

RoyalTeck Application Programming Interface

Partner Handbook
January 2022

VivoMeetings Partner APIs

Table of Contents

Introduction	5
High Level Architecture	5
Global Network	5
Application Architecture	7
Key Entities and Data Structure	8
Quick Start Guide	8
Provisioning a company & subscribing to a plan	9
Provisioning a meeting Host	9
Taking the user to their meeting dashboard	9
1.0 General	10
1.1 Request method/format	10
1.2 Response method/format	10
1.3 Roles	10
2.0 Reseller	10
3.0 Authentication	11
3.1 Authenticate	11
4.0 Companies	12
4.1 Create a Company	12
4.2 Get Company Details	13
4.3 Modify a Company	13
4.4 Get All Companies	15
5.0 Subscriptions	16
5.1 Fetch products	16
5.2 Create Subscription	17
5.2 Update Subscription	18
5.3 Suspend Subscription	19
5.4 Reactivate Subscription	20
5.5 Cancel Subscription	20
5.6 Fetch Subscriptions	21
6.0 Hosts	22

6.1 Create Host Account	22
6.2 Get Host Account Details	26
6.3 Edit Host Account	28
6.4 Delete Host Account	31
6.5 Welcome Email	32
6.6 Reset Password	32
6.7 Host Conferences	33
6.8 Host CDRs	35
6.9 Active Calls	37
6.10 Fetch all hosts in a company	37
7.0 Contacts	39
7.1 Create Contacts	39
7.2 Get Contact Details	41
7.3 Get Contact Details for Host	42
7.4 Edit Contact	44
7.5 Delete Contact	45
8.0 Conferences	45
8.1 Schedule Conference	46
8.2 Create Reservationless Conference	49
8.3 Update Conference	50
8.4 Fetch Conference	53
8.5 Fetch All Conferences	54
8.6 Cancel Conference	56
8.7 End Conference	57
8.8 Fetch Participant Details	57
8.9 Fetch Live Stream Viewer Count	58
8.10 Recurring Conferences	59
8.11 Error Messages	62
9.0 Embedding Vivomeetings into Your App	63
9.1 Embedding in a web application	63
9.2 Embedding a live stream player	65
9.3 Embedding options for Gaming and Watch Parties	65
9.4 Hiding and displaying controls	66
9.5 Meeting events	67
10.0 Signing in on behalf of Hosts (SSO)	69

10.1 SSO via POST or GET (current tab or new tab)	69
10.2 SSO via iFrame	70
11.0 Webhooks	71
11.1 Generalities	71
11.2 Conference Start	71
11.3 Conference End	71
11.4 Start Recording	72
11.5 Stop Recording	72
11.6 Start Streaming	72
11.7 Stop Streaming	72
11.8 Recording Ready	72
11.9 Transcription Ready	73
11.10 Enter Livestream	73
11.11 Leave Livestream	73
12.0 Other API Use Cases	73

Introduction

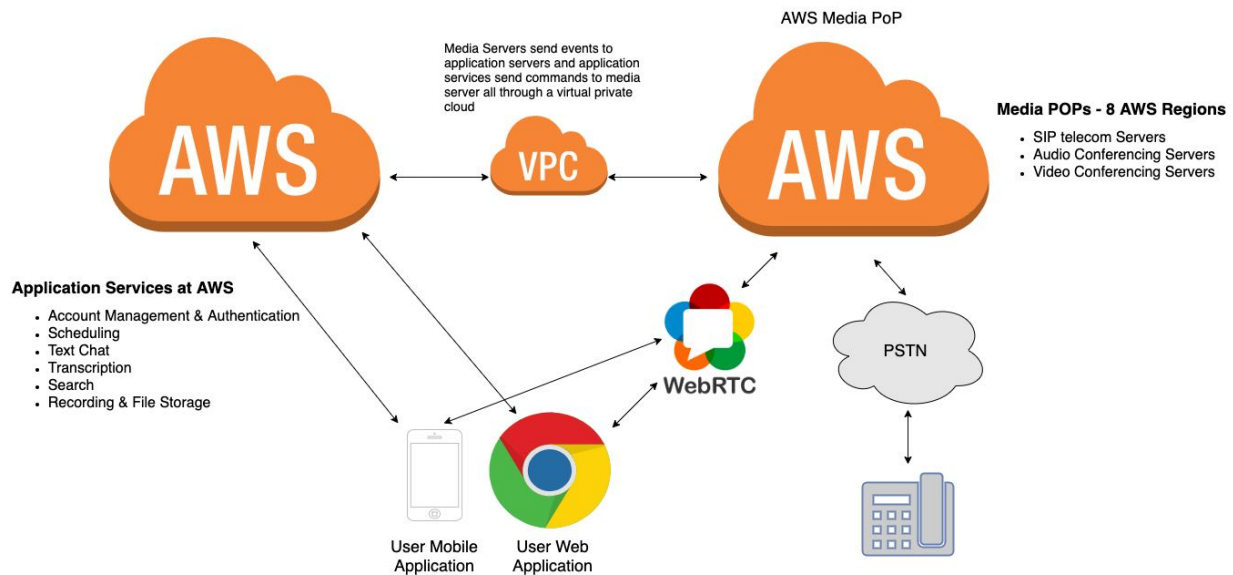
Vivomeetings is a real time communications tool commercialised as a white-label product for partners to deploy their own branded version. RoyalTeck retails parts of the solution under the brands VivoMeetings and CallToNet. Most often, white-label solutions follow the VivoMeetings model of service, which is similar to other enterprise offerings for scheduling, video meetings, webinars, transcriptions, recordings and physical meeting room support.

Numerous channel partners have deployed versions of ICC under their own brand, and sometimes with their own integrations and additional services.

This document outlines the interface characteristics of RoyalTeck’s platform for real time communication. The Application Programming Interface described within will provide methods by which developers may request and send instructions to and from the platform with the objective to integrate 3rd party applications, and extend ICC’s functionality.

High Level Architecture

ICC is a cloud-based, multi-tenant solution that provides scale and security. Below is a diagram outlining a high-level view of system architecture.



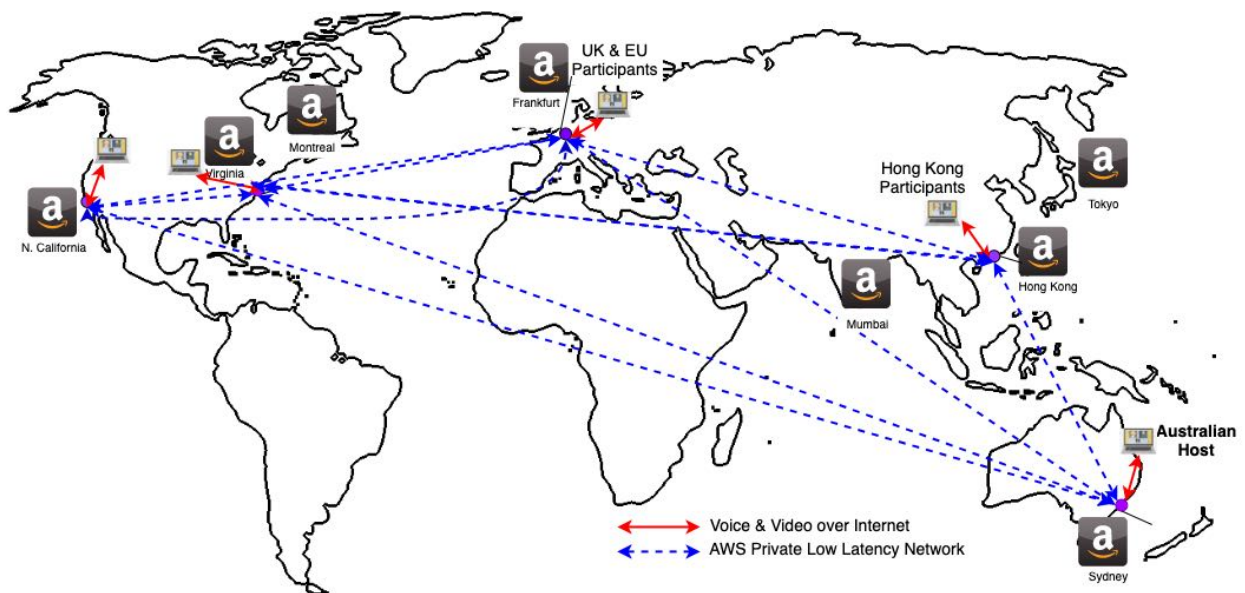
Global Network

The RoyalTeck has a global network of media Points of Presence (POPs). These POPs connect to each other using Amazon's high quality and low latency private network forming a network mesh architecture. The result is high quality voice and video for participants all across the globe.

Royalteck currently has POPs in the following countries:

- United States
- Canada
- Germany
- Mumbai
- Hong Kong
- Tokyo
- Sydney

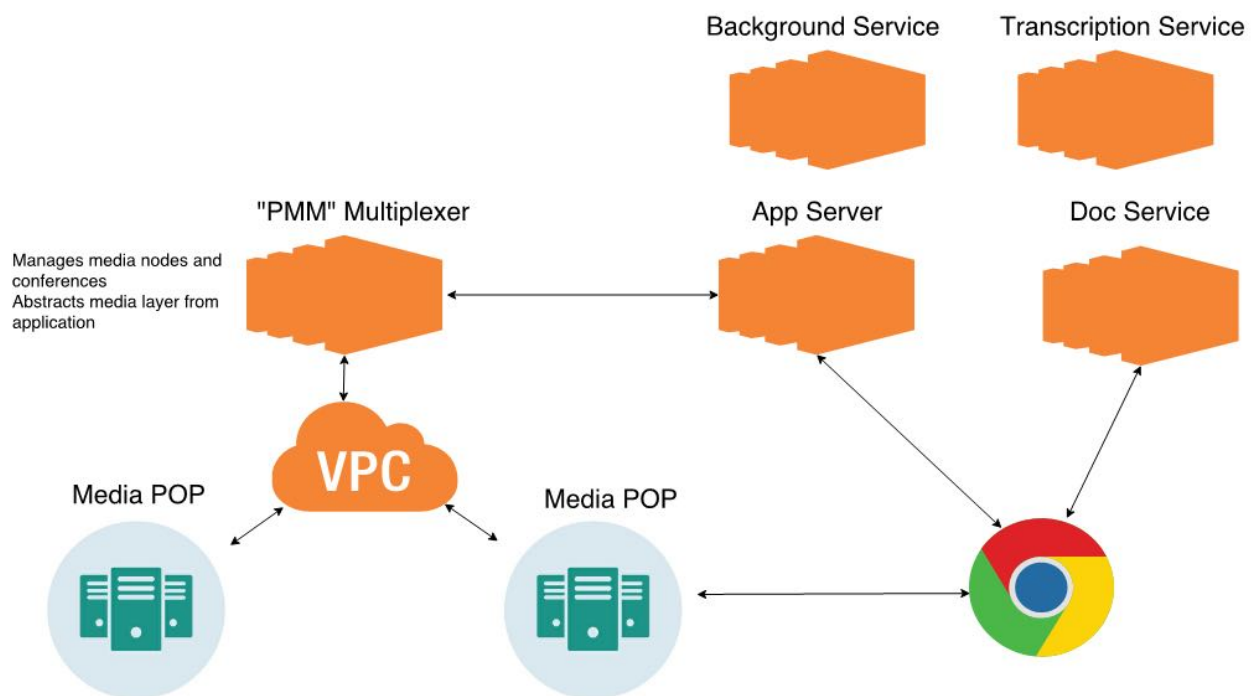
RoyalTeck can add a media POP to any AWS region.



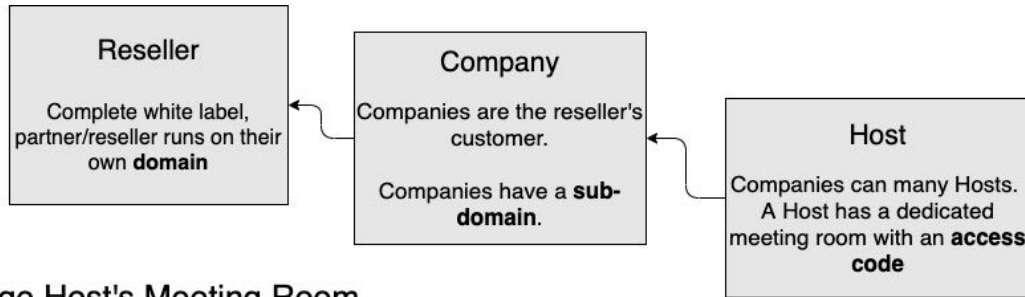
Application Architecture

Inside our AWS implementation are services to cover:

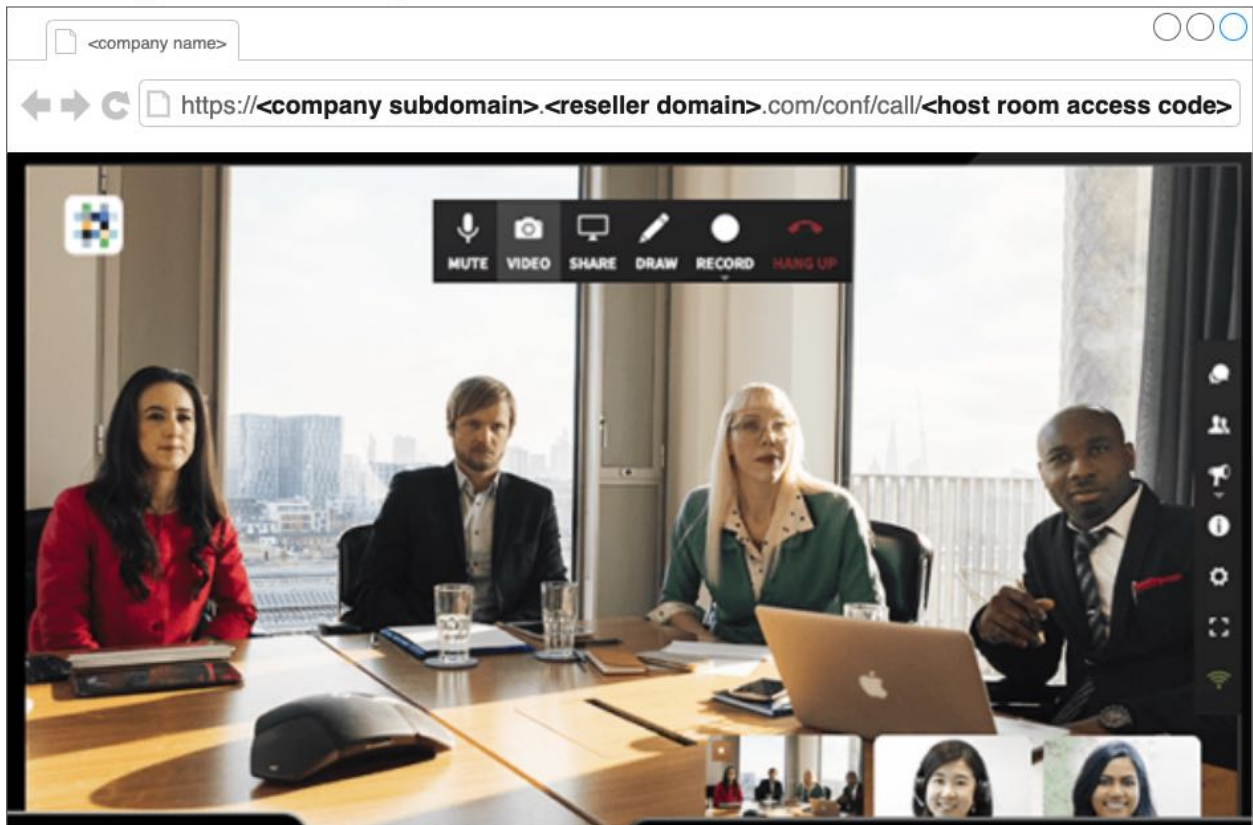
- Account management and authentication
- Scheduling
- Text Chat
- Search
- Recording and File Storage
- Transcription



Key Entities and Data Structure



Callbridge Host's Meeting Room



Quick Start Guide

Provisioning customers and their users is pretty straightforward using the APIs outlined in this document. As described in the diagram above our platform starts with the **Reseller**; typically you are the Reseller. If you have white label sub Resellers they are also Resellers on our platform. Below the **Reseller** there are **Companies**, these are your customers, and below companies there are meeting **Hosts**, they are the end users of the platform. You can follow the 3 simple steps below to quickly provision users on your new white label and direct them to their meetings dashboard.

1. Provisioning a company & subscribing to a plan

1. Somewhere in your admin interface would be a way to enable our video service for that customer/company. You would call "Create Company" outlined in [4.1 Create a Company](#).
2. Then you would subscribe to the company to a plan. The plans simply define which features on the video platform are available to the meeting Host. Plans can be completely customized for each Reseller. Subscribing to a plan is done using [5.2 Create Subscription](#).

2. Provisioning a meeting Host

1. Once the company has a subscription to a plan, then you can automatically or manually start creating meeting Hosts on our platform that belong to the company using [6.1 Create Host Account](#). Once the Host is created you're ready to direct that user to their white labeled meetings dashboard.

3. Taking the user to their meeting dashboard

1. In the response from the "Create Host" API is a "login_token_public_key". This login token will allow you to place a link or button on your site directing the user to their meetings dashboard without having to sign into our platform. See [10.1 SSO via POST or GET \(current tab or new tab\)](#).

Of course there are many more APIs and capabilities outlined in this document. The above steps simply serve as a quick and easy way to get started.

1.0 General

1.1 Request method/format

HTTPS POST, mime-type application/json, with the request body being a JSON-serialized dictionary (key-value pairs).

All requests except Authenticate must include, besides the parameters listed under Request at each individual method, an **auth_token** parameter, as received in the response to the Authenticate method. Unless otherwise indicated, all request Input parameters are required.

The FQDN for API requests will be that of a company, typically the initial admin company created for the reseller during onboarding. E.g. the entire URL for an authenticate request would be

https://reseller-admin.reseller.tld/enterprise_api/authenticate.

1.2 Response method/format

Response HTTP status code will indicate the overall status of the request:

- 200 - the request succeeded
- 401 - for Authenticate, that the supplied credentials are invalid; for other methods, that the auth_token parameter is missing, or that the associated session has expired (use Authenticate to get a new auth_token)
- 403 - attempting to act on objects beyond the scope of one's role (Reseller Admin beyond the current Reseller, Company Admin beyond the current Company)
- 404 - update, delete, and similar methods attempted where the indicated object does not exist
- 500 - internal server error

Response mime-type will be application/json, the body being a JSON-serialized dictionary including, besides the values listed under Response at each method, a 'message' with details of any errors.

1.3 Roles

- Host - may host conferences only.
- Company Admin - may host conferences, and via the web interface or this API, add/edit/delete hosts in his/her own Company
- Reseller Admin - may host conferences, and via the web interface or this API, add companies, manipulate subscriptions, and add/delete/hosts in any of the Reseller's Companies

Each method below indicates which roles may use that method.

2.0 Reseller

A reseller represents the organization selling conferencing as a white label of VivoMeetings. A reseller can sell directly to Companies (customers) or can have sub resellers below them.

A Reseller has the following attributes:

- **domain:** abc-conferencing.com for example

3.0 Authentication

Before calling other methods, API consumers must authenticate. The credentials will be those of a host created from the Admin UI at at least the Company Admin level. (It can coincide with a real host, but typically resellers will create a dedicated 'host' for API use.)

Also see [12. OAuth](#) for details on using OAuth with the API.

3.1 Authenticate

ResellerAdmin/CompanyAdmin

```
POST "/enterprise_api/authenticate"

Request body (application/json):
{
    "email": "api@partner.tld",
    "password": "Sup3rSecret!"
}

Response body (application/json):
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98"
}
```

Request:

- **email:** the email of a host in that company that has the ResellerAdmin or CompanyAdmin role
- **password:** password for the said host

Response:

- **auth_token:** The token to be included with all subsequent requests, as described in the General section above. An auth_token expires five minutes after its last successful use in any of the methods below (i.e. it is 'touched' on each use).

4.0 Companies

A Company represents the customer. A company could be any organization that subscribes to a conferencing service. A company has any number of Host Accounts. Host accounts are the end users.

A company stores basic details about the customer, can have customizations available to all Hosts on that company and holds the billing details.

4.1 Create a Company

/enterprise_api/company/create
ResellerAdmin Only

```
POST "/enterprise_api/company/create"

Request body (application/json):
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "subdomain": "joe",
    "name": "The Joe Company",
    "openid_provider": true
}

Response body (application/json):
{
    "id": 5
}
```

Request:

- **subdomain:** Every company has a subdomain that is used in their meeting urls, i.e royalteck.vivomeetings.com. The subdomain must be unique within the scope of each reseller domain.
- **name:** The name of the company as it should be displayed for billing purposes
- **openid_provider:** (optional) If this company is an openid-connect authentication provider, we will send users through your auth instead of our own. Please contact us before using this feature.

Response:

- **id:** id of the resulting new company

4.2 Get Company Details

ResellerAdmin

Retrieve details for a particular company

```
POST "/enterprise_api/company/fetch"

Request body (application/json):
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "company_id": 5
}

Response body (application/json):
{
    "subdomain": "joe",
    "name": "The Joe Company",
    "host_count": 3,
    "hosts": [6, 7, 8],
    "openid_provider": false
}
```

Request:

- **company_id**: The id of the company you wish to fetch details of.

Response:

- **subdomain**: Every company has a subdomain that is used in their meeting urls, i.e royalteck.vivomeetings.com. The subdomain must be unique within the scope of each reseller domain.
- **name**: The name of the company as it should be displayed for billing purposes
- **host_count**: Current number of hosts
- **hosts**: Array of host_id (q.v. in the Hosts section below), hosts in this company
- **openid_provider**: If this company is an openid-connect authentication provider, we will send users through your auth instead of our own. Please contact us before using this feature.

4.3 Modify a Company

ResellerAdmin Only

Change company subdomain and/or name

```
POST "/enterprise_api/company/update"
```

```
Request body (application/json):
```

```
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "company_id": 5,
    "subdomain": "jane",
    "name": "The Jane Company"
}
```

```
Response body (application/json):
```

```
{
    "subdomain": "jane",
    "name": "The Jane Company",
    "host_count": 3,
    "hosts": [6, 7, 8]
}
```

Request:

- **company_id**: The id of the company you wish to modify.
- **subdomain**: Every company has a subdomain that is used in their meeting urls, i.e royalteck.vivomeetings.com. The subdomain must be unique within the scope of each reseller domain.
and/or
- **name**: The name of the company as it should be displayed for billing purposes
and/or
- **openid_provider**: If this company is an openid-connect authentication provider, we will send users through your auth instead of our own. Please contact us before using this feature.

Response:

- **subdomain**: Every company has a subdomain that is used in their meeting urls, i.e royalteck.vivomeetings.com. The subdomain must be unique within the scope of each reseller domain.
- **name**: The name of the company as it should be displayed for billing purposes
- **host_count**: Current number of hosts
- **hosts**: Array of host_id (q.v. in the Hosts section below), hosts in this company

4.4 Get All Companies

ResellerAdmin

Retrieve details for all companies

```
POST "/enterprise_api/company/fetch_all"

Request body (application/json):
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98"
}

Response body (application/json):
[
  {
    "subdomain": "joe",
    "name": "The Joe Company",
    "host_count": 3,
    "hosts": [6, 7, 8],
    "openid_provider": false
  },
  {
    "subdomain": "jane",
    "name": "The Jane Company",
    "host_count": 4,
    "hosts": [11, 12, 13],
    "openid_provider": false
  }
]
```

Response: array of:

- **subdomain:** Every company has a subdomain that is used in their meeting urls, i.e royalteck.vivomeetings.com. The subdomain must be unique within the scope of each reseller domain.
- **name:** The name of the company as it should be displayed for billing purposes
- **host_count:** Current number of hosts
- **hosts:** Array of host_id (q.v. in the Hosts section below), hosts in this company
- **openid_provider:** If this company is an openid-connect authentication provider, we will send users through your auth instead of our own. Please contact us before using this feature.

5.0 Subscriptions

A company can have one or more plans, and zero or more add-ons at a given time. Having multiple plans enables companies to assign some hosts to one plan, and others to another plan. Each host individually can only be assigned to one plan at a time.

5.1 Fetch products

ResellerAdmin/CompanyAdmin

Fetch a list of all available products (plans and add-ons) for a reseller

```
POST "/enterprise_api/subscription/fetch_products"

Request body (application/json):
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "company_id": 5
}

Response body (application/json):
[
    {
        "slug": "small_plan",
        "name": "Small Conference Plan",
        "plan": true
    },
    {
        "slug": "large_plan",
        "name": "Large Conference Plan",
        "plan": true
    },
    {
        "slug": "recording",
        "name": "Recording Add-On",
        "plan": false
    },
]
```


Response, array of:

- **slug**: an alphabetic unique identifier for the product
- **name**: a human-readable name of the product
- **plan**: boolean, whether the product is a plan [true] or an add-on [false]

5.2 Create Subscription

ResellerAdmin/CompanyAdmin

Create a subscription to a plan or add-on for a particular company.

```
POST "/enterprise_api/subscription/create"
```

```
Request body (application/json):
```

```
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "company_id": 5,
    "slug": "small_plan"
}
```

```
Response body (application/json): "OK"
```

Request:

- **company_id**: The id of the company you wish to add this Subscription to.
- **slug**: alphabetic unique identifier for the product to add (see `fetch_products` above)
- **switch_plan (optional)**
 - If specified and set to the subscription ID of one of the company's current plans, that plan will be cancelled and used in to prorate the initial amount of the new plan (plan switching)
 - If set to null, the current plan(s) of the company will not be cancelled and instead the company will be subscribed to the provided plan in addition to their existing plan(s).
 - When omitted (no key/value not present in the request body), the default behavior of switching the company's current plan with the new plan will be applied.
- **host_limit**: A positive integer value that the new subscription's host limit should be set to. When subscribing to a product that is using the per-host pricing method, the `host_limit` value provided will be multiplied by the per-host price of the product to determine and set the total amount of the monthly subscription charges.

Response: "OK"

Partners can use the `company_id` and `slug` if follow up API calls to reference the specific subscription created for a company.

host_limit pricing details:

Product Pricing Method:	Details:
Flat Rate (\$ / month)	No <code>host_limit</code> parameter should be provided. The customer is charged the <code>product.billing_amount</code> each month. There is no restriction on the number of hosts that can be assigned to the plan/add-on.
Per-Host Pricing (\$ / host / month)	A <code>host_limit</code> parameter should be supplied, and will be multiplied by the <code>product.billing_amount</code> (per-host amount) to determine the total monthly charges for the subscription. The host limit represents the maximum number of hosts that can be assigned to the plan/add-on, assigning additional hosts will automatically increment the host limit, and increase monthly charges.

5.2 Update Subscription

ResellerAdmin/CompanyAdmin

```
PUT "/enterprise_api/subscription/update"
```

```
Request body (application/json):
```

```
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "company_id": 5,
    "slug": "small_plan",
    "host_limit": 10
}
```

```
Response body (application/json): "OK"
```

Request:

- **company_id**: The id of the company you wish to add this Subscription to.
- **slug**: alphabetic unique identifier for the product. This is used to identify the associated subscription to be updated.
- **host_limit**: A positive integer value that the new subscription's host limit should be set to. When subscribing to a product that is using the per-host pricing method, the `host_limit` value provided will be multiplied by the per-host price of the product to determine and set the total amount of the monthly subscription charges.

Response: "OK"

5.3 Suspend Subscription

ResellerAdmin Only

Upon suspending a subscription a Host will no longer have access to features associated with the Plan or Add-on. If suspending a Subscription to a Plan the Hosts will no longer be able to host meetings.

A suspended Subscription can be reactivated at any time.

```
POST "/enterprise_api/subscription/suspend"
```

```
Request body (application/json):
```

```
{
  "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
  "company_id": 5,
  "slug": "small_plan"
}
```

```
Response body (application/json): "OK"
```

Request:

- **company_id**: The id of the company you wish to suspend this Subscription for.
- **slug**: alphabetic unique identifier for the product to suspend (see `fetch_products` above)

Response: "OK"

5.4 Reactivate Subscription

ResellerAdmin Only

Reactivating a subscription will mean those Hosts have access to all the features associated with that subscription.

```
POST "/enterprise_api/subscription/reactivate"
```

```
Request body (application/json):
```

```
{
  "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
  "company_id": 5,
  "slug": "small_plan"
}
```

```
Response body (application/json): "OK"
```

Request:

- **company_id**: The id of the company you wish to reactivate this Subscription for.
- **slug**: alphabetic unique identifier for the product to reactivate (see `fetch_products` above)

Response: "OK"

5.5 Cancel Subscription

ResellerAdmin Only

Canceling a subscription is a permanent deactivation of that subscription. Hosts and Companies associated with that subscription can be subscribed to new subscriptions at any time but that particular subscription can not be reactivated once cancelled.

```
POST "/enterprise_api/subscription/cancel"
```

```
Request body (application/json):
```

```
{
  "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
  "company_id": 5,
  "slug": "small_plan"
}
```

```
Response body (application/json): "OK"
```

Request:

- **company_id**: The id of the company you wish to cancel this Subscription for/from.
- **slug**: alphabetic unique identifier for the product to cancel (see fetch_products above)

Response: "OK"

5.6 Fetch Subscriptions

ResellerAdmin Only

```
POST "/enterprise_api/subscription/fetch"
```

```
Request body (application/json):
```

```
{
  "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
  "company_id": 5
}
```

```
Response body (application/json):
```

```
[
  {
    "id": 100,
    "status": "Active",
    "host_limit": 10,
    "product": {
      "slug": "small_plan",
      "name": "Small Conference Plan",
      "plan": true
    }
  },
  {
    "id": 101,
    "status": "Active",
    "host_limit": 15,
    "product": {
```

```
    "slug": "recording",
    "name": "Recording Add-On",
    "plan": false
  }
}
```

Request:

- **company_id**: The id of the company you wish to fetch subscriptions for.

Response, array of:

- **id**: integer value uniquely identifying this subscription, globally.
- **host_limit**: integer indicating the assigned host limit (used for per-host priced subscriptions only)
- **product**
 - **slug**: an alphabetic unique identifier for the product
 - **name**: a human-readable name of the product
 - **plan**: boolean, whether the product is a plan [true] or an add-on [false]
- **status**: 'Active' or 'Suspended' (previously-cancelled subscriptions not included)

6.0 Hosts

A host is an account able to organize and host meetings and conference calls.

6.1 Create Host Account

ResellerAdmin/CompanyAdmin

Create a new Host

```
POST "/enterprise_api/host/create"
```

Request **body** (application/json):

```
{
  "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
  "company_id": 5,
  "name": "John Host",
  "email": "john@company.tld",
```

```
    "password": "SoSecret5!",
    "no_welcome_email": true,
    "role": "HOST",
    "openid_sub": "23081912eca",
    "subscription_ids": [102, 104],
    "moderator_pin": 9876543,
    "access_code": 1234567,
    "vivomeeting_streaming_enabled": true,
    "host_initiated_recording": true,
    "disable_emails": { "conference_invitations": true,
"conference_reminders": false, "conference_summaries": false,
"transcription_summaries": false }
}
```

Response **body** (application/json):

```
{
    "host_id": 9,
    "name": "John Host",
    "email": "john@company.tld",
    "primary_dialin_number": {"phone_number": "14165551234",
    "domestic_format": "(416) 555-1234",
    "international_format": "+1 416-555-1234",
    "country_code": "CA",
    "location": "Toronto",
    "id": null, "created_at": null, "updated_at": null,
    "feature_id": null, "locale": null, "carrier": null,
    "state": null, "billing_group": null, "display": null,
    "organization_id": null, "grouped_billing_id": null,
    "site_domain_id": null, "billing_code": null, "active": null,
    "referral_number": null },
    "meeting_url": "https://company.reseller.tld/conf/call/1234567",
    "role": "HOST",
    "openid_sub": "23081912eca",
    "access_code": { "number": "1234567",
    "id": null, "participant_id": null,
    "lock_version": null, "updated_at": null,
    "conference_call_id": null, "person_id": null,
    "account_id": null, "recording_id": null,
    "site_domain_id": null, "organization_id": null },
    "moderator_pin": {"number": "9876543",
    "id": null, "participant_id": null,
    "lock_version": null, "updated_at": null,
```

```

        "conference_call_id": null, "person_id": null,
        "account_id": null, "recording_id": null,
        "site_domain_id": null, "organization_id": null },
    "password_creation_link": "https://...",
    "login_token_public_key": "a1b2c3e4d5f6...",
        "plan_slug": "small_plan",
        "add_on_slugs": ["recording"],
    "subscription_ids": [110, 111],
        "sip_uri": "sip:1234567-0@sipgw.reseller.com",
        "disable_emails": { "conference_invitations": true,
"conference_reminders": false, "conference_summaries": false,
"transcription_summaries": false }
}

```

Request:

- **company_id**: The id of the company you wish to add the Host to. (Required when the API consumer is a Reseller Admin; ignored when a Company Admin: host will be created in Company Admin's own company)
- **name**: The name of Host that will be displayed when joining meetings.
- **email**: The email address of the Host. This will be used for invitations and reminders as well as authentication.
- **password**: optional, to set on creation
- **no_welcome_email**: whether to *not* send the host a welcome email (optional, boolean, should probably be used with the above, and/or see password_creation_link in the Response)
- **role**: one of RESELLER_ADMIN, COMPANY_ADMIN, HOST - Only Reseller Admins can grant the Reseller Admin role; Reseller and Company Admins can grant the Company Admin or Host roles. Defaults to HOST if omitted.
- **openid_sub**: Optional. the OpenID subject identifier used for OAuth logins using your system rather than our login infrastructure. If your platform does not support OpenID OAuth, please ignore this parameter.
- **moderator_pin**: Optional. Providing this parameter will request a custom moderator pin for this account. If the number is already taken, the request will fail. If omitted, a number will be generated automatically.
- **access_code**: Optional. Providing this parameter will request a custom access code for this account. If the number is already taken, the request will fail. If omitted, a number will be generated automatically.
- **vivommeting_streaming_enabled**: Optional. Providing this parameter will enable live streaming for the host account on account creation. This bypasses the need to enable it through the account settings.
- **host_initiated_recording**: Optional. Providing this parameter will change the trigger for starting recording or streaming to the host joining instead of having at least 2 participants in the conference.

- **subscription_ids:** Optional. The list of subscription ids that define which products the host is subscribed to.
 - Providing this property will set the host's subscriptions to the exact list provided. This means you can unsubscribe hosts from their subscriptions by omitting specific ids or by providing an empty array.
 - If this property is omitted, the host's existing subscriptions will be left intact.
 - IMPORTANT: The host limit for each subscription will automatically be incremented if required, so that the host limit remains greater than or equal to the number of hosts assigned to each subscription. Except in the case where hosts limit changes have been locked for the company, in which case an exception is raised.
 - NOTE:
 - Instead of passing `subscription_ids`, you can alternatively pass the following two optional arguments:
 - **plan_slug** the slug of the plan to assign the host to.
 - **add_on_slugs:** an array of product slugs of add-ons to assign the host to.
- **disable_emails:** Optional. Specifies which emails are enabled or disabled for the host, options are `conference_invitations`, `conference_reminders`, `conference_summaries`, `transcription_summaries` and these are set to true to disable and false to enable.

Response:

- **host_id:** The unique id of the host. This should be stored for all future APIs call on that host.
- **name:** The name of Host that will be displayed when joining meetings.
- **email:** The email address of the Host. This will be used for invitations and reminders as well as authentication.
- **primary_dialin_number:** The default dial-in number for the Hosts meeting invitations and reminders (`primary_dialin_number` is an object; `.phone_number` (raw e164), `.domestic_format` and `.international_format` represent the actual number to dial; choose the format that best suits your need; `country_code` and `city` are also included in case you wish to display to the end user; other properties can be safely ignored but are included here for the benefit of strict deserializers).
- **meeting_url:** The meeting url is the web location of the Host's meeting room
- **role:** one of RESELLER_ADMIN, COMPANY_ADMIN, HOST
- **access_code:** to be used by non-moderator conference participants (`access_code` is an object; `.number` is the actual access code; other properties can be safely ignored but are included here for the benefit of strict deserializers)
- **moderator_pin:** to be used by moderator (`moderator_pin` is an object; `.number` is the actual moderator_pin; other properties can be safely ignored but are included here for the benefit of strict deserializers)

- **password_creation_link**: can be sent by resellers themselves to new hosts to allow setting password (most useful in conjunction with `no_welcome_email: true` in the request)
- **login_token_public_key**: this is what you will need to submit to us to use our SSO feature. See section 8.0 for details.
- **plan_slug**: slug (unique identifier, see `subscription/fetch_products` above) for the plan the host is subscribed to
- **add_on_slugs**: slug[s] for the add-on[s], if any, the host is subscribed to
- **subscription_ids**: An array of integers identifying the plan/add-on subscriptions that the host is currently assigned to.
- **sip_uri**: If the host is subscribed to the SIP Gateway feature, this sip URI can be used by SIP endpoints such as those from Cisco and Polycom to connect to the host's meetings; null otherwise. Contact us to enable SIP Gateway.
- **disable_emails**: Shows the email settings for the host.

6.2 Get Host Account Details

`/enterprise_api/host/fetch`
ResellerAdmin/CompanyAdmin

Fetch the details of a particular host

```
POST "/enterprise_api/host/fetch"

Request body (application/json):
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "host_id": 9
}

Response body (application/json):
{
    "host_id": 9,
    "name": "John Host",
    "email": "john@company.tld",
    "primary_dialin_number": {"phone_number": "14165551234",
        "domestic_format": "(416) 555-1234",
        "international_format": "+1 416-555-1234",
        "country_code": "CA",
        "location": "Toronto",
    "id": null, "created_at": null, "updated_at": null,
    "feature_id": null, "locale": null, "carrier": null,
```

```

    "state": null, "billing_group": null, "display": null,
    "organization_id": null, "grouped_billing_id": null,
    "site_domain_id": null, "billing_code": null, "active": null,
    "referral_number": null },
    "meeting_url": "https://company.reseller.tld/conf/call/1234567",
    "role": "HOST",
    "openid_sub": "23081912eca",
    "access_code": { "number": "1234567",
    "id": null, "participant_id": null,
    "lock_version": null, "updated_at": null,
    "conference_call_id": null, "person_id": null,
    "account_id": null, "recording_id": null,
    "site_domain_id": null, "organization_id": null },
    "moderator_pin": {"number": "9876543",
    "id": null, "participant_id": null,
    "lock_version": null, "updated_at": null,
    "conference_call_id": null, "person_id": null,
    "account_id": null, "recording_id": null,
    "site_domain_id": null, "organization_id": null },
    "password_creation_link": "https://...",
    "login_token_public_key": "a1b2c3e4d5f6...",
    "plan_slug": "small_plan",
    "add_on_slugs": ["recording"],
    "subscription_ids": [110, 111],
    "sip_uri": "sip:1234567-0@sipgw.reseller.com",
    "disable_emails": { "conference_invitations": true,
    "conference_reminders": false, "conference_summaries": false,
    "transcription_summaries": false }
}

```

Request:

- **host_id**: unique identifier of the host

Response:

- **host_id**: The unique id of the host. This should be stored for all future APIs call on that host.
- **name**: The name of Host that will be displayed when joining meetings.
- **email**: The email address of the Host. This will be used for invitations and reminders as well as authentication.
- **primary_dialin_number**: The default dial-in number for the Hosts meeting invitations and reminders (primary_dialin_number is an object; .phone_number

(raw e164), `.domestic_format` and `.international_format` represent the actual number to dial; choose the format that best suits your need; `country_code` and `city` are also included in case you wish to display to the end user; other properties can be safely ignored but are included here for the benefit of strict deserializers).

- **meeting_url**: The meeting url is the web location of the Host's meeting room
- **role**: one of RESELLER_ADMIN, COMPANY_ADMIN, HOST
- **access_code**: to be used by non-moderator conference participants (`access_code` is an object; `.number` is the actual access code; other properties can be safely ignored but are included here for the benefit of strict deserializers)
- **moderator_pin**: to be used by moderator (`moderator_pin` is an object; `.number` is the actual moderator_pin; other properties can be safely ignored but are included here for the benefit of strict deserializers)
- **password_creation_link**: can be sent by resellers themselves to new hosts to allow setting password (most useful in conjunction with `no_welcome_email`: true in the request)
- **login_token_public_key**: this is what you will need to submit to us to use our SSO feature. See section 8.0 for details.
- **plan_slug**: slug (unique identifier, see `subscription/fetch_products` above) for the plan the host is subscribed to
- **add_on_slugs**: slug[s] for the add-on[s], if any, the host is subscribed to
- **subscription_ids**: An array of integers identifying the plan/add-on subscriptions that the host is currently assigned to.
- **sip_uri**: If the host is subscribed to the SIP Gateway feature, this sip URI can be used by SIP endpoints such as those from Cisco and Polycom to connect to the host's meetings; null otherwise. Contact us to enable SIP Gateway.
- **openid_sub**: The OpenID subject identifier used for OAuth logins using your system rather than our login infrastructure. If your platform does not support OpenID OAuth, this will be blank.
- **disable_emails**: Shows the email settings for the host.

6.3 Edit Host Account

`/enterprise_api/host/update`

ResellerAdmin/CompanyAdmin

Only `host_id` is required; all other input parameters optional. Omitted host properties will remain unchanged.

```
POST "/enterprise_api/host/update"
```

```
Request body (application/json):
```

```
{
  "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
  "host_id": 9,
  "name": "John ChangedName",
```

```
        "email": "newemail@company.tld",
        "role": "COMPANY_ADMIN",
        "openid_sub": "23051912fbd",
        "subscription_ids": [102, 104],
        "moderator_pin": 9876543,
        "access_code": 1234567,
        "disable_emails": { "conference_invitations": true,
"conference_reminders": false, "conference_summaries": false,
"transcription_summaries": false }
    }
}
```

Response **body** (application/json):

```
{
    "host_id": 9,
    "name": "John ChangedName",
    "email": "newemail@company.tld",
    "primary_dialin_number": {"phone_number": "14165551234",
        "domestic_format": "(416) 555-1234",
        "international_format": "+1 416-555-1234",
        "country_code": "CA",
        "location": "Toronto",
        "id": null, "created_at": null, "updated_at": null,
        "feature_id": null, "locale": null, "carrier": null,
        "state": null, "billing_group": null, "display": null,
        "organization_id": null, "grouped_billing_id": null,
        "site_domain_id": null, "billing_code": null, "active": null,
        "referral_number": null },
    "meeting_url": "https://company.reseller.tld/conf/call/1234567",
    "role": "COMPANY_ADMIN",
    "openid_sub": "23051912fbd",
    "access_code": { "number": "1234567",
        "id": null, "participant_id": null,
        "lock_version": null, "updated_at": null,
        "conference_call_id": null, "person_id": null,
        "account_id": null, "recording_id": null,
        "site_domain_id": null, "organization_id": null },
    "moderator_pin": {"number": "9876543",
        "id": null, "participant_id": null,
        "lock_version": null, "updated_at": null,
        "conference_call_id": null, "person_id": null,
        "account_id": null, "recording_id": null,
        "site_domain_id": null, "organization_id": null },
}
```

```
"password_creation_link": "https://...",
"login_token_public_key": "a1b2c3e4d5f6...",
  "plan_slug": "small_plan",
  "add_on_slugs": ["recording"],
  "subscription_ids": [110, 111],
  "sip_uri": "sip:1234567-0@sipgw.reseller.com",
  "disable_emails": { "conference_invitations": true,
"conference_reminders": false, "conference_summaries": false,
"transcription_summaries": false }
}
```

Request:

- **host_id**: id of the host
- **name**: The name of Host that will be displayed when joining meetings.
- **email**: The email address of the Host. This will be used for invitations and reminders as well as authentication.
- **role**: one of RESELLER_ADMIN, COMPANY_ADMIN, HOST - Only Reseller Admins can grant the Reseller Admin role; Reseller and Company Admins can grant the Company Admin or Host roles.
- **moderator_pin**: optional. Providing this parameter will request a custom moderator pin for this account. If the number is already taken, the request will fail. If omitted, a number will be generated automatically.
- **access_code**: optional. Providing this parameter will request a custom access code for this account. If the number is already taken, the request will fail. If omitted, a number will be generated automatically.
- **openid_sub**: Optional. the OpenID subject identifier used for OAuth logins using your system rather than our login infrastructure. If your platform does not support OpenID OAuth, please ignore this parameter.
- **subscription_ids**: Optional. The list of subscription ids that define which products the host is subscribed to.
 - Providing this property will set the host's subscriptions to the exact list provided. This means you can unsubscribe hosts from their subscriptions by omitting certain ids or by providing an empty array.
 - If this property is omitted, the host's existing subscriptions will be left intact.
 - IMPORTANT: The host limit for each subscription will automatically be incremented if required, so that the host limit remains greater than or equal to the number of hosts assigned to each subscription. Except in the case where hosts limit changes have been locked for the company, in which case an exception is raised.
 - NOTE:
 - Instead of passing `subscription_ids`, you can alternatively pass the following two optional arguments:

- **plan_slug** the slug of the plan to assign the host to.
 - **add_on_slugs**: an array of product slugs of add-ons to assign the host to.
- **disable_emails**: Optional. Specifies which emails are enabled or disabled for the host, options are conference_invitations, conference_reminders, conference_summaries, transcription_summaries and these are set to true to disable and false to enable.

Response:

- **name**: The name of Host that will be displayed when joining meetings.
- **email**: The email address of the Host. This will be used for invitations and reminders as well as authentication.
- **primary_dialin_number**: The default dial-in number for the Hosts meeting invitations and reminders (primary_dialin_number is an object; .phone_number (raw e164), .domestic_format and .international_format represent the actual number to dial; choose the format that best suits your need; country_code and city are also included in case you wish to display to the end user; other properties can be safely ignored but are included here for the benefit of strict deserializers).
- **meeting_url**: The meeting url is the web location of the Host's meeting room
- **role**: one of RESELLER_ADMIN, COMPANY_ADMIN, HOST
- **access_code**: to be used by non-moderator conference participants (access_code is an object; .number is the actual access code; other properties can be safely ignored but are included here for the benefit of strict deserializers)
- **moderator_pin**: to be used by moderator (moderator_pin is an object; .number is the actual moderator_pin; other properties can be safely ignored but are included here for the benefit of strict deserializers)
- **password_creation_link**: can be sent by resellers themselves to new hosts to allow setting password (most useful in conjunction with no_welcome_email: true in the request)
- **login_token_public_key**: this is what you will need to submit to us to use our SSO feature. See section 8.0 for details.
- **plan_slug**: slug (unique identifier, see subscription/fetch_products above) for the plan the host is subscribed to
- **add_on_slugs**: slug[s] for the add-on[s], if any, the host is subscribed to
- **subscription_ids**: An array of integers identifying the plan/add-on subscriptions that the host is currently assigned to.
- **sip_uri**: If the host is subscribed to the SIP Gateway feature, this sip URI can be used by SIP endpoints such as those from Cisco and Polycom to connect to the host's meetings; null otherwise. Contact us to enable SIP Gateway.
- **openid_sub**: The OpenID subject identifier used for OAuth logins using your system rather than our login infrastructure. If your platform does not support OpenID OAuth, this will be blank.
- **disable_emails**: Shows the email settings for the host.

6.4 Delete Host Account

ResellerAdmin/CompanyAdmin

```
POST "/enterprise_api/host/delete"
```

Request **body** (application/json):

```
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "host_id": 8
}
```

Request:

- **host_id**: unique identifier of the host

6.5 Welcome Email

ResellerAdmin/CompanyAdmin

Re-send the the welcome email of the Host

```
POST "/enterprise_api/host/resend_invitation"
```

Request **body** (application/json):

```
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "host_id": 8
}
```

Request:

- **host_id**: unique identifier of the host

6.6 Reset Password

ResellerAdmin/CompanyAdmin

Send a Host a password reset email

```
POST "/enterprise_api/host/reset_password"
```



```
Request body (application/json):
```

```
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "host_id": 8
}
```

Request:

- **host_id**: unique identifier of the host

6.7 Host Conferences

ResellerAdmin/CompanyAdmin

Details of the past conferences of the host.

```
POST "/enterprise_api/host/conferences"
```

```
Request body (application/json):
```

```
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "host_id": 8,
    "output": "summary",
    "time_zone": "UTC",
    "start_date": "2020-01-01",
    "end_date": "2020-12-31"
}
```

```
Response body (application/json):
```

```
[
    {
        "conference_id": 256,
        "start_time": "2020-09-29 11:10:00",
        "end_time": "2020-09-29 11:20:00",
        "total_participants": 5,
        "total_participant_minutes": 20,
        "meeting_summary_url": "https://test.domain.com/conf/summary/256"
    }
]
```

Request:

- **host_id**: unique identifier of the host
- **output**: summary or detail, type of data to be returned. Optional defaults to summary if omitted.
- **time_zone**: Time zone for the start/end date and for the times in the response. Optional, defaults to UTC if omitted. Values are those (e.g. US/Central, Europe/London, Asia/Tokyo, etc.) in the IANA Time Zone database (<https://www.iana.org/time-zones>)
- **start_date**: Filter by date. Optional.
- **end_date**: Filter by date: Optional.

Response:**Summary:**

- **conference_id**: Unique conference ID
- **start_time**: Conference start date/time, if **time_zone** was specified in the request will be in the requested **time_zone**, if omitted will be in UTC. Values are those (e.g. US/Central, Europe/London, Asia/Tokyo, etc.) in the IANA Time Zone database (<https://www.iana.org/time-zones>)
- **end_time**: Conference end date/time.
- **total_participants**: Total conference participants
- **total_participant_minutes**: Total minutes for all participants
- **meeting_summary_url**: URL to retrieve the meeting summary

Detail:

The detail output provides comprehensive data on the conference and the returned data will depend on the conference activity which includes:

- **Conference_id**: Unique conference ID
- **Subject**: Subject
- **Agenda**: Agenda
- **Start_time**: Date/time in the specified **time_zone** or defaults to UTC. Values are those (e.g. US/Central, Europe/London, Asia/Tokyo, etc.) in the IANA Time Zone database (<https://www.iana.org/time-zones>)
- **End_time**: Date/time
- **Day**: Day of the conference
- **Month**: Month of the conference
- **Year**: Year of the conference
- **Cancelled**: true or false, true if the conference was cancelled.
- **Participants**: Number of participants
- **Live_wall_posts**: List of comments made during the conference

- **Uploaded_documents:** List of uploaded documents
- **Recordings:** List of recordings

6.8 Host CDRs

ResellerAdmin/CompanyAdmin

Details of conference participants.

```
POST "/enterprise_api/host/cdrs"
```

```
Request body (application/json):
```

```
{
  "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
  "host_id": 8,
  "conference_id": 256,
  "time_zone": "UTC",
  "start_date": "2020-01-01",
  "end_date": "2020-12-31"
}
```

```
Response body (application/json):
```

```
[
  {
    "id": 69226060,
    "conference_id": 256,
    "host_id": 8,
    "host_name": "Host Name",
    "host_email": "host@vivomeetings.com",
    "start_time": "2020-09-29 11:10:00",
    "end_time": "2020-09-29 11:20:00",
    "participant_id": 71974494,
    "participant_name": "Participant Name 1",
    "participant_email": "participant1@vivomeetings.com",
    "from": "+15551236789",
    "to": "2342342",
    "minutes": 880
  },
  {
    "id": 69226061,
    "conference_id": 256,
    "host_id": 8,
```

```

        "host_name": "Host Name",
        "host_email": "host@vivomeetings.com",
        "start_time": "2020-09-29 11:10:00",
        "end_time": "2020-09-29 11:20:00",
        "participant_id": 71974494,
        "participant_name": "Participant Name 2",
        "participant_email": "participant2@vivomeetings.com",
        "from": "+15551236798",
        "to": "2342342",
        "minutes": 880
    }
]

```

Request:

- **host_id**: unique identifier of the host
- **conference_id**: unique identifier of a conference, Optional
- **time_zone**: Time zone for the start/end date and for the times in the response. Optional, defaults to UTC if omitted. Values are those (e.g. US/Central, Europe/London, Asia/Tokyo, etc.) in the IANA Time Zone database (<https://www.iana.org/time-zones>)
- **start_date**: Filter by date. Optional.
- **end_date**: Filter by date: Optional.

Response:

- **Id**: Unique CDR ID
- **conference_id**: Unique conference ID
- **host_id**: Unique host ID
- **host_name**: Host name
- **host_email**: Host email
- **start_time**: Participant's start date/time, if time_zone was specified in the request will be in the requested time_zone, if omitted will be in UTC. Values are those (e.g. US/Central, Europe/London, Asia/Tokyo, etc.) in the IANA Time Zone database (<https://www.iana.org/time-zones>)
- **end_time**: Conference end date/time.
- **participant_id**: Unique participant ID
- **participant_name**: Participant name
- **participant_email**: Participant email
- **from**: Number participant originates from
- **to**: Number participant connected to
- **minutes**: Total minutes for the participant

Note: Only the first 100 records will be returned in the response.

6.9 Active Calls

List of calls which a given person is currently participating in.

```
POST "/enterprise_api/host/active_calls"

Request body (application/json):
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "host_id": 8,
    "time_zone": "UTC"
}

Response body (application/json):
[
    {
        "conference_id": 256,
        "subject": "Call Subject",
        "agenda": "Call Agenda",
        "start_time": "2020-09-29 11:10:00"
    }
]
```

Request:

- **host_id**: unique identifier of the host

Response:

- **conference_id**: Unique conference ID
- **subject**: Call's subject
- **agenda**: Call's agenda
- **start_time**: Call's start date/time. If `time_zone` was specified in the request will be in the requested `time_zone`, if omitted will be in UTC. Values are those (e.g. US/Central, Europe/London, Asia/Tokyo, etc.) in the IANA Time Zone database (<https://www.iana.org/time-zones>)

6.10 Fetch all hosts in a company

Get a summary of all the hosts in a company.

```
POST "/enterprise_api/host/fetch_all"

Request body (application/json):
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "company_id": 8
}

Response body (application/json):
[
    {
        "host_id": 9,
        "name": "John Host",
        "email": "john@company.tld",
        "meeting_url": "https://company.reseller.tld/conf/call/1234567",
        "role": "COMPANY_ADMIN",
        "access_code": "1234567"
    },
    {
        "host_id": 10,
        "name": "Jane Host",
        "email": "jane@company.tld",
        "meeting_url": "https://company.reseller.tld/conf/call/1234568",
        "role": "HOST",
        "access_code": "1234568"
    }
]
```

Request:

- **company_id**: The id of the company whose hosts you wish to fetch. (Required when the API consumer is a Reseller Admin; ignored when a Company Admin: hosts will be fetched for Company Admin's own company)

Response: array of

- **host_id**: unique identifier of the host
- **name**: The name of Host that will be displayed when joining meetings.

- **email**: The email address of the Host. This will be used for invitations and reminders as well as authentication.
- **meeting_url**: The meeting url is the web location of the Host's meeting room
- **role**: one of RESELLER_ADMIN, COMPANY_ADMIN, HOST
- **access_code**: to be used by non-moderator conference participants, the access code itself as a string (not an object as with fetch)

7.0 Contacts

Contacts can be added to scheduled conferences.

7.1 Create Contacts

/enterprise_api/contact/create
ResellerAdmin/CompanyAdmin

Create contacts for a Host. This function allows the creation of multiple contacts at the same time.

```
POST "/enterprise_api/contact/create"

Request body (application/json):
{
  "auth_token": "8f946001-02b3-442b-9050-65b93c193f07",
  "host_id": 31,
  "contacts": [
    {
      "name": "John Smith",
      "email": "john@smith.com",
      "business_phone": "+12345678901",
      "mobile_phone": "+12345678902",
      "home_phone": "+12345678903",
      "other_phone": "+12345678904",
      "time_zone": "US/Eastern"
    },
    {
      "name": "James Brown",
      "email": "james@brown.com",
      "business_phone": "+12345678905",
      "mobile_phone": "+12345678906",
      "home_phone": "+12345678907",
```

```
        "other_phone": "+12345678908",
        "time_zone": "US/Central"
    }
]
}
```

Response **body** (application/json):

```
[
  {
    "contact_id": 35,
    "name": "John Smith",
    "email": "john@smith.com",
    "business_phone": "+12345678901",
    "mobile_phone": "+12345678902",
    "home_phone": "+12345678903",
    "other_phone": "+12345678904",
    "time_zone": "US/Eastern",
    "has_host": false
  },
  {
    "contact_id": 37,
    "name": "James Brown",
    "email": "james@brown.com",
    "business_phone": "+12345678905",
    "mobile_phone": "+12345678906",
    "home_phone": "+12345678907",
    "other_phone": "+12345678908",
    "time_zone": "US/Central",
    "has_host": false
  }
]
```

Request:

- **host_id**: The id of the host you wish to add the Contacts to. (Required when the API consumer is a Reseller Admin; ignored when a Company Admin: host will be created in Company Admin's own company)

A list of contacts is supplied:

- **name**: The name of the Contact
- **email**: The email address of the Contact.
- **business_phone**: optional, business phone number

- **mobile_phone**: optional, mobile phone number
- **home_phone**: optional, home phone number
- **other_phone**: optional, other phone number
- **time_zone**: optional, IANA time zone name. Values are those (e.g. US/Central, Europe/London, Asia/Tokyo, etc.) in the IANA Time Zone database (<https://www.iana.org/time-zones>)

Response:

A list of contacts is returned containing:

- **contact_id**: Unique id of the Contact.
- **name**: Name of the Contact.
- **email**: Email address of the contact
- **business_phone**: Business phone number
- **mobile_phone**: Mobile phone number
- **home_phone**: Home phone number
- **other_phone**: Other phone number
- **time_zone**: Time zone for the contact
- **has_host**: Indicates whether the contact is a host's contact details when true or a standalone contact when false.

7.2 Get Contact Details

/enterprise_api/contact/fetch
ResellerAdmin/CompanyAdmin

Fetch the details of a particular contact

```
POST "/enterprise_api/contact/fetch"

Request body (application/json):
{
    "auth_token": "25af2fe2-010e-4868-8be8-802b100d34a7",
    "host_id": 31,
    "contact_id": 35
}

Response body (application/json):
{
    "contact_id": 35,
    "name": "John Smith",
    "email": "john@smith.com",
```

```
    "business_phone": "+12345678901",
    "mobile_phone": "+12345678902",
    "home_phone": "+12345678903",
    "other_phone": "+12345678904",
    "time_zone": "US/Eastern",
    "has_host": false
}
```

Request:

- **host_id**: Unique identifier of the host
- **contact_id**: Unique identifier of the contact

Response:

- **contact_id**: Unique id of the contact.
- **name**: Name of the contact.
- **email**: Email address of the contact.
- **business_phone**: Business phone number
- **mobile_phone**: Mobile phone number
- **home_phone**: Home phone number
- **other_phone**: Other phone number
- **time_zone**: Time zone for the contact. Values are those (e.g. US/Central, Europe/London, Asia/Tokyo, etc.) in the IANA Time Zone database (<https://www.iana.org/time-zones>)
- **has_host**: Indicates whether the contact is a host's contact details when true or a standalone contact when false.

7.3 Get Contact Details for Host

/enterprise_api/contact/fetch_all
ResellerAdmin/CompanyAdmin

Fetch all contact details of a particular host

```
POST "/enterprise_api/contact/fetch_all"
```

```
Request body (application/json):
```

```
{
    "auth_token": "25af2fe2-010e-4868-8be8-802b100d34a7",
    "host_id": 31
}
```

```
Response body (application/json):
```

```
[
  {
    "contact_id": 35,
    "name": "John Smith",
    "email": "john@smith.com",
    "business_phone": "+12345678901",
    "mobile_phone": "+12345678902",
    "home_phone": "+12345678903",
    "other_phone": "+12345678904",
    "time_zone": "US/Eastern",
    "has_host": false
  },
  {
    "contact_id": 37,
    "name": "James Brown",
    "email": "james@brown.com",
    "business_phone": "+12345678905",
    "mobile_phone": "+12345678906",
    "home_phone": "+12345678907",
    "other_phone": "+12345678908",
    "time_zone": "US/Central",
    "has_host": false
  }
]
```

Request:

- **host_id**: Unique identifier of the host

Response:

Returns an array of contacts:

- **contact_id**: Unique id of the contact
- **name**: Name of the contact
- **email**: Email address of the contact
- **business_phone**: Business phone number
- **mobile_phone**: Mobile phone number
- **home_phone**: Home phone number
- **other_phone**: Other phone number
- **time_zone**: Time zone for the contact. Values are those (e.g. US/Central, Europe/London, Asia/Tokyo, etc.) in the IANA Time Zone database (<https://www.iana.org/time-zones>)

- **has_host**: Indicates whether the contact is a host's contact details when true or a standalone contact when false.

7.4 Edit Contact

/enterprise_api/contact/update
ResellerAdmin/CompanyAdmin

Only contact_id is required; all other input parameters optional. Omitted contact properties will remain unchanged.

```
POST "/enterprise_api/contact/update"
```

```
Request body (application/json):
```

```
{
  "auth_token": "25af2fe2-010e-4868-8be8-802b100d34a7",
  "host_id": 31,
  "contact_id": 35,
  "name": "John Smith Updated",
  "business_phone": "+12345678901",
  "mobile_phone": "+12345678902",
  "home_phone": "+12345678903",
  "other_phone": "+12345678904",
  "time_zone": "US/Eastern"
}
```

```
Response body (application/json):
```

```
{
  "contact_id": 35,
  "name": "John Smith Updated",
  "email": "john@smith.com",
  "business_phone": "+12345678901",
  "mobile_phone": "+12345678902",
  "home_phone": "+12345678903",
  "other_phone": "+12345678904",
  "time_zone": "US/Eastern"
}
```

Request:

- **host_id**: unique identifier of the host
- **contact_id**: unique id of the contact
- **name**: Contact Name

- **email:** Contact Email
- **time_zone:** optional, IANA time zone name. Values are those (e.g. US/Central, Europe/London, Asia/Tokyo, etc.) in the IANA Time Zone database (<https://www.iana.org/time-zones>)

Response:

- **contact_id:** unique id of the contact
- **name:** Contact Name
- **email:** Contact Email
- **business_phone:** Business phone number
- **mobile_phone:** Mobile phone number
- **home_phone:** Home phone number
- **other_phone:** Other phone number
- **time_zone:** IANA time zone name

7.5 Delete Contact

/enterprise_api/contact/delete
ResellerAdmin/CompanyAdmin

```
POST "/enterprise_api/contact/delete"
```

```
Request body (application/json):
```

```
{
  "auth_token": "25af2fe2-010e-4868-8be8-802b100d34a7",
  "host_id": 31,
  "contact_id": 37
}
```

Request:

- **host_id:** unique identifier of the host
- **contact_id:** unique id of the contact

8.0 Conferences

The details below are for the initial release of the conference API functions. You get the calendar integration and advanced Vivomeetings features but integrated into your own scheduling application. This will allow you to schedule meetings with advanced features like

- One-time access codes
- Security PINs
- Recurring Meetings (not yet implemented)
- Default dial-in numbers (not yet implemented)

8.1 Schedule Conference

ResellerAdmin/CompanyAdmin

Create a new scheduled conference

```
POST "/enterprise_api/conference/create"
```

```
Request body (application/json):
```

```
{
  "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
  "host_id": 1,
  "subject": "Test Conference",
  "agenda": "An agenda",
  "start": "2020-09-19 13:00:00",
  "time_zone": "Europe/London",
  "duration": 60,
  "auto_record": "none",
  "auto_stream": "none",
  "auto_transcribe": false,
  "one_time_access_code": true,
  "security_pin": "123456",
  "mute_mode": "conversation",
  "participants": [ 2, 3 ]
}
```

```
Response body (application/json):
```

```
{
  "conference_id": 183,
  "subject": "Test Conference",
  "agenda": "An agenda",
  "start": "2020-09-19 13:00:00",
  "end_time": "2020-09-19 14:00:00",
  "active": false
  "time_zone": "Europe/London",
  "duration": 60,
  "auto_record": "none",
  "auto_stream": "none",
  "auto_transcribe": false,
  "one_time_access_code": 987654321,
  "security_pin": "123456",
}
```

```
    "mute_mode": "conversation",
    "room_url":
      "https://subdomain.domain.tld/conf/call/987654321",
    "recordings": [],
    "participants": [ 1, 2, 3 ],
    "moderators": [ 1 ],
    "is_streaming": false,
  }
```

Request:

- **host_id**: unique identifier of the host
- **subject**: Subject
- **agenda**: Agenda
- **start**: Start time, this works in conjunction with the `time_zone` to set the correct time.
- **time_zone**: Time zone associated with the start time. Values are those (e.g. US/Central, Europe/London, Asia/Tokyo, etc.) in the IANA Time Zone database (<https://www.iana.org/time-zones>)
- **duration**: In minutes.
- **auto_record**: Optional. Indicates whether the conference should be recorded automatically and the format, allowable values are: none, audio, video. If omitted will default to none. Both record and stream must be the same value if recording and streaming together.
- **auto_stream**: Optional. Indicates whether the conference should be streamed automatically and the format, allowable values are: none, audio, video. If omitted will default to none. Both record and stream must be the same value if recording and streaming together.
- **auto_transcribe**: Optional. Whether the recording should automatically be transcribed. Valid values are true or false. If omitted will default to false.
- **one_time_access_code**: true or false, when set to true an access code will be returned in the response.
- **security_pin**: Pin required to access the conference
- **mute_mode**: Optional. Sets the initial participant mute settings. Options are:
 - conversation - Default, all unmuted
 - q&a - Participants muted, but can unmute
 - presentation - Participants muted and can not unmuteIf omitted will default to conversation.
- **participants**: List of contact IDs to be included as participants.
- **participant_emails**: Alternatively participants can be listed via emails with moderators specified. If a participant's emails is not on the host's contact list, it will be added:
- **[UNRELEASED] waiting_room**: true or false, when set to true any participants will need to be let in by the organizer

- **[UNRELEASED] secure_url**: true or false, when set to true the one_time_access_code field will be encrypted

```
POST "/enterprise_api/conference/create"
```

```
Request body (application/json):
```

```
{
  "participant_emails": [
    {"email": "bob@xxxx.com", "moderator": true},
    {"email": "alice@xxxx.com", "moderator": false}
  ]
}
```

Response:

- **conference_id**: id of the conference
- **subject**: Subject
- **agenda**: Agenda
- **start**: Start time, this works in conjunction with the time_zone to indicate the correct time.
- **end_time**: The scheduled end time, according to the specified duration.
- **active**: Whether the call is active (is scheduled for the current time or has live participants)
- **time_zone**: Time zone associated with the start time. Values are those (e.g. US/Central, Europe/London, Asia/Tokyo, etc.) in the IANA Time Zone database (<https://www.iana.org/time-zones>)
- **duration**: In minutes.
- **auto_record**: Indicates whether the conference will be recorded automatically and the format.
- **auto_stream**: Indicates whether the conference will be streamed automatically and the format.
- **auto_transcribe**: Whether the recording will be transcribed automatically.
- **one_time_access_code**: When the request has one_time_access_code set to true this field contains the conference access code.
- **security_pin**: Pin required to access the conference.
- **mute_mode**: The initial participant mute settings. Values as per the request.
- **participants**: List of participants in the conference, note the organizer is always included in the list. These are contact_id's that can be used with contact/fetch and other contact methods.
- **moderators**: List of moderators in the conference. These are contact_id's that can be used with contact/fetch and other contact methods.

- **room_url**: the URL for the meeting room. Can be navigated to in a browser to join the meeting.
- **recordings**: information about recordings made during the conference. For create, update, create/reservationless, will be an empty array [], as these return information about future conferences; however, see conference/fetch below for the substantive response for past conferences
- **is_streaming**: Boolean. Indicates if the conference is currently streaming

The above response is returned by the other API calls as well:

- update
- fetch
- fetch_all - Returns a list of these responses.

8.2 Create Reservationless Conference

ResellerAdmin/CompanyAdmin

Creates an active reservationless conference call immediately

```
POST "/enterprise_api/conference/create/reservationless"
```

```
Request body (application/json):
```

```
{
  "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
  "host_id": 183,
  "one_time_access_code": true,
  "waiting_room": true,
  "secure_url": true
}
```

```
Response body (application/json):
```

```
{
  "conference_id": 183,
  "subject": "Test Conference",
  "agenda": "null",
  "start": "2020-09-19 13:00:00",
  "end_time": "2020-09-19 14:00:00",
  "active": true
  "time_zone": "Europe/London",
  "duration": 60,
  "auto_record": "none",
  "auto_stream": "none",
}
```

```
    "auto_transcribe": false,
    "one_time_access_code": 987654321,
      "security_pin": "123456",
        "mute_mode": "conversation",
          "room_url":
            "https://subdomain.domain.tld/conf/call/987654321",
            "recordings": [],
            "participants": [ 1, 2, 3 ],
            "moderators": [ 1 ],
            "is_streaming": false,
  }
```

Request:

- **host_id**: unique identifier of the host
- **one_time_access_code**: true or false, when set to true an access code will be returned in the response.
- **[UNRELEASED] waiting_room**: true or false, when set to true any participants will need to be let in by the organizer
- **[UNRELEASED] secure_url**: true or false, when set to true the one_time_access_code field will be encrypted

Response:

- **conference_id**: id of the conference
- **subject**: Subject
- **agenda**: Agenda
- **start**: Start time, this works in conjunction with the time_zone to indicate the correct time.
- **end_time**: The scheduled end time, according to the specified duration.
- **active**: Whether the call is active (is scheduled for the current time or has live participants)
- **time_zone**: Time zone associated with the start time. Values are those (e.g. US/Central, Europe/London, Asia/Tokyo, etc.) in the IANA Time Zone database (<https://www.iana.org/time-zones>)
- **duration**: In minutes.
- **auto_record**: Indicates whether the conference will be recorded automatically and the format.
- **auto_stream**: Indicates whether the conference will be streamed automatically and the format.
- **auto_transcribe**: Whether the recording will be transcribed automatically.
- **one_time_access_code**: When the request has one_time_access_code set to true this field contains the conference access code.
- **mute_mode**: The initial participant mute settings.

- **participants:** List of participants in the conference, note the organizer is always included in the list. These are contact_id's that can be used with contact/fetch and other contact methods.
- **moderators:** List of moderators in the conference. These are contact_id's that can be used with contact/fetch and other contact methods.
- **room_url:** the URL for the meeting room. Can be navigated to in a browser to join the meeting.
- **recordings:** information about recordings made during the conference. For create, update, create/reservationless, will be an empty array [], as these return information about future conferences; however, see conference/fetch below for the substantive response for past conferences
- **is_streaming:** Boolean. Indicates if the conference is currently streaming

8.3 Update Conference

ResellerAdmin/CompanyAdmin

Update a scheduled conference

```
POST "/enterprise_api/conference/update"
```

Request **body** (application/json):

```
{
  "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
  "conference_id": 183,
  "subject": "Test Conference",
  "agenda": "An agenda",
  "start": "2020-09-19 13:00:00",
  "time_zone": "Europe/London",
  "duration": 60,
  "auto_record": "none",
  "auto_transcribe": false,
  "one_time_access_code": false,
  "mute_mode": "conversation",
  "disable_emails": [ "conference_reminders",
"conference_summaries", "transcription_summaries" ],
  "Participants": [ 2, 3 ]
}
```

Response **body** (application/json):

```
{
  "conference_id": 183,
```

```
        "subject": "Test Conference",
        "agenda": "An agenda",
        "start": "2020-09-19 13:00:00",
        "end_time": "2020-09-19 14:00:00",
        "active": false
        "time_zone": "Europe/London",
        "duration": 60,
        "auto_record": "none",
        "auto_transcribe": false,
        "one_time_access_code": 987654321,
        "security_pin": "123456",
        "mute_mode": "conversation",
        "room_url":
            "https://subdomain.domain.tld/conf/call/987654321",
        "recordings": [],
        "participants": [ 1, 2, 3 ],
        "moderators": [ 1 ],
        "is_streaming": false,
    }
```

Request:

- **conference_id**: unique identifier of the conference
- **subject**: Subject
- **agenda**: Agenda
- **start**: Start time, this works in conjunction with the `time_zone` to set the correct time.
- **end_time**: The scheduled end time, according to the specified duration.
- **active**: Whether the call is active (is scheduled for the current time or has live participants)
- **time_zone**: Time zone associated with the start time. Values are those (e.g. US/Central, Europe/London, Asia/Tokyo, etc.) in the IANA Time Zone database (<https://www.iana.org/time-zones>)
- **duration**: In minutes.
- **auto_record**: Optional. Indicates whether the conference should be recorded automatically and the format, allowable values are: none, audio, video. If omitted will default to none.
- **auto_stream**: Optional. Indicates whether the conference should be streamed automatically and the format, allowable values are: none, audio, video. If omitted will default to none. Both record and stream must be the same value if recording and streaming together.

- **auto_transcribe**: Optional. Whether the recording should automatically be transcribed. Valid values are true or false. If omitted will default to false.
- **one_time_access_code**: When the request has one_time_access_code set to true this field contains the conference access code.
- **security_pin**: Pin required to access the conference.
- **mute_mode**: Optional. Sets the initial participant mute settings. Options are:
 - conversation - Default, all unmuted
 - q&a - Participants muted, but can unmute
 - presentation - Participants muted and can not unmute
 If omitted will default to conversation.
- **participants**: List of contact IDs to be included as participants.
- **participant_emails**: Alternatively participants can be listed via emails with moderators specified. If a participant's emails is not on the host's contact list, it will be added:

```
POST "/enterprise_api/conference/update"
```

```
Request body (application/json):
```

```
{
  "participant_emails": [
    {"email": "bob@xxxx.com", "moderator": true},
    {"email": "alice@xxxx.com", "moderator": false}
  ]
}
```

Response:

Refer to the Schedule Conference response for field details.

8.4 Fetch Conference

ResellerAdmin/CompanyAdmin

Fetch a scheduled conference details

```
POST "/enterprise_api/conference/fetch"
```

```
Request body (application/json):
```

```
{
```

```
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "conference_id": 183
}

Response body (application/json):
{
    "conference_id": 183,
    "subject": "Test Conference",
    "agenda": "An agenda",
    "start": "2020-09-19 13:00:00",
    "end_time": "2020-09-19 14:00:00",
    "active": false
    "time_zone": "Europe/London",
    "duration": 60,
    "auto_record": "none",
    "auto_stream": "none",
    "auto_transcribe": false,
    "one_time_access_code": 987654321,
    "security_pin": "123456",
    "mute_mode": "conversation",
    "room_url":
        "https://subdomain.domain.tld/conf/call/987654321",
    "recordings": [{
        "id": 12345,
        "start_time": "2020-09-19 13:00:00",
        "end_time": "2020-09-19 14:00:00",
        "type": "video",
        "publishing_state": "published",
        "download_link":
            "https://subdomain.domain.tld/recording/inline/dc68df6c8a5.mp4",
        "participants": [ 1, 2, 3 ],
        "moderators": [ 1 ],
        "is_streaming": false,
    }
}
```

Request:

- **conference_id**: unique identifier of the conference

Response:

Refer to the Schedule Conference response for details of other fields, except:

- **recordings:** information about recordings made during the conference. An array of:
 - **id:** unique identifier for the recording
 - **start_time**
 - **end_time**
 - **type:** 'audio' or 'video'
 - **publishing_state:** 'processing' or 'published'. 'processing' indicates that the recording is still undergoing final post-processing (the conference has likely just ended). Repeat the fetch in a few minutes.
 - **download_link:** a link directly to the mp3 of an audio recording, or mp4 of a video recording (once the publishing state is 'published')

8.5 Fetch All Conferences

ResellerAdmin/CompanyAdmin

Fetch a list of all active scheduled conferences for a host. Please note that conferences scheduled more than 18 months in the future will not be returned.

```
POST "/enterprise_api/conference/fetch_all"
```

```
Request body (application/json):
```

```
{
  "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
  "host_id": 1
}
```

```
Response body (application/json):
```

```
[
  {
    "conference_id": 183,
    "subject": "Test Conference",
    "agenda": "An agenda",
    "start": "2020-09-19 13:00:00",
    "end_time": "2020-09-19 14:00:00",
    "active": false
    "time_zone": "Europe/London",
    "duration": 60,
    "auto_record": "none",
    "auto_transcribe": false,
    "one_time_access_code": 987654321,
  }
]
```

```

        "security_pin": "123456",
        "mute_mode": "conversation",
        "room_url":
            "https://subdomain.domain.tld/conf/call/987654321",
        "recordings": [],
        "participants": [ 1, 2, 3 ],
        "moderators": [ 1 ],
        "is_streaming": false,
    },
    {
        "conference_id": 184,
        "subject": "Test Conference 2",
        "agenda": "An agenda 2",
        "start": "2020-09-19 14:00:00",
        "end_time": "2020-09-19 15:00:00",
        "active": false
        "time_zone": "Europe/London",
        "duration": 60,
        "auto_record": "none",
        "auto_transcribe": false,
        "one_time_access_code": 987654321,
        "security_pin": "123456",
        "mute_mode": "conversation",
        "room_url":
            "https://subdomain.domain.tld/conf/call/987654321",
        "recordings": [],
        "participants": [ 1, 2, 3 ],
        "moderators": [ 1 ],
        "is_streaming": false,
    },
]

```

Request:

- **output:** if specified as “detail”, the calls will be returned with additional details. See **6.7 Host Conferences**

Response:

A list of scheduled conferences, refer to the Schedule Conference response for field details, and to the previous section, fetch, for details specifically about **recordings**.

8.6 Cancel Conference

ResellerAdmin/CompanyAdmin

Cancel a scheduled conference

```
POST "/enterprise_api/conference/cancel"
```

```
Request body (application/json):
```

```
{
  "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
  "conference_id": 183
}
```

```
Response body (application/json):
```

```
{
  "message": "OK"
}
```

Request:

- **conference_id**: unique identifier of the conference

Response:

- **message**: OK

8.7 End Conference

ResellerAdmin/CompanyAdmin

Ends an active conference

```
POST "/enterprise_api/conference/end"
```

```
Request body (application/json):
```

```
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "conference_id": 183
}
```

```
Response body (application/json):
```

```
{
    "message": "OK"
}
```

8.8 Fetch Participant Details

Get Contact (see section 7) details for participants and moderators in a conference

```
POST "/enterprise_api/conference/fetch_participant_details"
```

```
Request body (application/json):
```

```
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "conference_id": 183
}
```

```
Response body (application/json):
```

```
{
  participants: [
    {
      "contact_id": 35,
      "name": "John Smith",
      "email": "john@smith.com",
      "business_phone": "+12345678901",
      "mobile_phone": "+12345678902",
      "home_phone": "+12345678903",
      "other_phone": "+12345678904",
      "time_zone": "US/Eastern",
      "has_host": false
    }
  ]
}
```

```

    }
  ],
  moderators: [
    {
      "contact_id": 37,
      "name": "James Brown",
      "email": "james@brown.com",
      "business_phone": "+12345678905",
      "mobile_phone": "+12345678906",
      "home_phone": "+12345678907",
      "other_phone": "+12345678908",
      "time_zone": "US/Central",
      "has_host": true
    }
  ]
}

```

Request:

- **conference_id**: unique identifier of the conference

Response:

- **participants**: array of contact structures (see 7.2 for details of fields) for participants in the conference
- **moderators**: array of contact structures (see 7.2 for details of fields) for moderators in the conference

8.9 Fetch Live Stream Viewer Count

Get the current number of participants viewing the livestream

```
POST "/enterprise_api/conference/fetch_livestream_viewer_count"
```

Request **body** (application/json):

```

{
  "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
  "conference_id": 183
}

```

Response **body** (application/json):

```

{
  "count": "4"
}

```

Request:

- **conference_id**: unique identifier of the conference

Response:

- **count**: number of livestream viewers currently viewing the livestream

8.10 Stop Streaming

Stop livestreaming on the specified conference

```
POST "/enterprise_api/conference/stop_streaming"

Request body (application/json):
{
    "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
    "conference_id": 183
}

Response body (application/json):
{
    "message": "OK"
}
```

Request:

- **conference_id**: unique identifier of the conference

Response:

- **message**: OK

8.11 Recurring Conferences

Recurring conferences can be created by including a schedule parameter in the list of fields. The schedule field follows the iCalendar format for the RRULE field and allows flexibility in defining the schedule to be implemented.

The implementation supports a subset of the iCalendar RRULE options which are detailed below.

Available fields are:

FREQ: DAILY, WEEKLY or MONTHLY

INTERVAL: Interval between days, weeks or months.

UNTIL: End date for the recurring schedule, note this should just be date and will include the date in the schedule.

COUNT: Number of occurrences

BYDAY: For weekly this is MO, TU, WE, TH, FR, SA, SU, multiple days are supported. For Monthly this is the same set of days but prefixed by the week, so 2WE is the 2nd Wednesday.

BYMONTHDAY: Only applicable to monthly and specifies the day of the month, note negative numbers can be used to indicate the reference is from the end of the month rather than the start.

Example rules are:

Daily every other day

FREQ=DAILY;INTERVAL=2

Daily until specific date

FREQ=DAILY;UNTIL=20211231

Weekly on Monday and Wednesday

FREQ=WEEKLY;COUNT=10;BYDAY=MO,WE

Monthly, second Wednesday of the month, every other month

FREQ=MONTHLY;COUNT=10;INTERVAL=2;BYDAY=2WE

Monthly, last Friday of the month, every month

FREQ=MONTHLY;COUNT=10;INTERVAL=1;BYDAY=-1FR

Monthly, 10th day of the month

FREQ=MONTHLY;COUNT=10;INTERVAL=1;BYMONTHDAY=10

Monthly on the third-to-the-last day of the month, forever:

FREQ=MONTHLY;BYMONTHDAY=-3

Request and Response

```
POST "/enterprise_api/conference/create"
```

Request **body** (application/json):

```
{
  "auth_token": "dd6d5d51-9ea5-4c4c-82c8-677267aece98",
  "host_id": 1,
  "subject": "Test Conference",
  "agenda": "An agenda",
  "start": "2020-09-19 13:00:00",
  "time_zone": "Europe/London",
  "duration": 60,
  "auto_record": "none",
  "auto_transcribe": false,
  "one_time_access_code": true,
  "security_pin": "123456",
  "mute_mode": "conversation",
  "participants": [ 2, 3 ],
  "schedule": "FREQ=DAILY;INTERVAL=2"
}
```

Response **body** (application/json):

```
{
  "conference_id": 183,
  "subject": "Test Conference",
  "agenda": "An agenda",
  "start": "2020-09-19 13:00:00",
  "end_time": "2020-09-19 14:00:00",
  "active": false
  "time_zone": "Europe/London",
  "duration": 60,
  "auto_record": "none",
  "auto_transcribe": false,
  "one_time_access_code": 987654321,
  "security_pin": "123456",
  "mute_mode": "conversation",
  "participants": [ 1, 2, 3 ],
  "moderators": [ 1 ],
  "is_master": true,
  "master_id": 183,
  "schedule": "FREQ=DAILY;INTERVAL=2"
}
```

Extra fields appear in the response:

- **is_master**: Indicates this is the master conference, conferences which are part of this schedule will have this set to false.
- **master_id**: ID for the master conference, for the master this will be the same as the conference_id
- **schedule**: The schedule that this conference is part of. Note the initial request may vary from the output as the system converts from the iCal RRULE to internal for processing and then when outputting the conference details it converts from internal to iCal so the content and formatting may differ.

Updating

Updating the master conference schedule will update all conferences in the series.

Deleting

Deleting conferences created by the schedule will only remove the specific instance unless the master conference is deleted which will remove all the occurrences.

8.12 Error Messages

Below are error messages generated by the Schedule Conference API functions.

Error	Description
Recording feature not enabled	The host does not have the recording feature assigned.
Video recording feature not enabled	The host does not have the video recording feature assigned.
Transcription feature not enabled	The host does not have the transcription feature assigned.
One time access code feature not available	The host does not have the one time access code feature assigned.
Must enable one_time_access_code to use security_pin	The security PIN has to be used in conjunction with the One Time Access Code.
Participants must be an array of contacts'	The participant field must be an array.

Participants has one or more contact that does not exist in the host's contact list.	The participants must be set as contacts for the host.
--	--

Additionally there are errors generated when the schedule parameter is included:

Error	Description
Can only include schedule details in the master conference	Recurring conferences can be updated but only the master conference can update the schedule.
Failed to parse the schedule rule.	This is a general error generated when trying to parse the schedule rule and will include additional details on the problem.
Schedule FREQ must be DAILY, WEEKLY or MONTHLY	The FREQ field only supports DAILY, WEEKLY or MONTHLY.
Schedule MONTHLY BYDAY only supports one item	For monthly rules the BYDAY field only supports a single day.
Schedule MONTHLY BYMONTHDAY only supports one item	Only a single day in the month is supported.
Schedule MONTHLY supports BYDAY or BYMONTHDAY, cannot use both	Only BYDAY or BYMONTHDAY can be used they cannot be used at the same time.
Schedule MONTHLY BYDAY must include number and day	For monthly the BYDAY field must include the week number and day.
Schedule MONTHLY must include BYDAY or BYMONTHDAY	For monthly one of BYDAY or BYMONTHDAY must be included.
Schedule WEEKLY must include BYDAY	For weekly BYDAY must be included.

9.0 Embedding Vivomeetings into Your App

You can embed Vivomeetings in your own web application or native mobile application.

9.1 Embedding in a web application

You can embed any of the pages visible on Vivomeetings into your web application using an iframe. Please contact us beforehand so that we can authorize your domains to host our pages. In your iframe attributes, allow camera, microphone, and fullscreen so that our meeting room and/or livestream player works as intended.

Compatibility notes: The host page(s) must have a valid SSL certificate for our iframe to work correctly in Chrome. For our iframe to be usable in Internet Explorer, if it is nested more than one level deep (if our iframe is within another iframe, or deeper) all ancestors of our iframe must belong to the same host.

Optional parameters:

- **name:** string. If you provide this parameter when linking directly to a meeting room, the user will not be prompted to enter a name.
- **skip_join:** true/false. If you provide this parameter when linking directly to a meeting room, the user will not be presented with the video/audio device selection dialog. Instead they will join the meeting using their system default camera and microphone.
- **observer:** true/false. Join the video room but have your camera off and this participant will not have a video tile displayed to others. They can still hear and be heard by others.
- **mute:mic,camera.** If you want to mute either the camera or microphone by default when joining the participant into the meeting room use this comma delimited list. You can pass either 'camera', 'mic' or both 'camera,mic'.
- **view:gallery,bottom_speaker,left_side_speaker.** The default view for meetings is gallery view. You can override this by specifying 'bottom_speaker' or 'left_side_speaker'. 'bottom_speaker' is the preferred view for 1 on 1 video chats.

Example, to embed our meeting room:

```
<iframe
  allow="camera;microphone;fullscreen;autoplay;display-capture"
  src="[your-domain].com/conf/call/123456?
  name=Steve+Irwin&
  skip_join=true"
/>
```

Example, to embed our meeting room with the camera off

```
<iframe
  allow="camera;microphone;fullscreen;autoplay;display-capture"
  src="[your-domain].com/conf/call/123456?
  name=Steve+Irwin&
  skip_join=true&
  mute=camera"
/>
```

You can embed any of our other pages by the same means.

If you would like the user to be logged into our system while viewing our iframe, please sign them in using SSO (section 10.2 below)

9.2 Embedding a live stream player

Vivomeetings video conferences can be live streamed via HTTP & HLS. You can embed our live stream player into your web application or mobile application using an iframe. In your iframe attributes, allow autoplay and fullscreen so that our livestream player works as intended.

The livestream url `[your-domain].com/livestream/123456` where 123456 is the access code of the meeting room being live streamed.

Example, to embed our livestream player:

```
<iframe
  allow="fullscreen;autoplay"
  src="[your-domain].com/livestream/123456"
/>
```

9.3 Embedding options for Gaming and Watch Parties

Strip layout is an additional optional parameter you can include on an iframe that will render the video conference iframe in a thin bar you can place at the bottom of your application allowing for the rest of the screen to be devoted to your application.

Example, to embed our as a strip for gaming or watch parties:

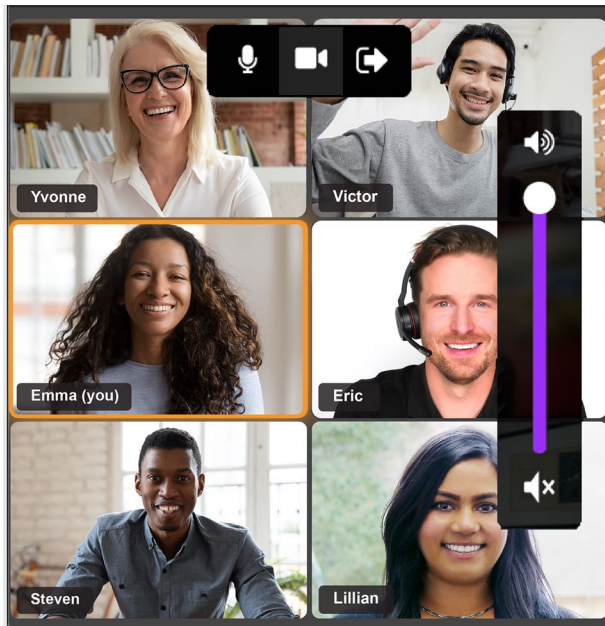
```
<iframe
  allow="camera;microphone;fullscreen;autoplay;display-capture"
```

```
src="[your-domain].com/conf/call/123456?  
strip_layout=true&  
skip_join=true”
```

/>



Example, for a watch party display volume control and hide most others:



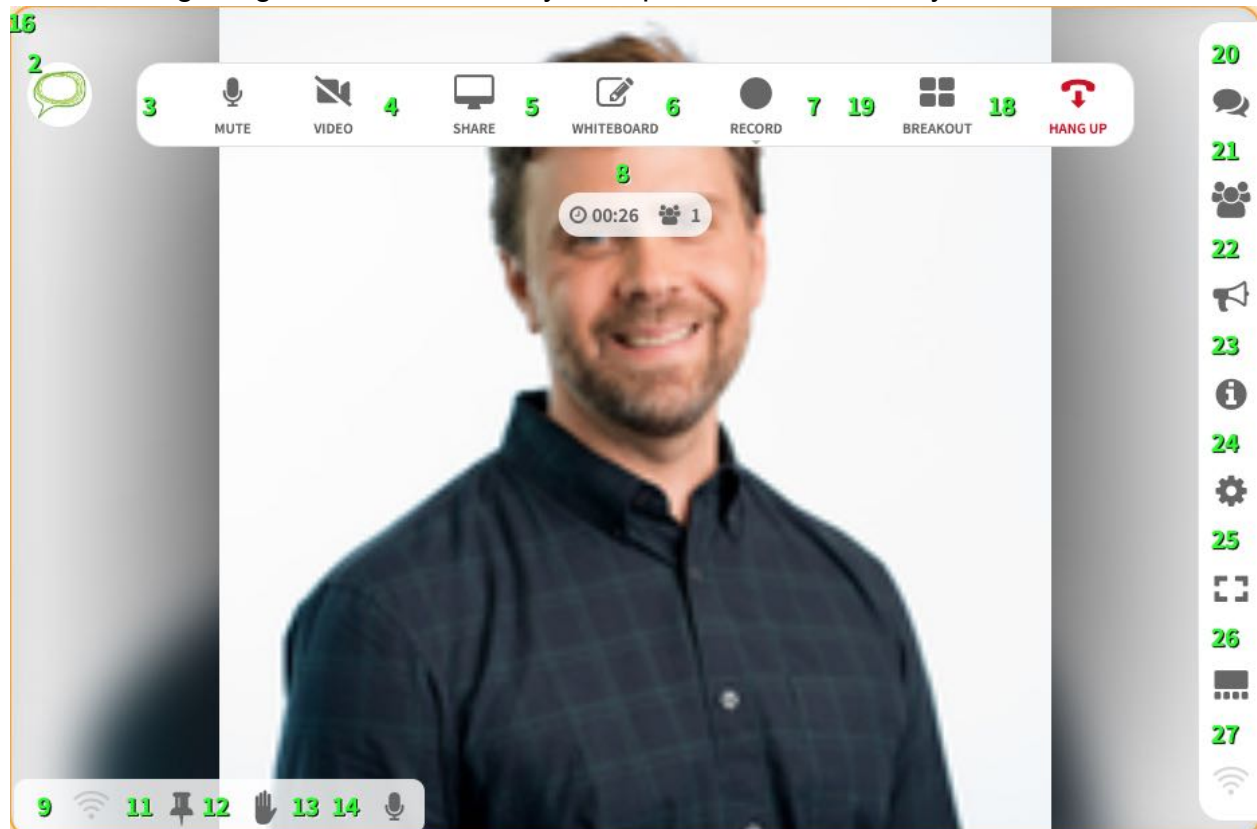
9.4 Hiding and displaying controls

Most UI controls displayed in the meeting room can be configured to be hidden for specific use cases. For example, a gaming or watch party application may want minimal UI controls presented to users. Watch party and 1 on 1 video chat applications may want a volume control which is normally hidden by default.

The following controls can be hidden or displayed:

- Screen Sharing
- Whiteboard
- Record
- Output volume
- Text Chat
- Participants
- Mute All
- Meeting Info
- Settings
- Full Screen
- Gallery View
- Connection Quality

The following image can be used as a key to request which controls you'd like to hide:



9.5 Meeting events

The meeting iframe will post messages about the participant state that your application can catch and act on. The messages that are currently supported relate to the participant leaving the meeting. New messages can be added, let us know what events you care about!

The iframe will post a message like this

```
window.parent.postMessage({type: 'vivomeetingsLeaveCallEvent', reason: 'left'})
```

The reason field can have the following values:

- 'left' - the user clicked the hang up button
- 'kicked' - the user was removed by the moderator or host
- 'blocked' - the user was removed by the moderator or host and is blocked from rejoining

Below is some sample code for catching and acting on these events

Sample code for catching and acting on these events

```
window.addEventListener("message", event => {
  if (event.data.type == 'vivomeetingsLeaveCallEvent' && event.data.reason ==
'left') {
    // they "left"
  }
}, false);
```

10.0 Signing in on behalf of Hosts (SSO)

Using the `host_id` and `login_token_public_key` available from the hosts endpoints (section 6.0) you can log hosts into our application seamlessly, without presenting them with a login screen. Certain UI elements will be hidden from these users, such as the change password and change email address features. Please note that these endpoints should be visited directly by the host/user, not by your server. Therefore you don't have to provide the API authorization token obtained from 3.0.

10.1 SSO via POST or GET (current tab or new tab)

To SSO a host in the current browser tab or a new tab, have your page (in the client's browser) submit a POST request. Hyperlinks submit as GET so to make it a POST you will have to insert the link in a `<form>` or as a javascript function on your page.

Required parameters:

- **host_id**: The account number of the user, retrieved from host endpoints (section 6.0)
- **login_token_public_key**: A host-specific authorization token, retrieved from host endpoints (section 6.0)
- **redirect_url**: What page the user should land on after logging in. This could be the dashboard or a specific meeting room. Can be a relative URL.

Optional parameters:

- **after_call_url**: If provided, the user will redirect to the provided URL after exiting a call. If it's not within our domain, you must provide a full URL with scheme (`http://` or `https://`)

Post Example:

```
POST "/auth"
```

```
Request body (application/json):
```

```
{
  "host_id": 8,
  "login_token_public_key": "a1b2c3d4e5f6",
  "redirect_url": "/conf/calls/upcoming",
  "after_call_url": "https://en.wikipedia.org"
}
```

GET Example:

```
<a
href="[your-domain].com/auth?host_id=8&login_token_public_key=a1b2c3d4e5f6&
redirect_url=/conf/create" />
```

10.2 SSO via iFrame

Please contact us before using this feature so that we can set up our response headers to properly authorize your host domains. To SSO through an iframe, use the /auth endpoint as the src attribute of the iframe.

Required parameters:

- **host_id:** The account number of the user, retrieved from host endpoints (section 6.0)
- **login_token_public_key:** A host-specific authorization token, retrieved from host endpoints (section 6.0)
- **redirect_url:** What page the user should land on after logging in. This could be the dashboard or a specific meeting room. Can be a relative URL.

Optional parameters:

- **after_call_url:** If provided, the user will redirect to the provided URL after exiting a call. If it's not within our domain, you must provide a full URL with scheme (http:// or https://)

Example:

```
<iframe allow="camera;microphone;fullscreen;display-capture"
src="[your-domain].com/auth?host_id=8&login_token_public_key=a1b2c3d4e5f6&r
edirect_url=/conf/calls/upcoming&after_call_url=https://en.wikipedia.org"
/>
```

11.0 Webhooks

We can POST, to a url of your choice, details of certain events related to conferences. Please contact us to set up your endpoint. We would need to know:

- The endpoint URL
- Which events (see 11.2 et seq) you are interested in. (Or all events.)

11.1 Generalities

Webhook events are sent as an HTTP post to the endpoint URL you specify. The request type is JSON and it has two keys:

- **event:** string, the type of event, *conference_start*, *conference_end*, etc. See values in the individual sections below
- **payload:** string, *escaped and serialized JSON* representing details of about the event, which depend on the type, see below.

```
POST
Request body (application/json):
{
  "event": "conference_end",
  "payload": "{\"conference_id\":1258835,\"host_id\":61803}"
}
```

11.2 Conference Start

- **event:** *conference_start*
- **payload**
 - **host_id:** unique identifier for the conference's host (can be used with host methods above)
 - **conference_id:** unique identifier for the conference (can be used with conference methods above)

11.3 Conference End

- **event:** *conference_end*
- **payload**
 - **host_id:** unique identifier for the conference's host (can be used with host methods above)
 - **conference_id:** unique identifier for the conference (can be used with conference methods above)

11.4 Start Recording

- **event:** start_recording
- **payload**
 - **host_id:** unique identifier for the conference's host (can be used with host methods above)
 - **conference_id:** unique identifier for the conference (can be used with conference methods above)

11.5 Stop Recording

- **event:** stop_recording
- **payload**
 - **host_id:** unique identifier for the conference's host (can be used with host methods above)
 - **conference_id:** unique identifier for the conference (can be used with conference methods above)

11.6 Start Streaming

- **event:** start_streaming
- **payload**
 - **host_id:** unique identifier for the conference's host (can be used with host methods above)
 - **conference_id:** unique identifier for the conference (can be used with conference methods above)

11.7 Stop Streaming

- **event:** stop_streaming
- **payload**
 - **host_id:** unique identifier for the conference's host (can be used with host methods above)
 - **conference_id:** unique identifier for the conference (can be used with conference methods above)

11.8 Recording Ready

- **event:** recording_ready
- **payload**
 - **host_id:** unique identifier for the conference's host (can be used with host methods above)
 - **conference_id:** unique identifier for the conference (can be used with conference methods above)
 - **recording_id:** unique identifier for the recording
 - **download_link:** a link directly to the mp3 of an audio recording, or mp4 of a video recording

11.9 Transcription Ready

- **event:** transcription_ready
- **payload**
 - **host_id:** unique identifier for the conference's host (can be used with host methods above)
 - **conference_id:** unique identifier for the conference (can be used with conference methods above)

11.10 Enter Livestream

- **event:** enter_livestream
- **payload**
 - **host_id:** unique identifier for the conference's host (can be used with host methods above)
 - **conference_id:** unique identifier for the conference (can be used with conference methods above)
 - **participant_id:** unique identifier for the participant

11.11 Leave Livestream

- **event:** leave_livestream
- **payload**
 - **host_id:** unique identifier for the conference's host (can be used with host methods above)
 - **conference_id:** unique identifier for the conference (can be used with conference methods above)
 - **participant_id:** unique identifier for the participant

12.0 OAuth

OAuth can be used with the Enterprise API so that an end user can authorize the application and a supplied token can be stored on the server for future use.

Typically a library is used to handle the OAuth process which involves:

- a. Redirecting the user to the OAuth page
- b. Receiving the confirmation that the user has authorized the application with an authorization code
- c. Requesting the access token and refresh token from the service using the authorization code
- d. Requesting new access tokens from the service using the refresh token

12.1 Data Required

To configure the application within the Vivomeetings system there are 2 pieces of information needed:

12.1.1 Service Name

This is the name displayed on the authorization page and should relate to your service, for example the Google Calendar service is configured for Vivomeetings and shown as:



Allow Google Calendar access to your account?

Google Calendar will be able to read your VivoMeeting meeting details.

DENY

AUTHORIZE

In the above example the logo is set based on the domain being used and this will already be configured, the text is constructed based on:

Allow Google Calendar access to your account?	The 'Google Calendar' is the service name and this should be set to the relevant name for your service.
Google Calendar will be able to read your Vivomeetings meeting details.	The Google Calendar service name is as above and the Vivomeetings will be replaced with the name configured in the database for your service.

12.1.2 Callback URL

One or more callback URL must be supplied so that once the user has authorized the application they can be sent back to your application.

The callback URL is sent along with the authorization request and must match the one stored exactly otherwise the authorization will be rejected.

12.2 Supplied Data

Vivomeetings will configure the application with the details above and supply the UID / client_id and secret to be used with the service.

12.3 Service URLs

The OAuth service uses these URLs:

Authorize	https://app.domain.com/oauth/authorize
Token	https://app.domain.com/oauth/token
Revoke	https://app.domain.com/oauth/revoke

12.4 Authorization Parameters

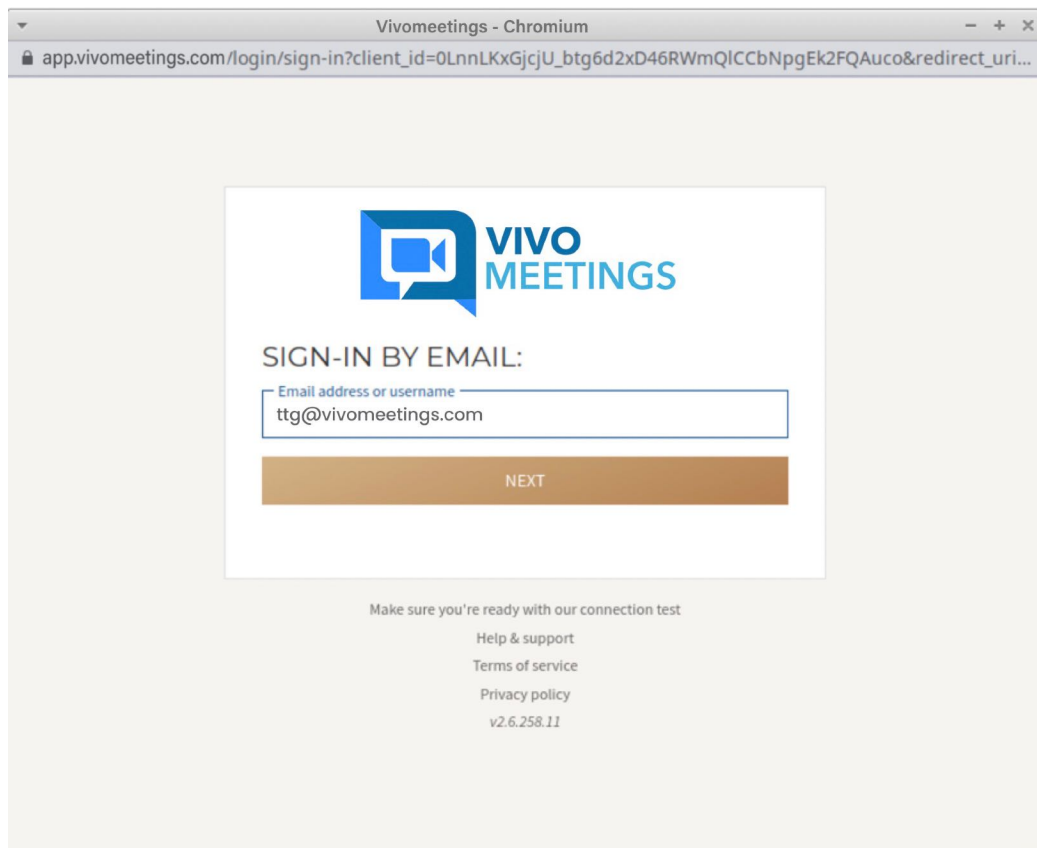
Depending on the library being used different configuration may be required to set up the required parameters. The underlying request should contain the parameters:

client_id	UID supplied by vivomeetings to identify your configuration
redirect_uri	The callback URL that was supplied to Vivomeetings and configured within the application
response_type	code
scope	read,write
state	Optional parameter which will be returned if you need to maintain some state data.

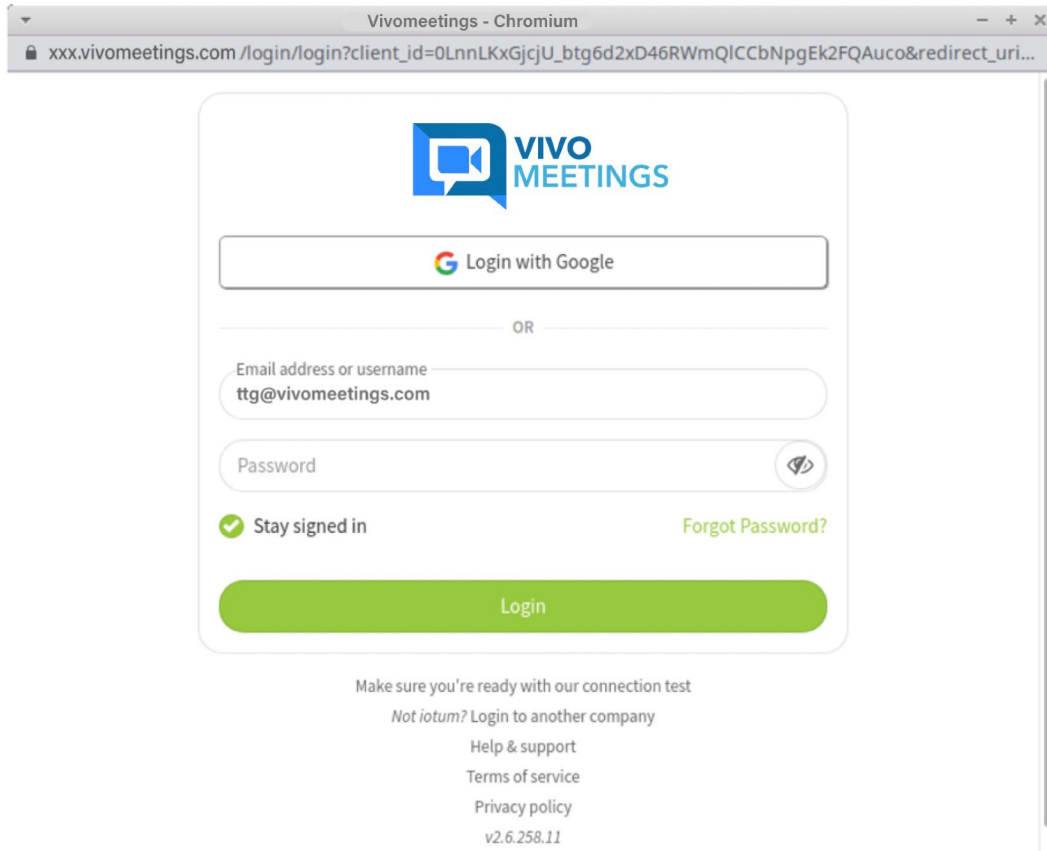
12.5 Logging In and Authorizing

Once the library is configured calling the authorize method will direct the user to the oauth/authorize URL.

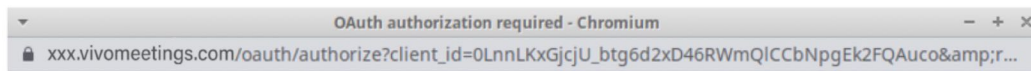
The first step is to log the user into the conference service through the usual login method, this example is from Vivomeetings:



The screenshot shows a Chromium browser window with the title "Vivomeetings - Chromium". The address bar contains the URL: `app.vivomeetings.com/login/sign-in?client_id=0LnnLKxGjcjU_btg6d2xD46RWmQlCCbNpgEk2FQAuco&redirect_uri...`. The main content area features the Vivomeetings logo (a blue speech bubble with a white camera icon) and the text "VIVO MEETINGS". Below the logo, it says "SIGN-IN BY EMAIL:". There is a text input field with the placeholder "Email address or username" and the value "ttg@vivomeetings.com". A brown button labeled "NEXT" is positioned below the input field. At the bottom of the page, there is a footer with the text "Make sure you're ready with our connection test", followed by links for "Help & support", "Terms of service", and "Privacy policy", and the version number "v2.6.258.11".



Once logged in the authorization screen will be displayed asking the user to Authorize access for the application.



Allow Google Calendar access to your account?

Google Calendar will be able to read your VivoMeetings meeting details.

DENY

AUTHORIZE

Clicking Authorize then redirects back to your application with the authorization code.

The library can then request the access and refresh tokens and store locally for use with the Enterprise API.

12.6 Supported Grant Types

The OAuth service supports:

12.6.1 Authorization Code Grant

Used when a client can securely store the client secret for example when the client runs on a server.

12.6.2 Implicit Grant

Used when a client cannot securely store the client secret for example when the client runs in a browser.

This flow returns the access token directly in the authorization response without the need to use the returned authorization code to request an access token.

For this mechanism the secret is not used and should not be stored in the application.

12.7 Using with the Enterprise API

12.7.1 Bearer Header

The Enterprise API accepts an `auth_token` parameter in the body of the request for normal authentication using user credentials to generate a short lived `auth_token` via an `authenticate` API request.

To use the OAuth credentials the `authenticate` request is not needed and the application can make calls by including a Bearer Authorization header instead:

```
Authorization: Bearer 09ba487fc3df...
```

The API detects the Bearer header parameter and retrieves the authorization associated with the token from the database and provides access to the resources for the specific user and level.

12.7.2 Refresh Token

The OAuth access tokens are short lived and the refresh token may be needed to generate a new access token, the library should handle this automatically.

To get a new access token the library should make a call to the `oauth/token` URL with the same parameters as the original request to get an access token with the changes:

grant_type: `refresh_token`

refresh_token: The refresh token value received from the original authorization

This will return a new access token.

12.7.3 Invalid Token

The application will need to allow for authorizations being revoked which make the stored token details invalid, this will result in a 401 Unauthorized response from the API. In this situation the end user will need to re-authorize the application.

13.0 Other API Use Cases

Almost any use case can be supported. Vivomeetings will work with you to provide the APIs you require to be successful.

Some examples may be:

- Create stand alone meetings
 - Some partners may want to create stand alone meetings
- Meeting events
 - Meeting starts / ends
 - Participant joins / leaves