**ZOHO Books API**

Zoho Books is an online accounting software developed by Zoho Corporation. It is designed to help small businesses manage their finances, invoicing, expenses, and more. It's possible that Zoho Books might have an API that allows developers to interact with its features programmatically.

The Zoho Books API allows you to perform all the operations that you do with our web client.

Zoho Books API is built using REST principles which ensures predictable URLs that makes writing applications easy. This API follows HTTP rules, enabling a wide range of HTTP clients to interact with the API.

Every resource is exposed as a URL. The URL of each resource can be obtained by accessing the API Root Endpoint.

**Organization ID**

In Zoho Books, your business is termed as an organization. If you have multiple businesses, you simply set each of those up as an individual organization. Each organization is an independent Zoho Books Organization with its own organization ID, base currency, time zone, language, contacts, reports, etc.

The parameter organization\_id along with the organization ID should be sent in with every API request to identify the organization.

The organization\_id can be obtained from the GET /organizations API’s JSON response. Alternatively, it can be obtained from the Manage Organizations page in the admin console:

Login to the Zoho Books admin console. Click the drop down with the organization's name as the label and click Manage Organizations.

**Multiple Data Centers**

Zoho Books is hosted at multiple data centers, and therefore available on different domains.

Zoho Books is available in the US,Europe,India,Australia and Japan.

**Data Center Domain Base App URI**

India .in <https://www.zohoapis.in/books/>

**API Call Limit**

API calls are limited to provide better quality of service and availability to all the users. The limits on total requests per day are listed below for each plan:

* Free Plan - 1000 API requests/day
* Standard Plan- 2000 requests/day
* Professional Plan- 5000 requests/day
* Premium Plan- 10000 requests/day
* Elite Plan- 10000 requests/day
* Ultimate Plan- 10000 requests/day

A threshold of 30 API requests per minute applies for all the APIs

**API Root End Point** : <https://www.zohoapis.in/books/v3>

**OAuth:**

Zoho REST APIs uses the OAuth 2.0 protocol to authorize and authenticate calls. It provides secure access to protect resources thereby reducing the hassle of asking for a username and password every time a user logs in. Follow the steps listed here, to access Zoho’s APIs using OAuth 2.0.

**Data Center Domain Base App URI**

India .in <https://accounts.zoho.in/>

Here's a general overview of how OAuth works with the Zoho Books API:

**1. Register Your App:** Before you can use OAuth with the Zoho Books API, you need to register your application in the Zoho Developer Console. This process involves providing basic information about your app and obtaining client credentials (client ID and client secret).

**2. User Authorization:** When a user wants to grant your app access to their Zoho Books account, your app redirects them to Zoho's authorization endpoint. This usually involves specifying the scope of access your app needs.

**3. User Consent:** At the authorization endpoint, the user is asked to log in to their Zoho account (if not already logged in) and approve the requested permissions. This step ensures that the user is aware of what data your app will access.

**4. Access Token Request:** After the user approves, Zoho generates an authorization code and redirects the user back to your app's redirect URL along with the code.

**5. Exchange Authorization Code for Access Token:** Your app uses the authorization code to make a secure request to Zoho's token endpoint, providing the client ID, client secret, authorization code, and redirect URL. In return, your app receives an access token and a refresh token.

**6. Using the Access Token:** Your app includes the access token in the Authorization header of API requests to the Zoho Books API. This token validates that your app is authorized to access the user's account data.

**7. Refreshing Access Tokens:** Access tokens typically have an expiration time. When the access token expires, your app can use the refresh token to request a new access token without requiring the user's intervention.

**8. Revoking Access:** Users can revoke your app's access at any time through their Zoho account settings.

**List of scopes available in Zoho Books :**

To access scopes related APIs we have four features CREATE,READ,UPDATE and DELETE.

Ex. contacts,settings,estimates,customer payments,credit notes etc.

**HTTP Method:**

Zoho Books API uses appropriate HTTP verbs for every action.

**1. GET:** The GET method is used to retrieve information from the Zoho Books API. You can use it to retrieve various types of data, such as invoices, expenses, customers, items, etc.

**Example**: GET /api/v3/invoices

**2. POST:** The POST method is used to create new resources in the Zoho Books API, such as invoices, expenses, or other records.

**Example:**

POST /api/v3/invoices

{

"customer\_id": "12345",

"date": "2023-08-29",

"total": 100.00

// other invoice data

}

**3. PUT:** The PUT method is used to update existing resources in the Zoho Books API. You would use this to update details of invoices, expenses, customers, etc.

**Example:**

PUT /api/v3/invoices/123

{

"total": 150.00

// other updated invoice data

}

**4. DELETE:** The DELETE method is used to remove resources from the Zoho Books API. This could involve deleting invoices, expenses, customers, etc.

**Example:** DELETE /api/v3/invoices/123

**Responses:**

Responses will be in the JSON format.

| **Node Name** | **Description** |
| --- | --- |
| code | Zoho Books error code. This will be zero for a success response and non-zero in case of an error. |
| message | Message for the invoked API. |
| resource name | Comprises the invoked API’s Data. |

Certain APIs also support csv and pdf formats. The required response format needs to be specified in the respective request’s Accept header or accept query parameter.

**Errors**

Zoho Books uses HTTP status codes to display the result of an API call. Generally, 2xx class codes are success codes, 4xx class codes occur when the information provided by the client is incorrect, and 5xx class codes indicate server side errors. The HTTP status codes that are commonly used are listed below.

| **HTTP Status Codes**  | **Status Code** | **Description** | | --- | --- | | 200 | OK | | 201 | Created | | 400 | Bad request | | 401 | Unauthorized (Invalid AuthToken) | | 404 | URL Not Found | | 405 | Method Not Allowed (Method you have called is not supported for the invoked API) | | 429 | Rate Limit Exceeded (API usage limit exceeded) | | 500 | Internal Error | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

**Pagination**

Zoho Books provides APIs to retrieve lists of contacts, plans and other resources - paginated to 200 items by default. The pagination information will be included in the list API response under the node name **page\_context.**

By default the first page will be listed. For navigating through pages, use the page parameter.

The per\_page parameter can be used to set the number of records that you want to receive in response.

**ZOHO Meeting APIs**

Integrate Zoho Meeting with third-party applications and websites using Zoho Meeting APIs. Based on the REST API, requests to resource URIs will result in JSON responses. Using our programmatic framework, you can develop customized applications or integrate Zoho Meeting with existing websites. It gives you the freedom to expand and build on your programmatic base to suit your needs.

API Root Endpoint: [https://meeting.zoho.in/api/v2](https://meeting.zoho.com/api/v2)

All Zoho Meeting APIs require a minimum of two mandatory headers.

1. **Authorization**- Authentication request url parameter.
2. **X-ZSOURCE-** Organization/Company Name

**Example:**

**$ curl -X GET 'https://meeting.zoho.in/api/v2/user.json' \**

**-H Authorization: Zoho-oauthtoken 6e80ed905055deb9fd2d12d71f518a80**

**-H X-ZSOURCE: {Company Name}**

**Authentication**

All Zoho Meeting APIs need to be authenticated using an OAuth token.

OAuth 2.0 is an open authorization protocol that grants third-party applications limited access to user accounts on an HTTP service. The authentication and authorization process is facilitated only between the end-user and the HTTP service. Zoho APIs use OAuth 2.0 to provide dependable security for your application data. OAuth 2.0 delegates authorization and authentication for web and desktop applications, as well as mobile devices.

You can obtain an OAuth token by following these steps:

1. Register your application
2. Get an authorization grant
3. Get an access token

**1. Register your application**

Before integrating Zoho APIs with your application using OAuth, you must register your application with Zoho. The Client ID is used by Zoho to verify the identity of the application and has public visibility.

**2. Get An Authorization Grant**

URL: [https://accounts.zoho.in/oauth/v2/auth](https://accounts.zoho.com/oauth/v2/auth)

Method: GET

**Params:**

* scope=Scopes for which token to be generated Eg : ZohoMeeting.meeting.READ,ZohoMeeting.meeting.CREATE
* client\_id=The client ID of the integrating app
* response\_type=code
* redirect\_uri= https://app.example.com/oauth (Redirect URL given during registration)
* access\_type=offline (The allowed values are offline and online)
* state=opaque string that will be returned in redirect url
* **prompt**=consent (Used to generate refresh token everytime)

**Available Scopes:**

* manageOrg- ZohoMeeting.manageOrg.READ
* meeting- ZohoMeeting.manageOrg.READ,UPDATE,CREATE,DELETE
* webinar - ZohoMeeting.manageOrg.READ,UPDATE,CREATE,DELETE

**Actions under each scope:**

User Details

* Get User Details API

Meeting API

* Get Meeting Details
* Create Meeting
* Edit Meeting
* Delete Meeting

Webinar API

* Get Webinar Details
* Create Webinar
* Edit Webinar
* Delete Webinar

**Example:**

**https://accounts.zoho.in/oauth/v2/auth?response\_type=code&client\_id=<client-id>&scope=ZohoMeeting.meeting.ALL&redirect\_uri=<url>&access\_type=offline**

**3. Get An Access Token**

After getting the authorization grant, post it to the below API to get the access token and the refresh token.

URL: [https://accounts.zoho.in/oauth/v2/token](https://accounts.zoho.com/oauth/v2/token)

Method : POST

**Params:**

* code=<Code obtained in the above step>
* client\_id=<The client ID of the integrating app>
* client\_secret=<Obtained during Client Registration>
* redirect\_uri=<Same URI that was used in the above step>
* grant\_type=authorization\_code

**Note:** expires\_in parameter in response contains expiry time of access\_token in milliseconds.

**Regenerating The Access Token With The Refresh Token**

Once the access token has expired, it can be regenerated from the refresh token by making a POST call to the API given below.

URL: [https://accounts.zoho.in/oauth/v2/token](https://accounts.zoho.com/oauth/v2/token)

**Params:**

* refresh\_token=<Refresh Token obtained in the above step>
* client\_id=<The client ID of the integrating app>
* client\_secret=<Obtained during Client Registration>
* redirect\_uri=<Same URI that was used in the above step>
* grant\_type=refresh\_token

**Revoking A Refresh Token**

A refresh token can be revoked by calling the API given below.

URL: [https://accounts.zoho.in/oauth/v2/token](https://accounts.zoho.com/oauth/v2/token)

**Note:** The Refresh Token will always be generated by the prompt=consent. The maximum number of Refresh Tokens is 20. Once the limit is reached, the first Refresh Token generated will be deleted.

Params: token = <Refresh token>

**Get User Details**

In Zoho Meeting, your business is termed as an organization. A **zsoid** key has to be used for all other APIs.

**NOTE:**

The expires\_in field in the response you provided indicates the duration for which the access token is valid, in seconds. In your case, the access token is valid for 3600 seconds, which is equivalent to 1 hour. Once this duration elapses, you will need to obtain a new access token to continue accessing the resources or performing actions that require authentication.

The expiration time of an access token is typically set by the service provider (Zoho Meeting in this case) to ensure security and limit the window of opportunity for unauthorized access in case the token is compromised.

Unfortunately, you cannot directly change the expiration time of an access token from the response itself. The expiration time is determined by the server issuing the token. If you find that the access token's expiration time is causing inconvenience for your use case, you might want to explore the following options:

**Token Refresh**: Many APIs provide a way to refresh an access token. You use a refresh token to obtain a new access token without requiring user interaction. This allows you to maintain long-lived sessions while periodically refreshing your access token.

**Reauthentication:** If you can't refresh the token, you might need to reauthenticate the user or your application to obtain a new access token when the current one expires.

**Contact the Service Provider:** If the default expiration time doesn't align with your use case and the service provider allows it, you could try reaching out to their support or checking their documentation to see if there's a way to request longer-lived access tokens.

**Meeting API**

Zoho Meeting lets you host instant or scheduled meetings, collaborate with audio, video, and screen sharing. You can use the Meeting API to get, create, update, and delete the meetings in your Zoho Meeting account.

**OAuth Scope:** ZohoMeeting.meeting.ALL

**Attributes**:

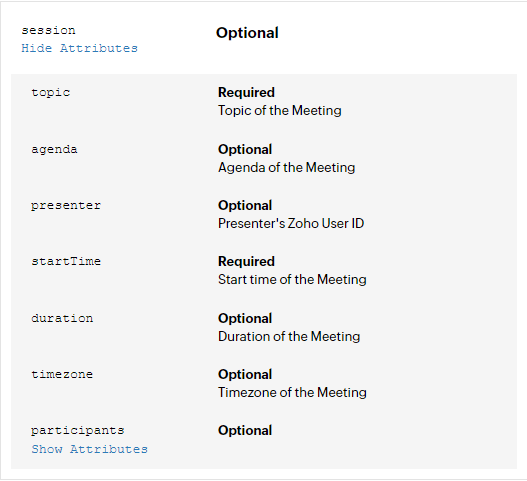
| meetingKey | **long**  Webinar Key generated by the server. This is used as an identifier. |
| --- | --- |
| presenter | **long**  Presenter Zoho User-ID |
| offset | **long**  Offset for timezone in Webinar scheduled |
| timezone | **string**  Timezone of the Webinar scheduled |
| accessCode | **string**  Access Code of the Webinar |
| creatorZuid | **long**  User Id of the Webinar creator |
| registrationLink | **string**  Registration link of the Webinar |
| agenda | **string**  Agenda of the Webinar. |
| startLink | **string**  Start link of the Webinar |
| duration | **long**  Duration of the Webinar scheduled |
| topic | **string**  Topic of the Webinar |
| sessionType | **string**  Agenda of the Webinar |
| dialinUrl | **string**  Dial in url of the Webinar |
| startTime | **string**  Start time of the Webinar |
| endTime | **string**  End time of the Webinar |

**Create a Meeting**

This API allows you to create a meeting by providing the details (title, date, time and duration) of the meeting in the Schedule meeting form.

**OAuth Scope:** ZohoMeeting.meeting.CREATE

**Attributes:**



**POST /{zsoid}/sessions.json**

**Request Example:**

curl -X POST https://meeting.zoho.com/api/v2/{zsoid}/sessions.json

-H "Content-Type: application/json;charset=UTF-8"

-H "Authorization: Zoho-oauthtoken ba4604e8e433g9c892e360d53463oec5"

-d JSONString='{

"session": {

"topic": "Monthly Marketing Meeting",

"agenda": "Points to get noted during meeting.",

"presenter": 123456789,

"startTime": "Jun 19, 2020 07:00 PM",

"duration": 3600000,

"timezone": "Asia/Calcutta",

"participants": [

{

"email": "dummy@email.com"

}

]

}

}'

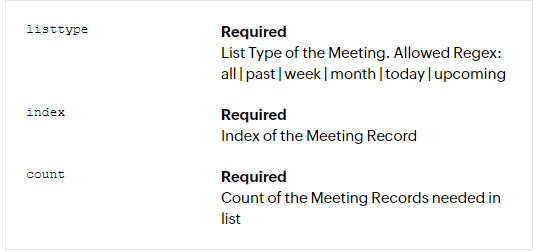
**Note:** Replace **zsoid** with user zoho user id and **Zoho-oauthtoken** with access token you generated while authentication.

**List of Meeting API**

You can use this API to get the list of meetings.

**​OAuth Scope:** ZohoMeeting.meeting.READ

**Arguments**



**GET /{zsoid}/sessions.json**

**Request Example:**

curl -X GET https://meeting.zoho.com/api/v2/{zsoid}/sessions.json

-H "Content-Type: application/json;charset=UTF-8"

-H "Authorization: Zoho-oauthtoken ba4604e8e433g9c892e360d53463oec5"

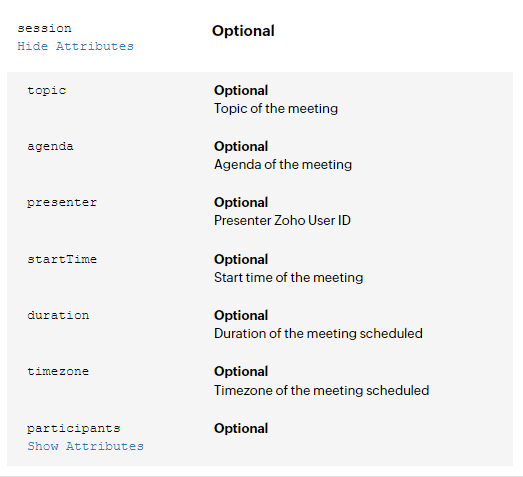
**Note:** Replace **zsoid** with user zoho user id and **Zoho-oauthtoken** with access token you generated while authentication.

**Edit Meeting API**

You can use this API to edit a meeting. You can change the topic, agenda, or time of a meeting you've scheduled at any time.

**OAuth Scope:** ZohoMeeting.meeting.UPDATE

**Arguments**



**PUT /{zsoid}/sessions/{meetingKey}.json**

**Request Example:**

curl -X PUT https://meeting.zoho.com/api/v2/{zsoid}/sessions/{meetingKey}.json

-H "Content-Type: application/json;charset=UTF-8"

-H "Authorization: Zoho-oauthtoken ba4604e8e433g9c892e360d53463oec5"

-d JSONString='{

"session": {

"topic": "Monthly Marketing Meeting",

"agenda": "Points to get noted during meeting.",

"presenter": 123456789,

"startTime": "Jun 19, 2020 07:00 PM",

"duration": 3600000,

"timezone": "Asia/Calcutta",

"participants": [

{

"email": "dummy@email.com"

}

]

}

}'

**Note:** Replace **zsoid** with user zoho user id, **Zoho-oauthtoken** with access token you generated while authentication, **meetingKey** with meeting id from zoho meeting of respective meeting you created .

**Get Meeting API**

You can use this API to get the details of a meeting. This includes the start link, join link, date and time, topic, host of the meeting and the email addresses of participants.

**OAuth Scope:** ZohoMeeting.meeting.READ

**GET /{zsoid}/sessions/{meetingKey}.json**

**Request Example:**

curl -X GET https://meeting.zoho.com/api/v2/{zsoid}/sessions/{meetingKey}.json

-H "Content-Type: application/json;charset=UTF-8"

-H "Authorization: Zoho-oauthtoken ba4604e8e433g9c892e360d53463oec5"

**Note:** Replace **zsoid** with user zoho user id, **Zoho-oauthtoken** with access token you generated while authentication, meetingKey with meeting id from zoho meeting of respective meeting you created .

**Delete Meeting API**

You can use this API to delete a meeting. Once you delete the meeting, it will no longer be available for you to host.

**OAuth Scope:** ZohoMeeting.meeting.DELETE

**DELETE /{zsoid}/sessions/{meetingKey}.json**

**Request Example:**

$ curl https://meeting.zoho.com/api/v2/{zsoid}/sessions/{meetingKey}.json

-X DELETE

-H "Content-Type: application/json;charset=UTF-8"

-H "Authorization: Zoho-oauthtoken ba4604e8e433g9c892e360d53463oec5"

**Note:** Replace **zsoid** with user zoho user id, **Zoho-oauthtoken** with access token you generated while authentication, meetingKey with meeting id from zoho meeting of respective meeting you created .

**Webinar API**

Using Zoho Webinar, you can broadcast live presentations or demos, interact and engage with your audience through Q&A, and launch live polls to gather opinions from them on any related topic. You can use this Webinar API to get, create, update, and delete the webinars in your Zoho Meeting account.

**OAuth Scope:** ZohoMeeting.webinar.ALL

**Create a Webinar**

**OAuth Scope:** ZohoMeeting.webinar.CREATE

**POST /{zsoid}/webinar.json**

**Request Example:**

$ curl - X POST https://meeting.zoho.com/api/v2/{zsoid}/webinar.json

-H "Content-Type: application/json;charset=UTF-8"

-H "Authorization: Zoho-oauthtoken ba4604e8e433g9c892e360d53463oec5"

-d JSONString='{

"session": {

"topic": "Monthly Marketing Webinar",

"agenda": "Description of webinar.",

"presenter": "7989202345",

"startTime": "Jun 19, 2023 04:00 PM",

"duration": 3600000,

"timezone": "Asia/Calcutta",

"participants": [

{

"email": "dummy@email.com"

}

]

}

}'

**Note:** Replace **zsoid** with user zoho user id, **Zoho-oauthtoken** with access token you generated while authentication.

**List of Webinar API**

**OAuth Scope:** ZohoMeeting.webinar.READ

**GET /{zsoid}/webinar.json**

**Request Example:**

$ curl -X GET https://meeting.zoho.com/api/v2/{zsoid}/webinar.json

-H "Content-Type: application/json;charset=UTF-8"

-H "Authorization: Zoho-oauthtoken ba4604e8e433g9c892e360d53463oec5"

**Note:** Replace **zsoid** with user zoho user id, **Zoho-oauthtoken** with access token you generated while authentication.

**Edit Webinar API**

**OAuth Scope:** ZohoMeeting.webinar.UPDATE

**PUT /{zsoid}/webinar/{webinarKey}.json**

**Request Example:**

$ curl -X PUT https://meeting.zoho.com/api/v2/{zsoid}/webinar/{webinarKey}.json

-H "Content-Type: application/json;charset=UTF-8"

-H "Authorization: Zoho-oauthtoken ba4604e8e433g9c892e360d53463oec5"

-d JSONString='{

"session": {

"topic": "Monthly Marketing Webinar",

"agenda": "Description of webinar.",

"presenter": "7989202345",

"startTime": "Jun 19, 2020 07:00 PM",

"duration": 3600000,

"timezone": "Asia/Calcutta",

"participants": [

{

"email": "dummy@email.com"

}

]

}

}'

**Note:** Replace **zsoid** with user zoho user id, **Zoho-oauthtoken** with access token you generated while authentication, **webinarKey** with webinar id from zoho meeting of respective webinar you created .

**Get Webinar API**

**OAuth Scope:** ZohoMeeting.webinar.READ

**GET /{zsoid}/webinar/{webinarKey}.json**

**Request Example:**

$ curl https://meeting.zoho.com/api/v2/{zsoid}/webinar/{webinarKey}.json

-H "Content-Type: application/json;charset=UTF-8"

-H "Authorization: Zoho-oauthtoken ba4604e8e433g9c892e360d53463oec5"

**Note:** Replace **zsoid** with user zoho user id, **Zoho-oauthtoken** with access token you generated while authentication, **webinarKey** with webinar id from zoho meeting of respective webinar you created .

**Delete Webinar API**

**OAuth Scope:** ZohoMeeting.webinar.DELETE

**DELETE /{zsoid}/webinar/{webinarKey}.json**

**Request Example:**

$ curl https://meeting.zoho.com/api/v2/{zsoid}/webinar/{webinarKey}.json

-X DELETE

-H "Content-Type: application/json;charset=UTF-8"

-H "Authorization: Zoho-oauthtoken ba4604e8e433g9c892e360d53463oec5"

**Note:** Replace **zsoid** with user zoho user id, **Zoho-oauthtoken** with access token you generated while authentication, **webinarKey** with webinar id from zoho meeting of respective webinar you created .

**Bulk Registration API**

Zoho Meeting lets bulk registrations happen at once. You can use this API and allow bulk registrations for your webinar, with all the necessary parameters for a successful registration.

**Method**: POST

**Example**: /meeting/api/v0/[zsoid]/register/import/[meetingKey].json?instanceId=[instanceId]&sendMail=true|false

**Note: All the attributes and arguments mentioned in the zoho meeting api are used in zoho webinar api too.**