": [{\"default\":\"Next\",\"translation\":\"Suiv.\",\"context\":\"survey_next\",\"resource_id\":\"1234\"}, {\"default\":\"Prev\",\"translation\":\"Préc.\",\"context\":\"survey_prev\",\"resource_id\":\"1234\"}, {\"default\":\"Done\",\"translation\":\"Terminé\",\"context\":\"survey_done\",\"resource_id\":\"1234\"}], {\"default\":\"Exit\",\"translation\":\"Quitter\",\"context\":\"survey_exit\",\"resource_id\":\"1234\"}, {\"default\":\"Thank you for completing our survey!\",\"translation\":\"Merci d'avoir répondu à notre sondage!\",\"context\":\"collector_disqualification\",\"resource_id\":\"1234\"}, {\"default\":\"Are you multilingual?\",\"translation\":\"Êtes-vous multilingue?\",\"context\":\"question_heading\",\"resource_id\":\"5678\"}, {\"default\":\"Yes\",\"translation\":\"Oui\",\"context\":\"question_option_text\",\"resource_id\":\"8912\"}, {\"default\":\"No\",\"translation\":\"Non\",\"context\":\"question_option_text\",\"resource_id\":\"3456\"}, {\"default\":\"My Survey\",\"translation\":\"Mon Sondage\",\"context\":\"survey_title_html\",\"resource_id\":\"1234\"}]}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/surveys/1234/languages/", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[POST /surveys/{survey_id}/languages/{language_code}](#)

Body parameter

```
{
  "enabled": true,
  "translations": [
    {
      "default": "Next",
      "translation": "Suiv.",
      "context": "survey_next",
      "resource_id": "1234"
    },
    {
      "default": "Prev",
      "translation": "Préc.",
      "context": "survey_prev",
      "resource_id": "1234"
    },
    {
      "default": "Done",
      "translation": "Terminé",
```

```

    "context": "survey_done",
    "resource_id": "1234"
},
{
    "default": "Exit",
    "translation": "Quitter",
    "context": "survey_exit",
    "resource_id": "1234"
},
{
    "default": "Thank you for completing our survey!",
    "translation": "Merci d'avoir répondu à notre sondage!",
    "context": "collector_disqualification",
    "resource_id": "1234"
},
{
    "default": "Are you multilingual?",
    "translation": "Êtes-vous multilingue?",
    "context": "question_heading",
    "resource_id": "5678"
},
{
    "default": "Yes",
    "translation": "Oui",
    "context": "question_option_text",
    "resource_id": "8912"
},
{
    "default": "No",
    "translation": "Non",
    "context": "question_option_text",
    "resource_id": "3456"
},
{
    "default": "My Survey",
    "translation": "Mon Sondage",
    "context": "survey_title_html",
    "resource_id": "1234"
}
]
}

```

REQUEST BODY FOR POST /SURVEYS/{SURVEY_ID}/LANGUAGES/{LANGUAGE_CODE}

Name	Type	Required	Description
enabled	boolean	false	If True, translation is available for respondents to select
translations	array	true	List of translation objects. Each object represents a string in the survey. At least one is required for POST and PATCH. See below for structure.

200 Response

```
{}
```

PATCH /surveys/{survey_id}/languages/{language_code}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"enabled\":true,\"translations\":[{\"\\\"default\\\":\\\"Next\\\",\\\"translation\\\":\\\"Suiv.\\\",\\\"context\\\":\\\"survey_next\\\",\\\"resource_id\\\":\\\"1234\\\"},{\\\"default\\\":\\\"Prev\\\",\\\"translation\\\":\\\"Préc.\\\",\\\"context\\\":\\\"survey_prev\\\",\\\"resource_id\\\":\\\"1234\\\"},{\\\"default\\\":\\\"Done\\\",\\\"translation\\\":\\\"Terminé\\\",\\\"context\\\":\\\"survey_done\\\",\\\"resource_id\\\":\\\"1234\\\"},{\\\"default\\\":\\\"Exit\\\",\\\"translation\\\":\\\"Quitter\\\",\\\"context\\\":\\\"survey_exit\\\",\\\"resource_id\\\":\\\"1234\\\"},{\\\"default\\\":\\\"Thank you for completing our survey!\\\",\\\"translation\\\":\\\"Merci d'avoir répondu à notre sondage!\\\",\\\"context\\\":\\\"collector_disqualification\\\",\\\"resource_id\\\":\\\"1234\\\"},{\\\"default\\\":\\\"Are you multilingual?\\\",\\\"translation\\\":\\\"Êtes-vous multilingue?\\\",\\\"context\\\":\\\"question_heading\\\",\\\"resource_id\\\":\\\"5678\\\"},{\\\"default\\\":\\\"Yes\\\",\\\"translation\\\":\\\"Oui\\\",\\\"context\\\":\\\"question_option_text\\\",\\\"resource_id\\\":\\\"8912\\\"},{\\\"default\\\":\\\"No\\\",\\\"translation\\\":\\\"Non\\\",\\\"context\\\":\\\"question_option_text\\\",\\\"resource_id\\\":\\\"3456\\\"},{\\\"default\\\":\\\"My Survey\\\",\\\"translation\\\":\\\"Mon Sondage\\\",\\\"context\\\":\\\"survey_title_html\\\",\\\"resource_id\\\":\\\"1234\\\"}]}"

```

```

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/v3/surveys/1234/languages/", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

PATCH /surveys/{survey_id}/languages/{language_code}

Body parameter

```
{
    "enabled": true,
    "translations": [
        {
            "default": "Next",
            "translation": "Suiv.",
            "context": "survey_next",
            "resource_id": "1234"
        },
        {
            "default": "Prev",
            "translation": "Préc.",
            "context": "survey_prev",
            "resource_id": "1234"
        },
        {
            "default": "Done",
            "translation": "Terminé",
            "context": "survey_done",
            "resource_id": "1234"
        },
        {
            "default": "Exit",
            "translation": "Quitter",
            "context": "survey_exit",
            "resource_id": "1234"
        },
        {
            "default": "Thank you for completing our survey!",
            "translation": "Merci d'avoir répondu à notre sondage!",
            "context": "collector_disqualification",
            "resource_id": "1234"
        },
        {
            "default": "Are you multilingual?",
            "translation": "Êtes-vous multilingue?",
            "context": "question_heading",
            "resource_id": "5678"
        },
        {
            "default": "Yes",
            "translation": "Oui",
            "context": "question_option_text",
            "resource_id": "8912"
        },
        {
            "default": "No",
            "translation": "Non",
            "context": "question_option_text",
            "resource_id": "3456"
        },
        {
            "default": "My Survey",
            "translation": "Mon Sondage",
            "context": "survey_title_html",
            "resource_id": "1234"
        }
    ]
}
```

REQUEST BODY FOR PATCH /SURVEYS/{SURVEY_ID}/LANGUAGES/{LANGUAGE_CODE}

Name	Type	Required	Description
enabled	boolean	false	If True, translation is available for respondents to select
translations	array	true	List of translation objects. Each object represents a string in the survey. At least one is required for POST and PATCH. See below for structure.

200 Response

```
{}
```

Contacts and Contact Lists

If your application is using email collectors or sms collectors to collect survey responses, these endpoints let you create contacts and contact lists to send survey invite messages to by passing a `contact_id` as an argument to `POST /collectors/{id}/messages/{id}/recipients`. NOTE: Contacts can also be created if they are passed directly to `/collectors/{id}/messages/{id}/recipients`.

GET /contact_lists

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/contact_lists", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`GET /contact_lists`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns all contact lists. Public App users need access to the **View Contacts** scope

`POST`: Creates a contact list, contacts can be sent survey invite messages using an email invite or sms collector, see Collectors and Invite Messages. Public App users need access to the **Create/Modify Contacts** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
per_page	string	false	Number of resources to return per page

Example Response for GET /contact_lists

```
{
  "data": [
    {
      "id": "1234",
      "name": "John Doe"
    }
  ]
}
```

```

        "name": "contact list name",
        "href": "https://api.surveymonkey.com/v3/contact_lists/1234"
    }
],
"per_page": 50,
"page": 1,
"total": 1,
"links": {
    "self": "https://api.surveymonkey.com/v3/contact_lists?page=1&per_page=50"
}
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	
data	object	
id	string	Contact list id
name	string	Contact list name
href	string	Resource API URL

POST /contact_lists

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"name\":\"My Contact list\"}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/contact_lists", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

[POST /contact_lists](#)

Used to create a new contact list

Body parameter

```
{
    "name": "My Contact list"
}
```

REQUEST BODY FOR POST /CONTACT_LISTS

Name	Type	Required	Description
name	string	false	Contact list name (default="New List")

Example Response for POST /contact_lists

```
{  
  "data": [  
    {  
      "id": "1234",  
      "name": "contact list name",  
      "href": "https://api.surveymonkey.com/v3/contact_lists/1234"  
    }  
,  
    {"per_page": 50,  
     "page": 1,  
     "total": 1,  
     "links": {  
       "self": "https://api.surveymonkey.com/v3/contact_lists?page=1&per_page=50"  
     }  
  }  
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	
data	object	
id	string	Contact list id
name	string	Contact list name
href	string	Resource API URL

GET /contact_lists/{contact_list_id}

Code samples

```
import http.client  
  
conn = http.client.HTTPSConnection("api.surveymonkey.com")  
  
headers = {  
  'Accept': "application/json",  
  'Authorization': "Bearer {access-token}"  
}  
  
conn.request("GET", "/v3/contact_lists/1234", headers=headers)  
  
res = conn.getresponse()  
data = res.read()  
  
print(data.decode("utf-8"))
```

`GET /contact_lists/{contact_list_id}`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns a contact list. Public App users need access to the **View Contacts** scope

`PATCH`: Modifies a contact list (updates any fields accepted as arguments to `POST` /contact_lists). Public App users need access to the **Create/Modify Contacts** scope

`PUT`: Replaces a contact list `POST` (same arguments and requirements as `POST` /contact_lists). Public App users need access to the **Create/Modify Contacts** scope

`DELETE`: Deletes a contact list. Public App users need access to the **Create/Modify Contacts** scope

Example Response for GET, PATCH, or PUT /contact_lists/{id}

```
{  
  "name": "My Contact List",  
  "id": "1234",  
  "href": "https://api.surveymonkey.com/v3/contact_lists/1234"  
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Contact list id
name	string	Contact list name
href	string	Resource API URL

PATCH /contact_lists/{contact_list_id}

Code samples

```
import http.client  
  
conn = http.client.HTTPSConnection("api.surveymonkey.com")  
  
payload = "{\"name\":\"string\"}"  
  
headers = {  
    'Content-Type': "application/json",  
    'Accept': "application/json",  
    'Authorization': "Bearer {access-token}"  
}  
  
conn.request("PATCH", "/v3/contact_lists/1234", payload, headers)  
  
res = conn.getresponse()  
data = res.read()  
  
print(data.decode("utf-8"))
```

`PATCH /contact_lists/{contact_list_id}`

`PATCH`: Modifies a contact list name

Example Response for PATCH /contact_lists/{id}

```
{  
  "name": "My Contact List",  
  "id": "1234",  
  "href": "https://api.surveymonkey.com/v3/contact_lists/1234"  
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Contact list id
name	string	Contact list name
href	string	Resource API URL

PUT /contact_lists/{contact_list_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"name\":\"string\"}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PUT", "/v3/contact_lists/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[PUT /contact_lists/{contact_list_id}](#)

PUT: Replaces a contact list [POST](#) (same arguments and requirements as [POST /contact_lists](#)). Public App users need access to the **Create/Modify Contacts** scope

Example Response for PUT /contact_lists/{contact_list_id}

```
{
  "name": "My Contact List",
  "id": "1234",
  "href": "https://api.surveymonkey.com/v3/contact_lists/1234"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Contact list id
name	string	Contact list name
href	string	Resource API URL

POST /contact_lists/{contact_list_id}/copy

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/contact_lists/1234/copy", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

POST /contact_lists/{contact_list_id}/copy

AVAILABLE METHODS

POST: Creates a copy of an existing contact list. Requires **Create/Modify Contacts** scope

Example Response for POST /contact_lists/{contact_list_id}/copy

```
{
  "name": "My Contact List",
  "id": "1234",
  "href": "https://api.surveymonkey.com/v3/contact_lists/1234"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Contact list id
name	string	Contact list name
href	string	Resource API URL

POST /contact_lists/{contact_list_id}/merge

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"list_id\":\"1234\"}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/contact_lists/1234/merge", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

POST /contact_lists/{contact_list_id}/merge

AVAILABLE METHODS

POST: Copies contacts in the list specified in the request body and adds to the list specified in the resource URL. Public App users need access to the **Create/Modify Contacts** scope

Body parameter

```
{  
    "list_id": "1234"  
}
```

REQUEST BODY FOR POST /CONTACT_LISTS/{CONTACT_LIST_ID}/MERGE

Name	Type	Required	Description
list_id	string	true	A single contact list ID to merged with the contact list id pass in the url

Example Response for POST /contact_lists/{contact_list_id}/merge

```
{  
    "name": "My Contact List",  
    "id": "1234",  
    "href": "https://api.surveymonkey.com/v3/contact_lists/1234"  
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Contact list id
name	string	Contact list name
href	string	Resource API URL

GET /contact_lists/{contact_list_id}/contacts

Code samples

```
import http.client  
  
conn = http.client.HTTPSConnection("api.surveymonkey.com")  
  
headers = {  
    'Accept': "application/json",  
    'Authorization': "Bearer {access-token}"  
}  
  
conn.request("GET", "/v3/contact_lists/1234/contacts", headers=headers)  
  
res = conn.getresponse()  
data = res.read()  
  
print(data.decode("utf-8"))
```

GET /contact_lists/{contact_list_id}/contacts

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns all contacts in a contact list. Public App users need access to the **View Contacts** scope

POST: Creates a new contact and adds them to a contact list, contacts can be sent survey invite messages using an email invite or sms collector, see Collectors and Invite Messages. Public App users need access to the **Create/Modify Contacts** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
per_page	string	false	Number of resources to return per page
status	string	false	Filter by status (default=active): <code>active</code> , <code>optout</code> or <code>bounced</code>
sort_by	string	false	Field used to sort returned contact list (default=email): <code>email</code> , <code>first_name</code> , <code>last_name</code> , <code>1</code> , ..., <code>6</code>
sort_order	string	false	Sort order with the following options ['ASC','asc', 'DESC','desc']
search_by	string	false	The search_bu parameter needs to be used with <code>search</code> . Field used to search the selected contact_list. user can seach by email, first_name, last_name, or phone_number
search	string	false	Query used to search, the search parameter needs to be used with <code>search_by</code>

ENUMERATED VALUES

Parameter	Value
status	active
status	optout
status	bounced
sort_by	email
sort_by	first_name
sort_by	last_name
sort_order	ASC
sort_order	asc
sort_order	DESC
sort_order	desc
search_by	email
search_by	first_name
search_by	last_name
search_by	phone_number

Response Example for GET and POST /contact_lists/{id}/contacts

```
{
  "id": "1234",
  "first_name": "John",
  "last_name": "Doe",
  "email": "test@surveymonkey.com",
  "phone_number": "+1 202 555 0156",
  "custom_fields": {
    "1": "Mr",
    "2": "Company",
    "3": "Address",
    "4": "City",
    "5": "Country",
    "6": "Phone Number"
  },
  "href": "https://api.surveymonkey.com/v3/contacts/1234"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Contact id
first_name	string	Contact first name
last_name	string	Contact last name
email	string	Contact email address
phone_number	string	Contact phone number
custom_fields	object	Contact custom fields
href	string	Resource API URL

POST /contact_lists/{contact_list_id}/contacts

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/contact_lists/1234/contacts", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[POST /contact_lists/{contact_list_id}/contacts](#)

Body parameter

```
{}
```

GET /contact_lists/{contact_list_id}/contacts/bulk

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/contact_lists/1234/contacts/bulk", headers=headers)

res = conn.getresponse()
data = res.read()
```

```
print(data.decode("utf-8"))
```

```
GET /contact_lists/{contact_list_id}/contacts/bulk
```

AVAILABLE METHODS

GET: Returns a list of all contacts in the list with all available fields. Public App users need access to the **View Contacts** scope

POST: Creates multiple contacts and adds them to a list, contacts can be sent survey invite messages using an email invite or sms collector, see Collectors and Invite Messages. Public App users need access to the **Create/Modify Contacts** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	integer	false	Page number. Defaults to 1
per_page	integer	false	Number of contacts to return per page
sort_by	string	false	sort_by either email, first_name, or last_name
status	string	false	Filter by status (default=active): active, optout, bounced
sort_order	string	false	Sort order e.g. ['ASC', 'DESC']
search_by	string	false	Field used to search returned contact list: email, first_name, last_name, 1, ..., 50
search	string	false	Query used to search

ENUMERATED VALUES

Parameter	Value
sort_by	email
sort_by	first_name
sort_by	last_name

Example Response for GET and POST /contacts_lists/{contact_list_id}/contacts/bulk

```
{
  "data": [
    {
      "id": "1",
      "first_name": "",
      "last_name": "",
      "email": "test@surveymonkey.com",
      "phone_number": "+1 202 555 0156",
      "href": "https://api.surveymonkey.com/v3/contacts/1"
    }
  ],
  "per_page": 50,
  "page": 1,
  "total": 1,
  "links": {
    "self": "https://api.surveymonkey.com/v3/contacts/bulk?page=1&per_page=50"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[object]	
id	string	Contact id
first_name	string	Contact first name

Name	Type	Description
last_name	string	Contact last name
email	string	Contact email address
phone_number	string	Contact phone number
href	string	Resource API URL
custom_fields	object	Custom fields
status	boolean	
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	

POST /contact_lists/{contact_list_id}/contacts/bulk

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"update_existing\":true,\"contacts\":[]}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/contact_lists/1234/contacts/bulk", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`POST /contact_lists/{contact_list_id}/contacts/bulk`

Body parameter

```
{
  "update_existing": true,
  "contacts": [
    {}
  ]
}
```

REQUEST BODY FOR POST /CONTACT_LISTS/{ID}/CONTACTS/BULK

Name	Type	Required	Description
update_existing	boolean	false	Whether or not to update existing contacts
contacts	[anyOf]	true	Contacts to add

Example Response for POST /contact_lists/{id}/contacts/bulk

```
{
  "succeeded": [
    {
      "first_name": "John",
      "last_name": "Doe",
      "id": "1234",
      "email": "test@surveymonkey.com",
      "phone_number": "+1 202 555 0156",
      "custom_fields": {
        "1": "Mr",
        "2": "Company",
        "3": "Address",
        "4": "City",
        "5": "Country",
        "6": "Phone Number"
      }
    }
  ],
  "invalids": [
  ],
  "existing": [
  ]
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
succeeded	array	List of successfully added contact objects
succeeded.id	string	Id for the contact
succeeded.email	string	Email address for the contact
succeeded.phone_number	string	Phone number for the contact
succeeded.first_name	string	First name for the contact
succeeded.last_name	string	Last name for the contact
invalids	array	List of invalid contact email addresses that were provided
existing	array	List of contact email addresses that have already been added

GET /contacts

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
  'Accept': "application/json",
  'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/contacts", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /contacts

AVAILABLE METHODS

- HEAD**: Checks if resource is available
- OPTIONS**: Returns available methods and options
- GET**: Returns a list of all contacts. Public App users need access to the **View Contacts** scope
- POST**: Creates a new contact, contacts can be sent survey invite messages using an email invite or sms collector, see Collectors and Invite Messages. Public App users need access to the **Create/Modify Contacts** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
per_page	string	false	Number of resources to return per page
status	string	false	Filter by status (default=active): <code>active</code> , <code>optout</code> or <code>bounced</code>
sort_by	string	false	Field used to sort returned contact list (default=email): <code>email</code> , <code>first_name</code> , <code>last_name</code> , <code>1</code> , ..., <code>6</code>
sort_order	string	false	Sort order with the following options ['ASC','asc', 'DESC','desc']
search_by	string	false	The search_bu parameter needs to be used with <code>search</code> . Field used to search the selected contact_list. user can search by email, first_name, last_name, or phone_number
search	string	false	Query used to search, the search parameter needs to be used with <code>search_by</code>

ENUMERATED VALUES

Parameter	Value
status	active
status	optout
status	bounced
sort_by	email
sort_by	first_name
sort_by	last_name
sort_order	ASC
sort_order	asc
sort_order	DESC
sort_order	desc
search_by	email
search_by	first_name
search_by	last_name
search_by	phone_number

Example Response for GET /contacts

```
{
  "data": [
    {
      "id": "1",
      "first_name": "",
      "last_name": "",
      "email": "test@surveymonkey.com",
      "phone_number": "+1 202 555 0156",
      "href": "https://api.surveymonkey.com/v3/contacts/1"
    }
  ],
  "per_page": 50,
  "total": 1
}
```

```

"page": 1,
"total": 1,
"links": {
  "self": "https://api.surveymonkey.com/v3/contacts/bulk?page=1&per_page=50"
}
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[object]	
id	string	Contact id
first_name	string	Contact first name
last_name	string	Contact last name
email	string	Contact email address
phone_number	string	Contact phone number
href	string	Resource API URL
custom_fields	object	Custom fields
status	boolean	
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	

POST /contacts

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"first_name\": \"Quinn\", \"last_name\": \"Doe\", \"email\": \"test@surveymonkey.com\", \"phone_number\": \"+1 202 555 0156\"}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/contacts", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

POST /contacts

Body parameter

```
{
  "first_name": "Quinn",
  "last_name": "Doe",
  "email": "test@surveymonkey.com",
  "phone_number": "+1 202 555 0156"
}
```

Example Response for POST /contacts

```
{
  "data": [
    {
      "id": "1",
      "first_name": "",
      "last_name": "",
      "email": "test@surveymonkey.com",
      "phone_number": "+1 202 555 0156",
      "href": "https://api.surveymonkey.com/v3/contacts/1"
    }
  ],
  "per_page": 50,
  "page": 1,
  "total": 1,
  "links": {
    "self": "https://api.surveymonkey.com/v3/contacts/bulk?page=1&per_page=50"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[object]	
id	string	Contact id
first_name	string	Contact first name
last_name	string	Contact last name
email	string	Contact email address
phone_number	string	Contact phone number
href	string	Resource API URL
custom_fields	object	Custom fields
status	boolean	
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	

GET /contacts/bulk

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/contacts/bulk", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /contacts/bulk`

AVAILABLE METHODS

`GET`: Returns a list of all contacts with all available fields. Public App users need access to the **View Contacts** scope

`POST`: Creates multiple contacts, contacts can be sent survey invite messages using an email invite or sms collector, see Collectors and Invite Messages. Public App users need access to the **Create/Modify Contacts** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	integer	false	Page number. Defaults to 1
per_page	integer	false	Number of contacts to return per page
sort_by	string	false	Number of contacts to return per page
status	string	false	Filter by status (default=active): <code>active</code> , <code>optout</code> , <code>bounced</code>
sort_order	string	false	Sort order e.g. ['ASC', 'DESC']
search_by	string	false	Field used to search returned contact list: : <code>email</code> , <code>first_name</code> , <code>last_name</code> , <code>1</code> , ..., <code>50</code>
search	string	false	Query used to search

ENUMERATED VALUES

Parameter	Value
sort_by	email
sort_by	first_name
sort_by	last_name

Example Response for `GET /contacts/bulk`

```
{
  "data": [
    {
      "id": "1",
      "first_name": "",
      "last_name": "",
      "email": "test@surveymonkey.com",
      "phone_number": "+1 202 555 0156",
      "href": "https://api.surveymonkey.com/v3/contacts/1"
    }
  ],
  "per_page": 50,
  "page": 1,
  "total": 1,
  "links": {
    "self": "https://api.surveymonkey.com/v3/contacts/bulk?page=1&per_page=50"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[object]	
id	string	Contact id
first_name	string	Contact first name
last_name	string	Contact last name
email	string	Contact email address
phone_number	string	Contact phone number
href	string	Resource API URL
custom_fields	object	Custom fields
status	boolean	
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	

POST /contacts/bulk

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"update_existing\":true,\"contacts\":[{}]}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/contacts/bulk", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

POST /contacts/bulk

Body parameter

```
{
  "update_existing": true,
  "contacts": [
    {}
  ]
}
```

REQUEST BODY FOR POST /CONTACTS/BULK

Name	Type	Required	Description
update_existing	boolean	false	Whether or not to update existing contacts
contacts	[anyOf]	true	Contacts to add

Example Response for POST /contacts/bulk

```
{  
  "data": [  
    {  
      "id": "1",  
      "first_name": "",  
      "last_name": "",  
      "email": "test@surveymonkey.com",  
      "phone_number": "+1 202 555 0156",  
      "href": "https://api.surveymonkey.com/v3/contacts/1"  
    }  
,  
    {"per_page": 50,  
     "page": 1,  
     "total": 1,  
     "links": {  
       "self": "https://api.surveymonkey.com/v3/contacts/bulk?page=1&per_page=50"  
     }  
  }  
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[object]	
id	string	Contact id
first_name	string	Contact first name
last_name	string	Contact last name
email	string	Contact email address
phone_number	string	Contact phone number
href	string	Resource API URL
custom_fields	object	Custom fields
status	boolean	
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	

GET /contacts/{contact_id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/contacts/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /contacts/{contact_id}`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns a contact. Public App users need access to the **View Contacts** scope

`PATCH`: Modifies a contact (updates any fields accepted as arguments to `POST` /contacts). Public App users need access to the **Create/Modify Contacts** scope

`PUT`: Replaces a contact (same arguments and requirements as `POST` /contacts). Public App users need access to the **Create/Modify Contacts** scope

`DELETE`: Deletes a contact. Public App users need access to the **Create/Modify Contacts** scope

Example Response for GET, POST, and PATCH /contacts/{contact_id}

```
{
    "first_name": "",
    "last_name": "",
    "email": "name@email.com",
    "phone_number": "+1 202 555 0156",
    "custom_fields": null,
    "id": "1",
    "status": "active",
    "href": "https://api.surveymonkey.com/v3/contacts/1"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Contact id
first_name	string	Contact first name
last_name	string	Contact last name
email	string	Contact email address
phone_number	string	Contact phone number
href	string	Resource API URL
custom_fields	object	Custom fields
status	boolean	

PATCH /contacts/{contact_id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"first_name\":\"Quinn\", \"last_name\":\"Doe\", \"email\":\"test@surveymonkey.com\", \"phone_number\":\"+1 202 555 0156\"}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/v3/contacts/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

PATCH /contacts/{contact_id}

[PATCH]: Modifies a contact (updates any fields accepted as arguments to **[POST]** /contacts. Public App users need access to the **Create/Modify Contacts** scope

Body parameter

```
{
  "first_name": "Quinn",
  "last_name": "Doe",
  "email": "test@surveymonkey.com",
  "phone_number": "+1 202 555 0156"
}
```

UPDATE A CONTACT INFORMATION

Name	Type	Required	Description
first_name	string	undefined	Contact first name
last_name	string	undefined	Contact last name
email	string	undefined	Contact email address. Required if no phone_number
phone_number	string	undefined	Contact phone_number. Required if no email

Example response for PATCH /contacts/{contact_id}

```
{
  "first_name": "",
  "last_name": "",
  "email": "name@email.com",
  "phone_number": "+1 202 555 0156",
  "custom_fields": null,
  "id": "1",
  "status": "active",
  "href": "https://api.surveymonkey.com/v3/contacts/1"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Contact id
first_name	string	Contact first name
last_name	string	Contact last name
email	string	Contact email address

Name	Type	Description
phone_number	string	Contact phone number
href	string	Resource API URL
custom_fields	object	Custom fields
status	boolean	

PUT /contacts/{contact_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"first_name\": \"Quinn\", \"last_name\": \"Doe\", \"email\": \"test@surveymonkey.com\", \"phone_number\": \"+1 202 555 0156\"}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PUT", "/v3/contacts/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`PUT /contacts/{contact_id}`

`PUT`: Replaces a contact (same arguments and requirements as `POST /contacts`). Public App users need access to the **Create/Modify Contacts** scope

Example response for `PUT /contacts/{contact_id}`

```
{
  "first_name": "",
  "last_name": "",
  "email": "name@email.com",
  "phone_number": "+1 202 555 0156",
  "custom_fields": null,
  "id": "1",
  "status": "active",
  "href": "https://api.surveymonkey.com/v3/contacts/1"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Contact id
first_name	string	Contact first name
last_name	string	Contact last name
email	string	Contact email address
phone_number	string	Contact phone number
href	string	Resource API URL

Name	Type	Description
custom_fields	object	Custom fields
status	boolean	

GET /contact_fields

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/contact_fields", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /contact_fields](#)

AVAILABLE METHODS

[HEAD](#): Checks if resource is available

[OPTIONS](#): Returns available methods and options

[GET](#): Returns a list of all custom contacts fields. Public App users need access to the **View Contacts** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	integer	false	Page number. Default is 1
per_page	integer	false	Number of contacts to return per page, Default is 50

Example Response for GET /contact_fields

```
{
  "page": 1,
  "per_page": 1,
  "total": 1,
  "data": [
    {
      "href": "https://api.surveymonkey.com/v3/contact_fields/1",
      "id": "1",
      "label": "Custom 1"
    }
  ],
  "links": {
    "self": "https://api.surveymonkey.com/v3/contact_fields?page=1&per_page=1"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	object	

Name	Type	Description
id	string	Contact id
label	string	Contact Field Label
href	string	Resource API URL
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	

GET /contact_fields/{contact_field_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/contact_fields/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /contact_fields/{contact_field_id}](#)

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns the specified custom contact field. Public App users need access to the **View Contacts** scope

PATCH: Modifies a custom contact field's label. Public App users need access to the **Create/Modify Contacts** scope

Example Response for GET /contact_fields/{id}

```
{
  "href": "https://api.surveymonkey.com/v3/contact_fields/1",
  "id": "1",
  "label": "Custom 1"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Contact id
label	string	Contact Field Label
href	string	Resource API URL

PATCH /contact_fields/{contact_field_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"label\":\"Custom 1\"}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/v3/contact_fields/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`PATCH /contact_fields/{contact_field_id}`

Body parameter

```
{
    "label": "Custom 1"
}
```

REQUEST BODY FOR PATCH /CONTACT_FIELDS/{CONTACT_FIELD_ID}

Name	Type	Required	Description
label	string	true	Label assigned to the custom contact field

Example response for PATCH /contact_fields/{id}

```
{
    "href": "https://api.surveymonkey.com/v3/contact_fields/1",
    "id": "1",
    "label": "Custom 1"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Contact id
label	string	Contact Field Label
href	string	Resource API URL

Collectors and Invite Messages

Collectors allow you to collect survey responses with a link to your survey. Many `types` of collectors are available through the API such as `sms`, `web link`, `email` and general `popup` collectors. Weblink collectors give you a survey URL, email invitation collectors and sms invitation collectors send survey invite messages with a survey URL via the `/messages` endpoints, popup collectors let you embed surveys on your website. A

variety of collector options are accepted as arguments to `/surveys/{id}/collectors`. All collectors except for weblink collectors and some collector options, for example, `is_branding_enabled=False` require a SurveyMonkey paid plan.

- 1 Your email address needs to be verified in order to successfully send messages through our email invitation collector. You can verify your address in your SurveyMonkey account.

GET /surveys/{survey_id}/collectors

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/collectors", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`GET /surveys/{survey_id}/collectors`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns a list of collectors for a given survey. Public App users need access to the **View Collectors** scope

`POST`: Creates an sms, weblink, email collector or popup collector for a given survey. Public App users need access to the **Create/Modify Collectors** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	integer	false	Which page of resources to return. Defaults to 1
per_page	integer	false	Number of resources to return per page
sort_by	string	false	Field used to sort returned collector list e.g. ['id', 'date_modified', 'type', 'status', 'name']
sort_order	string	false	Sort order e.g. ['ASC', 'DESC']
name	string	false	Nickname of collector to search against
start_date	string	false	Collectors must be created after this date. Date string in format YYYY-MM-DDTHH:MM:SS (no offset)
end_date	string	false	Collectors must be created before this date. Date string in format YYYY-MM-DDTHH:MM:SS (no offset)
include	string	false	Specify additional fields to return per collector: 'type', 'status', 'response_count', 'date_created', 'date_modified', 'url'

Example Response for `GET /surveys/{survey_id}/collectors`

```
{
  "data": [
    {
      "name": "Teams Poll",
      "id": "1",
      "href": "https://api.surveymonkey.com/v3/collectors/1"
    }
  ],
}
```

```

"per_page": 50,
"page": 1,
"total": 1,
"links": {
  "self": "https://api.surveymonkey.com/v3/surveys/309536605/collectors?page=1&per_page=50"
}
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	object	
id	string	Collector id
name	string	Collector Name
href	string	Resource API URL
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	

POST /surveys/{survey_id}/collectors

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"type\":\"string\",\"name\":\"string\",\"thank_you_page\":{\"is_enabled\":true,\"message\":\"string\",\"thank_you_message\":\"string\",\"disqualification_message\":\"string\",\"disqualification_url\":\"string\",\"close_date\":\"string\",\"closed_page_message\":\"string\",\"redirect_url\":\"string\",\"display_survey_results\":true,\"edit_response_type\":\"string\",\"anonymous_type\":\"string\",\"allow_multiple_responses\":true,\"password\":\"string\",\"sender_email\":\"string\",\"response_limit\":0,\"redirect_type\":\"string\",\"width\":0,\"height\":0,\"border_color\":\"#FFFFFF\",\"is_branding_enabled\":true,\"headline\":\"string\",\"message\":\"string\",\"sample_rate\":0,\"primary_button\":{\"bg_color\":\"#FFFFFF\",\"text_color\":\"#000000\",\"text\":\"string\"},\"secondary_button\":{\"bg_color\":\"#FFFFFF\",\"text_color\":\"#000000\",\"text\":\"string\"},\"respondent_authentication\":false,\"from_collector_id\":\"string\"}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/surveys/1234/collectors", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

POST /surveys/{survey_id}/collectors

Call Types:

Blank: Creating a collector from scratch

Copy: Creating a collector by Copying an existing collector

Body parameter

```
{
  "type": "string",
  "name": "string",
  "thank_you_page": {
    "is_enabled": true,
    "message": "string"
  },
  "thank_you_message": "string",
  "disqualification_message": "string",
  "disqualification_url": "string",
  "close_date": "string",
  "closed_page_message": "string",
  "redirect_url": "string",
  "display_survey_results": true,
  "edit_response_type": "string",
  "anonymous_type": "string",
  "allow_multiple_responses": true,
  "password": "string",
  "sender_email": "string",
  "response_limit": 0,
  "redirect_type": "string",
  "width": 0,
  "height": 0,
  "border_color": "#FFFFFF",
  "is_branding_enabled": true,
  "headline": "string",
  "message": "string",
  "sample_rate": 0,
  "primary_button": {
    "bg_color": "#FFFFFF",
    "text_color": "#000000",
    "text": "string"
  },
  "secondary_button": {
    "bg_color": "#FFFFFF",
    "text_color": "#000000",
    "text": "string"
  },
  "respondent_authentication": false,
  "from_collector_id": "string"
}
}
```

REQUEST BODY FOR POST /SURVEYS/{SURVEY_ID}/COLLECTORS

Name	Type	Required	Description
type	string	undefined	'sms', 'weblink', 'email'. Popup Collectors: 'popup_invitation', 'embedded_survey', 'popup_survey'
name	string	true	Collector name
thank_you_page	object	is_enabled	Settings for a custom thank you page. This is a replacement for <code>thank_you_message</code> and should be used instead of that. Using <code>thank_you_page</code> supersedes <code>thank_you_message</code> .
is_enabled	boolean	true	Whether to use the custom thank you message. This toggles the use of <code>message</code> .
message	string	false	Custom message for the thank you page (default='Thank you for completing our survey!'), regardless of whether a custom thank you message is enabled.
thank_you_message	string	false	Message for <code>thank you page</code> (default='Thank you for completing our survey!). This is an older form of <code>thank_you_page</code> and should not be used.
disqualification_message	string	false	Message for disqualification page (default='Thank you for completing our survey!')
disqualification_url	string	false	Redirect to this url for disqualification page
close_date	string	false	Close date of collector
closed_page_message	string	false	Message shown when a survey is closed (default='Thank you for completing our survey!')
redirect_url	string	false	Redirect to this url upon survey completion
display_survey_results	boolean	false	Shows respondents survey instant results when they complete the survey (default=False)

Name	Type	Required	Description
edit_response_type	string	false	When respondents can edit their response: 'until_complete', 'never', or 'always' (default='until_complete')
anonymous_type	string	false	Turns off IP tracking. For email collectors, also removes respondent email address and name from response: 'not_anonymous', 'partially_anonymous', 'fully_anonymous' (default='not_anonymous')
allow_multiple_responses	boolean	false	Allows respondents to take a survey more than once from the same browser on the same computer. Not available for <code>email</code> collectors (default=False)
password	string	false	Set a password to restrict access to your survey
sender_email	string	false	Sender email for email collectors
response_limit	integer	false	Sets the collector to close after specified number of responses are collected
redirect_type	string	false	Determines survey end page behavior: <code>url</code> (redirects to URL set in <code>redirect_url</code> or if none is set, shows standard SurveyMonkey thank you page), <code>close</code> (closes the survey window or tab), or <code>loop</code> (loops the survey back to the beginning, only available for <code>weblink</code> collectors with <code>allow_multiple_responses</code> :true)
width	integer	false	Sets the width of your popup. Minimum 300, maximum 700
height	integer	false	Sets the height of your popup. Minimum 300, maximum 700
border_color	string	false	Changes border color surrounding the survey
is_branding_enabled	boolean	false	Whether the popup has SurveyMonkey branding
headline	string	false	Only applies to 'popup_invitation' and 'popup_survey'. Represents the title of the modal window surrounding the survey
message	string	false	Only applies to 'popup_invitation'. Represents the message below the headline
sample_rate	integer	false	Only applies to 'popup_invitation' and 'popup_survey'. Determines the percentage of people who will see your popup. Must be between 1 and 100
primary_button	object	undefined	Only applies to 'popup_invitation'. Users who click the primary button will be taken to the beginning of your survey
bg_color	string	false	Primary button's background color
text_color	string	false	Primary button's text color
text	string	false	The text on the primary button
secondary_button	object	undefined	Only applies to 'popup_invitation'. The secondary button closes the popup. Once a user has seen the popup, they will not see it again
bg_color	string	false	Secondary button's background color
text_color	string	false	Secondary button's text color
text	string	false	The text on the secondary button
respondent_authentication	boolean	false	Indicates whether Respondent Authentication is required. This is used in conjunction with single sign-on (SSO) to require respondents to sign-in. You can use Respondent Authentication to send more secure internal surveys and track the people taking your survey with SSO metadata.
from_collector_id	string	false	Collector ID to copy collector from

Response for POST /surveys/{survey_id}/collectors

```
{
  "status": "open",
  "id": "1234",
  "type": "weblink",
  "name": "My Collector",
  "thank_you_page": {
```

```

    "is_enabled": true,
    "message": "Thank you for taking my survey."
},
"thank_you_message": "Thank you for taking my survey.",
"disqualification_message": "Thank you for taking my survey.",
"disqualification_url": "https://www.surveymonkey.com",
"close_date": "2038-01-01T00:00:00+00:00",
"closed_page_message": "This survey is currently closed.",
"redirect_url": "https://www.surveymonkey.com",
"display_survey_results": false,
"edit_response_type": "until_complete",
"anonymous_type": "not_anonymous",
"allow_multiple_responses": false,
"date_modified": "2015-10-06T12:56:55+00:00",
"url": "https://www.surveymonkey.com/r/2Q3RXZB",
"open": true,
"date_created": "2015-10-06T12:56:55+00:00",
"password_enabled": false,
"sender_email": null,
"response_limit": null,
"redirect_type": "url",
"respondent_authentication": false,
"href": "https://api.surveymonkey.com/v3/collectors/1234"
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
response_count	integer	
close_date	any	
anonymous_type	string	
type	string	
allow_multiple_responses	boolean	
id	string	
href	string	
closed_page_message	string	
redirect_type	string	
thank_you_page	object	Settings for a custom thank you page. This is a replacement for <code>thank_you_message</code> and should be used instead of that. Using <code>thank_you_page</code> supersedes <code>thank_you_message</code> .
is_enabled	boolean	Whether to use the custom thank you message. This toggles the use of <code>message</code> .
message	string	Custom message for the thank you page (default='Thank you for completing our survey!'), regardless of whether a custom thank you message is enabled.
thank_you_message	string	
sender_email	any	
password_enabled	boolean	
date_created	string	
edit_response_type	string	
disqualification_url	string	
url	any	
survey_id	string	

Name	Type	Description
response_limit	any	
date_modified	string	
disqualification_type	string	
name	string	
display_survey_results	boolean	
redirect_url	string	
status	string	
disqualification_message	string	
respondent_authentication	boolean	

GET /collectors/{collector_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/collectors/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /collectors/{collector_id}](#)

AVAILABLE METHODS

[HEAD](#): Checks if resource is available

[OPTIONS](#): Returns available methods and options

[GET](#): Returns a collector

[PATCH](#): Modifies a collector (updates any fields accepted as arguments to [POST /surveys/{survey_id}/collectors](#), with the addition of [status](#) and the removal of [type](#)). Public App users need access to the **Create/Modify Collectors** scope

[PUT](#): Replaces a collector (same arguments and requirements as [POST /surveys/{survey_id}/collectors](#), with the addition of [status](#) and the removal of [type](#)). Public App users need access to the **Create/Modify Collectors** scope

[DELETE](#): Deletes a collector. Public App users need access to the **Create/Modify Collectors** scope

Example Response for GET, PATCH, PUT /collectors/{collector_id}

```
{
  "status": "open",
  "id": "1234",
  "survey_id": "123456789",
  "type": "weblink",
  "name": "My Collector",
  "thank_you_message": "Thank you for taking my survey.",
  "thank_you_page": {
    "is_enabled": true,
    "message": "Thank you for taking my survey."
  },
  "disqualification_message": "Thank you for taking my survey.",
  "disqualification_url": "https://www.surveymonkey.com",
  "close_date": "2038-01-01T00:00:00+00:00",
```

```

"closed_page_message": "This survey is currently closed.",
"redirect_url": "https://www.surveymonkey.com",
"display_survey_results": false,
"edit_response_type": "until_complete",
"anonymous_type": "not_anonymous",
"allow_multiple_responses": false,
"date_modified": "2015-10-06T12:56:55+00:00",
"url": "https://www.surveymonkey.com/r/2Q3RXZB",
"date_created": "2015-10-06T12:56:55+00:00",
"sender_email": null,
"password_enabled": false,
"response_limit": 100,
"redirect_type": "url",
"respondent_authentication": false,
"href": "https://api.surveymonkey.com/v3/collectors/1234"
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
status	string	Collector status: 'open' or 'closed'
id	string	Collector id
survey_id	string	ID of the survey the collector belongs to
type	string	Collector type: 'sms', 'weblink' or 'email'
name	string	Name of the collector
thank_you_page	object	Settings for a custom thank you page. This is a replacement for <code>thank_you_message</code> and should be used instead of that. Using <code>thank_you_page</code> supersedes <code>thank_you_message</code> .
is_enabled	boolean	Whether to use the custom thank you message. This toggles the use of <code>message</code> .
message	string	Custom message for the thank you page (default='Thank you for completing our survey!'), regardless of whether a custom thank you message is enabled.
thank_you_message	string	Message for thank you page
disqualification_message	string	Message for disqualification page
disqualification_url	string	Redirect to this url for disqualification page
close_date	string	Close date of collector
closed_page_message	string	Message shown when someone visits a closed survey
redirect_url	string	Redirects respondent to this url upon survey completion
display_survey_results	boolean	Shows respondents survey instant results when they complete the survey
edit_response_type	string	When respondents can edit their response: 'until_complete', 'never', or 'always'
anonymous_type	string	Turns off IP tracking. For email collectors, also removes respondent email address and name from response: 'not_anonymous', 'partially_anonymous', 'fully_anonymous'
allow_multiple_responses	boolean	Allows respondents to take a survey more than once from the same browser on the same computer. Only available for <code>weblink</code> collectors.
date_modified	string	Date collector was last modified
url	string	If collector is a Web Collector (type 'weblink') then the url for the collector
date_created	string	Date collector was created
password_enabled	boolean	True if the collector is password protected
sender_email	string	Sender email for email collectors. User's email is used if null

Name	Type	Description
redirect_type	string	Determines survey end page behavior: <code>url</code> (redirects to URL set in <code>redirect_url</code> or if none is set, shows standard SurveyMonkey thank you page), <code>close</code> (closes the survey window or tab), or <code>loop</code> (loops the survey back to the beginning, only available for <code>weblink</code> collectors with <code>allow_multiple_responses</code> :true)
href	string	Resource API URL
width	integer	Width of your popup
height	integer	Height of your popup
border_color	Hex string	Border color surrounding the survey
is_branding_enabled	boolean	Whether the popup has SurveyMonkey branding
headline	string	Title of the modal window surrounding the survey
message	string	The message below the headline
sample_rate	string	The percentage of people who will see your popup
primary_button	object	The primary button of your popup
bg_color	Hex string	Primary button's background color
text_color	Hex string	Primary button's text color
text	string	The text on the primary button
secondary_button	object	The secondary button of your popup
bg_color	Hex string	Secondary button's background color
text_color	Hex string	Secondary button's text color
text	string	The text on the secondary button
respondent_authentication	boolean	Indicates whether Respondent Authentication is required. This is used in conjunction with single sign-on (SSO) to require respondents to sign-in. You can use Respondent Authentication to send more secure internal surveys and track the people taking your survey with SSO metadata.

PATCH /collectors/{collector_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"status\":\"open\",\"name\":\"string\",\"thank_you_message\":\"string\",\"thank_you_page\":{\"is_enabled\":true,\"message\":\"string\"},\"disqualification_message\":\"string\",\"disqualification_url\":\"string\",\"close_date\":\"string\",\"closed_page_message\":\"string\",\"redirect_url\":\"string\",\"display_survey_results\":true,\"edit_response_type\":\"string\",\"anonymous_type\":\"string\",\"allow_multiple_responses\":true,\"password\":\"string\",\"sender_email\":\"string\",\"response_limit\":0,\"redirect_type\":\"string\",\"width\":8,\"height\":8,\"border_color\":\"string\",\"is_branding_enabled\":true,\"headline\":\"string\",\"message\":\"string\",\"sample_rate\":0,\"primary_button\":{\"bg_color\":\"string\",\"text_color\":\"string\"},\"secondary_button\":{\"bg_color\":\"string\",\"text_color\":\"string\"},\"text\":\"string\"},\"respondent_authentication\":false}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/collectors/1234567890", payload, headers)

res = conn.getresponse()
data = res.read()

print(data)
```

```

conn.request("PATCH", "/v3/collectors/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

PATCH /collectors/{collector_id}

PATCH: Modifies a collector (updates any fields accepted as arguments to **POST** /surveys/{survey_id}/collectors, with the addition of **status** and the removal of **type**). Public App users need access to the **Create/Modify Collectors** scope

PUT /collectors/{collector_id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"status\":\"open\",\"name\":\"string\",\"thank_you_message\":\"string\",\"thank_you_page\":{\"is_enabled\":true,\"message\":\"string\"},\"disqualification_message\":\"string\",\"disqualification_url\":\"string\",\"close_date\":\"string\",\"closed_page_message\":\"string\",\"redirect_url\":\"string\",\"display_survey_results\":true,\"edit_response_type\":\"string\",\"anonymous_type\":\"string\",\"allow_multiple_responses\":true,\"password\":\"string\",\"sender_email\":\"string\",\"response_limit\":0,\"redirect_type\":\"string\",\"width\":0,\"height\":0,\"border_color\":\"string\",\"is_branding_enabled\":true,\"headline\":\"string\",\"message\":\"string\",\"sample_rate\":0,\"primary_button\":{\"bg_color\":\"string\",\"text_color\":\"string\",\"text\":\"string\"},\"secondary_button\":{\"bg_color\":\"string\",\"text_color\":\"string\",\"text\":\"string\"},\"respondent_authentication\":false}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PUT", "/v3/collectors/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

PUT /collectors/{collector_id}

PUT: Replaces a collector (same arguments and requirements as **POST** /surveys/{survey_id}/collectors, with the addition of **status** and the removal of **type**). Public App users need access to the **Create/Modify Collectors** scope

GET /collectors/{collector_id}/messages

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/collectors/1234/messages", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

GET /collectors/{collector_id}/messages

AVAILABLE METHODS

ⓘ Email and SMS Invitations created with the SurveyMonkey API are subject to our Anti Spam Policy. While you can edit the text surrounding the opt out link to better customize your invitation message, per our Anti-Spam Policy you must clearly explain what the opt out link does, and you should not attempt to hide the link in the message.

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns a message. Public App users need access to the **View Collectors** scope

POST: Creates a message. See `/collectors/{id}/messages/{id}/recipients` to add recipients and `/collectors/{id}/messages/{id}/send` to send or schedule. Public App users need access to the **Create/Modify Collectors** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
per_page	integer	false	Number of resources to return per page
page	integer	false	Which page of resources to return. Defaults to 1

Example Response for GET, PATCH, PUT `/collectors/{collector_id}/messages`

```
{
  "data": [
    {
      "status": "sent",
      "type": "invite",
      "id": "86554397",
      "href": "https://api.surveymonkey.com/v3/collectors/401356528/messages/86554397"
    }
  ],
  "per_page": 50,
  "page": 1,
  "total": 1,
  "links": {
    "self": "https://api.surveymonkey.com/v3/surveys/277387623/collectors/401356528/messages?page=1&per_page=50"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	object	
status	string	Message status
type	string	Message type
id	integer	Message id
href	string	Resource link
per_page	integer	
page	integer	
total	integer	
links	object	
self	string	

POST `/collectors/{collector_id}/messages`

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"type\": \"invite\", \"from_collector_id\": \"\", \"from_message_id\": \"\", \"include_recipients\": false, \"recipient_status\": \"\", \"subject\": \"Share your opinion with us\", \"body_text (for email invitations)\": \"We want your opinion!\", \"body_text (for sms messages)\": \"We want your opinion!\", \"body_html\": \"\", \"embed_first_question\": true, \"is_branding_enabled\": true}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/collectors/1234/messages", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

POST /collectors/{collector_id}/messages

 Only creators of a collector will be able to add messages to the collector.

Body parameter

```
{
  "type": "invite",
  "from_collector_id": "",
  "from_message_id": "",
  "include_recipients": false,
  "recipient_status": "",
  "subject": "Share your opinion with us",
  "body_text (for email invitations)": "We want your opinion!",
  "body_text (for sms messages)": "We want your opinion!",
  "body_html": "",
  "embed_first_question": true,
  "is_branding_enabled": true
}
```

REQUEST BODY FOR POST /COLLECTORS/{COLLECTOR_ID}/MESSAGES

Name	Type	Required	Description
type	string	true	Message type: 'sms', 'invite', 'reminder', or 'thank_you'
from_collector_id	string	false	Collector ID to copy message from
from_message_id	string	false	Message ID to copy from
include_recipients	boolean	false	Include recipients attached to existing message
recipient_status	string	false	Set of recipients to which we want to send messages. If type is 'reminder', acceptable values are: 'has_not_responded' or 'partially_responded', with the default being 'has_not_responded'. If type is 'thank_you', acceptable values are : 'completed', 'responded', or 'partially_responded', with the default being 'completed'
subject	string	false	Subject of the email message to be sent to recipients (default='We want your opinion')
body_text (for email invitations)	string	false	The plain text body of the email message to be sent to recipients. Message must include [SurveyLink], [OptOutLink], [PrivacyLink] and [FooterLink] and the opt-out and privacy links must be clearly labeled.
body_text (for sms messages)	string	false	Make your custom message 30 characters or less (including spaces) to keep your entire text invitation together as one text. Every text you send through SurveyMonkey will include the following default text, which counts toward the overall limit: Reply HELP for info. STOP to opt out. Msg&DataRatesMayApply

Name	Type	Required	Description
body_html	string	false	The html body of the email message to be sent to recipients. This overrides body_text. Message must include [SurveyLink], [OptOutLink], [PrivacyLink] and [FooterLink] and the opt-out and privacy links must be clearly labeled.
embed_first_question	boolean	false	If collector 'type' is 'invite' then setting to 'true' will embed first question into email body
is_branding_enabled	boolean	false	Whether the email has SurveyMonkey branding

Example response for POST /collectors/{collector_id}/messages

```
{
  "status": "not_sent",
  "body": "\n<html>\n\n<body style=\"margin:0; padding: 0;\">\n    <div align=\"center\">\n        <table border=\"0\" cellpadding=\"0\" cellspacing=\"0\" align=\"center\" width=\"100%\" style=\"font-family: Arial,Helvetica,sans-serif; max-width: 700px;\">\n            <tr bgcolor=\"#A7BC38\">\n                <td colspan=\"5\" height=\"40\"></td>\n            <tr>\n                <td width=\"20\"></td>\n                <td width=\"20\"></td>\n                <td align=\"center\" style=\"font-size: 29px; color:#FFF; font-weight: normal; letter-spacing: 1px; line-height: 1; text-shadow: -1px -1px 1px rgba(0, 0, 0, 0.2); font-family: Arial,Helvetica,sans-serif;\">Survey Title </td>\n                <td width=\"20\"></td>\n                <td width=\"20\"></td>\n            </tr>\n            <tr>\n                <td height=\"10\" colspan=\"5\"></td>\n                <td><p>We're conducting a survey and your input would be appreciated. Click the button below to start the survey. Thank you for your participation!</p></td>\n                <td></td>\n                <td></td>\n                <td colspan=\"3\" height=\"30\"><table border=\"0\" cellpadding=\"0\" cellspacing=\"0\" align=\"center\" style=\"background:#A7BC38; border-radius: 4px; border: 1px solid #BBBBBB; color:#FFFFFF; font-size:14px; letter-spacing: 1px; text-shadow: -1px -1px 1px rgba(0, 0, 0, 0.8); padding: 10px 18px;\">\n                    <tr>\n                        <td align=\"center\" valign=\"center\"><a href=\"[SurveyLink]\" target=\"_blank\" style=\"color: #FFFFFF; text-decoration:none;\">Begin Survey</a></td>\n                    </tr>\n                </table></td>\n            </tr>\n            <tr>\n                <td align=\"center\" style=\"color: #666666;font-size: 10px;\">\n                    <p>Please do not forward this email as its survey link is unique to you.\n                    <br><a href=\"[PrivacyLink]\" target=\"_blank\" style=\"color: #333333; text-decoration: underline;\">Privacy</a> | <a href=\"[OptOutLink]\" target=\"_blank\" style=\"color: #333333; text-decoration: underline;\">Unsubscribe</a> from this list</p>\n                <td align=\"center\" colspan=\"5\" height=\"20\"><td colspan=\"5\">[FooterLink]</td>\n            </tr>\n        </table>\n    </div>\n</body>\n</html>\n",
  "recipient_status": null,
  "is_branding_enabled": true,
  "href": "https://api.surveymonkey.com/v3/collectors/1234/messages/1234",
  "is_scheduled": false,
  "scheduled_date": null,
  "date_created": "2016-08-17T23:47:37+00:00",
  "type": "invite",
  "id": "31454399",
  "subject": "We want your opinion"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
status	string	Status of the message
body	string	Email body
recipient_status	string	Set of recipients to send to
subject	string	Subject of the email message to be sent to recipients
is_branding_enabled	boolean	Whether the email has SurveyMonkey branding
is_scheduled	boolean	Whether the email is scheduled
schedule_date	date	Date email is scheduled to send
date_created	date	Date email was created
type	string	Type of message
id	integer	Id of the message

GET /collectors/{collector_id}/messages/{message_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/collectors/1234/messages/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /collectors/{collector_id}/messages/{message_id}

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns a message. Public App users need access to the **View Collectors** scope

PATCH: Modifies a message (only subject, body_text, body_html, is_branding, and recipient_status if 'reminder' or 'thank_you', can be modified). Public App users need access to the **Create/Modify Collectors** scope

PUT: Replaces a message (only subject, body_text, body_html, is_branding, and recipient_status if 'reminder' or 'thank_you', can be modified). Public App users need access to the **Create/Modify Collectors** scope

DELETE: Deletes a message. Public App users need access to the **Create/Modify Collectors** scope

Tip Only subject, body_text, body_html, is_branding, and recipient_status if 'reminder' or 'thank_you', can be modified with PUT and PATCH.

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
per_page	integer	false	Number of resources to return per page
page	integer	false	Which page of resources to return. Defaults to 1

Example Response for GET, PATCH, PUT /collectors/{id}/messages{id}

```
{
  "status": "not_sent",
  "body": "\n<html>\n\n<body style=\"margin:0; padding: 0;\">\n      <div align=\"center\">\n          <table border=\"0\" cellpadding=\"0\" cellspacing=\"0\" align=\"center\" width=\"100%\" style=\"font-family: Arial,Helvetica,sans-serif; max-width: 700px;\">\n              <tr bgcolor=\"#A7BC38\">\n                  <td colspan=\"5\" height=\"40\"> </td> </tr>\n              <tr bgcolor=\"#A7BC38\">\n                  <td width=\"20\"> </td>\n                  <td width=\"20\"> </td>\n                  <td align=\"center\" style=\"font-size: 29px; color:#FFF; font-weight: normal; letter-spacing: 1px; line-height: 1; text-shadow: -1px -1px 1px rgba(0, 0, 0, 0.2); font-family: Arial,Helvetica,sans-serif;\">Survey Title </td>\n                  <td width=\"20\"> </td>\n                  <td width=\"20\"> </td> </tr>\n              <tr bgcolor=\"#A7BC38\">\n                  <td height=\"10\" colspan=\"5\"> </td> </tr>\n                  <tr>\n                      <td height=\"10\" colspan=\"5\"> </td> </tr>\n                      <td> </td>\n                      <td colspan=\"3\">We're conducting a survey and your input would be appreciated. Click the button below to start the survey. Thank you for your participation!</td> </td>\n                      <td> </td>\n                      <td colspan=\"3\" height=\"30\"> </td>\n                  </tr>\n                  <tr>\n                      <td colspan=\"3\"> </td>\n                      <td border=\"0\" cellpadding=\"0\" cellspacing=\"0\" align=\"center\" style=\"background:#A7BC38; border-radius: 4px; border: 1px solid #BBBBBB; color:#FFFFFF; font-size:14px; letter-spacing: 1px; text-shadow: -1px -1px 1px rgba(0, 0, 0, 0.8); padding: 10px 18px;\">\n                          <tr>\n                              <td align=\"center\" valign=\"center\"> <a href=\"[SurveyLink]\" target=\"_blank\" style=\"color:#FFF; text-decoration:none;\">Begin Survey</a> </td> </tr> </td>\n                          <td align=\"center\" valign=\"top\" style=\"color: #666666;font-size: 10px;\">\n                              <tr>\n                                  <td align=\"center\"> <a href=\"[OptOutLink]\" target=\"_blank\" style=\"color: #333333; text-decoration: underline;\">Privacy</a> | <a href=\"[UnsubscribeLink]\" target=\"_blank\" style=\"color: #333333; text-decoration: underline;\">Unsubscribe</a> from this list</a> </td>\n                              <td align=\"center\" colspan=\"5\">[FooterLink]</td> </tr>\n                          </td>\n                      <td align=\"center\" colspan=\"3\"> </td>\n                  </tr>\n              </table>\n          </div>\n      </body>\n  </html>",
  "recipient_status": null,
  "is_branding_enabled": true,
  "href": "https://api.surveymonkey.com/v3/collectors/1234/messages/1234",
  "is_scheduled": false,
  "scheduled_date": null,
  "date_created": "2016-08-17T23:47:37+00:00",
```

```

    "type": "invite",
    "id": "31454399",
    "subject": "We want your opinion"
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
status	string	Status of the message
body	string	Email body
recipient_status	string	Set of recipients to send to
subject	string	Subject of the email message to be sent to recipients
is_branding_enabled	boolean	Whether the email has SurveyMonkey branding
is_scheduled	boolean	Whether the email is scheduled
schedule_date	date	Date email is scheduled to send
date_created	date	Date email was created
type	string	Type of message
id	integer	Id of the message

PATCH /collectors/{collector_id}/messages/{message_id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"type\":\"invite\",\"from_collector_id\":\"\", \"from_message_id\":\"\", \"include_recipients\":false, \"recipient_status\":\"\", \"subject\":\"Share your opinion with us \", \"body_text (for email invitations)\":\"We want your opinion!\", \"body_text (for sms messages)\":\"We want your opinion!\", \"body_html\":\"\", \"embed_first_question\":true, \"is_branding_enabled\":true}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/v3/collectors/1234/messages/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

[PATCH /collectors/{collector_id}/messages/{message_id}](#)

PATCH: Modifies a message (only subject, body_text, body_html, is_branding, and recipient_status if ‘reminder’ or ‘thank_you’, can be modified). Public App users need access to the **Create/Modify Collectors** scope

Example response for PATCH /collectors/{collector_id}/messages/{message_id}

```
{
  "status": "not_sent",
  "body": "\n<html>\n<body style=\"margin:0; padding: 0;\">\n    <div align=\"center\">\n        <table border=\"0\" cellpadding=\"0\" cellspacing=\"0\" align=\"center\" width=\"100%\" style=\"font-family: Arial,Helvetica,sans-serif; max-width: 700px;\">\n            <tr bgcolor=\"#A7BC38\">\n                <td colspan=\"5\" height=\"40\"> </td> </tr>\n            <tr bgcolor=\"#A7BC38\">\n                <td width=\"20\"> </td>\n                <td width=\"20\"> </td>\n                <td align=\"center\" style=\"font-size: 29px; color:#FFF; font-weight: normal; letter-spacing: 1px; line-height: 1;\">\n                    text-shadow: -1px -1px 1px rgba(0, 0, 0, 0.2); font-family: Arial,Helvetica,sans-serif;\n                </td>\n                <td width=\"20\"> </td>\n            </tr>\n        </table>\n    </div>\n</body>\n</html>"
}
```

```

Survey Title </td>\n          <td width=\"20\"> </td>\n          <td width=\"20\"> </td> </tr>\n          <tr>\n            <td height=\"10\" colspan=\"5\"> </td> </tr>\n<p>We're conducting a survey and your input would be appreciated. Click the button below to start the survey. Thank you for your participation!</p> </td>\n          <td colspan=\"3\"> </td> </tr> \n          <tr>\n            <td colspan=\"5\" border=\"0\" cellpadding=\"0\" cellspacing=\"0\" height=\"30\" align=\"center\"> <table border=\"0\" cellpadding=\"0\" cellspacing=\"0\" align=\"center\"> <a href=\"[SurveyLink]\" target=\"_blank\" style=\"color: #FFFFFF; text-decoration:none;\">Begin Survey</a> </td> </tr> </table> </td> </tr>\n            <td align=\"center\" valign=\"center\"> <a href=\"[SurveyLink]\" target=\"_blank\" style=\"color: #FFFFFF; text-decoration:none;\">Begin Survey</a> </td> </tr> </table> </td> </tr>\n            <td align=\"center\" colspan=\"5\" height=\"30\"> <br><a href=\"[PrivacyLink]\" target=\"_blank\" style=\"color: #333333; text-decoration: underline;\">Privacy</a> | <a href=\"[OptOutLink]\" target=\"_blank\" style=\"color: #333333; text-decoration: underline;\">Unsubscribe</a> from this list</p> </td>\n            <td align=\"center\" colspan=\"5\" height=\"20\" colspan=\"5\"> </td> </tr> \n            <tr>\n              <td align=\"center\" colspan=\"5\">[FooterLink]</td> </tr>\n            <tr>\n              <td align=\"center\" colspan=\"5\" height=\"20\" colspan=\"5\"> </td> </tr> </table></div></body>\n          
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
status	string	Status of the message
body	string	Email body
recipient_status	string	Set of recipients to send to
subject	string	Subject of the email message to be sent to recipients
is_branding_enabled	boolean	Whether the email has SurveyMonkey branding
is_scheduled	boolean	Whether the email is scheduled
schedule_date	date	Date email is scheduled to send
date_created	date	Date email was created
type	string	Type of message
id	integer	Id of the message

PUT /collectors/{collector_id}/messages/{message_id}

Code samples

```

import http.client\n\nconn = http.client.HTTPSConnection("api.surveymonkey.com")\n\npayload = "{\"type\":\"invite\",\"from_collector_id\":\"\",\"from_message_id\":\"\",\"include_recipients\":false,\"recipient_status\":\"\",\"subject\":\"Share your opinion with us\",\"body_text (for email invitations)\":\"We want your opinion!\",\"body_text (for sms messages)\":\"We want your opinion!\",\"body_html\":\"\",\"embed_first_question\":true,\"is_branding_enabled\":true}\"\n\nheaders = {\n    'Content-Type': \"application/json\", \n    'Accept': \"application/json\", \n    'Authorization': \"Bearer {access-token}\" \n}\n\nconn.request(\"PUT\", \"/v3/collectors/1234/messages/1234\", payload, headers)
  
```

```

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

PUT /collectors/{collector_id}/messages/{message_id}

PUT: Replaces a message (only subject, body_text, body_html, is_branding, and recipient_status if ‘reminder’ or ‘thank_you’, can be modified). Public App users need access to the **Create/Modify Collectors** scope

Example response for PUT /collectors/{collector_id}/messages/{collector_id}

```
{
  "status": "not_sent",
  "body": "\n<html>\n\n<body style=\"margin:0; padding: 0;\">\n    <div align=\"center\">\n        <table border=\"0\" cellpadding=\"0\" cellspacing=\"0\" align=\"center\" width=\"100%\" style=\"font-family: Arial,Helvetica,sans-serif; max-width: 700px;\">\n            <tr bgcolor=\"#A7BC38\">\n                <td width=\"20\"> </td>\n                <td width=\"20\"> </td>\n                <td align=\"center\" style=\"font-size: 29px; color:#FFF; font-weight: normal; letter-spacing: 1px; line-height: 1; text-shadow: -1px -1px 1px rgba(0, 0, 0, 0.2); font-family: Arial,Helvetica,sans-serif;\">Survey Title </td>\n                <td width=\"20\"> </td>\n                <td width=\"20\"> </td>\n                <tr>\n                    <td height=\"10\" colspan=\"5\"> </td>\n                    <tr>\n                        <td height=\"10\" align=\"left\" valign=\"top\" style=\"color:#666666; font-size: 13px;\"> \n                            <p>We're conducting a survey and your input would be appreciated. Click the button below to start the survey. Thank you for your participation!</p>\n                            <td> </td>\n                            <td> </td>\n                            <td> </td>\n                            <td colspan=\"5\" height=\"30\"> </td>\n                        </tr>\n                    <tr>\n                        <td colspan=\"3\"> <table border=\"0\" cellpadding=\"0\" cellspacing=\"0\" align=\"center\" style=\"background:#A7BC38; border-radius: 4px; border: 1px solid #BBBBBB; color:#FFFFFF; font-size:14px; letter-spacing: 1px; text-shadow: -1px -1px 1px rgba(0, 0, 0, 0.8); padding: 10px 18px;\">\n                            <tr>\n                                <td align=\"center\" valign=\"center\"> <a href=\"[SurveyLink]\" target=\"_blank\" style=\"color:#FFFFFF; text-decoration:none;\">Begin Survey</a> </td>\n                            </tr>\n                        </table> </td>\n                        <td align=\"center\" colspan=\"5\" height=\"30\"> </td>\n                    </tr>\n                <tr>\n                    <td align=\"center\" colspan=\"5\"> <br><a href=\"[PrivacyLink]\" target=\"_blank\" style=\"color: #333333; text-decoration: underline;\">Privacy</a> | <a href=\"[OptOutLink]\" target=\"_blank\" style=\"color: #333333; text-decoration: underline;\">Unsubscribe</a> from this list</p> </td>\n                    <td align=\"center\" colspan=\"5\"> <br><a href=\"[FooterLink]\" target=\"_blank\" style=\"color: #999999;font-size: 10px;\">[FooterLink]</a> </td>\n                </tr>\n            </table>\n        </div>\n    </body>\n</html>\n",
  "recipient_status": null,
  "is_branding_enabled": true,
  "href": "https://api.surveymonkey.com/v3/collectors/1234/messages/1234",
  "is_scheduled": false,
  "scheduled_date": null,
  "date_created": "2016-08-17T23:47:37+00:00",
  "type": "invite",
  "id": "31454399",
  "subject": "We want your opinion"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
status	string	Status of the message
body	string	Email body
recipient_status	string	Set of recipients to send to
subject	string	Subject of the email message to be sent to recipients
is_branding_enabled	boolean	Whether the email has SurveyMonkey branding
is_scheduled	boolean	Whether the email is scheduled
schedule_date	date	Date email is scheduled to send
date_created	date	Date email was created
type	string	Type of message
id	integer	Id of the message

POST /collectors/{collector_id}/messages/{message_id}/send

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"scheduled_date\":\"2015-10-06T12:56:55+00:00\"}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/collectors/1234/messages/1234/send", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

POST /collectors/{collector_id}/messages/{message_id}/send

AVAILABLE METHODS

POST: Send or schedule to send an existing message to all message recipients. Targeted message must have a `status` of `not_sent`. See `/collectors/{id}/messages/{id}/recipients` to add recipients to a message and `/collectors/{id}/messages` to create messages. Public App users need access to the **Create/Modify Collectors** scope

Body parameter

```
{  
  "scheduled_date": "2015-10-06T12:56:55+00:00"  
}
```

REQUEST BODY FOR POST /COLLECTORS/{COLLECTOR_ID}/MESSAGES/{MESSAGE_ID}/SEND

Name	Type	Required	Description
scheduled_date	string	false	Date when the message should send. If not specified, message sends immediately

Example Response for POST /collectors/{collector_id}/messages{message_id}/send

```
{  
  "is_scheduled": true,  
  "scheduled_date": "2015-11-04T12:56:55+00:00",  
  "body": "<html>...</html>",  
  "subject": "We want your opinion",  
  "recipients": [  
    "1234",  
    "1234"  
  ],  
  "recipient_status": null,  
  "type": "invite"  
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
is_scheduled	boolean	If a message has been scheduled to send
scheduled_date	string	Date message was scheduled to be sent
body	string	The plain text body of the email or sms message to be sent to recipients.

Name	Type	Description
subject	string	Subject of the email message to be sent to recipients
recipients	array	List of recipient ids
type	string	Message type: 'sms', 'invite', 'reminder', or 'thank_you'
recipient_status	string	Recipient filter: 'reminder' or 'thank_you'

GET /collectors/{collector_id}/messages/{message_id}/recipients

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/collectors/1234/messages/1234/recipients", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /collectors/{collector_id}/messages/{message_id}/recipients

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns a list of recipients. Public App users need access to the **View Collectors** scope

POST: Creates a new recipient for the specified message. See `/collectors/{id}/messages/{id}/send` for sending. This method only available to messages of type **sms**, or **invite** if they have not already been sent. Public App users need access to the **Create/Modify Collectors** scope

💡 For POST: If the contact_id is provided (with contact_id), fetch the contact from the address book. If contact_id is not provided (without contact_id), allow passing of the email or phone number, name, and custom_fields. A new contact will only be created if an existing contact with the same email or phone number cannot be found. If custom_fields are provided, update the contact with those fields, otherwise keep existing custom_fields.

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
per_page	integer	false	Number of resources to return per page
page	integer	false	Which page of resources to return. Defaults to 1
include	string	false	Specify additional fields to return per recipient: <code>survey_response_status</code> , <code>mail_status</code> , <code>custom_fields</code> , <code>remove_link</code> , <code>extra_fields</code> , <code>survey_link</code>
sort_order	string	false	The sort order. If sorting is not needed, use <code>none</code> as it will allow the API to choose the most performant ordering plan.

ENUMERATED VALUES

Parameter	Value
sort_order	asc

Parameter	Value
sort_order	dsc
sort_order	none

Example Response for GET /collectors/{collector_id}/messages/{collector_id}/recipients

```
{
  "per_page": 1,
  "total": 1,
  "data": [
    {
      "phone_number": "+1 202 555 0156",
      "href": "https://api.surveymonkey.com/v3/collectors/1234/recipients/1234",
      "id": "1234",
      "email": "test@surveymonkey.com"
    }
  ],
  "page": 1,
  "links": {
    "self": "https://api.surveymonkey.com/v3/collectors/1234/recipients/1234?page=1&per_page=1"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	
data	object	
survey_response_status	string	If the recipient has completed the survey: <code>not_responded</code> , <code>partially_responded</code> , <code>completely_responded</code>
mail_status	string	If an invite message to the recipient has been: <code>sent</code> , <code>not_sent</code> , or is <code>processing</code>
custom_fields	object	Contact details for recipient
id	string	Recipient's id
remove_link	string	Unsubscribe link
extra_fields	object	Extra fields
survey_link	string	Link to the survey

POST /collectors/{collector_id}/messages/{message_id}/recipients

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"email\":\"test@surveymonkey.com\", \"phone_number\":\"+1 202 555 0156\", \"first_name\":\"John\", \"last_name\":\"Doe\", \"custom_fields\":{}, \"extra_fields\":{}, \"duplicates\": \"add\", \"contact_id\": \"147044\"}"
headers = { }
```

```

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/collectors/1234/messages/1234/recipients", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`POST /collectors/{collector_id}/messages/{message_id}/recipients`

Body parameter

```
{
    "email": "test@surveymonkey.com",
    "phone_number": "+1 202 555 0156",
    "first_name": "John",
    "last_name": "Doe",
    "custom_fields": {},
    "extra_fields": {},
    "duplicates": "add",
    "contact_id": "147044"
}
```

REQUEST BODY FOR POST /COLLECTORS/{COLLECTOR_ID}/MESSAGES/{MESSAGE_ID}/RECIPIENTS

Name	Type	Required	Call	Description
email	string	undefined	<code>without contact_id</code>	Email of the recipient. Required if email collector
phone_number	string	undefined	<code>without contact_id</code>	Phone number of the recipient. Required if sms collector. Should otherwise be excluded.
first_name	string	false	<code>without contact_id</code>	First name of the recipient
last_name	string	false	<code>without contact_id</code>	Last name the recipient
custom_fields	object	undefined	<code>without contact_id</code>	Custom fields for the recipient contact. Only allows <code>string</code> object value type.
extra_fields	object	undefined	<code>without contact_id</code>	Extra fields needed for the message body. Supported object value types include: <code>string</code> , <code>number</code> , <code>date string</code>
duplicates	string	false	<code>with contact_id</code> <code>without contact_id</code>	Intended behavior while adding duplicate recipient (add (default), add_if_not_in_message, add_if_not_in_collector)
contact_id	string	false	<code>with contact_id</code>	Id of the contact

Example Response for POST /collectors/{collector_id}/messages/{message_id}/recipients

```
{
    "survey_response_status": "not_responded",
    "mail_status": "not_sent",
    "email": "test@surveymonkey.com",
    "phone_number": "+1 202 555 0156",
    "first_name": "John",
    "last_name": "Doe",
    "custom_fields": {
        "1": "Mr",
        "2": "Company",
        "3": "Address",
        "4": "City",
        "5": "Country",
        "6": "Phone Number"
    },
    "id": "1234",
    "remove_link": "https://www.surveymonkey.com/optout?sm=1234",
    "extra_fields": {
        "favorite_color": "Red",
    }
}
```

```

    "second_favorite_color": "Black",
    "third_favorite_color": "Green"
  },
  "survey_link": "https://www.surveymonkey.com/r/?sm=1234",
  "href": "https://api.surveymonkey.com/v3/collectors/1234/recipients/1234"
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
first_name	string	First name of the recipient
last_name	string	Last name of the recipient
email	string	Email of the recipient
survey_response_status	string	If the recipient has completed the survey: <code>not_responded</code> , <code>partially_responded</code> , <code>completely_responded</code>
mail_status	string	If an invite message to the recipient has been: <code>sent</code> , <code>not_sent</code> , or is <code>processing</code>
custom_fields	object	Contact details for recipient
id	string	Recipient's id
remove_link	string	Unsubscribe link
extra_fields	object	Extra fields
survey_link	string	Link to the survey
href	string	Link to Recipient Resource

POST /collectors/{collector_id}/messages/{message_id}/recipients/bulk

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"contact_ids\":[],\"contact_list_ids\":[],\"contacts\":[{\"email\":\"test@surveymonkey.com\",\"phone_number\":\"+1 202 5555 0156\",\"first_name\":\"John\",\"last_name\":\"Doe\"},\"custom_fields\":{},\"extra_fields\":{}}]"

headers = {
  'Content-Type': "application/json",
  'Accept': "application/json",
  'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/collectors/1234/messages/1234/recipients/bulk", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`POST /collectors/{collector_id}/messages/{message_id}/recipients/bulk`

- If `custom_fields` are provided for any one contact, all contacts will have their `custom_fields` updated. In other words, if `contact_1`'s `custom_fields` is provided, but `contact_2`'s `custom_fields` is not provided, `contact_2`'s previous `custom_fields` will be cleared after this API call.

AVAILABLE METHODS

`POST`: Creates multiple recipients. Public App users need access to the **Create/Modify Collectors** scope

Body parameter

```
{
  "contact_ids": [],
  "contact_list_ids": [],
  "contacts": [
    {
      "email": "test@surveymonkey.com",
      "phone_number": "+1 202 555 0156",
      "first_name": "John",
      "last_name": "Doe",
      "custom_fields": {},
      "extra_fields": {}
    }
  ]
}
```

REQUEST BODY FOR POST /COLLECTORS/{COLLECTOR_ID}/MESSAGES/{MESSAGE_ID}/RECIPIENTS/BULK

Name	Type	Required	Description
contact_ids	array	false	Contact ids
contact_list_ids	array	false	Contact list ids
contacts	[object]	false	New contacts
email	string	false	Contact's email address
phone_number	string	false	Contact's phone number
first_name	string	false	Contact's first name
last_name	string	false	Contact's last name
custom_fields	object	undefined	Custom fields for contact. Only allows <code>string</code> object value type.
extra_fields	object	undefined	Extra fields needed for the message body. Supported object value types include: <code>string</code> , <code>number</code> , <code>date</code> , <code>string</code>

Example Response for POST /collectors/{collector_id}/messages/{message_id}/recipients/bulk

```
{
  "succeeded": [
    {
      "phone_number": "+1 202 555 0156",
      "id": "1234",
      "email": "test@surveymonkey.com",
      "href": "https://api.surveymonkey.com/v3/collectors/1234/recipients/1234"
    }
  ],
  "invalids": [],
  "existing": [],
  "bounced": [],
  "opted_out": [],
  "duplicate": []
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
succeeded	array	List of successfully added recipient objects
succeeded.id	string	Contact id for the recipient
succeeded.email	string	Email address for the recipient
succeeded.phone_number	string	Phone number for the recipient

Name	Type	Description
succeeded.href	string	API resource URL for the recipient
invalids	array	List of invalid recipient email addresses that were provided
existing	array	List of recipients email addresses that have already been added
bounced	array	List of recipients email addresses that have previously bounced
opted_out	array	List of recipients email addresses that have opted out of receiving emails
duplicate	array	List of recipients email addresses recipients that were provided

GET /collectors/{collector_id}/recipients

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/collectors/1234/recipients", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /collectors/{collector_id}/recipients](#)

AVAILABLE METHODS

[HEAD](#): Checks if resource is available

[OPTIONS](#): Returns available methods and options

[GET](#): Returns a list of recipients. Public App users need access to the **View Collectors** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
per_page	string	false	Number of resources to return per page
include	string	false	Specify additional fields to return per recipient: survey_response_status , mail_status , custom_fields , remove_link , extra_fields , survey_link
sort_order	string	false	The sort order. If sorting is not needed, use none as it will allow the API to choose the most performant ordering plan.

ENUMERATED VALUES

Parameter	Value
sort_order	asc
sort_order	dsc
sort_order	none

Example Response for GET /collectors/{collector_id}/recipients

```
{
  "per_page": 1,
  "total": 1,
  "data": [
    {
      "phone_number": "+1 202 555 0156",
      "href": "https://api.surveymonkey.com/v3/collectors/1234/recipients/1234",
      "id": "1234",
      "email": "test@surveymonkey.com"
    }
  ],
  "page": 1,
  "links": {
    "self": "https://api.surveymonkey.com/v3/collectors/1234/recipients/1234?page=1&per_page=1"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	
data	object	
survey_response_status	string	If the recipient has completed the survey: <code>not_responded</code> , <code>partially_responded</code> , <code>completely_responded</code>
mail_status	string	If an invite message to the recipient has been: <code>sent</code> , <code>not_sent</code> , or is <code>processing</code>
custom_fields	object	Contact details for recipient
id	string	Recipient's id
remove_link	string	Unsubscribe link
extra_fields	object	Extra fields
survey_link	string	Link to the survey

GET /surveys/{survey_id}/collectors/{collector_id}/recipients/{recipient_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
  'Accept': "application/json",
  'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/collectors/1234/recipients/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

```
GET /collectors/{collector_id}/recipients/{recipient_id}
```

AVAILABLE METHODS

- HEAD**: Checks if resource is available
- OPTIONS**: Returns available methods and options
- GET**: Returns a recipient. Public App users need access to the **View Collectors** scope
- DELETE**: Deletes a recipient. Public App users need access to the **Create/Modify Collectors** scope

Example Response for GET /surveys/{survey_id}/collectors/{collector_id}/recipients/{recipient_id}

```
{
  "first_name": "John",
  "last_name": "Doe",
  "email": "test@surveymonkey.com",
  "survey_response_status": "not_responded",
  "mail_status": "not_sent",
  "custom_fields": {
    "1": "Mr",
    "2": "Company",
    "3": "Address",
    "4": "City",
    "5": "Country",
    "6": "Phone Number"
  },
  "id": "1234",
  "remove_link": "https://www.surveymonkey.com/optout?sm=1234",
  "extra_fields": {
    "favorite_color": "Red",
    "second_favorite_color": "Black",
    "third_favorite_color": "Green"
  },
  "survey_link": "https://www.surveymonkey.com/r/?sm=1234"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
first_name	string	First name of the recipient
last_name	string	Last name of the recipient
email	string	Email of the recipient
survey_response_status	string	If the recipient has completed the survey: <code>not_responded</code> , <code>partially_responded</code> , <code>completely_responded</code>
mail_status	string	If an invite message to the recipient has been: <code>sent</code> , <code>not_sent</code> , or is <code>processing</code>
custom_fields	object	Contact details for recipient
id	string	Recipient's id
remove_link	string	Unsubscribe link
extra_fields	object	Extra fields
survey_link	string	Link to the survey

GET /collectors/{collector_id}/stats

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")
```

```

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/collectors/1234/stats", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /collectors/{collector_id}/stats`

Same as `/collectors/{collector_id}/messages/{message_id}/stats` but returns stats for all messages sent from the collector.

GET **/surveys/{survey_id}/collectors/{collector_id}/messages/{message_id}/stats**

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/collectors/1234/messages/1234/stats", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /collectors/{collector_id}/messages/{message_id}/stats`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns a stats for a collector's message

Example Response for `GET /surveys/{survey_id}/collectors/{collector_id}/messages/{message_id}/stats`

```
{
    "survey_response_status": {
        "completely_responded": 0,
        "not_responded": 0,
        "partially_responded": 0
    },
    "mail_status": {
        "opened": 0,
        "opted_out": 0,
        "not_sent": 1,
        "sent": 0,
        "bounced": 0,
        "link_clicked": 0
    },
    "recipients": 1
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
survey_response_status	object	
completely_responded	integer	Count of recipients who have completed a survey response
not_responded	integer	Count of recipients who have not started the survey
partially_responded	integer	Count of recipients who have begun the survey but not completed it
mail_status	object	
opened	integer	Count of recipients that have opened the message
opted_out	integer	Count of recipients that've clicked on the opt out link
not_sent	integer	Count of recipients that've been added but their message has not been delivered
sent	integer	Count of recipients that messages have been sent to
bounced	integer	Count of recipients with messages that bounced
link_clicked	integer	Count of messages where the included survey link was clicked on
recipients	integer	Count of recipients included in the stats

Survey Responses

These endpoints let you create responses directly via the API and view responses you have collected through any collector type.

GET /surveys/{survey_id}/responses

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys//responses", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`GET /surveys/{id}/responses`

AVAILABLE METHODS

`GET`: Returns a list of responses. Public App users need access to the **View Responses** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
custom_variables	string	false	A list of custom variables to filter for as key-value pairs. Surround each key and value with single quote characters. Key and value must be separated from one another with a URL encoded equal sign (%3D). Separate each key-value pair with a URL encoded comma (%2C).

Example response for GET /surveys/{survey_id}/responses

```
{}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	object	
id	string	Response id
href	string	Response API URL
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	

GET /collectors/{collector_id}/responses

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/collectors//responses", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /collectors/{id}/responses](#)

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns a list of responses. Public App users need access to the **View Responses** scope

POST: Creates a response. Public App users need access to the **Create/Modify Responses** scope

DELETE: Deletes all the responses from the Collector. Public App users need access to the **Create/Modify Responses** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	integer	false	Page of resources to return. Defaults to 1
per_page	integer	false	Number of resources to return per page
start_created_at	undefined	false	Responses started after this date
end_created_at	undefined	false	Responses started before this date

Name	Type	Required	Description
start_modified_at	undefined	false	Responses modified after this date
end_modified_at	undefined	false	Responses modified before this date
status	string	false	Status of the response: <code>completed</code> , <code>partial</code> , <code>overquota</code> , <code>disqualified</code>
email	string	false	Email of the recipient
first_name	string	false	First Name of the recipient
last_name	string	false	Last Name of the recipient
ip	string	false	The IP the response was taken from
custom	string	false	The custom value associated with the response
total_time_max	integer	false	The maximum amount of time spent on the response
total_time_min	integer	false	The minimum amount of time spent on the response
total_time_units	string	false	Unit of time for total_time_min and total_time_max: <code>second</code> , <code>minute</code> , <code>hour</code>
sort_order	string	false	Sort order: <code>ASC</code> or <code>DESC</code>
sort_by	string	false	Field used to sort returned responses: <code>date_modified</code>
custom_variables	string	false	A list of custom variables to filter for as key-value pairs. Surround each key and value with single quote characters. Key and value must be separated from one another with a URL encoded equal sign (%3D). Separate each key-value pair with a URL encoded comma (%2C).

Example response for GET /collectors/{collector_id}/responses

{}
{}

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	
data	object	
id	string	Response id
href	string	URL for the response resource
survey_id	string	ID of the survey the response was taken for
collector_id	string	ID of the collector the response was taken for
recipient_id	string	ID of the recipient (only for email collectors). See collector recipient
language	string	The language in which the respondent took the survey. Only available for multilingual surveys when using Power BI and Tableau integrations.

Name	Type	Description
total_time	integer	Total time in seconds spent on the survey
custom_value	string	Custom value associated with a response
edit_url	string	Weblink to the survey taking page to edit the response
analyze_url	string	Weblink to the analyze page to view the response
ip_address	string	IP Address the response was taken from
custom_variables	object	Values to any available custom variables in the survey
response_status	string	Status of the response: <code>completed</code> , <code>partial</code> , <code>overquota</code> , or <code>disqualified</code>
collection_mode	string	The collection mode of the response: <code>default</code> , <code>preview</code> , <code>data_entry</code> , <code>survey_preview</code> , or <code>edit</code>
date_created	DateString	Date the response was created
date_modified	object	Date the response was last modified
pages	[object]	Pages from the survey and their associated responses
id	integer	The ID of the page with responses
questions	[object]	The questions on that page with responses
id	integer	ID of the question with responses
variable_id	integer	ID of the random assignment variable for the question
answers	[object]	The answers for the question with responses
choice_id	integer	The choice selected
row_id	integer	The row selected
col_id	integer	The column selected
other_id	integer	The other text choice selected
text	string	Any open ended text
download_url	string	Download URL for <code>file_upload</code> questions
human_download_url	string	Download URL for <code>file_upload</code> questions, intended for humans to use. This is a non-expiring link but requires logging in.
download_url_error	string	An error describing why a file is not downloadable for a <code>file_upload</code> question
content_type	string	MIME type for <code>download_url</code>
choice_metadata	object	
weight	string	The weight associated with the choice, if present
tag_data	object	Tag data only applies for open ended/other responses
hexcolor	string	Hex color for tag
label	string	Label for tag
tag_type	string	Tag type value of user or sentiment. *sentiment requires the sentiment analysis flag enabled on the survey

POST /collectors/{collector_id}/responses

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"custom_variables\":{},\"custom_value\":\"\",\"date_created\":\"string\",\"response_status\":\"completed\",\"ip_address\":\"\",\"recipient_id\":1234,\"pages\":[{\"id\":\"1234\"},\"questions\":[{\"id\":\"1234\"},\"variable_id\":\"\",\"answers\":[{\"choice_id\":\"123456\",\"row_id\":\"\",\"col_id\":\"\",\"other_id\":\"\",\"text\":\"\"}]]]}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/collectors//responses", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

POST /collectors/{id}/responses

Body parameter

```
{
  "custom_variables": {},
  "custom_value": "",
  "date_created": "string",
  "response_status": "completed",
  "ip_address": "",
  "recipient_id": 1234,
  "pages": [
    {
      "id": "1234",
      "questions": [
        {
          "id": "1234",
          "variable_id": "",
          "answers": [
            {
              "choice_id": "123456",
              "row_id": "",
              "col_id": "",
              "other_id": "",
              "text": ""
            }
          ]
        }
      ]
    }
  ]
}
```

REQUEST BODY FOR POST /COLLECTORS/{COLLECTOR_ID}/RESPONSES

Name	Type	Required	Description
custom_variables	object	undefined	Values to any available custom variables in the survey
custom_value	string	false	A custom value to attach to the response for a weblink collector
response_status	string	false	Status of the response: <code>completed</code> , <code>partial</code> , <code>overquota</code> , <code>disqualified</code>
ip_address	string	false	IP Address the response was taken from
recipient_id	integer	false	The recipient ID from an email collector. Required for email collectors. See collector recipient
pages	[object]	true	Pages from the survey and their associated responses
id	string	true	The ID of the page with responses

Name	Type	Required	Description
questions	[object]	true	The questions on that page with responses
id	string	true	ID of the question with responses
variable_id	string	false	ID of the random assignment variable for the question
answers	[object]	true	The answers for the question with responses. See formatting question types
choice_id	string	false	The choice selected
row_id	string	false	The row selected
col_id	string	false	The column selected
other_id	string	false	The other text choice selected
text	string	false	Any open ended text

Example response for POST /collectors/{collector_id}/responses

{}
{}

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Response id
href	string	URL for the response resource
survey_id	string	ID of the survey the response was taken for
collector_id	string	ID of the collector the response was taken for
recipient_id	string	ID of the recipient (only for email collectors). See collector recipient
total_time	integer	Total time in seconds spent on the survey
custom_value	string	Custom value associated with a response
edit_url	string	Weblink to the survey taking page to edit the response
analyze_url	string	Weblink to the analyze page to view the response
ip_address	string	IP Address the response was taken from
custom_variables	object	Values to any available custom variables in the survey
response_status	string	Status of the response: <code>completed</code> , <code>partial</code> , <code>overquota</code> , or <code>disqualified</code>
collection_mode	string	The collection mode of the response: <code>default</code> , <code>preview</code> , <code>data_entry</code> , <code>survey_preview</code> , or <code>edit</code>
date_created	DateString	Date the response was created
date_modified	object	Date the response was last modified
pages	[object]	Pages from the survey and their associated responses
id	integer	The ID of the page with responses
questions	[object]	The questions on that page with responses

Name	Type	Description
id	integer	ID of the question with responses
variable_id	integer	ID of the random assignment variable for the question
answers	[object]	The answers for the question with responses
choice_id	integer	The choice selected
row_id	integer	The row selected
col_id	integer	The column selected
other_id	integer	The other text choice selected
text	string	Any open ended text
download_url	string	Download URL for <code>file_upload</code> questions. This url expires after about 12 hours.
human_download_url	string	Download URL for <code>file_upload</code> questions, intended for humans to use. This is a non-expiring link but requires logging in.
content_type	string	MIME type for <code>download_url</code>

GET /surveys/{survey_id}/responses/bulk

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys//responses/bulk", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`GET /surveys/{id}/responses/bulk`

Same as `/collectors/{survey_id}/responses/bulk`

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
custom_variables	string	false	A list of custom variables to filter for as key-value pairs. Surround each key and value with single quote characters. Key and value must be separated from one another with a URL encoded equal sign (%3D). Separate each key-value pair with a URL encoded comma (%2C).

GET /collectors/{collector_id}/responses/bulk

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")
```

```

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/collectors//responses/bulk", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /collectors/{id}/responses/bulk`

AVAILABLE METHODS

`GET`: Retrieves a list of full expanded responses, including answers to all questions. Public App users need access to the **View Response Details** scope

Note that responses for file upload questions are indicated with an expiring URL. The URL will have a query-string parameter called `Expires` which indicates the epoch timestamp of the expiration. The expiration is about 12 hours from the time of the request.

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	integer	false	Page of resources to return. Defaults to 1
per_page	integer	false	Number of resources to return per page. Max of 100 allowed per page. Defaults to 50
simple	boolean	false	When ‘true’, this includes the question and answer text, in addition to IDs.
collector_ids	string	false	Only include responses for this list of collector IDs
start_created_at	string	false	Responses started after this date
end_created_at	string	false	Responses started before this date
start_modified_at	string	false	Responses modified after this date
end_modified_at	string	false	Responses modified before this date
status	string	false	Status of the response: <code>completed</code> , <code>partial</code> , <code>overquota</code> , <code>disqualified</code>
email	string	false	Email of the recipient
first_name	string	false	First Name of the recipient
last_name	string	false	Last Name of the recipient
ip	string	false	The IP the response was taken from
custom	string	false	The custom value associated with the response
total_time_max	integer	false	The maximum amount of time spent on the response
total_time_min	integer	false	The minimum amount of time spent on the response
total_time_units	string	false	Unit of time for total_time_min and total_time_max: <code>second</code> , <code>minute</code> , or <code>hour</code>
sort_order	string	false	Sort order: <code>ASC</code> or <code>DESC</code>
sort_by	string	false	Field used to sort returned responses: <code>date_modified</code>
page_ids	string	false	List of survey pages to filter on. Returns all pages if not provided
question_ids	string	false	List of survey questions to filter on. Returns all questions if not provided
custom_variables	string	false	A list of custom variables to filter for as key-value pairs. Surround each key and value with single quote characters. Key and value must be separated from one another with a URL encoded equal sign (%3D). Separate each key-value pair with a URL encoded comma (%2C).

Example response for GET /collectors/{collector_id}/responses/bulk

```
{}
```

```
{}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	
data	object	
id	string	Response id
href	string	URL for the response resource
survey_id	string	ID of the survey the response was taken for
collector_id	string	ID of the collector the response was taken for
recipient_id	string	ID of the recipient (only for email collectors). See collector recipient
language	string	The language in which the respondent took the survey. Only available for multilingual surveys when using Power BI and Tableau integrations.
total_time	integer	Total time in seconds spent on the survey
custom_value	string	Custom value associated with a response
edit_url	string	Weblink to the survey taking page to edit the response
analyze_url	string	Weblink to the analyze page to view the response
ip_address	string	IP Address the response was taken from
custom_variables	object	Values to any available custom variables in the survey
response_status	string	Status of the response: <code>completed</code> , <code>partial</code> , <code>overquota</code> , or <code>disqualified</code>
collection_mode	string	The collection mode of the response: <code>default</code> , <code>preview</code> , <code>data_entry</code> , <code>survey_preview</code> , or <code>edit</code>
date_created	DateString	Date the response was created
date_modified	object	Date the response was last modified
pages	[object]	Pages from the survey and their associated responses
id	integer	The ID of the page with responses
questions	[object]	The questions on that page with responses
id	integer	ID of the question with responses
variable_id	integer	ID of the random assignment variable for the question
answers	[object]	The answers for the question with responses

Name	Type	Description
choice_id	integer	The choice selected
row_id	integer	The row selected
col_id	integer	The column selected
other_id	integer	The other text choice selected
text	string	Any open ended text
download_url	string	Download URL for <code>file_upload</code> questions
human_download_url	string	Download URL for <code>file_upload</code> questions, intended for humans to use. This is a non-expiring link but requires logging in.
download_url_error	string	An error describing why a file is not downloadable for a <code>file_upload</code> question
content_type	string	MIME type for <code>download_url</code>
choice_metadata	object	
weight	string	The weight associated with the choice, if present
tag_data	object	Tag data only applies for open ended/other responses
hexcolor	string	Hex color for tag
label	string	Label for tag
tag_type	string	Tag type value of user or sentiment. *sentiment requires the sentiment analysis flag enabled on the survey

GET /surveys/{survey_id}/responses/{response_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/responses/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`GET /surveys/{survey_id}/responses/{response_id}`

Same as GET /collectors/{collector_id}/responses/{response_id}

PATCH /surveys/{survey_id}/responses/{response_id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"custom_variables\":{},\"custom_value\":\"\",\"date_created\":\"string\",\"response_status\":\"completed\",\"ip_address\":\"\",\"recipient_id\":1234,\"pages\":[{\"id\":\"1234\"},\"questions\":[{\"id\":\"1234\"},\"variable_id\":\"\",\"answers\":[{\"choice_id\":\"123456\",\"row_id\":\"\",\"col_id\":\"\",\"other_id\":\"\",\"text\":\"\"}]]}]}""

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/v3/surveys/1234/responses/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

PATCH /surveys/{survey_id}/responses/{response_id}

Same as PATCH /collectors/{collector_id}/responses/{response_id}

PUT /surveys/{survey_id}/responses/{response_id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"custom_variables\":{},\"custom_value\":\"\",\"date_created\":\"string\",\"response_status\":\"completed\",\"ip_address\":\"\",\"recipient_id\":1234,\"pages\":[{\"id\":\"1234\"},\"questions\":[{\"id\":\"1234\"},\"variable_id\":\"\",\"answers\":[{\"choice_id\":\"123456\",\"row_id\":\"\",\"col_id\":\"\",\"other_id\":\"\",\"text\":\"\"}]]}]}""

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PUT", "/v3/surveys/1234/responses/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

PUT /surveys/{survey_id}/responses/{response_id}

Same as PUT /collectors/{collector_id}/responses/{response_id}

GET /collectors/{collector_id}/responses/{response_id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/collectors/1234/responses/1234", headers=headers)

```

```

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /collectors/{collector_id}/responses/{response_id}`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns a response. Public App users need access to the **View Responses** scope

`PATCH`: Modifies a response (updates any fields accepted as arguments to `POST /surveys/{id}/responses`). Public App users need access to the **Create/Modify Responses** scope

`PUT`: Replaces a response (same arguments and requirements as `POST /surveys/{id}/responses`). Public App users need access to the **Create/Modify Responses** scope

`DELETE`: Deletes a response. Public App users need access to the **Create/Modify Responses** scope

Example response for `GET /collectors/{collector_id}/responses/{response_id}`

```
{}
```

```
{}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Response id
href	string	URL for the response resource
survey_id	string	ID of the survey the response was taken for
collector_id	string	ID of the collector the response was taken for
recipient_id	string	ID of the recipient (only for email collectors). See collector recipient
total_time	integer	Total time in seconds spent on the survey
custom_value	string	Custom value associated with a response
edit_url	string	Weblink to the survey taking page to edit the response
analyze_url	string	Weblink to the analyze page to view the response
ip_address	string	IP Address the response was taken from
custom_variables	object	Values to any available custom variables in the survey
response_status	string	Status of the response: <code>completed</code> , <code>partial</code> , <code>overquota</code> , or <code>disqualified</code>
collection_mode	string	The collection mode of the response: <code>default</code> , <code>preview</code> , <code>data_entry</code> , <code>survey_preview</code> , or <code>edit</code>
date_created	DateString	Date the response was created
date_modified	object	Date the response was last modified
pages	[object]	Pages from the survey and their associated responses
id	integer	The ID of the page with responses
questions	[object]	The questions on that page with responses
id	integer	ID of the question with responses
variable_id	integer	ID of the random assignment variable for the question

Name	Type	Description
answers	[object]	The answers for the question with responses
choice_id	integer	The choice selected
row_id	integer	The row selected
col_id	integer	The column selected
other_id	integer	The other text choice selected
text	string	Any open ended text
download_url	string	Download URL for <code>file_upload</code> questions. This url expires after about 12 hours.
human_download_url	string	Download URL for <code>file_upload</code> questions, intended for humans to use. This is a non-expiring link but requires logging in.
content_type	string	MIME type for <code>download_url</code>

PATCH /collectors/{collector_id}/responses/{response_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"custom_value\":\"\", \"date_created\": \"string\", \"response_status\": \"completed\", \"ip_address\": \"\", \"pages\": [{\"id\": \"1234\", \"questions\": [{\"id\": \"1234\", \"variable_id\": \"\", \"answers\": [{\"choice_id\": \"123456\", \"row_id\": \"\", \"col_id\": \"\", \"other_id\": \"\", \"text\": \"\"}]}]}]}

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/v3/collectors/1234/responses/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`PATCH /collectors/{collector_id}/responses/{response_id}`

`PATCH`: Modifies a response (updates any fields accepted as arguments to `POST /surveys/{survey_id}/responses`). Public App users need access to the **Create/Modify Responses** scope

PUT /collectors/{collector_id}/responses/{response_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"custom_variables\":{}, \"custom_value\": \"\", \"date_created\": \"string\", \"response_status\": \"completed\", \"ip_address\": \"\", \"pages\": [{\"id\": \"1234\", \"questions\": [{\"id\": \"1234\", \"variable_id\": \"\", \"answers\": [{\"choice_id\": \"123456\", \"row_id\": \"\", \"col_id\": \"\", \"other_id\": \"\", \"text\": \"\"}]}]}]}
```

```
}

conn.request("PUT", "/v3/collectors/1234/responses/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

PUT /collectors/{collector_id}/responses/{response_id}

PUT: Replaces a response (same arguments and requirements as [POST /surveys/{id}/responses](#)). Public App users need access to the **Create/Modify Responses** scope

GET /surveys/{survey_id}/responses/{response_id}/details

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/responses/1234/details", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /surveys/{survey_id}/responses/{response_id}/details

Same as /collectors/{id}/responses/{id}/details

GET /collectors/{collector_id}/responses/{response_id}/details

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/collectors/1234/responses/1234/details", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /collectors/{collector_id}/responses/{response_id}/details

AVAILABLE METHODS

GET: Retrieve a full expanded response, including answers to all questions. Public App users need access to the **View Response Details** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page_ids	string	false	List of survey pages to filter on. Returns all pages if not provided
question_ids	string	false	List of survey questions to filter on. Returns all questions if not provided
simple	boolean	false	When 'true', this includes the question and answer text, in addition to IDs.

Example response for GET /collectors/{id}/responses/details

```
{}
```

```
{}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Response id
href	string	URL for the response resource
survey_id	string	ID of the survey the response was taken for
collector_id	string	ID of the collector the response was taken for
recipient_id	string	ID of the recipient (only for email collectors). See collector recipient
total_time	integer	Total time in seconds spent on the survey
custom_value	string	Custom value associated with a response
edit_url	string	Weblink to the survey taking page to edit the response
analyze_url	string	Weblink to the analyze page to view the response
ip_address	string	IP Address the response was taken from
custom_variables	object	Values to any available custom variables in the survey
response_status	string	Status of the response: completed, partial, overquota, or disqualified
collection_mode	string	The collection mode of the response: default, preview, data_entry, survey_preview, or edit
date_created	DateString	Date the response was created
date_modified	object	Date the response was last modified
pages	[object]	Pages from the survey and their associated responses
id	integer	The ID of the page with responses
questions	[object]	The questions on that page with responses
id	integer	ID of the question with responses
variable_id	integer	ID of the random assignment variable for the question
answers	[object]	The answers for the question with responses
choice_id	integer	The choice selected
row_id	integer	The row selected
col_id	integer	The column selected

Name	Type	Description
other_id	integer	The other text choice selected
text	string	Any open ended text
download_url	string	Download URL for <code>file_upload</code> questions. This url expires after about 12 hours.
human_download_url	string	Download URL for <code>file_upload</code> questions, intended for humans to use. This is a non-expiring link but requires logging in.
content_type	string	MIME type for <code>download_url</code>

Response Counts and Trends

The following endpoints let you view how many respondents answered a question a particular way and trends in how many respondents answered a question a particular way in a given time period, for example a week.

Tip: Trends are not available for file_upload, slider, presentation, demographic, matrix_menu, or datetime question types and trends object will return empty for these questions.

GET /surveys/{survey_id}/rollups

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys//rollups", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`GET /surveys/{id}/rollups`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns rollups for all questions in a survey. Public App users need access to the **View Response Details** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
collector_ids	string	false	Limits responses to those from specified collector ids
start_created_at	undefined	false	Limits responses to those created after date specified
end_created_at	undefined	false	Limits responses to those created before date specified
start_modified_at	undefined	false	Limits responses to those last modified after date specified. Date string in format YYYY-MM-DDTHH:MM:SS (no offset)

Name	Type	Required	Description
end_modified_at	undefined	false	Limits responses to those last modified before date specified. Date string in format YYYY-MM-DDTHH:MM:SS (no offset)
status	string	false	Limits responses to those of a certain status. Accepts: completed, partial, overquota, disqualifie d
email	string	false	Limits responses to those associated with specified email address. NOTE: Responses must come from an email collector.
first_name	string	false	Limits responses to those associated with specified first name. NOTE: Responses must come from an email collector.
last_name	string	false	Limits responses to those associated with specified last name. NOTE: Responses must come from an email collector.
ip	string	false	Limits responses to those associated with specified IP address
custom	string	false	Limits responses to those associated with specified value for custom field 1. NOTE: Responses must come from an email collector.
total_time_max	string	false	Limits responses to those that took respondents less time to complete than specified time
total_time_min	string	false	Limits responses to those that took respondents more time to complete than specified time
total_time_units	string	false	Determines which unit of time total_time_max and total_time_min use. Accepts: second, minute, hour

Example response for GET /surveys/{survey_id}/rollups

{}
{}

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	object	
subtype	string	Question subtype. See question types
href	string	Resource URL for a question's rollup
id	string	Question identifier
family	string	Question family. See question types
summary	object	
answered	integer	Returns the count of respondents who answered the question
skipped	integer	Returns the count of respondents who skipped the question
stats	object	Returns basic statistics for close ended questions
other_answered	integer	Returns the count of responses who answered the other choice, if available
choices	object	Returns question's choices with the id and count for each choice

GET /surveys/{survey_id}/pages/{page_id}/rollups

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/pages/1234/rollups", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`GET /surveys/{survey_id}/pages/{page_id}/rollups`

Same as `/surveys/{id}/rollups` but returns rollups for only the survey page specified.

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
start_modified_at	undefined	false	Date string in format YYYY-MM-DDTHH:MM:SS (no offset)
end_modified_at	undefined	false	Date string in format YYYY-MM-DDTHH:MM:SS (no offset)

GET /surveys/{survey_id}/pages/{page_id}/questions/{question_id}/rollups

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/pages/1234/questions/1234/rollups", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`GET /surveys/{survey_id}/pages/{page_id}/questions/{question_id}/rollups`

Same as `/surveys/{id}/rollups` but returns rollups for only the survey question specified.

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
start_modified_at	undefined	false	Date string in format YYYY-MM-DDTHH:MM:SS (no offset)
end_modified_at	undefined	false	Date string in format YYYY-MM-DDTHH:MM:SS (no offset)

GET /surveys/{survey_id}/trends

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys//trends", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

GET /surveys/{id}/trends

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns answer counts for a particular time periods. Public App users need access to the **View Response Details** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
first_respondent	string	false	Limits responses to those completed after the specified respondent id
last_respondent	string	false	Limits responses to those completed before the specified respondent id
trend_by	string	false	Sets the period of time trends are broken into. Accepts: <code>year</code> , <code>quarter</code> , <code>month</code> , <code>week</code> , <code>day</code> , <code>hour</code>
start_modified_at	undefined	false	Date string in format YYYY-MM-DDTHH:MM:SS (no offset)
end_modified_at	undefined	false	Date string in format YYYY-MM-DDTHH:MM:SS (no offset)

Example response for GET /surveys/{survey_id}/trends

```
{}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	object	
id	string	Trend id
href	string	Response API URL
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	

GET /surveys/{survey_id}/pages/{page_id}/trends

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/pages/1234/trends", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

GET /surveys/{survey_id}/pages/{page_id}/trends

Same as /surveys/{id}/trends but returns trends for only the survey page specified.

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
start_modified_at	undefined	false	Date string in format YYYY-MM-DDTHH:MM:SS (no offset)
end_modified_at	undefined	false	Date string in format YYYY-MM-DDTHH:MM:SS (no offset)

GET /surveys/{survey_id}/pages/{page_id}/questions/{question_id}/trends

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/pages/1234/questions/1234/trends", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

GET /surveys/{survey_id}/pages/{page_id}/questions/{question_id}/trends

Same as /surveys/{id}/trends but returns trends for only the survey question specified.

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
start_modified_at	undefined	false	Date string in format YYYY-MM-DDTHH:MM:SS (no offset)
end_modified_at	undefined	false	Date string in format YYYY-MM-DDTHH:MM:SS (no offset)

Webhooks

Create webhooks that subscribe to various events in SurveyMonkey. You can create a webhook to POST a callback when:

A survey response is completed ('response_completed')
A survey response is disqualified ('response_disqualified')
A survey response is updated ('response_updated') - one webhook event is fired for each page of the survey
A respondent begins a survey ('response_created')
A response is deleted ('response_deleted')
A response is over a survey's quota ('response_overquota')
A survey is created ('survey_created')
A survey is updated ('survey_updated')
A survey is deleted ('survey_deleted')
A collector is created ('collector_created')
A collector is updated ('collector_updated')
A collector is deleted ('collector_deleted')
An app is installed ('app_installed')
An app is uninstalled ('app_uninstalled')

GET /webhooks

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/webhooks", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /webhooks

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns a list of webhooks. Public App users need access to the **View Webhooks** scope

POST: Create a webhook, see below for callback format. Public App users need access to the **Create/Modify Webhooks** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
per_page	string	false	Number of resources to return per page. Defaults to 50
event_type	string	false	Filter the webhook via the event type

ENUMERATED VALUES

Parameter	Value
event_type	response_completed
event_type	response_updated
event_type	response_disqualified
event_type	response_created
event_type	response_deleted
event_type	response_overquota
event_type	collector_created

Parameter	Value
event_type	collector_updated
event_type	collector_deleted
event_type	survey_created
event_type	survey_updated
event_type	survey_deleted
event_type	app_installed
event_type	app_uninstalled

Example response for GET /webhooks

```
{
  "data": [
    {
      "id": "1234",
      "name": "My Webhook",
      "href": "https://api.surveymonkey.com/v3/webhooks/123"
    }
  ],
  "per_page": 50,
  "page": 1,
  "total": 1,
  "links": {
    "self": "https://api.surveymonkey.com/v3/webhooks?page=1&per_page=50"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[object]	Information about each webhook
id	string	Webhook id
name	string	Webhook Name
href	string	Webhook API URL
per_page	integer	Number of resources returned per page
page	integer	Page of the resource returned
total	integer	Number of shared resources returned
links	object	Contains this endpoint's links
self	string	The link associated with this endpoint

POST /webhooks

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"name\":\"My Webhook\",\"subscription_url\":\"https://surveymonkey.com/webhook_reciever\",\"authorization\":\"xyz\",\"verify_ssl\":true,\"event_type\":\"response_compl
```

```

eted\", \"object_type\": \"survey\", \"object_ids\":[\"1234\", \"5678\"]}]"
headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/webhooks", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

POST /webhooks

AVAILABLE METHODS

POST: Create a webhook, see below for callback format. Public App users need access to the **Create/Modify Webhooks** scope

Body parameter

```
{
  "name": "My Webhook",
  "subscription_url": "https://surveymonkey.com/webhook_reciever",
  "authorization": "xyz",
  "verify_ssl": true,
  "event_type": "response_completed",
  "object_type": "survey",
  "object_ids": [
    "1234",
    "5678"
  ]
}
```

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
bypass_ping	boolean	false	Ignores ping of the subscription url. Default is False

REQUEST BODY FOR POST /WEBHOOKS

Name	Type	Required	Description
name	string	true	Webhook name
subscription_url	string	true	Subscription url that events are sent to. This url value must be unique to each webhook you create and is not allowed to be associated with any other webhooks in your account, regardless of event type specified. Url must accept a HEAD request and return a 200.
authorization	string	false	Authorization header to pass when events are sent
verify_ssl	boolean	false	Verify the connection. Defaults to True
event_type	string	true	Event type that the webhook listens to: <code>response_completed</code> , <code>response_updated</code> , <code>response_disqualified</code> , <code>response_created</code> , <code>response_deleted</code> , <code>response_overquota</code> , <code>collector_created</code> , <code>collector_updated</code> , <code>collector_deleted</code> , <code>survey_created</code> , <code>survey_updated</code> , <code>survey_deleted</code> , <code>app_installed</code> , <code>app_uninstalled</code>
object_type	string	false	Object type to filter events by: <code>survey</code> , <code>collector</code> , <code>app</code> . NOTES: If <code>event_type</code> is <code>collector_created</code> , do not set <code>object_type</code> to <code>collector</code> . <code>survey_created</code> event type should not have an <code>object_type</code> , and all other event types should have an <code>object_type</code> . <code>survey_updated</code> and <code>survey_deleted</code> event types should have <code>object_type</code> be <code>survey</code> . <code>app</code> should only be used for <code>app_installed</code> and <code>app_uninstalled</code>
object_ids	[string]	false	Object ids of the object type to filter events by. For example, if the object type is <code>survey</code> , survey ids should be in this field. If <code>object_type</code> is defined, <code>object_ids</code> must be defined.

Example response for POST /webhooks

```
{
  "id": "1234",
  "name": "My Webhook",
```

```

    "event_type": "response_completed",
    "object_type": "survey",
    "object_ids": [
        "1234",
        "5678"
    ],
    "subscription_url": "https://surveymonkey.com/webhook_reciever",
    "href": "https://api.surveymonkey.com/v3/webhooks/123"
}

```

RESPONSE SCHEMA

Status Code 201

Name	Type	Description
id	string	Webhook id
name	string	Webhook name
event_type	string	Event type that the webhook listens to: <code>response_completed</code> , <code>response_updated</code> , <code>response_disqualified</code> , <code>response_created</code> , <code>response_deleted</code> , <code>response_overquota</code> , <code>collector_created</code> , <code>collector_updated</code> , <code>collector_deleted</code> , <code>survey_created</code> , <code>survey_updated</code> , <code>survey_deleted</code> , <code>app_installed</code> , <code>app_uninstalled</code>
object_type	string	Object type to filter events by: <code>survey</code> , <code>collector</code> , <code>app</code>
object_ids	[string]	Object ids of the object type to filter events by. For example, if the object type is <code>survey</code> , survey ids should be in this field.
subscription_url	string	Subscription url that callback events are sent to
authorization	string	Authorization header to pass when events are sent
href	string	Resource API URL

GET /webhooks/{webhook_id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/webhooks/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /webhooks/{webhook_id}`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns a webhook. Public App users need access to the **View Webhooks** scope

`PATCH`: Modifies a webhook (updates any fields accepted as arguments to `POST` /webhooks, except `app_installed` and `app_uninstalled` event types).

Public App users need access to the **Create/Modify Webhooks** scope

`PUT`: Replaces a webhook (same arguments and requirements as `POST` /webhooks). Public App users need access to the **Create/Modify Webhooks** scope

`DELETE`: Deletes a webhook. Public App users need access to the **Create/Modify Webhooks** scope

Example response for GET /webhooks/{webhook_id}

```
{
  "id": "1234",
  "name": "My Webhook",
  "event_type": "response_completed",
  "object_type": "survey",
  "object_ids": [
    "1234",
    "5678"
  ],
  "subscription_url": "https://surveymonkey.com/webhook_reciever",
  "href": "https://api.surveymonkey.com/v3/webhooks/123"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Webhook id
name	string	Webhook name
event_type	string	Event type that the webhook listens to: <code>response_completed</code> , <code>response_updated</code> , <code>response_disqualified</code> , <code>response_created</code> , <code>response_deleted</code> , <code>response_overquota</code> , <code>collector_created</code> , <code>collector_updated</code> , <code>collector_deleted</code> , <code>survey_created</code> , <code>survey_updated</code> , <code>survey_deleted</code> , <code>app_installed</code> , <code>app_uninstalled</code>
object_type	string	Object type to filter events by: <code>survey</code> , <code>collector</code> , <code>app</code>
object_ids	[string]	Object ids of the object type to filter events by. For example, if the object type is <code>survey</code> , survey ids should be in this field.
subscription_url	string	Subscription url that callback events are sent to
authorization	string	Authorization header to pass when events are sent
href	string	Resource API URL

PATCH /webhooks/{webhook_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"name\":\"My Webhook\",\"subscription_url\":\"https://surveymonkey.com/webhook_reciever\",\"authorization\":\"xyz\",\"verify_ssl\":true,\"event_type\":\"response_completed\",\"object_type\":\"survey\",\"object_ids\":[\"1234\",\"5678\"]}"

headers = {
  'Content-Type': "application/json",
  'Accept': "application/json",
  'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/v3/webhooks/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`PATCH /webhooks/{webhook_id}`

`PATCH`: Modifies a webhook (updates any fields accepted as arguments to `POST`/webhooks, except `app_installed` and `app_uninstalled` event types). Public App users need access to the **Create/Modify Webhooks** scope

Body parameter

```
{
  "name": "My Webhook",
  "subscription_url": "https://surveymonkey.com/webhook_reciever",
  "authorization": "xyz",
  "verify_ssl": true,
  "event_type": "response_completed",
  "object_type": "survey",
  "object_ids": [
    "1234",
    "5678"
  ]
}
```

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
bypass_ping	boolean	false	Ignores ping of the subscription url. Default is False

REQUEST BODY FOR PATCH /WEBHOOKS

Name	Type	Required	Description
name	string	false	Webhook name
subscription_url	string	false	Subscription url that events are sent to. This url value must be unique to each webhook you create and is not allowed to be associated with any other webhooks in your account, regardless of event type specified. Url must accept a HEAD request and return a 200.
authorization	string	false	Authorization header to pass when events are sent
verify_ssl	boolean	false	Verify the connection. Defaults to True
event_type	string	false	Event type that the webhook listens to: response_completed, response_updated, response_disqualified, response_created, response_deleted, response_overquota, collector_created, collector_updated, collector_deleted, survey_created, survey_updated, survey_deleted
object_type	string	false	Object type to filter events by: survey. NOTES: If event_type is collector_created, do not set object_type to collector. survey_updated and survey_deleted event types should have object_type be survey. survey_created should not have an object_type defined.
object_ids	[string]	false	Object ids of the object type to filter events by. For example, if the object type is survey, survey ids should be in this field.

PUT /webhooks/{webhook_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"name\":\"My Webhook\",\"subscription_url\":\"https://surveymonkey.com/webhook_reciever\",\"authorization\":\"xyz\",\"verify_ssl\":true,\"event_type\":\"response_completed\",\"object_type\":\"survey\",\"object_ids\":[\"1234\",\"5678\"]}"

headers = {
  'Content-Type': "application/json",
  'Accept': "application/json",
  'Authorization': "Bearer {access-token}"
}

conn.request("PUT", "/v3/webhooks/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`PUT /webhooks/{webhook_id}`

`PUT`: Replaces a webhook (same arguments and requirements as `POST` /webhooks). Public App users need access to the **Create/Modify Webhooks** scope

Body parameter

```
{  
  "name": "My Webhook",  
  "subscription_url": "https://surveymonkey.com/webhook_reciever",  
  "authorization": "xyz",  
  "verify_ssl": true,  
  "event_type": "response_completed",  
  "object_type": "survey",  
  "object_ids": [  
    "1234",  
    "5678"  
  ]  
}
```

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
bypass_ping	boolean	false	Ignores ping of the subscription url. Default is False

Webhook Callbacks

Our webhook callbacks are sent as POST requests and include the following request headers that can be used to verify the request is coming from SurveyMonkey:

Sm-Apikey: Your Client ID or API key (if you're using OLD Authentication)

Sm-Signature: # HMAC digest hashed with sha1

Verifying the request

```
import hashlib  
import hmac  
import base64  
  
def generate_signature(undecodedPayload, api_key, api_secret):  
    """  
    :type undecodedPayload: str The response body from the webhook exactly as you received it  
    :type api_key: str Your API Key for your app (if you have one) otherwise your Client ID  
    :type api_secret: str Your API Secret for your app  
    :return: str  
    """  
    signature = hmac.new(  
        key='%(api_key)s%(api_secret)s' % (api_key, api_secret).encode('utf-8'),  
        msg=undecodedPayload.encode('utf-8'),  
        digestmod=hashlib.sha1)  
  
    signature_digest = signature.digest()  
  
    return base64.b64encode(signature_digest).decode('utf-8')  
  
# Compare the signature generated from the request body against the one in the request headers where  
# `request` is the HTTP request received for the event. Attributes to access body/headers may vary by framework  
hmac.compare_digest(generate_signature(request.body, api_key, api_secret), request.headers['Sm-Signature'])
```

Example Collector Event Data

```
{  
  "name": "My Webhook",  
  "filter_type": "survey",  
  "filter_id": "123456789",  
  "event_type": "response_completed",  
  "event_id": "123456789",  
  "object_type": "response",  
  "object_id": "123456",  
  "timestamp": "2018-01-12T12:00:00Z",  
  "version": 1  
}
```

```

"event_datetime": "2016-01-01T21:56:31.182613+00:00",
"resources": {
  "collector_id": "123456789",
  "survey_id": "123456789",
  "user_id": "123456789"
}
}

```

Example Response Event Data

```

{
  "name": "My Webhook",
  "filter_type": "collector",
  "filter_id": "123456789",
  "event_type": "response_completed",
  "event_id": "123456789",
  "object_type": "response",
  "object_id": "123456",
  "event_datetime": "2016-01-01T21:56:31.182613+00:00",
  "resources": {
    "respondent_id": "123456789",
    "recipient_id": "123456789",
    "collector_id": "123456789",
    "survey_id": "123456789",
    "user_id": "123456789"
  }
}

```

CALLBACK EVENT

Name	Description	Data Type
name	Webhook name	String
filter_type	Which kind of object the webhook set to filter events by: <code>survey</code> or <code>collector</code>	String-ENUM
filter_id	The id of the object triggering the event	String
event_type	Event type that the webhook listens to: <code>response_completed</code> , <code>response_disqualified</code> , or <code>response_updated</code>	String-ENUM
event_id	Event id	String
object_type	Type of object that event occurred for	String
object_id	id of object that event occurred for	String
event_datetime	ISO 8601 string of date/time that the event occurred	String
resources	Ids associated with the event. Depending on the webhook configuration can include: <code>respondent_id</code> , <code>recipient_id</code> , <code>collector_id</code> , <code>survey_id</code> , and <code>user_id</code>	Object

Benchmarks

Benchmark data comes from SurveyMonkey customers who used Question Bank questions in their surveys. Organizations of all shapes and sizes use Question Bank, like local coffee shops, schools, and Fortune 500 companies. Benchmarks give context to your survey results by allowing you to compare your results to others who used the same Question Bank questions as you.

- ⓘ Benchmarks are available as an add on only for Private apps. If you are interested in benchmarks please contact: benchmarksales@surveymonkey.com.

GET /benchmark_bundles

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/benchmark_bundles", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /benchmark_bundles

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods

GET: Returns a list of benchmark bundles you have access to

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
country	integer	false	Country locale, this is set to “US” by default

Example response for GET /benchmark_bundles

```
{
  "per_page": 5,
  "total": 5,
  "data": [
    {
      "href": "https://api.surveymonkey.com/v3/benchmark_bundles/nps_education",
      "id": "nps_education",
      "title": "All Education"
    },
    {
      "href": "https://api.surveymonkey.com/v3/benchmark_bundles/nps_higher_education",
      "id": "nps_higher_education",
      "title": "Higher Education"
    },
    {
      "href": "https://api.surveymonkey.com/v3/benchmark_bundles/nps_schools",
      "id": "nps_schools",
      "title": "K-12 Schools"
    },
    {
      "href": "https://api.surveymonkey.com/v3/benchmark_bundles/nps_sports_instruction",
      "id": "nps_sports_instruction",
      "title": "Sports Instruction"
    },
    {
      "href": "https://api.surveymonkey.com/v3/benchmark_bundles/nps_health_human_services",
      "id": "nps_health_human_services",
      "title": "Health and Human Services"
    }
  ],
  "page": 1,
  "links": {
    "self": "https://api.surveymonkey.com/v3/benchmark_bundles?page=1&per_page=5"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[allOf]	
id	string	Benchmark id
name	string	Benchmark Name
href	string	Benchmark API URL
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	

GET /benchmark_bundles/{bundle_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/benchmark_bundles/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /benchmark_bundles/{bundle_id}](#)

AVAILABLE METHODS

[HEAD](#): Checks if resource is available

[OPTIONS](#): Returns available methods and options

[GET](#): Returns the questions and details included in a given bundle

Example response for GET /benchmark_bundles/{bundle_id}

```
{
  "title": "Consumer Goods and Services",
  "details": {
    "bullets": [
      "Includes retail companies, restaurants, and other consumer based industries",
      "2,600 responses per question (average)"
    ],
    "subtext": "Responses from 40+ organizations*"
  },
  "questions": [
    {
      "answer_order": "normal",
      "id": "25651",
      "answers": {
        "choices": [
          {
            "label": "A company that sells products or services directly to consumers"
          }
        ]
      }
    }
  ]
}
```

```

        "text": "Strongly Disagree"
    },
    {
        "text": "Disagree"
    },
    {
        "text": "Neutral/Neither agree nor disagree"
    },
    {
        "text": "Agree"
    },
    {
        "text": "Strongly Agree"
    }
]
},
{
    "title": "My organization is dedicated to diversity and inclusiveness."
}
],
{
    "id": "test_bundle"
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Benchmark Bundle id
title	string	Benchmark Bundle name
details	object	Contains bullets array for description bullets, and a subtext string.
questions	array	List of questions associated with the bundle
href	string	Resource API URL

GET /benchmark_bundles/{id}/analyze

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/benchmark_bundles/1234/analyze", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /benchmark_bundles/{bundle_id}/analyze`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns the benchmark for the given bundle

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
question_ids	string	false	List of questions for which to receive analytics.
percentile_start (optional)	integer	false	Starting percentile to filter by (default=0).
percentile_end (optional)	integer	false	Ending percentile to filter by (default=100).

Example response for GET /benchmark_bundles/{id}

```
{
  "date_end": "2015-12-31T00:00:00+00:00",
  "date_start": "2015-01-01T00:00:00+00:00",
  "id": "test_bundle",
  "benchmarks": [
    {
      "num_data_points": 50,
      "distribution_values": {
        "1": 0.0333333333,
        "2": 0.0555555555,
        "3": 0.2222222222,
        "4": 0.4444444444,
        "5": 0.2222222222
      },
      "question": {
        "answer_order": "normal",
        "id": "25651",
        "answers": {
          "choices": [
            {
              "text": "Strongly Disagree"
            },
            {
              "text": "Disagree"
            },
            {
              "text": "Neutral/Neither agree nor disagree"
            },
            {
              "text": "Agree"
            },
            {
              "text": "Strongly Agree"
            }
          ]
        },
        "title": "My organization is dedicated to diversity and inclusiveness."
      },
      "num_responses": 1000,
      "quartile_values": {
        "0": 1.9,
        "25": 8.2,
        "50": 5.211,
        "75": 2.011,
        "100": 5.4343
      }
    }
  ]
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Benchmark Bundle id
date_start	string	Start date for included responses in the benchmark
date_end	string	End date for included responses in the benchmark
benchmarks	array	List of benchmark results for each question in the bundle, with the question details

GET /surveys/{survey_id}/pages/{page_id}/questions/{question_id}/benchmark

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/surveys/1234/pages/1234/questions/1234/benchmark", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /surveys/{survey_id}/pages/{page_id}/questions/{question_id}/benchmark

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods

GET: Returns the benchmark for the given question

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
percentile_start (optional)	integer	false	Starting percentile to filter by (default=0).
percentile_end (optional)	integer	false	Ending percentile to filter by (default=100).

Example response for GET /surveys/{survey_id}/pages/{page_id}/questions/{question_id}/benchmark

```
{
  "date_end": "2015-12-31T00:00:00+00:00",
  "date_start": "2015-01-01T00:00:00+00:00",
  "benchmark": {
    "num_responses": 12921174,
    "num_data_points": 76147,
    "quartile_values": {
      "0": -100,
      "25": 2.8,
      "50": 40,
      "75": 67.6,
      "100": 100
    },
    "distribution_values": {
      "0": 0.2603693321,
      "100": 0.5344724075,
      "-100": 0.2051582605
    },
    "data_point_type_distribution": {
      "respondent_data_points": 0,
      "entity_data_points": 100
    }
  },
  "id": "327620251"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Benchmark question id

Name	Type	Description
date_start	string	Start date for included responses in the benchmark
date_end	string	End date for included responses in the benchmark
benchmark	object	Benchmark result for the question

Organizations

Use these endpoints to get information about an organization, including roles and workgroups in your team account.

GET /workgroups/{id}/members

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/workgroups/1234/members", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /workgroups/{id}/members

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns the members in a workgroup. Public App users need access to the **View Workgroup Members** scope

POST: Creates a new workgroup member. Public App users need access to the **Create/Modify Workgroup Members** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
per_page	string	false	Number of resources to return per page

Example response for GET /workgroups/{id}/members

```
{
  "per_page": 50,
  "total": 1,
  "data": [
    {
      "workgroup_id": "71d9d408d1914c9ca85fffcda8330d675",
      "status": "active",
      "is_workgroup_owner": true,
      "created_at": "2018-04-21T21:21:46",
      "updated_at": "2018-04-21T21:21:46",
      "role_assignment_id": "a1af2174db7c40c796f3b069d7efbc63",
      "id": "1234567"
    }
  ]
}
```

```

],
"page": 1,
"links": {
  "self": "https://api.surveymonkey.com/v3/workgroups/71d9d408d1914c9ca85ffcda8330d675/members?page=1&per_page=50"
}
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	
data	object	
id	string	The user ID of the member
workgroup_id	string	ID of the member's workgroup
is_workgroup_owner	boolean	Whether the member is an owner of the workgroup
role_assignment_id	string	An ID referencing the assigned role of the member
status	string	The status of the member, either “active” or “pending”
created_at	string	Datetime when the member was created
updated_at	string	Datetime when the member was last updated

POST /workgroups/{id}/members

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"user_id\": \"1234\", \"is_workgroup_owner\": false}"

headers = {
  'Content-Type': "application/json",
  'Accept': "application/json",
  'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/workgroups//members", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

POST /workgroups/{id}/members

Body parameter

```
{
  "user_id": "1234",
}
```

```
        "is_workgroup_owner": false
    }
```

REQUEST BODY FOR POST /WORKGROUPS/{ID}/MEMBERS

Name	Type	Required	Description
user_id	string	true	The user ID of the new member to add
is_workgroup_owner	boolean	true	Whether the new member will be an owner of the workgroup

Example response for POST /workgroups/{id}/members

```
{
    "workgroup_id": "71d9d408d1914c9ca85ffcd8330d675",
    "status": "active",
    "is_workgroup_owner": true,
    "created_at": "2018-04-21T21:21:46",
    "updated_at": "2018-04-21T21:21:46",
    "role_assignment_id": "a1af2174db7c40c796f3b069d7efbc63",
    "id": "1234567"
}
```

RESPONSE SCHEMA

Status Code 201

Name	Type	Description
id	string	The user ID of the member
workgroup_id	string	ID of the member's workgroup
is_workgroup_owner	boolean	Whether the member is an owner of the workgroup
role_assignment_id	string	An ID referencing the assigned role of the member
status	string	The status of the member, either "active" or "pending"
created_at	string	Datetime when the member was created
updated_at	string	Datetime when the member was last updated

POST /workgroups/{id}/members/bulk

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"members\":[{\"user_id\":\"1234\", \"is_workgroup_owner\":false}]}""

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/workgroups/1234/members/bulk", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

POST /workgroups/{id}/members/bulk

AVAILABLE METHODS

- HEAD**: Checks if resource is available
- OPTIONS**: Returns available methods and options
- POST**: Creates many new workgroup members. Public App users need access to the **Create/Modify Workgroup Members** scope

Body parameter

```
{  
  "members": [  
    {  
      "user_id": "1234",  
      "is_workgroup_owner": false  
    }  
  ]  
}
```

REQUEST BODY FOR POST /WORKGROUPS/{ID}/MEMBERS/BULK

Name	Type	Required	Description
members	[object]	true	List of members to add
user_id	string	true	The user ID of the new member to add
is_workgroup_owner	boolean	true	Whether the new member will be an owner of the workgroup

Example response for POST /workgroups/{id}/members/bulk

```
{  
  "per_page": 50,  
  "total": 1,  
  "data": [  
    {  
      "workgroup_id": "71d9d408d1914c9ca85ffcd8330d675",  
      "status": "active",  
      "is_workgroup_owner": true,  
      "created_at": "2018-04-21T21:21:46",  
      "updated_at": "2018-04-21T21:21:46",  
      "role_assignment_id": "a1af2174db7c40c796f3b069d7efbc63",  
      "id": "1234567"  
    }  
  ],  
  "page": 1,  
  "links": {  
    "self": "https://api.surveymonkey.com/v3/workgroups/71d9d408d1914c9ca85ffcd8330d675/members?page=1&per_page=50"  
  }  
}
```

RESPONSE SCHEMA

Status Code 201

Name	Type	Description
links	object	
self	string	
page	integer	
per_page	integer	
total	integer	
data	object	
id	string	The user ID of the member
workgroup_id	string	ID of the member's workgroup
is_workgroup_owner	boolean	Whether the member is an owner of the workgroup

Name	Type	Description
role_assignment_id	string	An ID referencing the assigned role of the member
status	string	The status of the member, either “active” or “pending”
created_at	string	Datetime when the member was created
updated_at	string	Datetime when the member was last updated

GET /workgroups/{workgroup_id}/members/{member_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/workgroups/1234/members/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /workgroups/{workgroup_id}/members/{member_id}

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns a member. Public App users need access to the **View Workgroup Members** scope

PATCH: Modifies a member (updates any fields accepted as arguments to **POST** /workgroups/{workgroup_id}/members). Public App users need access to the **Create/Modify Workgroup Members** scope

DELETE: Deletes a member from a workgroup. Public App users need access to the **Create/Modify Workgroup Members** scope

Example response for GET /workgroups/{workgroup_id}/members/{member_id}

```
{
  "workgroup_id": "71d9d408d1914c9ca85ffcd8330d675",
  "status": "active",
  "is_workgroup_owner": true,
  "created_at": "2018-04-21T21:21:46",
  "updated_at": "2018-04-21T21:21:46",
  "role_assignment_id": "a1af2174db7c40c796f3b069d7efbc63",
  "id": "1234567"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	The user ID of the member
workgroup_id	string	ID of the member's workgroup
is_workgroup_owner	boolean	Whether the member is an owner of the workgroup
role_assignment_id	string	An ID referencing the assigned role of the member
status	string	The status of the member, either “active” or “pending”

Name	Type	Description
created_at	string	Datetime when the member was created
updated_at	string	Datetime when the member was last updated

PATCH /workgroups/{workgroup_id}/members/{member_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"user_id\": \"1234\", \"is_workgroup_owner\": false}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/v3/workgroups/1234/members/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

PATCH /workgroups/{workgroup_id}/members/{member_id}

Body parameter

```
{
  "user_id": "1234",
  "is_workgroup_owner": false
}
```

REQUEST BODY FOR PATCH /WORKGROUPS/{WORKGROUP_ID}/MEMBERS/{MEMBER_ID}

Name	Type	Required	Description
user_id	string	true	The user ID of the new member to add
is_workgroup_owner	boolean	true	Whether the new member will be an owner of the workgroup

Example response for PATCH /workgroups/{workgroup_id}/members/{member_id}

```
{
  "workgroup_id": "71d9d408d1914c9ca85ffcd8330d675",
  "status": "active",
  "is_workgroup_owner": true,
  "created_at": "2018-04-21T21:21:46",
  "updated_at": "2018-04-21T21:21:46",
  "role_assignment_id": "a1af2174db7c40c796f3b069d7efbc63",
  "id": "1234567"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	The user ID of the member

Name	Type	Description
workgroup_id	string	ID of the member's workgroup
is_workgroup_owner	boolean	Whether the member is an owner of the workgroup
role_assignment_id	string	An ID referencing the assigned role of the member
status	string	The status of the member, either "active" or "pending"
created_at	string	Datetime when the member was created
updated_at	string	Datetime when the member was last updated

GET /organizations

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/organizations", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /organizations](#)

AVAILABLE METHODS

[HEAD](#): Checks if resource is available

[OPTIONS](#): Returns available methods and options

[GET](#): Returns properties of your organizations (group, team) that pertain to workgroups functionality. Public App users need access to the **View Workgroups** scope

[POST](#): Initializes workgroups functionality for your organization (group, team). Only needs to be done once (and is done implicitly the first time a workgroup is created on surveymonkey.com). Public App users need access to the **Create/Modify Workgroups** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
per_page	string	false	Number of resources to return per page. Defaults to 50
page	string	false	Which page of resources to return. Defaults to 1

Example response for GET /organizations

```
{
  "data": [
    {
      "id": "830279",
      "default_role_id": "489432b6e47e41958af16757787a36a8",
      "users_can_create_workgroups": false,
      "updated_at": "2023-01-23T22:15:15"
    }
  ],
  "per_page": 50,
  "page": 1,
  "total": 1,
  "links": {
    "self": "https://api.surveymonkey.com/v3/organizations?per_page=50&page=1"
  }
}
```

```
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[object]	Contains the organizations and their information
id	string	ID of your organization (group, team)
default_role_id	string	A reference to the default role which gets applied as the default for all new workgroups.
users_can_create_workgroups	boolean	Indicates whether non-admin users in the organization can create and manage their own workgroups.
updated_at	string	Date and time when the organization record was last updated.
per_page	integer	Number of resources returned per page
page	integer	Page of the resource returned
total	integer	Number of shared resources returned
links	object	Contains this endpoint's links
self	string	The link associated with this endpoint

POST /organizations

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/v3/organizations", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`POST /organizations`

`POST`: Initializes workgroups functionality for your organization (group, team). Only needs to be done once (and is done implicitly the first time a workgroup is created on surveymonkey.com). Public App users need access to the **Create/Modify Workgroups** scope

Body parameter

```
{}
```

Example response for POST /organizations

```
{
  "id": "830279",
  "default_role_id": "489432b6e47e41958af16757787a36a8",
```

```

    "users_can_create_workgroups": false,
    "updated_at": "2023-01-23T22:15:15"
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	ID of your organization (group, team)
default_role_id	string	A reference to the default role which gets applied as the default for all new workgroups.
users_can_create_workgroups	boolean	Indicates whether non-admin users in the organization can create and manage their own workgroups.
updated_at	string	Datetime when the organization record was last updated.

GET /organization/{organization_id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/organizations/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

[GET /organizations/{organization_id}](#)

AVAILABLE METHODS

[HEAD](#): Checks if resource is available

[OPTIONS](#): Returns available methods and options

[GET](#): Returns properties of your organization you specify that pertain to workgroups functionality. Public App users need access to the **View Workgroups** scope

[PATCH](#): Modifies the workgroups functionality for your organization (group, team). Public App users need access to the **Create/Modify Workgroups** scope

Example response for GET /organizations

```
{
    "id": "830279",
    "default_role_id": "489432b6e47e41958af16757787a36a8",
    "users_can_create_workgroups": false,
    "updated_at": "2023-01-23T22:15:15"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	ID of your organization (group, team)
default_role_id	string	A reference to the default role which gets applied as the default for all new workgroups.

Name	Type	Description
users_can_create_workgroups	boolean	Indicates whether non-admin users in the organization can create and manage their own workgroups.
updated_at	string	Datetime when the organization record was last updated.

PATCH /organization/{organization_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"id\":\"string\",\"default_role_id\":\"string\",\"users_can_create_workgroups\":true,\"updated_at\":\"string\"}"

headers = {
    'Content-Type': "application/json",
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/v3/organizations/1234", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[PATCH /organizations/{organization_id}](#)

PATCH: Modifies the workgroups functionality for your organization (group, team). Public App users need access to the **Create/Modify Workgroups** scope

Body parameter

```
{
  "id": "string",
  "default_role_id": "string",
  "users_can_create_workgroups": true,
  "updated_at": "string"
}
```

REQUEST BODY FOR PATCH /ORGANIZATION/{ORGANIZATION_ID}

Name	Type	Required	Description
id	string	undefined	ID of your organization (group, team)
default_role_id	string	undefined	A reference to the default role which gets applied as the default for all new workgroups.
users_can_create_workgroups	boolean	undefined	Indicates whether non-admin users in the organization can create and manage their own workgroups.
updated_at	string	undefined	Datetime when the organization record was last updated.

Example response for GET /organizations

```
{
  "id": "830279",
  "default_role_id": "489432b6e47e41958af16757787a36a8",
  "users_can_create_workgroups": false,
  "updated_at": "2023-01-23T22:15:15"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	ID of your organization (group, team)
default_role_id	string	A reference to the default role which gets applied as the default for all new workgroups.
users_can_create_workgroups	boolean	Indicates whether non-admin users in the organization can create and manage their own workgroups.
updated_at	string	Datetime when the organization record was last updated.

GET /roles

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/roles", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /roles

AVAILABLE METHODS

HEAD: Checks if resource is available

OPTIONS: Returns available methods and options

GET: Returns the list of user roles in an organization. Public App users need access to the **View Roles** scope

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
per_page	string	false	Number of resources to return per page. Defaults to 50
page	string	false	Which page of resources to return. Defaults to 1

Example Response for GET /roles

```
{
  "per_page": 50,
  "total": 2,
  "data": [
    {
      "is_enabled": true,
      "created_at": "2017-04-13T19:57:40",
      "description": "",
      "privileges": [
        "design.read_only",
        "collect.read_only",
        "analyze.read_only"
      ],
      "updated_at": "2017-07-11T13:04:35",
      "is_system_role": true,
      "id": "a1af2174db7c40c796f3b069d7efbc63",
      "name": "Viewer"
    }
  ]
}
```

```

},
{
  "is_enabled": true,
  "created_at": "2017-04-13T19:57:40",
  "description": "",
  "privileges": [
    "design.full_access",
    "collect.full_access",
    "analyze.full_access"
  ],
  "updated_at": "2017-06-26T13:12:11",
  "is_system_role": true,
  "id": "731fc0d072de6426fba74ec7751aa6eab",
  "name": "Full Access"
}
],
"page": 1,
"links": {
  "self": "https://api.surveymonkey.com/v3/roles?page=1&per_page=50"
}
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[object]	Contains the shared resources with their information
is_enabled	boolean	Whether the role is active for an organization. Disabled roles grant no privileges and cannot be assigned to members"
created_at	string	Datetime when the role was created
description	string	Description of the returned role
privileges	[string]	An array of scoped privileges granted by the role. For example, the privilege “design.full_access” grants access to all actions within the design features of surveys
updated_at	string	
is_system_role	boolean	Indicates whether the role is a system-defined (as opposed to a custom one created within your organization)
id	string	ID of the returned role
name	string	Name of the returned role
per_page	integer	Number of resources returned per page
page	integer	Page of the resource returned
total	integer	Number of shared resources returned
links	object	Contains this endpoint’s links
self	string	The link associated with this endpoint

Errors

Get a list of errors or lookup an error by its id. See also error codes.

GET /errors

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/errors", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

`GET /errors`

AVAILABLE METHODS

`HEAD`: Checks if resource is available

`OPTIONS`: Returns available methods and options

`GET`: Returns a list of known errors

OPTIONAL QUERY STRINGS FOR GET

Name	Type	Required	Description
page	string	false	Which page of resources to return. Defaults to 1
per_page	string	false	Number of resources to return per page. Defaults to 50

Example response for `GET /errors`

```
{
  "data": [
    {
      "id": "1234",
      "message": "This is an error.",
      "href": "https://api.surveymonkey.com/v3/errors/1234",
      "name": "Known Error",
      "docs": "https://developer.surveymonkey.com/api/v3/#error-codes"
    }
  ],
  "per_page": 50,
  "page": 1,
  "total": 1,
  "links": {
    "self": "https://api.surveymonkey.com/v3/errors?page=1&per_page=50"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
data	[object]	A list of error types
id	string	Error id
message	string	Error explanation
href	string	Error API URL
name	string	Error name
docs	string	Error documentation page
per_page	integer	Number of resources returned per page
page	integer	Page of the resource returned

Name	Type	Description
total	integer	Number of errors returned
links	object	Contains this endpoint's links
self	string	The link associated with this endpoint

GET /errors/{error_id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/v3/errors/1234", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /errors/{error_id}](#)

AVAILABLE METHODS

[HEAD](#): Checks if resource is available

[OPTIONS](#): Returns available methods and options

[GET](#): Returns the details of a known error

Example response for GET /errors/{error_id}

```
{
  "id": "1234",
  "message": "This is an error.",
  "href": "https://api.surveymonkey.com/v3/errors/1234",
  "name": "Known Error",
  "docs": "https://developer.surveymonkey.com/api/v3/#error-codes"
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	Error id
message	string	Error explanation
href	string	Error API URL
name	string	Error name
docs	string	Error documentation page

Scroll down for code samples, example requests and responses. Select a language for code samples from the tabs above or the mobile navigation menu.

To get your SurveyMonkey account set up with a System for Cross-Domain Identity Management (SCIM) see Manage Accounts with SCIM. SCIM version 2.0 is supported.

1 To use SCIM API, you will need to generate a separate access token from the rest of the standard SurveyMonkey APIs. Follow Set up SCIM Provisioning for more details.

1 Accessing the SurveyMonkey SCIM API without one of our IDP integrations is not recommended.

Base URLs:

<https://api.surveymonkey.com/scim/v2>

Users

APIs to manage SCIM Users

GET /Users

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/scim+json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/scim/v2/Users", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /Users](#)

AVAILABLE METHODS

[GET](#): Returns a list of Users managed by SCIM

[POST](#): Creates a User

200 Response

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0>ListResponse"
  ],
  "totalResults": 100,
  "itemsPerPage": 10,
  "startIndex": 1,
  "Resources": [
    {
      "id": "c70b2780-a6c8-440f-8524-8ef975c13a94",
      "externalId": "johnsmith12345",
      "schemas": [
        [
          "urn:ietf:params:scim:schemas:core:2.0:User"
        ]
      ],
      "name": {
        "givenName": "John",
        "familyName": "Smith"
      },
      "active": true
    }
  ]
}
```

```

        "urn:ietf:params:scim:schemas:core:2.0:User",
        "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
    ],
    "name": {
        "familyName": "Smith",
        "givenName": "John"
    },
    "userName": "jsmith@example.com",
    "active": true,
    "emails": [
        {
            "primary": true,
            "type": "work",
            "value": "jsmith@example.com"
        }
    ],
    "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
        "employeeNumber": "12345",
        "costCenter": "001",
        "department": "IT",
        "division": "IT"
    },
    "meta": {
        "resourceType": "User",
        "created": "2021-06-12T16:25:00Z",
        "lastModified": "2021-06-12T16:25:00Z",
        "location": "https://api.surveymonkey.com/scim/v2/Users/c70b2780-a6c8-440f-8524-8ef975c13a94"
    }
}
]
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
schemas	[string]	A list of SCIM object Schemas present in the response
totalResults	integer	The total number of SCIM Users
itemsPerPage	integer	The number of SCIM Users returned in each page
startIndex	integer	The 1-based index of the returned page
Resources	array	A list of User objects
id	string	A unique identifier for the SCIM User. Defined by SurveyMonkey
externalId	string	A identifier for the SCIM User. Defined by provisioning client
schemas	[string]	A list of SCIM object Schemas present in the response
name	object	
familyName	string	The family name of the User
givenName	string	The given name of the User
userName	string	The username used to login for this user
active	boolean	A boolean value indicating the User's status
emails	[object]	
primary	boolean	A boolean value indicating if this email is the user's primary email address
type	string	Indicates if the email is "work", "home", or "other". Only emails with type "work" will be stored in SurveyMonkey
value	string	The email address for the User

Name	Type	Description
urn:ietf:params:scim:schemas:extension:enterprise:2.0:User	object	
employeeNumber	string	The employee number for the User
costCenter	string	The cost center identifier for the User
department	string	The department for the User
division	string	The division for the User. This maps to Divisions in SurveyMonkey. Read more here https://help.surveymonkey.com/articles/en_US/kb/divisions
meta	object	
resourceType	string	The type of resource that was returned. For the /Users api this will be "User"
created	string	The "DateTime" that the User was added
lastModified	string	The most recent "DateTime" that the User was updated
location	string	The URI of the User

POST /Users

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "{\"schemas\": [[\"urn:ietf:params:scim:schemas:core:2.0:User\"], \"urn:ietf:params:scim:schemas:extension:enterprise:2.0:User\"]], \"externalId\": \"johnsmith12345\", \"name\": {\"familyName\": \"Smith\", \"givenName\": \"John\"}, \"userName\": \"jsmith@example.com\", \"active\": true, \"emails\": [{\"primary\": true, \"type\": \"work\", \"value\": \"jsmith@example.com\"}], \"urn:ietf:params:scim:schemas:extension:enterprise:2.0:User\": {\"employeeNumber\": \"12345\", \"costCenter\": \"001\", \"department\": \"IT\", \"division\": \"IT\"}}"

headers = {
    'Content-Type': "application/scim+json",
    'Accept': "application/scim+json",
    'Authorization': "Bearer {access-token}"
}

conn.request("POST", "/scim/v2/Users", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

POST /Users

Body parameter

```
{
  "schemas": [
    [
      "urn:ietf:params:scim:schemas:core:2.0:User",
      "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
    ],
    {
      "externalId": "johnsmith12345",
      "name": {
        "familyName": "Smith",
        "givenName": "John"
      },
      "userName": "jsmith@example.com",
      "active": true,
      "emails": [
        {
          "primary": true,
          "type": "work",
          "value": "jsmith@example.com"
        }
      ]
    }
  ]
}
```

```
{
  "primary": true,
  "type": "work",
  "value": "jsmith@example.com"
}
],
{
  "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
    "employeeNumber": "12345",
    "costCenter": "001",
    "department": "IT",
    "division": "IT"
  }
}
```

REQUEST BODY FOR POST /USERS

Name	Type	Required	Description
schemas	[string]	true	A list of SCIM object Schemas present in the request
externalId	string	false	A identifier for the SCIM User. Defined by provisioning client
familyName	string	false	The family name of the User
givenName	string	false	The given name of the User
userName	string	true	The username used to login for this user
active	boolean	false	A boolean value indicating the User's status
primary	boolean	true	A boolean value indicating if this email is the user's primary email address
type	string	true	Indicates if the email is "work", "home", or "other". Only emails with type "work" will be stored in SurveyMonkey
value	string	true	The email address for the User
employeeNumber	string	false	The employee number for the User
costCenter	string	false	The cost center identifier for the User
department	string	false	The department for the User
division	string	false	The division for the User. This maps to Divisions in SurveyMonkey. Read more here https://help.surveymonkey.com/articles/en_US/kb/divisions

200 Response

```
{
  "id": "c70b2780-a6c8-440f-8524-8ef975c13a94",
  "externalId": "johnsmith12345",
  "schemas": [
    [
      "urn:ietf:params:scim:schemas:core:2.0:User",
      "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
    ]
  ],
  "name": {
    "familyName": "Smith",
    "givenName": "John"
  },
  "userName": "jsmith@example.com",
  "active": true,
  "emails": [
    {
      "primary": true,
      "type": "work",
      "value": "jsmith@example.com"
    }
  ],
  "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
    "employeeNumber": "12345",
    "costCenter": "001",
    "department": "IT",
    "division": "IT"
  }
}
```

```

        "division": "IT"
    },
    "meta": {
        "resourceType": "User",
        "created": "2021-06-12T16:25:00Z",
        "lastModified": "2021-06-12T16:25:00Z",
        "location": "https://api.surveymonkey.com/scim/v2/Users/c70b2780-a6c8-440f-8524-8ef975c13a94"
    }
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	A unique identifier for the SCIM User. Defined by SurveyMonkey
externalId	string	A identifier for the SCIM User. Defined by provisioning client
schemas	[string]	A list of SCIM object Schemas present in the response
name	object	
familyName	string	The family name of the User
givenName	string	The given name of the User
userName	string	The username used to login for this user
active	boolean	A boolean value indicating the User's status
emails	[object]	
primary	boolean	A boolean value indicating if this email is the user's primary email address
type	string	Indicates if the email is "work", "home", or "other". Only emails with type "work" will be stored in SurveyMonkey
value	string	The email address for the User
urn:ietf:params:scim:schemas:extension:enterprise:2.0:User	object	
employeeNumber	string	The employee number for the User
costCenter	string	The cost center identifier for the User
department	string	The department for the User
division	string	The division for the User. This maps to Divisions in SurveyMonkey. Read more here https://help.surveymonkey.com/articles/en_US/kb/divisions
meta	object	
resourceType	string	The type of resource that was returned. For the /Users api this will be "User"
created	string	The "DateTime" that the User was added
lastModified	string	The most recent "DateTime" that the User was updated
location	string	The URI of the User

GET /Users/{id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/scim+json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/scim/v2/Users/c70b2780-a6c8-440f-8524-8ef975c13a94", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))

```

GET /Users/{id}

AVAILABLE METHODS

- GET**: Returns a User
- PUT**: Replaces all provided attributes of a User
- DELETE**: Deletes a User
- PATCH**: Updates provided attributes of a User

200 Response

```
{
  "id": "c70b2780-a6c8-440f-8524-8ef975c13a94",
  "externalId": "johnsmith12345",
  "schemas": [
    [
      "urn:ietf:params:scim:schemas:core:2.0:User",
      "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
    ]
  ],
  "name": {
    "familyName": "Smith",
    "givenName": "John"
  },
  "user_name": "jsmith@example.com",
  "active": true,
  "emails": [
    {
      "primary": true,
      "type": "work",
      "value": "jsmith@example.com"
    }
  ],
  "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
    "employeeNumber": "12345",
    "costCenter": "001",
    "department": "IT",
    "division": "IT"
  },
  "meta": {
    "resourceType": "User",
    "created": "2021-06-12T16:25:00Z",
    "lastModified": "2021-06-12T16:25:00Z",
    "location": "https://api.surveymonkey.com/scim/v2/Users/c70b2780-a6c8-440f-8524-8ef975c13a94"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	A unique identifier for the SCIM User. Defined by SurveyMonkey
externalId	string	A identifier for the SCIM User. Defined by provisioning client
schemas	[string]	A list of SCIM object Schemas present in the response

Name	Type	Description
name	object	
familyName	string	The family name of the User
givenName	string	The given name of the User
userName	string	The username used to login for this user
active	boolean	A boolean value indicating the User's status
emails	[object]	
primary	boolean	A boolean value indicating if this email is the user's primary email address
type	string	Indicates if the email is "work", "home", or "other". Only emails with type "work" will be stored in SurveyMonkey
value	string	The email address for the User
urn:ietf:params:scim:schemas:extension:enterprise:2.0:User	object	
employeeNumber	string	The employee number for the User
costCenter	string	The cost center identifier for the User
department	string	The department for the User
division	string	The division for the User. This maps to Divisions in SurveyMonkey. Read more here https://help.surveymonkey.com/articles/en_US/kb/divisions
meta	object	
resourceType	string	The type of resource that was returned. For the /Users api this will be "User"
created	string	The "DateTime" that the User was added
lastModified	string	The most recent "DateTime" that the User was updated
location	string	The URI of the User

PUT /Users/{id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "null"

headers = {
    'Content-Type': "application/scim+json",
    'Accept': "application/scim+json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PUT", "/scim/v2/Users/c70b2780-a6c8-440f-8524-8ef975c13a94", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

`PUT /Users/{id}`

AVAILABLE METHODS

`PUT`: Replaces all provided attributes of a User

Body parameter

null

200 Response

```
{
  "id": "c70b2780-a6c8-440f-8524-8ef975c13a94",
  "externalId": "johnsmith12345",
  "schemas": [
    [
      "urn:ietf:params:scim:schemas:core:2.0:User",
      "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
    ]
  ],
  "name": {
    "familyName": "Smith",
    "givenName": "John"
  },
  "userName": "jsmith@example.com",
  "active": true,
  "emails": [
    {
      "primary": true,
      "type": "work",
      "value": "jsmith@example.com"
    }
  ],
  "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
    "employeeNumber": "12345",
    "costCenter": "001",
    "department": "IT",
    "division": "IT"
  },
  "meta": {
    "resourceType": "User",
    "created": "2021-06-12T16:25:00Z",
    "lastModified": "2021-06-12T16:25:00Z",
    "location": "https://api.surveymonkey.com/scim/v2/Users/c70b2780-a6c8-440f-8524-8ef975c13a94"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	A unique identifier for the SCIM User. Defined by SurveyMonkey
externalId	string	A identifier for the SCIM User. Defined by provisioning client
schemas	[string]	A list of SCIM object Schemas present in the response
name	object	
familyName	string	The family name of the User
givenName	string	The given name of the User
userName	string	The username used to login for this user
active	boolean	A boolean value indicating the User's status
emails	[object]	

Name	Type	Description
primary	boolean	A boolean value indicating if this email is the user's primary email address
type	string	Indicates if the email is "work", "home", or "other". Only emails with type "work" will be stored in SurveyMonkey
value	string	The email address for the User
urn:ietf:params:scim:schemas:extension:enterprise:2.0:User	object	
employeeNumber	string	The employee number for the User
costCenter	string	The cost center identifier for the User
department	string	The department for the User
division	string	The division for the User. This maps to Divisions in SurveyMonkey. Read more here https://help.surveymonkey.com/articles/en_US/kb/divisions
meta	object	
resourceType	string	The type of resource that was returned. For the /Users api this will be "User"
created	string	The "DateTime" that the User was added
lastModified	string	The most recent "DateTime" that the User was updated
location	string	The URI of the User

PATCH /Users/{id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

payload = "null"

headers = {
    'Content-Type': "application/scim+json",
    'Accept': "application/scim+json",
    'Authorization': "Bearer {access-token}"
}

conn.request("PATCH", "/scim/v2/Users/c70b2780-a6c8-440f-8524-8ef975c13a94", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[PATCH /Users/{id}](#)

AVAILABLE METHODS

[PATCH](#): Updates provided attributes of a User

Body parameter

null

200 Response

```
{
  "id": "c70b2780-a6c8-440f-8524-8ef975c13a94",
  "externalId": "johnsmith12345",
  "schemas": [
    [
      "urn:ietf:params:scim:schemas:core:2.0:User",
      "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
    ]
  ],
  "name": {
    "familyName": "Smith",
    "givenName": "John"
  },
  "userName": "jsmith@example.com",
  "active": true,
  "emails": [
    {
      "primary": true,
      "type": "work",
      "value": "jsmith@example.com"
    }
  ],
  "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
    "employeeNumber": "12345",
    "costCenter": "001",
    "department": "IT",
    "division": "IT"
  },
  "meta": {
    "resourceType": "User",
    "created": "2021-06-12T16:25:00Z",
    "lastModified": "2021-06-12T16:25:00Z",
    "location": "https://api.surveymonkey.com/scim/v2/Users/c70b2780-a6c8-440f-8524-8ef975c13a94"
  }
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	A unique identifier for the SCIM User. Defined by SurveyMonkey
externalId	string	A identifier for the SCIM User. Defined by provisioning client
schemas	[string]	A list of SCIM object Schemas present in the response
name	object	
familyName	string	The family name of the User
givenName	string	The given name of the User
userName	string	The username used to login for this user
active	boolean	A boolean value indicating the User's status
emails	[object]	
primary	boolean	A boolean value indicating if this email is the user's primary email address
type	string	Indicates if the email is "work", "home", or "other". Only emails with type "work" will be stored in SurveyMonkey
value	string	The email address for the User
urn:ietf:params:scim:schemas:extension:enterprise:2.0:User	object	
employeeNumber	string	The employee number for the User
costCenter	string	The cost center identifier for the User

Name	Type	Description
department	string	The department for the User
division	string	The division for the User. This maps to Divisions in SurveyMonkey. Read more here https://help.surveymonkey.com/articles/en_US/kb/divisions
meta	object	
resourceType	string	The type of resource that was returned. For the /Users api this will be "User"
created	string	The "DateTime" that the User was added
lastModified	string	The most recent "DateTime" that the User was updated
location	string	The URI of the User

Schemas

GET /Schemas

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/scim+json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/scim/v2/Schemas", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /Schemas](#)

AVAILABLE METHODS

[GET](#): Returns a list of Schemas supported by the SCIM API

200 Response

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0>ListResponse"
  ],
  "totalResults": 2,
  "itemsPerPage": 2,
  "startIndex": 1,
  "Resources": [
    {
      "id": "urn:ietf:params:scim:schemas:core:2.0:User",
      "name": "User",
      "description": "User Account",
      "attributes": [
        {
          "name": "externalId",
          "type": "string",
          "returned": "default",
          "required": false,
          "mutability": "readWrite",
          "maxLength": null,
          "minLength": null
        }
      ]
    }
  ]
}
```

```

        "caseExact": false
    }
],
"meta": {
    "resourceType": "Schema",
    "location": "https://api.surveymonkey.com/scim/v2/Schemas/urn:ietf:params:scim:schemas:core:2.0:User"
}
}
]
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
schemas	[string]	
totalResults	integer	Number of total results returned in the list
itemsPerPage	integer	Number of items returned per page in the response
startIndex	integer	Start index of the return page
Resources	array	
id	string	SCIM Schema identifier for a User object
name	string	Name of this SCIM object
description	string	Description of the SCIM User object
attributes	[object]	Attributes supported by the User object
name	string	Name of the attribute
type	string	Data type for the attribute
returned	string	Whether this attribute is returned by the SCIM API
required	boolean	Whether this attribute is required
mutability	string	Whether the attribute is read/write
caseExact	boolean	Whether exact case is required when used in queries
meta	object	
resourceType	string	Type of SCIM resource
location	string	URL to this resource

GET /Schemas/{id}

Code samples

```

import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/scim+json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/scim/v2/Schemas/urn%3Aietf%3Aparams%3Ascim%3Aschemas%3Acore%3A2.0%3AUser", headers=headers)

res = conn.getresponse()

```

```
data = res.read()  
  
print(data.decode("utf-8"))
```

GET /schemas/{id}

AVAILABLE METHODS

GET: Returns a Schema supported by the SCIM API

200 Response

```
{  
  "id": "urn:ietf:params:scim:schemas:core:2.0:User",  
  "name": "User",  
  "description": "User Account",  
  "attributes": [  
    {  
      "name": "externalId",  
      "type": "string",  
      "returned": "default",  
      "required": false,  
      "mutability": "readWrite",  
      "caseExact": false  
    }  
  ],  
  "meta": {  
    "resourceType": "Schema",  
    "location": "https://api.surveymonkey.com/scim/v2/Schemas/urn:ietf:params:scim:schemas:core:2.0:User"  
  }  
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
id	string	SCIM Schema identifier for a User object
name	string	Name of this SCIM object
description	string	Description of the SCIM User object
attributes	[object]	Attributes supported by the User object
name	string	Name of the attribute
type	string	Data type for the attribute
returned	string	Whether this attribute is returned by the SCIM API
required	boolean	Whether this attribute is required
mutability	string	Whether the attribute is read/write
caseExact	boolean	Whether exact case is required when used in queries
meta	object	
resourceType	string	Type of SCIM resource
location	string	URL to this resource

ResourceTypes

GET /ResourceTypes

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/scim+json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/scim/v2/ResourceTypes", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /ResourceTypes

AVAILABLE METHODS

GET: Returns a list of ResourceTypes supported by the SCIM API

200 Response

```
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0>ListResponse"
  ],
  "totalResults": 2,
  "itemsPerPage": 2,
  "startIndex": 1,
  "Resources": [
    {
      "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:ResourceType"
      ],
      "id": "User",
      "name": "User",
      "endpoint": "/User",
      "description": "User Account",
      "schema": "urn:ietf:params:scim:schemas:core:2.0:User",
      "schemaExtensions": [
        {
          "schema": "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User",
          "required": true
        }
      ],
      "meta": {
        "resourceType": "ResourceType",
        "location": "https://api.surveymonkey.com/scim/v2/ResourceTypes/User"
      }
    }
  ]
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
schemas	[string]	
totalResults	integer	Number of total results returned in the list
itemsPerPage	integer	Number of items returned per page in the response
startIndex	integer	Start index of the return page
Resources	array	
schemas	[string]	

Name	Type	Description
id	string	SCIM Resource Type identifier for a User object
name	string	Name of the SCIM Resource User
endpoint	string	Resource Type SCIM endpoint for the SCIM Resource User
description	string	Description of the SCIM Resource User
schema	string	SCIM Schema identifier for the User Resource
schemaExtensions	[object]	SCIM Schema extensions for the User Resource
schema	string	Schema Id of the supported SCIM extension for the User Resource
required	boolean	Whether the SCIM extension is required for the User Resource
meta	object	
resourceType	string	Type of SCIM resource
location	string	URL to this resource

GET /ResourceTypes/{id}

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/scim+json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/scim/v2/ResourceTypes/User", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

[GET /ResourceTypes/{id}](#)

AVAILABLE METHODS

[GET](#): Returns a Schema supported by the SCIM API

200 Response

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:ResourceType"
  ],
  "id": "User",
  "name": "User",
  "endpoint": "/User",
  "description": "User Account",
  "schema": "urn:ietf:params:scim:schemas:core:2.0:User",
  "schemaExtensions": [
    {
      "schema": "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User",
      "required": true
    }
  ],
  "meta": {
    "resourceType": "ResourceType",
    "location": "https://api.surveymonkey.com/scim/v2/ResourceTypes/User"
  }
}
```

```
}
```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
schemas	[string]	
id	string	SCIM Resource Type identifier for a User object
name	string	Name of the SCIM Resource User
endpoint	string	Resource Type SCIM endpoint for the SCIM Resource User
description	string	Description of the SCIM Resource User
schema	string	SCIM Schema identifier for the User Resource
schemaExtensions	[object]	SCIM Schema extensions for the User Resource
schema	string	Schema Id of the supported SCIM extension for the User Resource
required	boolean	Whether the SCIM extension is required for the User Resource
meta	object	
resourceType	string	Type of SCIM resource
location	string	URL to this resource

ServiceProviderConfig

GET /ServiceProviderConfig

Code samples

```
import http.client

conn = http.client.HTTPSConnection("api.surveymonkey.com")

headers = {
    'Accept': "application/scim+json",
    'Authorization': "Bearer {access-token}"
}

conn.request("GET", "/scim/v2/ServiceProviderConfig", headers=headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

GET /ServiceProviderConfig

AVAILABLE METHODS

GET: Returns the SCIM API Service Provider configuration

200 Response

```
{
  "schemas": "urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig",
  "documentationUri": "https://developer.surveymonkey.com/api/v3/?shell#scim-api",
  "patch": {
```

```

    "supported": true
},
"filter": {
    "supported": true,
    "maxResults": 1000
},
"changePassword": {
    "supported": false
},
"sort": {
    "supported": false
},
"etag": {
    "supported": false
},
"authenticationSchemes": {
    "name": "OAuth Bearer Token",
    "description": "Authentication scheme using the OAuth Bearer Token Standard",
    "specUri": "http://www.rfc-editor.org/info/rfc6750",
    "documentationUri": "https://help.surveymonkey.com/en/surveymonkey/teams/scim/#set-up",
    "type": "oauthbearertoken",
    "primary": true
},
"meta": {
    "location": "https://api.surveymonkey.com/scim/v2/ServiceProviderConfig",
    "resourceType": "ServiceProviderConfig",
    "created": "2021-01-04T00:00:00Z",
    "lastModified": "2021-01-04T00:00:00Z"
}
}
}

```

RESPONSE SCHEMA

Status Code 200

Name	Type	Description
schemas	string	SCIM Schema identifier for ServiceProviderConfig
documentationUri	string	URL to the public documentation for the SCIM API
patch	object	
supported	boolean	Whether or not the PATCH operation is supported by the SCIM API
filter	object	
supported	boolean	Whether or not the filter operation is supported by the GET API(s)
maxResults	integer	Max number of results that will be returned by the GET API(s)
changePassword	object	
supported	boolean	Whether or not changing a User's password is supported by the SCIM API
sort	object	
supported	boolean	Whether or not the sort query parameter is supported by the GET API(s)
etag	object	
supported	boolean	Whether or not etag is supported
authenticationSchemes	object	
name	string	OAuth Bearer Token
description	string	Authentication scheme using the OAuth Bearer Token Standard
specUri	string	Link to the SCIM RFC
documentationUri	string	Link to documentation on how to generate SCIM API bearer token
type	string	Type of authentication supported by the SCIM API

Name	Type	Description
primary	boolean	Whether or not this is the primary authentication for the SCIM API
meta	object	
location	string	Location of this object
resourceType	string	The type of SCIM resource
created	string	The date-time of when this SCIM resource was created
lastModified	string	The date-time of when this SCIM resource was last modified

API Libraries & SDKs

Feedback SDKs for iOS and Android

Our mobile feedback SDKs let app developers:

Embed SurveyMonkey surveys in iOS or Android apps using a SDK collector
Schedule an intercept to prompt users to take a survey at a custom time interval
Bind to survey submissions and customize subsequent flows for a user based on their response
Use custom variables to record additional data in survey submissions

Get the SDK, sample projects, and full documentation on Github:

iOS SDK
Android SDK

► [Run in Postman](#)