API V2 REFERENCE DOCUMENT

API v2 Reference

API v2 Reference

The API is organized around REST. Our API is designed to have predictable, resource-oriented URLs, and to use HTTP response codes to indicate API errors. We use built-in HTTP features, like HTTP authentication and HTTP verbs, which can be understood by off-the-shelf HTTP clients. JSON will be returned in all responses from the API, including errors.

Endpoint

```
https://web.vboxx.nl:<port>/api/2
```

All requests to the VBoxxCloud API must be performed over HTTPS. Non-HTTPS requests will be rejected.

Responses

VBoxxCloud uses conventional HTTP response codes to indicate success or failure of an API request. In general, codes in the 2xx range indicate success, codes in the 4xx range indicate an error that resulted from the provided information (i.e. a required parameter was missing or invalid, the request is malformed, etc.), and codes in the 5xx range indicate an error with the VBoxxCloud server.

HTTP Status Code Summary

- **200** OK - Everything worked as expected.
- **304** Not Modified - The response has not changed since a previous request.
- **400** Bad Request - The request is missing a required parameter, includes an invalid parameter value, or is otherwise malformed.
- **401** Unauthorized - No valid authentication provided.
- **403** Forbidden - The request was refused because the authenticated user does not have access.
- **404** Not Found - The requested resource does not exist.
- **405** Method Not Allowed - An invalid HTTP method (GET, POST, etc.) was used.
- **409** Conflict - The request results in a conflict with the current state of the resource.
- **410** Gone - The requested resource did exist, but was permanently deleted.
- **500, 502, 503, 504** Server errors - Something went wrong on VBoxxCloud's end.

JSON

All API responses return a JSON object. For consistency, this is also true even in cases where the HTTP status code conveys the full response. For example, some API methods simply indicate that they were successful with an HTTP 200 status code. In those cases, the response will be:

HTTP status code: 200

Response body:

```
{
    "status": "ok"
}
```

Another case is when an HTTP 304 status code is returned, indicating that the client should use their cached response from a previous request.

HTTP status code: 304

Response body:

```
{
    "status": "not_modified"
}
```

Exact responses may be found in the details of each method.

Errors

In the case of an error, the API will use an appropriate HTTP status code in the response. JSON with details of the type of error that occurred and, when available, a human-readable description of the error will also be returned.

The following are common errors that may be encountered when working with the API:

Invalid request

HTTP status code: 400

Response body:

```
{
    "error": "invalid_request"
}
```

Access denied

HTTP status code: 401

Response body:

```
{
    "error": "access_denied"
}
```

Forbidden

HTTP status code: 403

Response body:

```
{
    "error": "forbidden"
}
```

Not found

HTTP status code: 404

Response body:

```
{
    "error": "not_found"
}
```

Unknown error

HTTP status code: 500

Response body:

```
{
    "error": "unknown"
}
```

Request is using an insecure transport

HTTP status code: 400

Response body:

```
{
    "error": "insecure_transport",
    "error_description": "Requests MUST utilize https."
}
```

Request is missing a required parameter

```
HTTP status code: 400

Response body:

{
    "error": "invalid_request",
    "error_description": "Missing required parameter: <param>"
}
```

Request contains an invalid parameter value

```
HTTP status code: 400

Response body:

{
    "error": "invalid_request",
    "error_description": "Invalid value for parameter: <param>"
}
```

Date is not properly formatted

```
HTTP status code: 400

Response body:

{
    "error": "invalid_date_format",
    "error_description": "Invalid date format. The expected format is: YYYY-MM-DD"
}
```

Date/time is not properly formatted

HTTP status code: 400

Response body:

```
{

    "error": "invalid_datetime_format",

    "error_description": "Invalid datetime format. The expected format is: YYYY-MM-DDTHH:MM:SS"

}
```

Errors specific to certain methods may be found in the method details.

Versioning

Major API versions are selected based on the endpoint URL. It is our intention to maintain backwards compatibility within a major version throughout its lifetime, except where security issues are involved. The minor API version in use can be determined by calling the version method.

Authentication

Unless otherwise noted, all VBoxxCloud API methods require authentication via OAuth2. Currently it supports the *password* and *refresh token* grant types, with *authorization code* and *client credential* grants coming in the near future.

The OAuth2 provider is not specific to an API version. For details on obtaining, refreshing, and revoking OAuth2 access tokens see the OAuth2 documentation.

General

Version

The version method returns the exact API version available in the format <major>.<minor>.<revision>. Authentication is not required.

```
GET /api/2/version
```

Response

```
HTTP status code: 200

Response body:

{

    "version": "2.0.9"

}
```

File system

Root object

Attributes

- **type** *string*

  "root"

- **id** *integer*

  The unique identifier of the root

- **name** *string*

The name of the root

- **path***string*

  "/"

- **is_locked***boolean*

  Whether the root is locked

- **root_type***string*

  The type of the root: "sync", "share", or "backup"

- **space_used***integer*

  The total size of the root in bytes

- **space_used_formatted***string*

  The total size of the root in human friendly formatting

- **children***list*

  A list of child [files](#) and [folders](#)

- **hash***string*

  A hash of the root metadata used for results caching

Example

```
{
    "children": [
        ...child files and folders...
    ],
    "hash": "<hash>",
    "id": 1,
    "is_locked": false,
    "name": "Example Files",
    "path": "/",
    "root_type": "sync",
    "space_used": 10000,
    "space_used_formatted": "10k",
    "type": "root"
}
```

File object

Attributes

- **type** *string*

  "file"

- **id** *integer*

  The unique identifier of the file

- **revision_id** *integer*

The unique revision identifier of the file

- **root_id***integer*

The identifier of the root containing the file

- **path***string*

The absolute path of the file within the root

- **is_deleted***boolean*

Whether the file is deleted

- **created***string*

The date/time in UTC the file was created in the format YYYY-MM-DDTHH:MM:SS

- **modified***string*

The date/time in UTC the file was last modified in the format YYYY-MM-DDTHH:MM:SS

- **size***integer*

The size of the file in bytes

- **size_formatted***string*

The size of the file in human friendly formatting

- **is_locked** *boolean*

Whether the file is locked

Example

```json
{
    "created": "2015-01-01T01:02:03",
    "id": 1,
    "is_deleted": false,
    "is_locked": false,
    "modified": "2015-01-01T02:03:04",
    "path": "/path/to/file.ext",
    "revision_id": 1,
    "root_id": 1,
    "size": 12205,
    "size_formatted": "11.92k",
    "type": "file"
}
```

Folder object

Attributes

- **type** *string*

  "folder"

- **id** *integer*

  The unique identifier of the folder

- **root_id** *integer*

  The identifier of the root containing the folder

- **path** *string*

  The absolute path of the folder within the root

- **is_deleted** *boolean*

  Whether the folder is deleted

- **is_locked** *boolean*

  Whether the folder is locked

- **children** *list*

A list of child *files* and *folders*

- **hash**_string_

A hash of the folder metadata used for results caching

Example

```json
{
    "children": [
        ...child files and folders...
    ],
    "hash": "<hash>",
    "id": 1,
    "is_deleted": false,
    "is_locked": false,
    "path": "/path/to/name",
    "root_id": 1,
    "type": "folder"
}
```

File share object

Attributes

- **type**_string_

  "file_share"

- **id**_integer_

  The unique identifier of the file share

- **file_id**_integer_

  The identifier of the shared file, or null if it's a shared folder

- **folder_id**_integer_

  The identifier of the shared folder, or null if it's a shared file

- **root_id**_integer_

  The identifier of the root containing the file or folder

- **hash**_string_

  The share hash used to create the unique share link

- **expires**_string_

The date/time in UTC the share expires in the format YYYY-MM-DDTHH:MM:SS or null if it has no expiration

- read_access*boolean*

Whether the user has read access. May not be present if there is no user context.

- write_access*boolean*

Whether the user has write access. May not be present if there is no user context.

- delete_access*boolean*

Whether the user has delete access. May not be present if there is no user context.

Example

```json
{
    "creator_id": 2,
    "delete_access": false,
    "expires": null,
    "file_id": 4,
    "folder_id": null,
    "hash": "06d0d4833ea65c",
    "id": 7,
    "read_access": true,
    "root_id": 3,
    "type": "share",
    "write_access": false }
```

File methods

Get file metadata

```
GET /api/2/files/<root_id>/<file_id>
```

Response

```
HTTP status code: 200

Response body:

{
    ...a file object...
}
```

Download a file

```
GET /api/2/files/<root_id>/<file_id>/download
```

Response

```
HTTP status code: 200

Response body:

<binary file data>
```

Share a file

```
POST /api/2/files/<root_id>/<file_id>/share
```

POST Fields

- **login_required**: whether users must log in to access the share. "true" or "false". Default "false".
- **expires**: a date in the format YYYY-MM-DD the share expires. Defaults to no expiration.
- **subscribers**: a comma-separated list of subscribers by email address.
- **notify_subscribers**: "true" or "false". Default "false".
- **download_limit**: the total number of downloads allowed for the share. Defaults to unlimited.

- **download_notify**: whether you want to be notified of downloads. "true" or "false". Default "false".

Rename a file

```
POST /api/2/files/<root_id>/<file_id>/rename
```

POST Fields

- **name**: the new file name. *Required.*

Errors

Invalid file name

```
HTTP status code: 400

Response body:

{

    "error": "invalid_name"

}
```

A file with the same name already exists

```
HTTP status code: 409

Response body:

{

    "error": "name_conflict"

}
```

Move a file

```
POST /api/2/files/<root_id>/<file_id>/move
```

POST Fields

- **to_folder_id**: the destination folder ID. To move to the root, omit this parameter.

Errors

A file with the same name already exists in the destination

```
HTTP status code: 409

Response body:

{

    "error": "name_conflict"

}
```

Lock a file

```
POST /api/2/files/<root_id>/<file_id>/lock
```

POST Fields

- **duration**: the number of minutes after which the lock will be automatically released.

Response

```
HTTP status code: 200
```

Unlock a file

```
POST /api/2/files/<root_id>/<file_id>/unlock
```

Response

```
HTTP status code: 200
```

Delete a file

```
POST /api/2/files/<root_id>/<file_id>/delete
```

Response

```
HTTP status code: 200
```

Folder method

Get folder metadata

```
GET /api/2/files/<root_id>/folder/<folder_id>
```

Query Parameters

- **include_children**: "true" or "false". Default "true".
- **include_deleted**: "true" or "false". Default "true".
- **hash**: the hash value returned by the last call to this method. Only used when include_children is "true".

Response

```
HTTP status code: 200

Response body:

{

    ...a folder object...

}
```

children is a list of file and folder metadata objects within the folder. It is only present when include_children is "true".

The hash value represents a signature for the response data. If the hash value in the response would be identical to the hash value passed by the client, then HTTP 304 Not Modified will be returned. hash is only present when include_children is "true".

Upload a file to a folder

POST /api/2/files/<root_id>/folder/<folder_id>/upload

POST Fields

- **file**: *Required.*

Response

HTTP status code: 200

Response body:

```
{
    ...file metadata...
}
```

Errors

No file received

HTTP status code: 400

Response body:

```
{
    "error": "no_file_received",
    "error_description": "Missing required file: file"
}
```

File rejected due to company policy

HTTP status code: 409

Response body:

```
{
    "error": "policy_error"
}
```

Invalid file name

```
HTTP status code: 400

Response body:

{

    "error": "invalid_name"

}
```

A file with the same name already exists

```
HTTP status code: 409

Response body:

{

    "error": "name_conflict"

}
```

Create a subfolder

```
POST /api/2/files/<root_id>/folder/<folder_id>/create_folder
```

POST Fields

- **name**: the new folder name. *Required.*

Response

```
HTTP status code: 200

Response body:

{

...folder metadata...

}
```

Errors

Invalid folder name

HTTP status code: 400

Response body:

```
{

    "error": "invalid_name"

}
```

Folder name is too long

HTTP status code: 400

Response body:

```
{

    "error": "name_too_long"

}
```

A folder with the same name already exists

HTTP status code: 409

Response body:

```
{

    "error": "name_conflict"

}
```

Download a folder

```
GET /api/2/files/<root_id>/folder/<folder_id>/download
```

Response

HTTP status code: 200

Response body:

`<binary zip file data>`

Share a folder

```
POST /api/2/files/<root_id>/folder/<folder_id>/share
```

POST Fields

- **login_required**: whether users must log in to access the share. "true" or "false". Default "false".
- **expires**: a date in the format YYYY-MM-DD the share expires. Defaults to no expiration.
- **subscribers**: a comma-separated list of subscribers by email address.
- **notify_subscribers**: "true" or "false". Default "false".
- **download_limit**: the total number of downloads allowed for the share. Defaults to unlimited.
- **download_notify**: whether you want to be notified of downloads. "true" or "false". Default "false".

Rename a folder

```
POST /api/2/files/<root_id>/folder/<folder_id>/rename
```

POST Fields

- **name**: the new folder name. *Required*.

Errors

Invalid folder name

```
HTTP status code: 400

Response body:

{
    "error": "invalid_name"
}
```

Folder name is too long

```
HTTP status code: 400

Response body:

{

    "error": "name_too_long"

}
```

A folder with the same name already exists

```
HTTP status code: 409

Response body:

{

    "error": "name_conflict"

}
```

Move a folder

```
POST /api/2/files/<root_id>/folder/<folder_id>/move
```

POST Fields

- **to_folder_id**: the destination folder ID. To move to the root, omit this parameter.

Errors

A folder with the same name already exists in the destination

```
HTTP status code: 409

Response body:

{

    "error": "name_conflict"

}
```

Lock a folder

POST /api/2/files/<root_id>/folder/<folder_id>/lock

POST Fields

- **duration**: the number of minutes after which the lock will be automatically released.

Response

HTTP status code: 200

Unlock a folder

POST /api/2/files/<root_id>/folder/<folder_id>/unlock

Response

HTTP status code: 200

Delete a folder

POST /api/2/files/<root_id>/folder/<folder_id>/delete

Response

HTTP status code: 200

Root methods

Get root metadata

GET /api/2/files/<root_id>

Query Parameters

- **include_children**: "true" or "false". Default "true".
- **include_deleted**: "true" or "false". Default "true".
- **hash**: the hash value returned by the last call to this method. Only used when include_children is "true".

Response

HTTP status code: 200

Response body:

{

   ...a root object...

}

Possible values for root_type are "sync", "share", and "backup".

children is a list of file and folder metadata objects at the top level of the root. It is only present when include_children is "true".

The hash value represents a signature for the response data. If the hash value in the response would be identical to the hash value passed by the client, then HTTP 304 Not Modified will be returned. hash is only present when include_children is "true".

HTTP status code: 304

Upload a file to a root

POST /api/2/files/<root_id>/upload

POST Fields

- **file**: *Required.*

Response

HTTP status code: 200

Response body:

{

...file metadata...

}

Errors

No file received

HTTP status code: 400

Response body:

```
{

    "error": "no_file_received",

    "error_description": "Missing required file: file"

}
```

File rejected due to company policy

HTTP status code: 409

Response body:

```
{

    "error": "policy_error"

}
```

Invalid file name

HTTP status code: 400

Response body:

```
{

    "error": "invalid_name"

}
```

A file with the same name already exists

```
HTTP status code: 409

Response body:

{

    "error": "name_conflict"

}
```

Create a folder in a root

```
POST /api/2/files/<root_id>/create_folder
```

POST Fields

- **name**: the new folder name. *Required.*

Response

```
HTTP status code: 200

Response body:

{

    ...folder metadata...

}
```

Errors

Invalid folder name

```
HTTP status code: 400

Response body:

{

    "error": "invalid_name"

}
```

Folder name is too long

```
HTTP status code: 400
```

Response body:

```
{

    "error": "name_too_long"

}
```

A folder with the same name already exists

HTTP status code: 409

Response body:

```
{

    "error": "name_conflict"

}
```

Search files and folders

GET /api/2/files/<root_id>/search

Query Parameters

- **q**: the search term. *Required.*

Response

HTTP status code: 200

Response body:

```
{

   "results": [

       ...file and folder metadata...

   ]

}
```

List recently modified files

```
GET /api/2/files/<root_id>/modified_since
```

Query Parameters

- **since**: a date/time in UTC in the format YYYY-MM-DDTHH:MM:SS. *Required.*
- **include_deleted**: "true" or "false". Default "true".

Response

```
HTTP status code: 200

Response body:

{

    "results": [

        ...file metadata...

    ]

}
```

Lock a root

```
POST /api/2/files/<root_id>/lock
```

POST Fields

- **duration**: the number of minutes after which the lock will be automatically released.

Response

```
HTTP status code: 200
```

Unlock a root

```
POST /api/2/files/<root_id>/unlock
```

Response

```
HTTP status code: 200
```

Errors

The following errors may occur for any file system method and are not explicity mentioned in the method descriptions.

Authenticated user does not have access to the root

```
HTTP status code: 403

Response body:

{
    "error": "forbidden"
}
```

Root, Folder, or File does not exist

```
HTTP status code: 404

Response body:

{
    "error": "not_found"
}
```

Root, Folder, or File is locked

```
HTTP status code: 409

Response body:

{
    "error": "resource_locked"
}
```

Root is no longer accessible because it was deleted

```
HTTP status code: 410

Response body:

{

    "error": "root_deleted",

    "error_description": "Root was previously deleted."

}
```

The file server is busy or temporarily unavailable

```
HTTP status code: 503

Response body:

{

    "error": "temporarily_unavailable",

    "error_description": "Service is temporarily unavailable.""

}
```

Person object

A person object represents an VBoxxCloud user account.

Attributes

- **type**_string_

  "person"

- **id**_integer_

  The unique identifier for the person

- **email**_string_

  The person's email address

- **username**_string_

  The person's username

- **company_id**_integer_

  The identifier of the company to which the person belongs

- **first_name**_string_

  The person's first name

- last_name*string*

  The person's last name

- display_name*string*

  The person's name formatted for display

- root_id*integer*

  The identifier of the person's sync root

- roots*list*

  A list of [root objects](root-objects) to which the person is subscribed

- space_quota*integer*

  The person's space quota in bytes

- space_quota_formatted*string*

  The person's space quota in human friendly formatting

- space_usage*integer*

  The amount of space used by the person in bytes

- space_usage_formatted*string*

The amount of space used by the person in human friendly formatting

- **can_share** *boolean*

Whether the person's company policy allows them to create shares

- **company_policy** *object*

The expanded company [policy object](#)

Example

```json
{
    "can_share": true,
    "company_policy": {
        ...
    },
    "display_name": "First Last",
    "email": "user@example.com",
    "first_name": "First",
    "id": 1,
    "last_name": "Last",
    "root_id": 1,
    "roots": [
        ...
    ],
    "space_quota": 10000,
    "space_quota_formatted": "10k",
    "space_usage": 100,
    "space_usage_formatted": "100b",
```

```
    "type": "person",

    "username": ""

}
```

Person methods

Get a person

The person method returns information about the authenticated user or a user with a given ID or email address.

```
GET /api/2/person

GET /api/2/person/<person_id>

GET /api/2/person/<email>
```

Response

```
HTTP status code: 200

Response body:

{

...a person object...

}
```

Create a person

```
POST /api/2/person/create
```

POST Fields

- company_id:
- email:
- first_name:
- last_name:
- password:
```

- generate_password:
- pw_expires:
- webdav:
- space_quota:
- mobile_phone:
- site_admin:
- system_admin:
- create_root:
- dept_shares:
- group_ids:
- quota_50:
- quota_80:
- quota_85:
- quota_90:
- quota_95:
- quota_100:
- send_welcome_email:

## Response

Returns the created person.

HTTP status code: 200

Response body:

{

 ...a person object...

}

## Update a person

POST /api/2/person/<person_id>/update

## Response

Returns the updated person.

HTTP status code: 200

Response body:

{

```
...a person object...

}
```

Delete a person

```
POST /api/2/person/<person_id>/delete
```

POST Fields

- **remove_server_files**:
- **remove_user_files**: whether to delete the user's account root.
- **remove_dept_files**: whether to delete files owned by the user in team shares.

Response

```
HTTP status code: 200
```

Create an account sync root

Create a sync root for an account if one does not already exist.

```
POST /api/2/person/<person_id>/roots/create
```

POST Fields

Optional

- **webdav**: whether to enable webdav access. Default "false".

Response

Returns the created root.

```
HTTP status code: 200

Response body:

{

...a root object... }
```

List recent activity for a person

```
GET /api/2/person/<person_id>/activity
```

Query Parameters

- **offset**: zero-based offset from which to start listing activity records. Default 0.

Response

```
HTTP status code: 200

Response body:

{

    "offset": 0,

    "results": [

        ...up to 100 activity objects in reverse chronological order...

    ],

    "total": 42

}
```

- **offset**: the offset given in the method call or zero.
- **results**: a list of activity objects in reverse chronological order.
- **total**: the total number of activity records available for the person.

Errors

The following errors may occur for any person method and are not explicitly mentioned in the method descriptions.

Authenticated user does not have access to view or modify the person

```
HTTP status code: 403

Response body:

{

    "error": "forbidden"

}
```

Person does not exist

HTTP status code: 404

Response body:

```
{
    "error": "not_found"
}
```

Organization

Organization object

An organization represents a collection people, machines, roots, and other child organizations. Access control, policies, and most system settings are applied at the organization level.

Attributes

- **type***string*

	"organization"

- **id***integer*

	The unique identifier for the organization

- **parent_id***integer*

	The unique identifier of the parent organization

- **name***string*

	The organization name

- **slug***string*

  The URL friendly organization name

- **description***string*

  A description of the organization

- **email***string*

  The organization's contact email address

- **hostname***string*

  The hostname your organization's agents will use to connect

- **policy***object*

  A [policy object](#)

- **created***string*

  The date/time in UTC the organization was created in the format YYYY-MM-DDTHH:MM:SS

- **email_server_id***integer*
- **bandwidth_throttle***integer*
- **throttle_exception_start***string*
- **throttle_exception_end***string*
- **throttle_exception_dow***list of integers*
- **throttle_exception_days***string*
- **throttle_exception_throttle***integer*

- **throttled** *boolean*

  Whether bandwidth throttling is enabled for the organization

- **active** *boolean*

  Whether the organization is active

- **plan_id** *integer*
- **trial_until** *string*

  The date/time in UTC the organization's trial period expires in the format YYYY-MM-DDTHH:MM:SS

- **subscription_uuid** *string*
- **share_disclaimer** *string*
- **default_encryption** *integer*
- **email_templates** *boolean*

- **privacy_mode** *boolean*

  Whether privacy mode is enabled for the organization

Example

```
{
    "active": true,
    "bandwidth_throttle": null,
    "created": "2013-10-08T05:45:40",
    "default_encryption": 1,
    "description": "",
    "email": "organization@example.com",
    "email_server_id": null,
```

```json
    "email_templates": false,
    "hostname": "",
    "id": 2,
    "name": "Example Organization",
    "parent_id": 1,
    "plan_id": null,
    "policy": {
        ...a policy object...
    },
    "privacy_mode": false,
    "share_disclaimer": null,
    "slug": "example-organization",
    "subscription_uuid": null,
    "throttle_exception_days": null,
    "throttle_exception_dow": null,
    "throttle_exception_end": null,
    "throttle_exception_start": null,
    "throttle_exception_throttle": null,
    "throttled": false,
    "trial_until": null,
    "type": "organization"
}
```

Policy object

Attributes

- **type** *string*

  "policy"

- **company_id** *integer*

  The identifier of the organization to which the policy belongs

- **ad_enabled** *boolean*
- **admin_browse_files** *boolean*

  Whether administrators can browse files

- **admin_browse_remote** *boolean*

  Whether administrators can browser remote files

- **admin_create_users** *boolean*

  Whether administrators can create users

- **backups_enabled** *boolean*

  Whether backups are enabled

- **branding_enabled** *boolean*

Whether custom branding is enabled

- change_password_frequency
- deactivate_token_frequency
- excluded_extensions *string*

A comma-delimited list of file extensions not allowed to be uploaded

- file_server_enabled *boolean*
- locked_extensions *string*

A comma-delimited list of file extensions

- max_file_size *integer*

The maximum uploaded file size in XX

- monthly_cost_cents *integer*
- monthly_cost_currency *string*
- num_orgs_maximum *integer*

The maximum number of child organizations allowed

- num_users_maximum *integer*

The maximum number of users allowed

- num_users_minimum *integer*

The minimum number of users allowed

- **psa_enabled***boolean*

  Whether PSA integration is enabled

- **purge_deleted***boolean*

  Whether deleted files are automatically purged

- **purge_deleted_frequency**

  The frequency at which deleted files are automatically purged

- **require_mobile_lock***boolean*

  Whether mobile app users are required to use a PIN lock

- **require_two_step_auth***boolean*

  Whether users are required to enable two step verification

- **secure_shares***boolean*

  Whether file shares are required to be private

- **service_plans_enabled***boolean*
- **space_quota***integer*

  The maximum space allowed to be used by the organization

- **trial_length_days***integer*

- **trim_revisions***boolean*

  Whether file revisions are automatically trimmed

- **trim_revisions_x**
- **user_create_backups***boolean*

  Whether users are allowed to create backup roots

- **user_create_shares***boolean*

  Whether users are allowed to share files

- **user_lock_files***boolean*

  Whether users are allowed to lock files

- **user_purge_deleted***boolean*

  Whether users are allowed to purge deleted files

- **user_trim_revisions***boolean*

  Whether users are allowed to trim file revisions

- **webdav_enabled***boolean*

  Whether WebDAV access is enabled

Example

```json
{
    "ad_enabled": true,
    "admin_browse_files": true,
    "admin_browse_remote": true,
    "admin_create_users": true,
    "backups_enabled": true,
    "branding_enabled": true,
    "change_password_frequency": null,
    "company_id": 2,
    "deactivate_token_frequency": "",
    "excluded_extensions": ".$$,.$db,.113,.3g2,.3gp,.3gp2,.3gpp,.3mm,.a,.abf,.abk,.afm,.ani,.ann,.asf,.avi,.avs,.bac,.bak,.bck,.bcm,.bdb,.bdf,.bkf,.bkp,.bmk,.bsc,.bsf,.cab,.cf1,.chm,.chq,.chw,.cnt,.com,.cpl,.cur,.dev,.dfont,.dll,.dmp,.drv,.dv,.dvd,.dvr,.dvr-ms,.eot,.evt,.exe,.ffa,.ffl,.ffo,.ffx,.flc,.flv,.fnt,.fon,.ftg,.fts,.fxp,.gid,.grp,.hdd,.hlp,.hxi,.hxq,.hxr,.hxs,.ico,.idb,.idx,.ilk,.img,.inf,.ini,.ins,.ipf,.iso,.isp,.its,.jar,.jse,.kbd,.kext,.key,.lex,.lib,.library-ms,.lnk,.log,.lwfn,.m1p,.m1v,.m2p,.m2v,.m4v,.mem,.mkv,.mov,.mp2,.mp2v,.mp4,.mpe,.mpeg,.mpg,.mpv,.mpv2,.msc,.msi,.msm,.msp,.mst,.ncb,.nt,.nvram,.o,.obj,.obs,.ocx,.old,.ost,.otf,.pch,.pf,.pfa,.pfb,.pfm,.pnf,.pol,.pref,.prf,.prg,.prn,.pvs,.pwl,.qt,.rdb,.reg,.rll,.rox,.sbr,.scf,.scr,.sdb,.shb,.suit,.swf,.swp,.sys,.theme,.tivo,.tmp,.tms,.ttc,.ttf,.v2i,.vbe,.vga,.vgd,.vhd,.video,.vmc,.vmdk,.vmsd,.vmsn,.vmx,.vxd,.win,.wpk",
    "file_server_enabled": false,
    "locked_extensions": ".doc,.docx,.xls,.xlsx,.ppt,.pptx,.pdf,.txt,.xlsb,.xlsm,.csv,.docm,.dotx,.dotm,.pub,.wpd,.odt,.ott,.oth,.odm,.ots,.odp,.odg,.otp,.odf,.oxt,.odc,.ods,.vdx,.vsx,.vtx,.one",
    "max_file_size": 1,
    "monthly_cost_cents": 0,
    "monthly_cost_currency": "",
    "num_orgs_maximum": 10,
    "num_users_maximum": 0,
    "num_users_minimum": 0,
    "psa_enabled": true,
    "purge_deleted": false,
    "purge_deleted_frequency": "",
    "require_mobile_lock": true,
    "require_two_step_auth": false,
    "secure_shares": false,
```

    "service_plans_enabled": false,

    "space_quota": "1073741824",

    "trial_length_days": 30,

    "trim_revisions": false,

    "trim_revisions_x": "",

    "type": "policy",

    "user_create_backups": true,

    "user_create_shares": true,

    "user_lock_files": true,

    "user_purge_deleted": false,

    "user_trim_revisions": false,

    "webdav_enabled": true

}

Organization methods

Get an organization

GET /api/2/organization/<organization_id>

GET /api/2/organization/<license_key>

Response

HTTP status code: 200

Response body:

{

    ...an organization object...

}

Create an organization

POST /api/2/organization/<organization_id>/organizations/create

POST Fields

Required

- **name**: the organization name.
- **email**: the organization's contact email address.
- **slug**:

Optional

- **description**: a description of the organization. Defaults to empty.
- **parent_id**: the ID of the parent organization. Defaults to the root organization.
- **hostname**: the hostname the organization's agents will use to connect.
- **trial_until**: a date/time in the format YYYY-MM-DDTHH:MM:SS the organization trial period expires. Defaults to no trial period.
- **share_disclaimer**:
- **login_required**: whether users must log in to access the share. "true" or "false". Default "false".
- **expires**: a date in the format YYYY-MM-DD the share expires. Defaults to no expiration.
- **subscribers**: a comma-separated list of subscribers by email address.
- **notify_subscribers**: "true" or "false". Default "false".
- **download_limit**: the total number of downloads allowed for the share. Defaults to unlimited.
- **download_notify**: whether you want to be notified of downloads. "true" or "false". Default "false".
- **space_quota**: space quota in bits. Default 107374182400.
- **max_file_size**: max file size in MB. Default 300.
- **excluded_extensions**: excluded extensions. Default ".$$,.$db,.113,.3g2,.3gp,.3gp2,.3gpp,.3mm,.a,.abf,.abk,.afm,.ani,.ann,.asf,.avi,.avs,.bac,.bak,.bck,.bcm,.bd2,.bdb,.bdf,.bkf,.bkp,.bmk,.bsc,.bsf,.cab,.cf1,.chm,.chq,.chw,.cnt,.com,.cpl,.cur,.dbs,.dev,.dfont,.dll,.dmp,.drv,.dv,.dvd,.dvr,.dvr-ms,.eot,.evt,.exe,.ffa,.ffl,.ffo,.ffx,.flc,.flv,.fnt,.fon,.ftg,.fts,.fxp,.gid,.grp,.hdd,.hlp,.hxi,.hxq,.hxr,.hxs,.ico,.idb,.idx,.ilk,.img,.inf,.ini,.ins,.ipf,.iso,.isp,.its,.jar,.jse,.kbd,.kext,.key,.lex,.lib,.library-ms,.lnk,.log,.lwfn,.m1p,.m1v,.m2p,.m2v,.m4v,.mem,.mkv,.mov,.mp2,.mp2v,.mp4,.mpe,.mpeg,.mpg,.mpv,.mpv2,.msc,.msi,.msm,.msp,.mst,.ncb,.nt,.nvram,.o,.obj,.obs,.ocx,.old,.ost,.otf,.pch,.pd6,.pf,.pfa,.pfb,.pfm,.pnf,.pol,.pref,.prf,.prg,.prn,.pst,.pvs,.pwl,.QBA,.QBA.TLG,.QBW,.QBW.TLG,.qt,.rdb,.reg,.rll,.rox,.sbr,.scf,.scr,.sdb,.shb,.suit,.swf,.swp,.sys,.theme,.tivo,.tmp,.tms,.ttc,.ttf,.v2i,.vbe,.vga,.vgd,.vhd,.video,.vmc,.vmdk,.vmsd,.vmsn,.vmx,.vxd,.win,.wpk".
- **user_trim_revisions**: allow users to erase revisions? Default "false".
- **trim_revisions**: auto-erase revisions? Default "false".
- **trim_revisions_x**: erase revisions for files unchanged after a certain number of days. Default.
- **user_purge_deleted**: allow users to erase deleted files? Default "false".
- **purge_deleted**: auto-erase deleted files? Default "false".

- **purge_deleted_frequency**: erase deleted files after a certain number of days. Default never.
- **deactivate_token_frequency**: deactivate API tokens after a certain number of days. Default 30.
- **user_create_backups**: allow users to create their own backups? Default "true".
- **user_create_shares**: allow users to share files? Default "true".
- **secure_shares**: force new share links to require login? Default "false".
- **user_overwrite_collisions**: allow users to overwrite collisions? Default "false".
- **user_lock_files**: allow users to lock files? Default "false".
- **locked_extensions**: use filesystem permissions to enforce locks on. Default ".doc,.docx,.xls,.xlsx,.ppt,.pptx,.pdf,.txt,.xlsb,.xlsm,.csv,.docm,.dotx,.dotm,.pub,.wpd,.odt,.ott,.oth,.odm,.ots,.odp,.odg,.otp,.odf,.oxt,.odc,.ods,.vdx,.vsx,.vtx,.one".
- **admin_browse_files**: let organization admins browse user files? Default "true".
- **admin_browse_remote**: let organization admins browse remote files? Default "true".
- **admin_create_users**: let organization admins create users? Default "true".
- **change_password_frequency**: force password change after a certain number of days. Default never.
- **require_two_step_auth**: require two-step authentication? Default "false".
- **num_users_minimum**: min number of users. Default 0.
- **num_users_maximum**: max number of users. Default none.
- **num_orgs_maximum**: max number of suborganizations. Default 10.
- **backups_enabled**: enable backup creation? Default "true".
- **branding_enabled**: enable branding support? Default "true".
- **webdav_enabled**: enable WebDAV support? Default "true".
- **psa_enabled**: enable PSA support? Default "true".
- **ad_enabled**: enable directory server authentication support? Default "true".
- **file_server_enabled**: enable File Server Enablement? Default "true".
- **trial_length_days**: trial length in days. Default 30.
- **service_plans_enabled**: enable service plans? Default "false".
- **require_mobile_lock**: require passcode lock on mobile devices? Default "false".
- **web_preview_enabled**: allow users to preview files on the web? Default "true".

Response

Returns the created organization.

HTTP status code: 200

Response body:

{

    ...an organization object...

}

Update an organization

```
POST /api/2/organization/<organization_id>/update
```

Response

Returns the updated organization.

```
HTTP status code: 200

Response body:

{

    ...an organization object...

}
```

Delete an organization

```
POST /api/2/organization/<organization_id>/delete
```

Update an organization's policy

```
POST /api/2/organization/<organization_id>/policy/update
```

Response

Returns the updated organization.

```
HTTP status code: 200

Response body:

{

    ...an organization object...

}
```

List an organization's child organizations

```
GET /api/2/organization/<organization_id>/organizations
```

Query Parameters

- **offset**: zero-based offset from which to start listing child organizations. Defaults to 0.

Response

```
HTTP status code: 200

Response body:

{

    "offset": 0,

    "results": [

        ...up to 100 organization objects...

    ],

    "total": 42

}
```

- **offset**: the offset given in the method call or zero.
- **results**: a list of organization objects.
- **total**: the total number of child organizations.

Root methods

List an organization's roots

```
GET /api/2/organization/<organization_id>/roots
```

Query Parameters

- **offset**: zero-based offset from which to start listing roots. Defaults to 0.

Response

```
HTTP status code: 200
```

Response body:

```
{
    "offset": 0,
    "results": [
        ...up to 100 root objects...
    ],
    "total": 42
}
```

- **offset**: the offset given in the method call or zero.
- **results**: a list of root objects.
- **total**: the total number of roots.

Get a root

GET /api/2/organization/<organization_id>/root/<root_id>

Response

```
HTTP status code: 200
Response body:
{
    ...a root object...
}
```

List an organization's shares

GET /api/2/organization/<organization_id>/shares

Query Parameters

- **offset**: zero-based offset from which to start listing shares. Defaults to 0.

Response

HTTP status code: 200

Response body:

```
{
    "offset": 0,
    "results": [
        ...up to 100 root objects...
    ],
    "total": 42
}
```

- **offset**: the offset given in the method call or zero.
- **results**: a list of root objects.
- **total**: the total number of shares.

Create a share

POST /api/2/organization/<organization_id>/shares/create

POST Fields

Required

- **name**: the share name.

Optional

- **description**: a description of the share. Defaults to empty.
- **auto_lock**: enable auto-locking of files in supported applications? Default "false".
- **notify**: notify new subscribers? Default "true".
- **subscribers**: a JSON list of tuples with the first value being a person ID and the second value being a machine ID or subscription type, "all", "future", "web", or "webdav".
- **group_subscribers**: a JSON list of tuples with the first value being a group ID and the second value being a subscription type, "web", "webdav", or "machines".
- **external_subscribers**: a JSON list external subscriber email addresses. These must be existing accounts in a different organization.

Response

Returns the created share.

HTTP status code: 200

Response body:

{

    ...a root object...

}

Get a share

```
GET /api/2/organization/<organization_id>/share/<root_id>
```

Response

HTTP status code: 200

Response body:

```
{

...a root object...

}
```

Update a share

```
POST /api/2/organization/<organization_id>/share/<root_id>/update
```

POST Fields

Optional

- **company_id**: a company ID to which the share will be moved.
- **name**: the share name.
- **description**: a description of the share.
- **auto_lock**: enable auto-locking of files in supported applications?
- **notify**: notify new subscribers?

Response

Returns the updated share.

HTTP status code: 200

Response body:

```
{

    ...a root object...

}
```

List share subscribers

GET /api/2/organization/<organization_id>/share/<root_id>/subscribers

Query Parameters

- **include_from_group**: include individual subscribers from subscribed groups? Defaults to "false".

Response

HTTP status code: 200

Response body:

```
{
    "external_subscribers": {"one@example.com": "invited", "two@example.com": "accepted"},
    "from_group": [[2, "web"], [2, "webdav"], [2, "future"]],
    "group_subscribers": [[1, "web"], [1, "webdav"], [1, "machines"]],
    "subscribers": [[1, "web"], [3, "web"], [3, "future"]]
}
```

- **external_subscribers**: a dictionary of external subscribers keyed by email address with values of "invited", "accepted", or "declined" indicating the invitee's response.
- **from_group**: a list of tuples with the first value being a person ID and the second value being a machine ID or subscription type, "web", "webdav", or "future".
- **group_subscribers**: a list of tuples with the first value being a group ID and the second value being a subscription type, "web", "webdav", or "machines".
- **subscribers**: a list of tuples with the first value being a person ID and the second value being a machine ID or subscription type, "future", "web", or "webdav".

Update share subscribers

POST /api/2/organization/<organization_id>/share/<root_id>/subscribers/update

POST Fields

Optional

- **subscribers**: a JSON list of tuples with the first value being a person ID and the second value being a machine ID or subscription type, "all", "future", "web", or "webdav".
- **group_subscribers**: a JSON list of tuples with the first value being a group ID and the second value being a subscription type, "web", "webdav", or "machines".
- **external_subscribers**: a JSON list external subscriber email addresses. These must be existing accounts in a different organization.
- **remove_files**: remove files from unsubscribed agents? Default "false".

Response

HTTP status code: 200

Delete a share

POST /api/2/organization/<organization_id>/share/<root_id>/delete

POST Fields

Optional

- **remove_files**: remove files from subscribed agents? Default "false".

Response

HTTP status code: 200

Machine methods

List an organization's machines

GET /api/2/organization/<organization_id>/machines

Query Parameters

- **offset**: zero-based offset from which to start listing machines. Defaults to 0.

Response

HTTP status code: 200

Response body:

```
{
    "offset": 0,
    "results": [
        ...up to 100 machine objects...
    ],
"total": 42
}
```

- **offset**: the offset given in the method call or zero.
- **results**: a list of machine objects.
- **total**: the total number of machines.

User methods

List an organization's users

GET /api/2/organization/<organization_id>/persons

Query Parameters

- **offset**: zero-based offset from which to start listing people. Defaults to 0.

Response

HTTP status code: 200

Response body:

```
{

    "offset": 0,

    "results": [

        ...up to 100 person objects...

    ],

    "total": 42

}
```

- **offset**: the offset given in the method call or zero.
- **results**: a list of person objects.
- **total**: the total number of people.

List an organization's guests

```
GET /api/2/organization/<organization_id>/guests
```

Query Parameters

- **offset**: zero-based offset from which to start listing guests. Defaults to 0.

Response

HTTP status code: 200

Response body:

```
{

    "offset": 0,

    "results": [

        ...up to 100 guest objects...

    ],

    "total": 42
```

```
}
```

- **offset**: the offset given in the method call or zero.
- **results**: a list of guest objects.
- **total**: the total number of guests.

Authentication source methods

List an organization's authentication sources

```
GET /api/2/organization/<organization_id>/auth_sources
```

Query Parameters

- **offset**: zero-based offset from which to start listing child organizations. Defaults to 0.

Response

```
HTTP status code: 200

Response body:

{
    "offset": 0,
    "results": [
        ...up to 100 authentication source objects...
    ],
    "total": 42
}
```

- **offset**: the offset given in the method call or zero.
- **results**: a list of authentication source objects.
- **total**: the total number of authentication sources.

Activity methods

List recent activity for an organization

GET /api/2/organization/<organization_id>/activity

Query Parameters

- **offset**: zero-based offset from which to start listing activity records. Defaults to 0.

Response

HTTP status code: 200

Response body:

```
{
    "offset": 0,
    "results": [
        ...up to 100 activity objects in reverse chronological order...
    ],
    "total": 42
}
```

- **offset**: the offset given in the method call or zero.
- **results**: a list of activity objects in reverse chronological order.
- **total**: the total number of activity records available for the organization.

Errors

The following errors may occur for any organization method and are not explicitly mentioned in the method descriptions.

Authenticated user does not have access to view or modify the organization

HTTP status code: 403

Response body:

```
{
```

```
    "error": "forbidden"

}
```

Organization does not exist

```
HTTP status code: 404

Response body:

{

    "error": "not_found"

}
```

Machine object

A machine represents a device used to connect to VBoxxCloud, such as a computer, phone, or tablet.

Attributes

- **type**_string_

  "machine"

- **id**_integer_

  The unique identifier for the machine

- **dns_name**_string_

  A descriptive name for the machine, such as the host name or device manufacturer and model

- **nickname**_string_
- **last_login**_string_

  The date/time in UTC the machine last logged in in the format YYYY-MM-DDTHH:MM:SS. Used by the desktop agent.

- **guid**_string_

  A globally unique identifier assigned to the machine

- **created**_string_

The date/time in UTC the machine was created in in the format YYYY-MM-DDTHH:MM:SS

- **os_type**_string_

The machine's system type, such as "win", "osx", "android", "ios", or "winphone"

- **os_version**_string_

The machine's system version, such as "5.1.0"

- **agent_version**_string_

The agent version installed on the machine. Used by the desktop agent.

- **locked**_boolean_
- **bandwidth_throttle**_integer_
- **throttle_exception_days**_string_
- **throttle_exception_dow**_list of integers_
- **throttle_exception_start**_string_
- **throttle_exception_end**_string_
- **throttle_exception_throttle**_integer_
- **throttled**_boolean_

Whether bandwidth throttling is enabled for the machine

- **machine_type**_string_

The machine type

- last_disconnect *string*

  The date/time in UTC the machine last disconnected in in the format YYYY-MM-DDTHH:MM:SS. Used by the desktop agent.

- health_report_period_minutes

- manual_collisions *boolean*

Example

```json
{
    "agent_version": null,
    "bandwidth_throttle": null,
    "created": "2015-01-01T01:02:03",
    "dns_name": "Google Nexus 6",
    "guid": "794f7f67-3bd6-407c-9274-be735b880143",
    "health_report_period_minutes": null,
    "id": 1,
    "last_disconnect": null,
    "last_login": null,
    "locked": false,
    "machine_type": "mobile",
    "manual_collisions": true,
    "nickname": null,
    "os_type": "android",
    "os_version": "5.1.0",
    "throttle_exception_days": null,
    "throttle_exception_dow": null,
    "throttle_exception_end": null,
    "throttle_exception_start": null,
    "throttle_exception_throttle": null,
```

```
    "throttled": false,

    "type": "machine"

}
```

Machine mapping object

A machine mapping represents a path on a file server enabled machine mapped to a root.

Attributes

- **type** *string*

  "machine_mapping"

- **id** *integer*

  The unique identifier for the mapping

- **machine_id** *integer*

  The identifier of the file server enabled machine to which the root is mapped

- **name** *string*

  The mapping name

- **path** *string*

The path on the file server enabled machine to which the root is mapped

- **person_id** *integer*

The identifier of the account if the mapped root is an account root

- **root_id** *integer*

The identifier of the mapped root

- **share** *boolean*

Whether the mapped root is a share

Example

```
{
    "id": 1234,
    "machine_id": 1,
    "name": "Example Share",
    "path": "C:\Example1",
    "person_id": null,
    "root_id": 2,
    "share": true,
    "type": "machine_mapping"
}
```

Backup object

A backup represents a backup root mapped to a path on a file server enabled machine.

Attributes

- **type***string*

  "backup"

- **id***integer*

  The unique identifier for the backup

- **machine_id***integer*

  The identifier of the file server enabled machine to which the root is mapped

- **path***string*

  The path on the file server enabled machine from which files are backed up

Example

```
{
    "id": 1234,
    "machine_id": 1,
    "path": "C:\\Example",
    "type": "backup"
}
```

Machine methods

Get a machine

GET /api/2/machine/<machine_id>

Response

HTTP status code: 200

Response body:

{

   ...a machine object...

}

Get a machine's status

GET /api/2/machine/<machine_id>/status

Response

HTTP status code: 200

Response body:

```
{
  "results": {
    "subscriptions": [
      {
        "excl_files": 1,
        "rsub_id": 1234,
        "synced": true,
        "synced_folder": "C:\\Example"
      }
    ],
    "synced": true,
```

```
    "transfers": []
  }
}
```

File Server Enablement methods

Enable file server on a machine

```
POST /api/2/machine/<machine_id>/server/enable
```

Response

```
HTTP status code: 200
```

Disable file server on a machine

```
POST /api/2/machine/<machine_id>/server/disable
```

Response

```
HTTP status code: 200
```

List files on a file server enabled machine

```
POST /api/2/machine/<machine_id>/ls
```

POST Fields

Optional

- **path**: the directory path to list, or none to list drives. Default none.
- **username**: the username if path is a network path and requires authentication. Default none.
- **password**: the password if path is a network path and requires authentication. Default none.

Response

HTTP status code: 200

Response body:

```json
{
    "results": [
        "Example1",
        "Example2"
    ]
}
```

List mapped paths on a file server enabled machine

```
GET /api/2/machine/<machine_id>/mappings
```

Response

HTTP status code: 200

Response body:

```json
{
    "results": [
        ...machine_mapping objects...
    ]
}
```

Map a path on a file server enabled machine to a root

POST /api/2/machine/<machine_id>/mappings/create

POST Fields

Required

- **root_id**: the identifier of the root to be mapped.
- **path**: the path on the file server enabled machine to which the root will be mapped.

Optional

- **username**: the username if path is a network path and requires authentication. Default none.
- **password**: the password if path is a network path and requires authentication. Default none.

Response

Returns the created mapping.

HTTP status code: 200

Response body:

{

    ...a machine_mapping object...

}

Get a machine mapping

GET /api/2/machine/<machine_id>/mapping/<mapping_id>

Response

HTTP status code: 200

Response body:

{

    ...a machine_mapping object...

}

Delete a machine mapping

```
POST /api/2/machine/<machine_id>/mapping/<mapping_id>/delete
```

Response

```
HTTP status code: 200
```

Backup methods

List backups

```
GET /api/2/machine/<machine_id>/backups
```

Response

```
HTTP status code: 200
Response body:
{
    "results": [
        ...backup objects...
    ]
}
```

Create a backup

```
POST /api/2/machine/<machine_id>/backups/create
```

POST Fields

Required

- **path**: the path on the file server enabled machine to which the root will be mapped.

Optional

- **username**: the username if path is a network path and requires authentication. Default none.
- **password**: the password if path is a network path and requires authentication. Default none.

Response

Returns the created backup.

HTTP status code: 200

Response body:

{

   ...a backup object...

}

Get a backup

GET /api/2/machine/<machine_id>/backup/<root_id>

Response

HTTP status code: 200

Response body:

{

   ...a backup object...

}

Restore a backup

POST /api/2/machine/<machine_id>/backup/<root_id>/restore

POST Fields

Required

- **path**: the path on the file server enabled machine to which the backup will be restored.

Optional

- **to_machine_id**: the id of the machine to which the backup will be restored. Defaults to restore to the machine the backup is on.

Response

Returns the updated backup object.

HTTP status code: 200

Response body:

```
{
    ...a backup object...
}
```

Delete a backup

POST /api/2/machine/<machine_id>/backup/<root_id>/delete

Response

HTTP status code: 200

Errors

The following errors may occur for any machine method and are not explicitly mentioned in the method descriptions.

Authenticated user does not have access to view or modify the machine

HTTP status code: 403

Response body:

```
{
    "error": "forbidden"
}
```

Machine does not exist

HTTP status code: 404

Response body:

```
{
    "error": "not_found"
}
```

Guest

Guest object

A guest represents a restricted account with limited access to shared files.

Attributes

- **type** *string*

  "guest"

- **id** *integer*

  The unique identifier for the guest

- **company_id** *integer*

  The identifier of the company to which the guest belongs

- **creator_id** *integer*

  The identifier of the person who created the guest

- **email** *string*

  The guest's email address

- **first_name** *string*

  The guest's first name

- **last_name** *string*

  The guest's last name

- **active** *boolean*

  Whether the guest account is active

- **created** *string*

  The date/time in UTC the guest was created in the format YYYY-MM-DDTHH:MM:SS

Example

```json
{
    "active": true,
    "company_id": 1,
    "created": "2015-01-01T01:02:03",
    "creator_id": 1,
    "email": "guest@example.com",
    "first_name": "First",
    "id": 1,
    "last_name": "Last",
    "type": "guest"
}
```

Guest methods

Get a guest

Retrieve a guest by ID or email address.

```
GET /api/2/guest/<guest_id>

GET /api/2/guest/<email>
```

Response

```
HTTP status code: 200

Response body:

{

    ...a guest object...

}
```

Create a guest

```
POST /api/2/guest/create
```

POST Fields

Required

- **company_id**: the identifier of the company in which to create the guest.
- **first_name**: the guest's first name.
- **last_name**: the guest's last name.
- **email**: the guest's email address.
- **password**: the guest's password. Required if generate_password is "false".
- **generate_password**: whether to automatically generate a password. Required if password is not provided. Default "false".

Optional

- **send_welcome_email**: send a welcome email to the guest? Default "false".
- **pw_expires**: number of hours until the password expires. Valid options are empty (never), "6", "12", and "24". Default never.

Response

Returns the created guest.

HTTP status code: 200

Response body:

```
{

    ...a guest object...

}
```

Update a guest

POST /api/2/guest/<guest_id>/update

POST Fields

Optional

- **first_name**: the guest's first name.
- **last_name**: the guest's last name.
- **email**: the guest's email address.
- **password**: the guest's password.
- **pw_expires**: number of hours until the password expires. Valid options are empty (never), "6", "12", and "24". Default never.

Response

Returns the updated guest.

HTTP status code: 200

Response body:

```
{

    ...a guest object...

}
```

Delete a guest

```
POST /api/2/guest/<guest_id>/delete
```

Response

```
HTTP status code: 200
```

Get files and folders shared with a guest

```
GET /api/2/guest/<guest_id>/shares
```

Response

```
HTTP status code: 200

Response body:

{

    "results": [

        ...file share objects...

    ]

}
```

Activity object

An activity represents a logged action including details of the acting entity and the affected entity.

Attributes

- **type**_string_

  "activity"

- **id**_integer_

  The unique identifier for the activity

- **actor_type**_string_

  The entity type triggering the activity

- **actor_text**_string_

  The description of the entity triggering the activity

- **actor_company_id**_integer_
- **actor_machine_id**_integer_
- **actor_person_id**_integer_
- **actor_guest_id**_integer_
- **action_text**_string_

  Additional description of the action taken

- **acted_on_type**_string_

The entity type being acted upon

- **acted_on_text** *string*

The description of the entity type being acted upon

- **acted_on_company_id** *integer*
- **acted_on_machine_id** *integer*
- **acted_on_person_id** *integer*
- **acted_on_guest_id** *integer*
- **acted_on_root_id** *integer*
- **acted_on_group_id** *integer*
- **data_text** *string*

The description of the action taken

- **activity_type_id** *integer*

The identifier of the activity type for this activity

- **created** *string*

The date/time in UTC the activity occurred in the format YYYY-MM-DDTHH:MM:SS

- **processed** *boolean*

Example

```
{
    "acted_on_company_id": 1,
    "acted_on_group_id": null,
    "acted_on_guest_id": null,
```

```
    "acted_on_machine_id": null,

    "acted_on_person_id": null,

    "acted_on_root_id": null,

    "acted_on_text": "Example Company",

    "acted_on_type": "site",

    "action_text": null,

    "activity_type_id": 8,

    "actor_company_id": null,

    "actor_guest_id": null,

    "actor_machine_id": null,

    "actor_person_id": 1,

    "actor_text": "First Last",

    "actor_type": "person",

    "created": "2015-01-01T01:02:03",

    "data_text": "increased from 4.00G to 100.00G",

    "id": 123,

    "processed": true,

    "type": "activity"

}
```

Activity type object

Attributes

- **type** *string*

  "activity_type"

- **id** *integer*

The unique identifier for the activity type

- **activity** *string*

The activity description

- **machine_scope** *boolean*

Whether the activity is performed within the scope of a machine

- **user_scope** *boolean*

Whether the activity is performed within the scope of a user

- **system_scope** *boolean*

Whether the activity is performed at the system level

Example

```json
{
    "activity": "added organization",
    "id": 3,
    "machine_scope": false,
    "system_scope": false,
    "type": "activity_type",
    "user_scope": false
}
```

Activity methods

Get a list of activity types

GET /api/2/activity/types

Response

HTTP status code: 200

Response body:

```
{
    "results": [
        ...a list of activity objects...
    ]
}
```

Get an activity record

GET /api/2/activity/<activity_id>

Response

HTTP status code: 200

Response body:

```
{
    ...an activity object...
}
```

Create an activity record

POST /api/2/activity/create

Response

Returns the created activity record.

```
HTTP status code: 200

Response body:

{

    ...an activity object...

}
```

Errors

The following errors may occur for any activity method and are not explicitly mentioned in the method descriptions.

Authenticated user does not have access to view or modify the activity or activity type

```
HTTP status code: 403

Response body:

{

    "error": "forbidden"

}
```

Activity or activity type does not exist

```
HTTP status code: 404

Response body:

{

    "error": "not_found"

}
```

Authentication source object

An authentication source represents a directory server from which users may be imported.

Attributes

- **type** *string*

  "auth_source"

- **id** *integer*

  The unique identifier for the authentication source

- **host** *string*

  The hostname of the directory server

- **domain** *string*

  The domain name on the directory server to use for importing users

- **login** *string*

  The user name used to access the directory server

- **port** *integer*

  The port number of the directory server

- **protocol**_integer_
- **ssl**_boolean_

Whether the directory server uses SSL/TLS

- **machine_id**_integer_

Example

```
{
    "domain": "example.com",
    "host": "localhost",
    "id": 1,
    "login": "example",
    "machine_id": null,
    "port": null,
    "protocol": null,
    "ssl": false,
    "type": "auth_source"
}
```