
MC COAL Documentation

Release

Thomas Bohmbach, Jr.

October 10, 2014

1 Features	3
2 Documentation	5
2.1 COAL Installation	5
2.2 World Hosting	7
2.3 API	8
HTTP Routing Table	61

MC COAL is an open-source project aimed at helping to build and maintain community around multi-player [Minecraft](#) worlds. It can also optionally provide powerful and simple Minecraft multi-player server hosting and management.

COAL is a web application. It is written in python and runs on the [Google Cloud Platform](#). It is licensed under the [GNU AFFERO GENERAL PUBLIC LICENSE Version 3](#)

Features

- Server status display including in-game time, weather, and currently logged in players
- Live chat and searchable chat history
- Chat into the game from the web
- Player session information
- Share screen shots
- Detailed and well-documented API
- Easily host and manage powerful multi-player servers on Google Compute Engine
- Works with vanilla Minecraft multi-player servers

2.1 COAL Installation

2.1.1 Prerequisites

- A free [Google Cloud Platform](#) account.
- Optional But Recommended: A free [Github](#) account.

2.1.2 Deploy and Configure COAL

1. Create a [Google Cloud Project](#) for your new COAL installation. Take note of the Project ID you select (e.g. *[my-project-id]*).
2. Optionally, in *Settings*, enable billing. COAL should be able to run comfortably under the daily free App Engine quota for most relatively small, lightly-populated worlds.
3. Fork, clone, or download the [MC COAL code repository](#). Make sure any new fork or clone is a private repository as it will contain sensitive information (like the `main_SECRET_KEY`).

Note: The master/trunk of the [MC COAL code repository](#) will always contain the latest tagged, stable release. Ongoing (potentially unstable) development will be done on branches.

4. Change the application name (i.e. `mc-coal`) in the first line of your repository's `app.yaml` to the Project ID you created above.
5. Update your installation settings in your repository's `appengine_config.py` file:
 1. Change the `main_SECRET_KEY` value to a unique random string. You can use this [random.org link](#) to generate a unique string value.
 2. Change the `main_TITLE` and `main_DESCRIPTION` to strings that are appropriate for your COAL installation.
 3. You do not have to change `oauth_SECRET_KEY` or `oauth_TOKEN_EXPIRES_IN`. The defaults will work just fine.
6. Complete the sub-steps below if you intend to host worlds on Google Compute Engine:
 1. In *Settings* enable billing if you haven't already. There is no daily free Google Compute Engine quota.

2. In APIs, make sure that Google Compute Engine, Google Cloud Storage and Google Cloud Storage JSON API are ON.
3. In Permissions make note of the Google Compute Engine service account email address. This should be of the form [project number]@developer.gserviceaccount.com or [project number]@project.gserviceaccount.com. Also, make sure there is an entry for [my-project-id]@appspot.gserviceaccount.com. If it isn't there, add it as a owner member.
4. Add the Google Compute Engine service account email address to the acl section of queue.yaml. When finished, the file should look something like this

```
queue:
- name: default
  rate: 5/s

- name: controller
  mode: pull
  acl:
  - user_email: 1234567890@developer.gserviceaccount.com
```

7. Deploy your modified code in either of two ways:

- Use the [Push-to-Deploy Release Pipeline](#) to deploy automatically from your github repository.
 1. Finish the steps under [Setting up a Release Pipeline](#)

Note: You do not have to complete Step 1 (install Git on your local system) if you intend to deploy only from a Github-hosted fork of the [MC COAL code repository](#).

2. Click the “Connect to a repository hosted on Github” button.
 3. After authenticating with github, choose the clone or fork of the repository you created in step 3 above.
 4. Choose the “Deploy Source Only” option.
 5. Optionally enter your email address to receive status updates for your deployments.
 6. Make a change to any file in your repository (for instance, add a blank line or comment to appengine_config.py) and save/push the change. This should initiate a deploy. If you entered your email address when setting up the Release Pipeline you should get an email when the deploy completes.
- If you are familiar with [Google Cloud Platform python development](#), feel free to use the [App Engine python developer tools](#)
8. After the deployment has completed, browse to your COAL administrator page at [https://\[my-project-id\].appspot.com/admin](https://[my-project-id].appspot.com/admin).

Note: It may take a few minutes after an initial successful deployment for database indexes to build. If you get a 500 error response when browsing your COAL site right after deployment, you might have to wait a few minutes for the indexes to finish building.

Warning: For bootstrapping purposes, the first user to request this page is made an administrator, so make sure to do this as soon as possible.

2.2 World Hosting

Next, you'll set up your minecraft world(s). There are two options: let your COAL host your world on Google Compute Engine (easy!) or host your world elsewhere (more work for you!). You can mix both kinds of hosted worlds on a single COAL install.

2.2.1 Hosting Worlds On Google Compute Engine

1. Define a new minecraft version in Admin/Define New Minecraft Version/URL. Enter a version name (i.e. 1.7.10) and the URL where the Minecraft Server JAR for that version can be downloaded (i.e. `https://s3.amazonaws.com/Minecraft.Download/versions/1.7.10/minecraft_server.1.7.10`).

Note: A list of all available versions and server JAR download links is available at mcversions.net. COAL has been tested with versions as far back as 1.4.7.

2. Create a new server in Admin/Create GCE-Hosted World. Then hit the play button to start the server. This can take a few minutes if a GCE instance has to be started up for the first time.
3. When the world status is "Playing" the IP address of the server will be shown. Use this IP address to connect your minecraft client to the new world.
4. Play! No additional infrastructure set up needed.
5. Additional administrator settings are available in Admin/Configure. Here you can modify settings such as the type of machine instance to use (which determines the speed of the CPU and amount of memory available), the size of the disk (larger disks are faster), the number of saved game versions to keep in the cloud, and whether to use a static IP address.

Note: Changes made on the Admin Configuration page won't be live until a new GCE instance is started. To shut down the currently running instance, click the "Kill Instance" button on the Admin page. To start a new instance, hit "Play" for one of your worlds.

Warning: Make sure all worlds are paused before killing the instance. Failure to do so may result in corrupted world files. Note that large worlds can take a few minutes to shutdown and save.

2.2.2 Hosting Worlds On Other Servers

If you already have a Minecraft multi-player world running on a UNIX-based server you can connect it to your COAL.

Hosting Server Prerequisites

The following libraries are required to be installed on the server that is hosting your world.

- Python 2.7
- pytz
- pyyaml
- requests

- NBT

Agent Installation

1. Create a new directory called `mc-coal` in your minecraft server's directory (the one with `server.properties`).
2. Download the following files from your COAL into the new `mc-coal` directory:

```
wget https://[my-project-id].appspot.com/mc/timezones.py -o timezones.py
wget https://[my-project-id].appspot.com/mc/mc_coal_agent.py -o mc_coal_agent.py
```

3. Download the following files from your COAL into your minecraft server's directory

```
wget https://[my-project-id].appspot.com/mc/log4j2.xml -o log4j2.xml
wget https://[my-project-id].appspot.com/mc/mc-start.sh -o mc-start.sh
wget https://[my-project-id].appspot.com/mc/mc-stop.sh -o mc-stop.sh
```

Run Agent

1. Create a new COAL world by clicking the Admin/Create External-Server-Hosted World and note the Agent Client ID and Agent Secret.
2. On your Minecraft server host, in the `mc-coal` directory, run `mc_coal_agent.py` with the `coal_host`, `agent_client_id`, and `agent_secret` for your server:

```
[~/minecraft-server/mc-coal] $ python mc_coal_agent.py --coal_host=[my-project-id].appspot.c
2014-01-01 23:00:01 : main      INFO    Monitoring './server.log' and reporting to '[my-proje
```

3. Edit the java configuration parameters within the `mc-start.sh` script as desired.
4. Start the minecraft server with the `mc-start.sh` script:

```
[~/minecraft-server] $ ./mc-start.sh
2014-01-21 22:15:09,540 DEBUG Generated plugins in 0.000023000 seconds
...
2014-01-21 22:15:09,588 DEBUG Shutting down OutputStreamManager SYSTEM_OUT
2014-01-21 22:15:09,588 DEBUG Reconfiguration completed
```

5. To stop the minecraft server later, use the `mc-stop.sh` script:

```
[~/minecraft_server] $ ./mc-stop.sh
Stopping MineCraft Server PID=5989
2014-01-22 22:12:19,540 DEBUG ServletContext not present - WebLookup not added
2014-01-22 22:12:19,541 DEBUG Shutting down FileManager server.log
MineCraft shutdown complete.
```

2.3 API

The API v1 web services allow read and limited write access to various resources in the COAL datastore.

2.3.1 Common

Secured Resources

Clients making calls to the API on behalf of a user require a bearer access token which can be acquired via a simple *authorization* flow.

GET `/api/v1/` (*resource*)

Request Headers

- **Authorization** – An *access token* using the “Bearer” scheme as specified in RFC6750: *Authorization Request Header Field*. The user that granted authorization for the access token will be considered the “authenticated user” for resources that expect one.

Status Codes

- **401 Unauthorized** – Invalid or no *Authorization* request header provided.
- **403 Forbidden** – The authorization was not granted by an active user.

Example:

```
GET /api/v1/(resource) HTTP/1.1
Authorization: Bearer 8wB8QtpULBVNuL2mqBaWdIRWX30qKtIK3E5QbOWP
```

POST `/api/v1/` (*resource*)

Request Headers

- **Authorization** – An *access token* using the “Bearer” scheme as specified in RFC6750: *Authorization Request Header Field*. The user that granted authorization for the access token will be considered the “authenticated user” for resources that expect one.

Status Codes

- **401 Unauthorized** – Invalid or no *Authorization* request header provided.
- **403 Forbidden** – The authorization was not granted by an active user.

Example:

```
POST /api/v1/(resource) HTTP/1.1
Authorization: Bearer 8wB8QtpULBVNuL2mqBaWdIRWX30qKtIK3E5QbOWP
```

Status Codes

- **200 OK**

The body will be a JSON object whose contents are resource specific:

```
{
  "key1": value1,
  "key2": value2,
  ...
}
```

- **201 Created**

The body will be a JSON object whose contents are resource specific:

```
{
  "key1": value1,
  "key2": value2,
  ...
}
```

- **400 Bad Request**

The body will be a JSON object of the form:

```
{
  "errors": "This was a bad request because..."
}
```

The `errors` string is resource and error specific.

- **403 Forbidden** – The body will be empty.
- **404 Not Found** – The body will be empty.
- **405 Method Not Allowed** – The body will be empty.
- **500 Internal Server Error**

The body will be a JSON object of the form:

```
{
  "errors": "This request failed because..."
}
```

The `errors` string is resource and error specific.

Request Parameters

Resources that accept request parameters expect them in the request body using the `application/x-www-form-urlencoded` or `application/json` formats with a character encoding of UTF-8.

Timestamps

Unless otherwise specified, all timestamps are of the form `%Y-%m-%d %H:%M:%S %Z-%z` (see [Python strftime formatting](#)) and returned as UTC unless otherwise noted.

Example timestamp:

```
"2013-04-14 19:55:22 UTC-0000"
```

List Resources

Some resources return a list of results that can span requests. These resources all take a common set of query parameters and return a common set of response data to help iterate through large lists of data.

GET `/api/v1/` (*list_resource*)

Query Parameters

- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.

Status Codes

- **200 OK** – Successfully called the *list_resource*.

Response Data

- **cursor** – If more results are available, this root level response value will be the next cursor string to be passed back into this resource to grab the next set of results. If no more results are available, this field will be absent.

Example first request:

```
GET /api/v1/(list_resource)?size=5 HTTP/1.1
```

Example first response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "results": ["result1", "result2", "result3", "result4", "result5"],
  "cursor": "hsajkhasjkdy8y3h3h8fhih38djhhdjdj"
}
```

Example second request:

```
GET /api/v1/(list_resource)?size=5&cursor=hsajkhasjkdy8y3h3h8fhih38djhhdjdj HTTP/1.1
```

Example second response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "results": ["result6", "result7", "result8"]
}
```

2.3.2 Authorization

Clients making calls to the *API* on behalf of a user require an *access token* which can be acquired via the following (simplified) *OAuth 2.0* flow.

1. *Register* the client via `POST /oauth/v1/register` and record the *client configuration* contained in the response. These *client configuration* values should be kept secure.
2. *Redirect* the user's browser to `GET /oauth/v1/auth` with the required *client configuration* values in the query parameters, including a `redirect_uri`. If the user grants the authorization to the client, the user's browser will be redirected to the `redirect_uri` with an *authorization code*.
3. *Request* `POST /oauth/v1/token` with the *authorization code* plus other *client configuration* values and record the *access token* and *refresh token* contained in the response.
4. *Refresh* the *access token* if it expires or otherwise becomes invalid using the *refresh token*.

Client Registration

Clients *register* with the *API* using an *open registration lifecycle*. Most client registration and configuration endpoints (`POST /oauth/v1/register`, `GET /oauth/v1/clients/(client_id)`,

and `PUT /oauth/v1/clients/(client_id)` return an `application/json` response body that is an object with the client configuration as top-level members:

Client Configuration

- **client_id** – The client id.
- **redirect_uris** – A list of redirect URIs (strings) for use in other oauth flows. Specifically, one of these URIs must always be used whenever a `redirect_uri` is required.
- **scope** – A space separated list of scope values that the client can use when requesting access tokens.
- **client_secret** – The client secret for use in other oauth flows.
- **client_secret_expires_at** – Time at which the `client_secret` will expire or 0 if it will not expire. The time is represented as the number of seconds from `1970-01-01T0:0:0Z` as measured in UTC until the date/time.
- **registration_access_token** – The access token that is used at the client configuration endpoint to perform subsequent operations upon the client registration through the client configuration endpoints (`GET /oauth/v1/clients/(client_id)`, `PUT /oauth/v1/clients/(client_id)`, and `DELETE /oauth/v1/clients/(client_id)`).
- **registration_client_uri** – The fully qualified URL of the client configuration endpoint for this client. The client **MUST** use this URL as given when communicating with the client configuration endpoint.
- **client_name** – (*optional*) – The human-readable name of the client to be presented to the user.
- **client_uri** – (*optional*) – The URL of the homepage of the client.
- **logo_uri** – (*optional*) – The URL that references a logo for the client.

POST /oauth/v1/register

JSON Parameters

- **redirect_uris** (*string_array*) – An array of redirect URIs for use in other oauth flows.
- **client_id** (*string*) – (*optional*) – A requested client id. If a client is already registered with the same client id, a unique client id based on the requested one will be created instead. If this parameter is omitted, a completely random client id will be created.
- **client_name** (*string*) – (*optional*) – The human-readable name of the client to be presented to the user.
- **client_uri** (*string*) – (*optional*) – The URL of the homepage of the client.
- **logo_uri** (*string*) – (*optional*) – The URL that references a logo for the client.
- **scope** (*string*) – (*optional*) – A space separated list of scope values that the client can use when requesting access tokens. Currently, the only valid value is `"data"`.

Status Codes

- **201 Created** – Successfully created a new client. The application/json response body will be an object with the *client configuration* as top-level members.
- **400 Bad Request** – The application/json response body will be an object with the error information as top-level members:

Response Data

- **error** – The error. Possible values are `invalid_request` and `server_error`.

Example request:

```
POST /oauth/v1/register HTTP/1.1
```

```
{
  "redirect_uris": ["http://example.com/callback"],
  "client_id": "my_example_app",
  "client_name": "My Example Application",
  "client_uri": "http://example.com",
  "logo_uri": "http://example.com/logo.png",
  "scope": "data"
}
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
```

```
{
  "client_id": "my_example_app",
  "redirect_uris": ["http://example.com/callback"],
  "scope": "data",
  "client_secret": "bdv8HtrspbJh5F5K01AUkD0l8KAyYcfsDQoTk1au",
  "client_secret_expires_at": 0,
  "registration_access_token": "VlhLNF2vifRspohNr7gBcbc005khEqADalH1PYE",
  "registration_client_uri": "https://my-coal.org/oauth/v1/clients/my_example_app",
  "client_name": "My Example Application",
  "client_uri": "http://example.com",
  "logo_uri": "http://example.com/logo.png"
}
```

Client Configuration

The client configuration endpoint is a protected resource that is provisioned by the server to facilitate viewing, updating, and deleting a client’s registered information. If a client ever forgets its *client configuration* values, they can be retrieved via `GET /oauth/v1/clients/(client_id)` as long as the client knows its `registration_client_uri` and `registration_access_token`.

The location of this endpoint is communicated to the client through the `registration_client_uri` member of the `POST /oauth/v1/register` response. Authorization for this endpoint requires that the client’s `registration_access_token` be set in the request Authorization header field using the “Bearer” scheme as specified in [RFC6750: Authorization Request Header Field](#).

GET /oauth/v1/clients/(client_id)

Read the current configuration of the client (*client_id*).

Request Headers

- **Authorization** – The client’s `registration_access_token` using the “Bearer” scheme as specified in RFC6750: [Authorization Request Header Field](#).

Response Headers

- **WWW-Authenticate** – If there is a problem with authorization, the value will be `Bearer error="invalid_token"` as specified in RFC6750: [WWW-Authenticate Response Header Field](#).

Status Codes

- **200 OK** – Successfully returned the client configuration. The `application/json` response body will be an object with the `client_configuration` as top-level members. Some of these values, including the `client_secret`, `client_secret_expires_at`, and `registration_access_token`, may be different from those in the initial `POST /oauth/v1/register` response. If there is a new client secret and/or registration access token in the response, the client must immediately discard its previous client secret and/or registration access token. The value of the `client_id` will not change from the initial `POST /oauth/v1/register` response.
- **401 Unauthorized** – Invalid or no Authorization request header provided. The `WWW-Authenticate` response header will contain the error.

Example request:

```
GET /oauth/v1/clients/my_example_app HTTP/1.1
Authorization: Bearer VlhLNF2vifRspohNr7gBcbc005khEqADalH1PYE
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "client_id": "my_example_app",
  "redirect_uris": ["http://example.com/callback"],
  "scope": "data",
  "client_secret": "bdv8HtrspbJh5F5K01AUkD018KAyYcfsDQoTk1au",
  "client_secret_expires_at": 0,
  "registration_access_token": "VlhLNF2vifRspohNr7gBcbc005khEqADalH1PYE",
  "registration_client_uri": "https://my-coal.org/oauth/v1/clients/my_example_app",
  "client_name": "My Example Application",
  "client_uri": "http://example.com",
  "logo_uri": "http://example.com/logo.png"
}
```

PUT /oauth/v1/clients/ (*client_id*)
Update the configuration of the client (*client_id*).

Request Headers

- **Authorization** – The client’s `registration_access_token` using the “Bearer” scheme as specified in RFC6750: [Authorization Request Header Field](#).

Response Headers

- **WWW-Authenticate** – If there is a problem with authorization, the value will be `Bearer error="invalid_token"` as specified in [RFC6750: WWW-Authenticate Response Header Field](#).

JSON Parameters

- **client_id** (*string*) – The client id. If not correct, a [400 Bad Request](#) `invalid_client_id` response will result.
- **redirect_uris** (*string_array*) – The new client redirect URIs.
- **client_secret** (*string*) – The client secret. If this value does not match the current client secret, a [400 Bad Request](#) `invalid_request` response will result.
- **scope** (*string*) – (*optional*) – A space separated list of scope values. If there are new values that are not part of the current scope, a [400 Bad Request](#) `invalid_request` response will result. Note that this means a client can remove scope values, but can never add them. If not present, the client scope will be unmodified.
- **client_name** (*string*) – (*optional*) – The new human-readable name of the client. If not present, the client name will be set to `null`.
- **client_uri** (*string*) – (*optional*) – The new URL of the homepage of the client. If not present, the homepage URL will be set to `null`.
- **logo_uri** (*string*) – (*optional*) – The new URL that references a logo for the client. If not present, the logo URL will be set to `null`.

Status Codes

- [200 OK](#) – Successfully updated the client configuration. The `application/json` response body will be an object with the new *client configuration* as top-level members. Some of these values, including the `client_secret`, `client_secret_expires_at`, and `registration_access_token`, may be different from those in the initial `POST /oauth/v1/register` response. If there is a new client secret and/or registration access token in the response, the client must immediately discard its previous client secret and/or registration access token. The value of the `client_id` will not change from the initial `POST /oauth/v1/register` response.
- [400 Bad Request](#) – The `application/json` response body will be an object with the error information as top-level members:

Response Data

- **error** – The error. Possible values are `invalid_request`, `invalid_client_id`, and `server_error`.
- [401 Unauthorized](#) – Invalid or no Authorization request header provided. The `WWW-Authenticate` response header may be set and contain the error.

Example request:

```
PUT /oauth/v1/clients/my_example_app HTTP/1.1
Authorization: Bearer VlhLNF2vifRspohNr7gBcbc005khEqADa1H1PYE
```

```
{
  "client_id": "my_example_app",
  "redirect_uris": ["http://example.com/v2/callback"],
  "client_secret": "bdv8HtrspbJh5F5K0lAUkD0l8KAYYcfsDQoTk1au",
  "scope": "data",
  "client_name": "My Example Application v2",
  "client_uri": "http://example.com/v2",
  "logo_uri": "http://example.com/logo_v2.png",
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "client_id": "my_example_app",
  "redirect_uris": ["http://example.com/v2/callback"],
  "scope": "data",
  "client_secret": "bdv8HtrspbJh5F5K0lAUkD0l8KAYYcfsDQoTk1au",
  "registration_access_token": "VlhLNF2vifRspohNr7gBcbc005khEqADalHlPYE",
  "registration_client_uri": "https://my-coal.org/oauth/v1/clients/my_example_app",
  "client_name": "My Example Application v2",
  "client_uri": "http://example.com/v2",
  "logo_uri": "http://example.com/logo_v2.png"
}
```

DELETE /oauth/v1/clients/ (*client_id*)

Remove the client and all grants and tokens associated with it (*client_id*).

Request Headers

- **Authorization** – The client’s `registration_access_token` using the “Bearer” scheme as specified in RFC6750: [Authorization Request Header Field](#).

Response Headers

- **WWW-Authenticate** – If there is a problem with authorization, the value will be `Bearer error="invalid_token"` as specified in RFC6750: [WWW-Authenticate Response Header Field](#).

Status Codes

- **204 No Content** – Successfully deprovisioned the client.
- **401 Unauthorized** – Invalid or no `Authorization` request header provided. The `WWW-Authenticate` response header may be set and contain the error.

Example request:

```
DELETE /oauth/v1/clients/my_example_app HTTP/1.1
Authorization: Bearer VlhLNF2vifRspohNr7gBcbc005khEqADalHlPYE
```

Example response:

```
HTTP/1.1 204 No Content
```

Authorization Code

Clients are granted a unique, one-time-use authorization code in response to an explicit, web-based authorization grant from a logged-in user.

GET /oauth/v1/auth

A user-facing web UI to prompt the user to grant or deny OAuth access for a client.

Query Parameters

- **client_id** – The client id to authorize.
- **redirect_uri** – The fully qualified URL that the user’s browser will redirect to with the access code or error. This must be one of the URIs in the client’s configuration `redirect_uris`.
- **response_type** – This should always be `code` when requesting an access code.
- **scope** – The scope for the authorization code request. Must always be `data`.

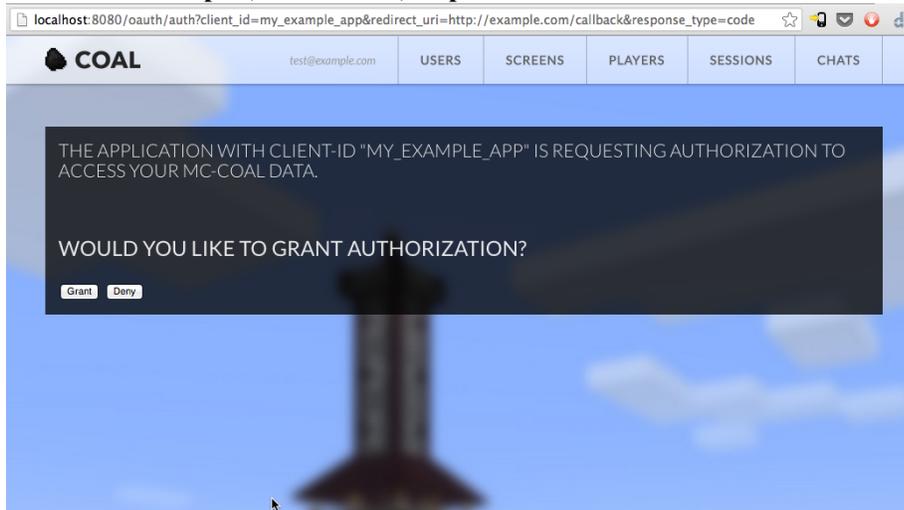
Status Codes

- **302 Found** – If the user grants authorization, the user’s browser will redirect to the `redirect_uri` with the authorization code passed via the `code` query parameter.
- **302 Found** – If the user denies authorization or an error occurs, the user’s browser will redirect to the `redirect_uri` with the error passed via the `error` query parameter.

Example (user browser) request:

```
GET /oauth/v1/auth?client_id=my_example_app&redirect_uri=http://example.com/callback&response_type=code
```

Example (user browser) response:



If the user grants authorization to the client, a **302 Found** response is returned to the user’s browser. The `Location` header in the response is set to the `redirect_uri` with the `code` query parameter set to the authorization code:

```
HTTP/1.1 302 Found
Location: http://example.com/callback?code=Yehb6FW0cPgnTUWtHwPcgBEojQjhU619YfshnqVd
```

If the user denies authorization to the client, a [302 Found](#) response is returned to the user's browser. The `Location` header in the response is set to the `redirect_uri` with the `error` query parameter set:

```
HTTP/1.1 302 Found
Location: http://example.com/callback?error=access_denied
```

Access Token

Clients use an *authorization code* to acquire an *access token* and a *refresh token*. These tokens are unique and tied to both the client and the user that granted the authorization code. Authorization for *secured Data APIs* requires that a valid access token be set in the request `Authorization` header field using the "Bearer" scheme as specified in [RFC6750: Authorization Request Header Field](#).

POST /oauth/v1/token

The client acquires tokens by making a request to the token endpoint, posting the following parameters in the request body using the `application/x-www-form-urlencoded` or `application/json` formats with a character encoding of UTF-8.

Form Parameters

- **client_id** – The client id.
- **client_secret** – The current client secret.
- **grant_type** – Should be `authorization_code` to convert an authorization code into an access token.
- **code** – The authorization code.
- **redirect_uri** – The fully qualified redirect URL. This must be one of the URIs in the client's configuration `redirect_uris`.
- **scope** – The scope for the access token. Must always be `data`.

Status Codes

- **200 OK** – Successfully converted the authorization code into access and refresh tokens. The `application/json` response body will be an object with the token information as top-level members:

Response Data

- **access_token** – The access token.
 - **refresh_token** – The refresh token.
 - **expires_in** – The lifetime in seconds of the access token.
 - **token_type** – Will always be `Bearer`
- **400 Bad Request** – The `application/json` response body will be an object with the error information as top-level members:

Response Data

- **error** – The error. Possible values are:
 - * `invalid_request` – Missing parameters.
 - * `unsupported_grant_type` – Incorrect grant type.
 - * `invalid_grant` – Incorrect access code or redirect URI.

- * `invalid_client` – Incorrect client id or client secret.
- * `invalid_scope` – Incorrect scope.
- * `server_error` – Generic server error.

Example request

```
POST /oauth/v1/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded

client_id=my_example_app&
client_secret=bdv8HtrspbJh5F5K01AUkD018KAyYcfsDQoTk1au&
grant_type=authorization_code&
code=YEhb6FWOcPgnTUWtHwPcgBEojQjhU619YfshnqVd&
redirect_uri=http%3A%2F%2Fexample.com%2Fcallback&
scope=data
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "access_token": "wIt7U1cpa5B4Rqbbvie6Mye1sWiwAjZ7H7kAXIjK",
  "token_type": "Bearer",
  "expires_in": 3600,
  "refresh_token": "PuFZ2Hyu6R6eIAxVG9Y4j4kFRYsCapISTR0n3AUM"
}
```

Refresh Token

When an access token expires, or otherwise becomes invalid, a one-time-use refresh token can be used to generate a new set of tokens (access and refresh).

POST /oauth/v1/token

The client acquires tokens by making a request to the token endpoint, posting the following parameters in the request body using the `application/x-www-form-urlencoded` or `application/json` formats with a character encoding of UTF-8.

Form Parameters

- `client_id` – The client id.
- `client_secret` – The current client secret.
- `grant_type` – Should be `refresh_token` to generate a new set of tokens.
- `refresh_token` – The refresh token.
- `scope` – The scope for the access token. Must always be `data`.

Status Codes

- `200 OK` – Successfully generated a new set of access and refresh tokens. The `application/json` response body will be an object with the token information as top-level members:

Response Data

- **access_token** – The access token.
 - **refresh_token** – The refresh token.
 - **expires_in** – The lifetime in seconds of the access token.
 - **token_type** – Will always be `Bearer`
- **400 Bad Request** – The `application/json` response body will be an object with the error information as top-level members:

Response Data

- **error** – The error. Possible values are:
 - * `invalid_request` – Missing parameters.
 - * `unsupported_grant_type` – Incorrect grant type.
 - * `invalid_grant` – Incorrect refresh token.
 - * `invalid_client` – Incorrect client id or client secret.
 - * `invalid_scope` – Incorrect scope.
 - * `server_error` – Generic server error.

Example request

```
POST /oauth/v1/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded

client_id=my_example_app&
client_secret=bdv8HtrspbJh5F5K0lAUkD0l8KAyYcfsDQoTk1au&
grant_type=refresh_token&
refresh_token=PuFZ2Hyu6R6eIAxVG9Y4j4kFRYsCapISTR0n3AUM&
scope=data
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "access_token": "vByKXlrmJzA0tD9t27B9Gf9szoA55JYBuMkvbs8f",
  "token_type": "Bearer",
  "expires_in": 3600,
  "refresh_token": "9e97DujgPxnplnr4OkYn8Qsr9QdhSQXwED96BRzs"
}
```

2.3.3 Users

GET /api/v1/users

Get a *list* of all users ordered by created timestamp.

Query Parameters

- **size** – The number of results to return per call (Default: 10. Maximum: 50).

- **cursor** – The cursor string signifying where to start the results.

Status Codes

- **200 OK** – Successfully queried the users.

Response Data

- **users** – The list of users.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **users** is an object of user information.

User

- **key** – The user key.
- **usernames** – The user's minecraft usernames. Empty list if the user has not claimed a minecraft username.
- **email** – The user's email.
- **nickname** – The user's nickname.
- **active** – A boolean indicating whether the user is active.
- **admin** – A boolean indicating whether the user is an admin.
- **last_coal_login** – The timestamp of the user's last COAL login.
- **created** – The user's creation timestamp.
- **updated** – The user's updated timestamp.

Example request:

```
GET /api/v1/users HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "users": [
    {
      "usernames": ["gumptionthomas"],
      "updated": "2013-04-14 18:37:35 CDT-0500",
      "created": "2013-03-04 15:05:52 CST-0600",
      "admin": true,
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchivbgw",
      "active": true,
      "last_coal_login": "2013-04-13 14:03:33 CDT-0500",
      "nickname": "thomas",
      "email": "t@gmail.com"
    },
  ],
}
```

```

    "usernames": "[]",
    "updated": "2013-03-14 17:23:09 CDT-0500",
    "created": "2013-03-04 17:43:37 CST-0600",
    "admin": false,
    "key": "ahRzfmdbXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchiZdQw",
    "active": true,
    "last_coal_login": null,
    "nickname": "jennifer",
    "email": "j@gmail.com"
  },
  {
    "usernames": ["quazifene"],
    "updated": "2013-04-14 18:56:59 CDT-0500",
    "created": "2013-03-04 17:53:12 CST-0600",
    "admin": true,
    "key": "ahRzfmdbXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchiBfQw",
    "active": true,
    "last_coal_login": "2013-04-12 14:04:39 CDT-0500",
    "nickname": "mark",
    "email": "m@gmail.com"
  }
]
}

```

GET `/api/v1/users/` (*key*)

Get the information for the user (*key*).

Parameters

- **key** – The requested user’s key. (*required*) To reference the authenticated user, use `self`.

Status Codes

- **200 OK** – Successfully read the user.

Response Data See *User response data*

Example request:

```
GET /api/v1/users/ahRzfmdbXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchivbgw HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```

{
  "username": ["gumptionthomas"],
  "updated": "2013-04-14 18:37:35 CDT-0500",
  "created": "2013-03-04 15:05:52 CST-0600",
  "admin": true,
  "key": "ahRzfmdbXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchivbgw",
  "active": true,
  "last_coal_login": "2013-04-13 14:03:33 CDT-0500",
  "nickname": "thomas",
  "email": "t@gmail.com"
}

```

Example request:

```
GET /api/v1/users/self HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "username": ["gumptionthomas"],
  "updated": "2013-04-14 18:37:35 CDT-0500",
  "created": "2013-03-04 15:05:52 CST-0600",
  "admin": true,
  "key": "ahRzfmdbXB0aW9uLWlpbmVjcmFmdHILCxEVXNlchivbgw",
  "active": true,
  "last_coal_login": "2013-04-13 14:03:33 CDT-0500",
  "nickname": "thomas",
  "email": "t@gmail.com"
}
```

2.3.4 Servers

GET /api/v1/servers

Get a *list* of all servers ordered by created timestamp.

Query Parameters

- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.

Status Codes

- **200 OK** – Successfully queried the servers.

Response Data

- **servers** – The list of servers.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **servers** is an object of server status information.

Server Status

- **key** – The server key.
- **name** – The server name.
- **gce** – A boolean indicating whether the server is hosted by MC-COAL on a Google Compute Engine server.
- **running_version** – The actual version of minecraft that was last running.

- **address** – The IP address (including the port, if necessary) of the server. If `gce` is `false` and the address isn't passed to the agent at startup, this field will be `null`.
- **status** – A string indicating the status of the minecraft server. Possible values are `RUNNING`, `STOPPED`, `QUEUED_START`, `QUEUED_STOP`, or `UNKNOWN`.
- **last_ping** – The timestamp of the last agent ping.
- **server_day** – An integer indicating the number of game days since the start of the level.
- **server_time** – An integer indicating the game time of day. 0 is sunrise, 6000 is mid day, 12000 is sunset, 18000 is mid night, 24000 is the next day's 0.
- **is_raining** – A boolean indicating whether it is raining. If this value is `null` the status is unknown.
- **is_thundering** – A boolean indicating whether it is thundering. If this value is `null` the status is unknown.
- **created** – The server's creation timestamp.
- **updated** – The server's updated timestamp.

Example request:

```
GET /api/v1/servers HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "servers": [
    {
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "name": "My World",
      "gce": false,
      "running_version": "1.5.1",
      "address": null,
      "last_ping": "2013-04-14 19:55:22 CDT-0500",
      "updated": "2013-04-14 19:55:22 CDT-0500",
      "status": "RUNNING",
      "created": "2013-03-04 15:05:53 CST-0600"
      "server_day": 15744,
      "server_time": 19767,
      "is_raining": true,
      "is_thundering": true
    },
    {
      "key": "IZCxIGQ2xpZW50IG1tYy1jb2FsLWFnZW50DA",
      "name": "My PVP World",
      "gce": true,
      "running_version": "1.5.1",
      "address": "1.2.3.4:56789",
      "last_ping": "2013-04-14 19:55:43 CDT-0500",
```

```

    "updated": "2013-04-14 19:55:43 CDT-0500",
    "status": "STOPPED",
    "created": "2013-03-04 15:07:00 CST-0600",
    "server_day": 15223,
    "server_time": 14141,
    "is_raining": false,
    "is_thundering": false
  }
]
}

```

POST /api/v1/servers

Create a new server. Only an authenticated administrator can call this service.

Form Parameters

- **name** – The server name. (*required*)
- **gce** – A boolean indicating whether the server should be hosted by MC-COAL on Google Compute Engine. (*required*)

Status Codes

- 201 Created – Successfully created the server.

Response Data See *Server status response*

- 403 Forbidden – The authenticated user is not an administrator.

Example request:

```
POST /api/v1/servers HTTP/1.1
```

```
name=Brave+New+World&gce=true&version=1.7.4&memory=256M
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
```

```

{
  "key": "bbhd871bXB0aW9uLW1pbmVj26GhY",
  "name": "Brave New World",
  "gce": true,
  "running_version": null,
  "address": null,
  "last_ping": null,
  "updated": "2014-01-14 17:33:00 CST-0600",
  "status": "UNKNOWN",
  "created": "2014-01-13 12:00:00 CST-0600",
  "server_day": null,
  "server_time": null,
  "is_raining": null,
  "is_thundering": null
}

```

GET /api/v1/servers/ (key)

Get status information for the server (*key*).

Parameters

- **key** – The requested server’s key. (*required*)

Status Codes

- **200 OK** – Successfully read the server.

Response Data See *Server status response*

Example request:

```
GET /api/v1/servers/bbhd871bXB0aW9uLW1pbmVj26GhY HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "key": "bbhd871bXB0aW9uLW1pbmVj26GhY",
  "name": "Brave New World",
  "gce": true,
  "running_version": null,
  "address": null,
  "last_ping": null,
  "updated": "2014-01-14 17:33:00 CST-0600",
  "status": "UNKNOWN",
  "created": "2014-01-13 12:00:00 CST-0600",
  "server_day": null,
  "server_time": null,
  "is_raining": null,
  "is_thundering": null
}
```

POST /api/v1/servers/ (*key*)

Update the settings for the server (*key*). Only an authenticated administrator can call this service.

Parameters

- **key** – The requested server’s key. (*required*)

Form Parameters

- **name** – The server name. (*required*)

Status Codes

- **200 OK** – Successfully updated the settings.

Response Data See *Server status response*

Example request:

```
POST /api/v1/servers/bbhd871bXB0aW9uLW1pbmVj26GhY HTTP/1.1
```

```
name=Not+So+Brave+New+World
```

Example response:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
```

```
{
  "key": "bbhd871bXB0aW9uLW1pbmVj26GhY",
  "name": "Not So Brave New World",
  "gce": true,
  "running_version": null,
  "address": null,
  "last_ping": null,
  "updated": "2014-01-16 15:00:00 CST-0600",
  "status": "UNKNOWN",
  "created": "2014-01-13 12:00:00 CST-0600",
  "server_day": null,
  "server_time": null,
  "is_raining": null,
  "is_thundering": null
}
```

GET /api/v1/servers/ (key) /properties

Get the minecraft server properties for the GCE-hosted server (*key*). If the requested server is not a GCE sever (i.e. *gce* is false), a 404 Not Found will be returned.

Parameters

- **key** – The requested server’s key. (*required*)

Status Codes

- 200 OK – Successfully read the server properties. The response will be an object with the current minecraft properties for the server.

Server Properties

- **key** – The server key.
- **server_port** – The minecraft server port to use (null indicates first available).
- **version** – The minecraft server version.
- **memory** – The amount of memory dedicated to the server. Possible values are 256M, 512M, 1G, 2G, 3G, or 4G.
- **operator** – The minecraft username of the initial operator of the server.
- **idle_timeout** – Number of idle minutes before the server is automatically paused (zero means never)
- **motd** – The message of the day.
- **white_list** – A boolean indicating whether the server whitelist is enabled.
- **gamemode** – An integer indicating the game mode. Possible values are 0 (Survival), 1 (Creative), and 2 (Adventure).
- **force_gamemode** – A boolean indicating whether players are forced to join in the default game mode.
- **level_type** – The type of map for the server. Possible values are DEFAULT, FLAT, LARGEBIOMES, and AMPLIFIED.

- **level_seed** – The seed for the server world.
- **generator_settings** – The settings used to customize Superflat world generation.
- **difficulty** – An integer indicating the server difficulty. Possible values are 0 (Peaceful), 1 (Easy), 2 (Normal), and 3 (Hard).
- **pvp** – A boolean indicating whether the server is PvP.
- **hardcore** – A boolean indicating whether the server is in hardcore mode.
- **allow_flight** – A boolean indicating whether users can use flight while in Survival mode.
- **allow_nether** – A boolean indicating whether players can travel to the Nether.
- **max_build_height** – The maximum height in which building is allowed (Min: 0, Max: 1024).
- **generate_structures** – A boolean indicating whether to generate structures.
- **spawn_npcs** – A boolean indicating whether to spawn villagers.
- **spawn_animals** – A boolean indicating whether to spawn animals.
- **spawn_monsters** – A boolean indicating whether to spawn monsters.
- **view_distance** – An integer indicating the number of chunks of world data the server sends the client (Min: 3, Max: 15).
- **player_idle_timeout** – An integer indicating the number of minutes before an idle player is kicked (zero means never) (Min: 0, Max: 60).
- **max_players** – An integer indicating the maximum number of players that can play on the server at the same time.
- **spawn_protection** – An integer radius of the spawn protection area (Min: 0, Max: 24).
- **enable_command_block** – A boolean indicating whether to enable command blocks.
- **snooper_enabled** – A boolean indicating whether to send snoop data regularly to snoop.minecraft.net.
- **resource_pack** – The URL (if any) to prompt clients to download a resource pack from.
- **op_permission_level** – An integer indicating the operator permission level. Possible values are 0 (Can bypass spawn protection), 1 (Can use `/clear`, `/difficulty`, `/effect`, `/gamemode`, `/gamerule`, `/give`, and

/tp, and can edit command blocks), and 2 (Can use /ban, /deop, /kick, and /op).

Example request:

```
GET /api/v1/servers/bbhd871bXB0aW9uLW1pbmVj26GhY/properties HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "key": "bbhd871bXB0aW9uLW1pbmVj26GhY",
  "server_port": null,
  "version": "1.7.4",
  "memory": "256M",
  "operator": "gumptionthomas",
  "idle_timeout": 5,
  "motd": "It's a brave new world out there",
  "white_list": true,
  "gamemode": 0,
  "force_gamemode": false,
  "level_type": "DEFAULT",
  "level_seed": "",
  "generator_settings": "",
  "difficulty": "1",
  "pvp": false,
  "hardcore": false,
  "allow_flight": false,
  "allow_nether": true,
  "max_build_height": 256,
  "generate_structures": true,
  "spawn_npcs": true,
  "spawn_animals": true,
  "spawn_monsters": true,
  "view_distance": 10,
  "player_idle_timeout": 0,
  "max_players": 20,
  "spawn_protection": 16,
  "enable_command_block": false,
  "snooper_enabled": true,
  "resource_pack": "",
  "op_permission_level": 3
}
```

POST /api/v1/servers/(key)/properties

Update the minecraft server properties for the GCE-hosted server (*key*). If the requested server is not a GCE sever (i.e. `gce` is `false`), a 404 Not Found will be returned. Only an authenticated administrator can call this service.

Parameters

- **key** – The requested server’s key. (*required*)

Form Parameters

- **server_port** – The minecraft server port to use (‘ ’ [empty string] indicates first available).

- **version** – The minecraft version to use for the server.
- **memory** – The amount of memory to dedicate to the server. Possible values are 256M, 512M, 1G, 2G, 3G, or 4G.
- **operator** – The minecraft username of the initial operator for the server.
- **idle_timeout** – The number of idle minutes before the server is automatically paused (zero means never)
- **motd** – The message of the day.
- **white_list** – A boolean indicating whether the server whitelist should be enabled.
- **gamemode** – An integer indicating the game mode. Possible values are 0 (Survival), 1 (Creative), and 2 (Adventure).
- **force_gamemode** – A boolean indicating whether players should be forced to join in the default game mode.
- **level_type** – The type of map for the server. Possible values are `DEFAULT`, `FLAT`, `LARGEBIOMES`, and `AMPLIFIED`.
- **level_seed** – The seed for the server world.
- **generator_settings** – The settings used to customize Superflat world generation.
- **difficulty** – An integer indicating the server difficulty. Possible values are 0 (Peaceful), 1 (Easy), 2 (Normal), and 3 (Hard).
- **pvp** – A boolean indicating whether the server should be PvP.
- **hardcore** – A boolean indicating whether the server should be in hardcore mode.
- **allow_flight** – A boolean indicating whether users can use flight while in Survival mode.
- **allow_nether** – A boolean indicating whether players can travel to the Nether.
- **max_build_height** – The maximum height in which building is allowed (Min: 0, Max: 1024).
- **generate_structures** – A boolean indicating whether to generate structures.
- **spawn_npcs** – A boolean indicating whether to spawn villagers.
- **spawn_animals** – A boolean indicating whether to spawn animals.
- **spawn_monsters** – A boolean indicating whether to spawn monsters.
- **view_distance** – An integer indicating the number of chunks of world data the server sends the client (Min: 3, Max: 15).
- **player_idle_timeout** – An integer indicating the number of minutes before an idle player is kicked (zero means never) (Min: 0, Max: 60).
- **max_players** – An integer indicating the maximum number of players that can play on the server at the same time.
- **spawn_protection** – An integer radius of the spawn protection area (Min: 0, Max: 24).

- **enable_command_block** – A boolean indicating whether to enable command blocks.
- **snooper_enabled** – A boolean indicating whether to send snoop data regularly to snoop.minecraft.net.
- **resource_pack** – The URL (if any) to prompt clients to download a resource pack from.
- **op_permission_level** – An integer indicating the operator permission level. Possible values are 0 (Can bypass spawn protection), 1 (Can use `/clear`, `/difficulty`, `/effect`, `/gamemode`, `/gamerule`, `/give`, and `/tp`, and can edit command blocks), and 2 (Can use `/ban`, `/deop`, `/kick`, and `/op`).

Status Codes

- **200 OK** – Successfully updated the server properties. The response will be an object with the new minecraft properties for the server.

Response Data See *Server properties response*

Example request:

```
POST /api/v1/servers/bbhd871bXB0aW9uLW1pbmVj26GhY/properties HTTP/1.1
```

```
memory=1G&gamemode=1&level_type=FLAT&spawn_monsters=false&motd=Maybe+not+that+brave
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "key": "bbhd871bXB0aW9uLW1pbmVj26GhY",
  "server_port": null,
  "version": "1.7.4",
  "memory": "1G",
  "operator": "gumptionthomas",
  "idle_timeout": 5,
  "motd": "Maybe not that brave",
  "white_list": true,
  "gamemode": 1,
  "force_gamemode": false,
  "level_type": "FLAT",
  "level_seed": "",
  "generator_settings": "",
  "difficulty": "1",
  "pvp": false,
  "hardcore": false,
  "allow_flight": false,
  "allow_nether": true,
  "max_build_height": 256,
  "generate_structures": true,
  "spawn_npcs": true,
  "spawn_animals": true,
  "spawn_monsters": false,
  "view_distance": 10,
```

```
"player_idle_timeout": 0,  
"max_players": 20,  
"spawn_protection": 16,  
"enable_command_block": false,  
"snooper_enabled": true,  
"resource_pack": "",  
"op_permission_level": 3  
}
```

POST /api/v1/servers/ (*key*) /queue/play

Place the GCE-hosted server (*key*) in the play queue. If the requested server is not a GCE sever (i.e. `gce` is `false`), a 404 Not Found will be returned. Any authenticated user can call this service.

Parameters

- **key** – The requested server’s key. (*required*)

Status Codes

- 200 OK – The server was already playing or queued to play. No action taken.
- 202 Accepted – Successfully queued the server to play.

Example request:

```
POST /api/v1/servers/bbhd871bXB0aW9uLW1pbmVj26GhY/queue/play HTTP/1.1
```

Example response:

```
HTTP/1.1 202 Accepted
```

Note: To determine when the server is ready to play (i.e. for minecraft clients to connect), call `GET /api/v1/servers/ (key)`. The response property `status` will be `RUNNING` and `address` will contain the server’s IP address.

POST /api/v1/servers/ (*key*) /queue/pause

Place the GCE-hosted server (*key*) in the pause queue. If the requested server is not a GCE sever (i.e. `gce` is `false`), a 404 Not Found will be returned. Only an authenticated administrator can call this service.

Parameters

- **key** – The requested server’s key. (*required*)

Status Codes

- 200 OK – The server was already paused or queued to pause. No action taken.
- 202 Accepted – Successfully queued the server to pause.

Example request:

```
POST /api/v1/servers/bbhd871bXB0aW9uLW1pbmVj26GhY/queue/pause HTTP/1.1
```

Example response:

```
HTTP/1.1 202 Accepted
```

Note: To determine when the server is paused (i.e. when minecraft clients can no longer connect), call `GET /api/v1/servers/(key)`. The response property `status` will be `STOPPED` or `UNKNOWN`.

2.3.5 Players

GET `/api/v1/servers/(server_key)/players`

Get a *list* of all minecraft players on the server (*server_key*). Results are ordered by username.

Parameters

- **server_key** – The target server’s key. (*required*)

Query Parameters

- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.

Status Codes

- **200 OK** – Successfully queried the players.

Response Data

- **players** – The list of players.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **players** is an object of player information.

Player

- **key** – The player key.
- **server_key** – The player’s server key.
- **username** – The player’s minecraft username.
- **user_key** – The player’s user key. `null` if the player is not mapped to a user.
- **last_login** – The timestamp of the player’s last minecraft login. `null` if the player has not logged in.
- **last_session_duration** – The player’s last session duration in seconds. `null` if the player has not logged in.
- **is_playing** – A boolean indicating whether the player is currently logged into the minecraft server.

Example request:

```
GET /api/v1/servers/ahRzfm1bXB0aW9uLW1pbmVjcmFmdH/players HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "players": [
    {
      "username": "gumptionthomas",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchiivbgw",
      "last_login": "2013-04-13 20:50:34 CDT-0500",
      "last_session_duration": 8126,
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIgl1bnG9iYWxfc2VydmVyDAsSB1BsYX1lc",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "is_playing": false
    },
    {
      "username": "quazifene",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchiBfQw",
      "last_login": "2013-04-13 21:21:30 CDT-0500",
      "last_session_duration": 6821,
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIuCXIGU2VydmVyIgl1bnG9iYWxfc2VydmVyDAsSB1BsYX1lc",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "is_playing": false
    }
  ]
}
```

GET `/api/v1/servers/{server_key}/players/{key_username}` Get the information for the player (*key_username*) on the server (*server_key*).

Parameters

- **server_key** – The target server’s key. (*required*)
- **key_username** – The requested player’s key or minecraft username. (*required*)

Status Codes

- **200 OK** – Successfully read the player.

Response Data See *Player response data*

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/players/gumptionthomas HTTP/1.1
```

OR

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/players/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIgl1bnG9iYWxfc2VydmVyDAsSB1BsYX1lc
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "username": "gumptionthomas",
  "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchiivbgw",
```

```

    "last_login": "2013-04-13 20:50:34 CDT-0500",
    "last_session_duration": 8126,
    "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1BsYX11ciIOZ",
    "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
    "is_playing": false
  }

```

2.3.6 Sessions

GET /api/v1/servers/ (*server_key*) /sessions

Get a *list* of all minecraft play sessions on the server (*server_key*) ordered by descending login timestamp.

Parameters

- **server_key** – The target server’s key. (*required*)

Query Parameters

- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.
- **since** – Return sessions with a login timestamp since the given date-time (inclusive). This parameter should be of the form YYYY-MM-DD HH:MM:SS and is assumed to be UTC.
- **before** – Return sessions with a login timestamp before this date-time (exclusive). This parameter should be of the form YYYY-MM-DD HH:MM:SS and is assumed to be UTC.

Status Codes

- 200 OK – Successfully queried the play sessions.

Response Data

- **sessions** – The list of play sessions.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **sessions** is an object of play session information.

Session

- **key** – The play session key.
- **server_key** – The play session’s server key.
- **username** – The minecraft username associated with the play session.
- **player_key** – The player key. `null` if the username is not mapped to a player.
- **user_key** – The user key. `null` if the username is not mapped to a player or the player is not mapped to a user.

- **login_timestamp** – The timestamp of the play session start. It will be reported in the agent’s timezone.
- **logout_timestamp** – The timestamp of the play session end. It will be reported in the agent’s timezone.
- **duration** – The length of the play session in seconds.
- **login_logline_key** – The login log line key. May be null.
- **logout_logline_key** – The logout log line key. May be null.
- **created** – The creation timestamp.
- **updated** – The updated timestamp.

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/sessions HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "sessions": [
    {
      "username": "gumptionthomas",
      "updated": "2013-04-13 23:06:01 CDT-0500",
      "logout_timestamp": "2013-04-13 23:06:00 CDT-0500",
      "login_timestamp": "2013-04-13 20:50:34 CDT-0500",
      "created": "2013-04-13 20:50:35 CDT-0500",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXN1chivbgw",
      "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1",
      "login_logline_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCzIGU2VydmVyIglnbG9iYWxfc2VydmV",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIsCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSC1BsYX1TZ",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "duration": 8126,
      "logout_logline_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCzIGU2VydmVyIglnbG9iYWxfc2VydmV",
    },
    {
      "username": "vesicular",
      "updated": "2013-04-13 20:20:21 CDT-0500",
      "logout_timestamp": "2013-04-13 20:20:19 CDT-0500",
      "login_timestamp": "2013-04-13 19:48:28 CDT-0500",
      "created": "2013-04-13 19:48:29 CDT-0500",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXN1chjkLww",
      "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIuCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1",
      "login_logline_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCzIGU2VydmVyIglnbG9iYWxfc2VydmV",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIsCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSC1BsYX1TZ",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "duration": 1911,
      "logout_logline_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCzIGU2VydmVyIglnbG9iYWxfc2VydmV",
    }
  ]
}
```

GET `/api/v1/servers/ (server_key) /sessions/`

key Get the information for the play session (*key*) on the server (*server_key*).

Parameters

- **server_key** – The target server’s key. (*required*)
- **key** – The requested play session’s key. (*required*)

Status Codes

- **200 OK** – Successfully read the play session.

Response Data See *Play session response data*

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/sessions/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH
```

Example response:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
  "username": "gumptionthomas",
  "updated": "2013-04-13 23:06:01 CDT-0500",
  "logout_timestamp": "2013-04-13 23:06:00 CDT-0500",
  "login_timestamp": "2013-04-13 20:50:34 CDT-0500",
  "created": "2013-04-13 20:50:35 CDT-0500",
  "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchivbgw",
  "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1BsYX",
  "login_logline_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAs",
  "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIsCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSC1BsYX1TZnZa",
  "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
  "duration": 8126,
  "logout_logline_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDA"
}
```

GET `/api/v1/servers/ (server_key) /players/`

key_username/sessions Get a *list* of a player’s (*key_username*) minecraft play sessions on the server (*server_key*) ordered by descending login timestamp.

Parameters

- **server_key** – The target server’s key. (*required*)
- **key_username** – The requested player’s key or minecraft username. (*required*)

Query Parameters

- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.
- **since** – Return sessions with a login timestamp since the given date-time (inclusive). This parameter should be of the form YYYY-MM-DD HH:MM:SS and is assumed to be UTC.

- **before** – Return sessions with a login timestamp before this date-time (exclusive). This parameter should be of the form YYYY-MM-DD HH:MM:SS and is assumed to be UTC.

Status Codes

- **200 OK** – Successfully queried the play sessions.

Response Data

- **sessions** – The list of the player's play sessions.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **sessions** is an object of play session information. See *Play session response data*

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/players/gumptionthomas/session HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "sessions": [
    {
      "username": "gumptionthomas",
      "updated": "2013-04-15 22:31:43 CDT-0500",
      "logout_timestamp": "2013-04-15 22:31:42 CDT-0500",
      "login_timestamp": "2013-04-15 22:31:18 CDT-0500",
      "created": "2013-04-15 22:31:19 CDT-0500",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXN1chivbgw",
      "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1",
      "login_logline_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCzIGU2VydmVyIglnbG9iYWxfc2VydmVy",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIsCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSC1BsYX1lTz",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "duration": 24,
      "logout_logline_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCzIGU2VydmVyIglnbG9iYWxfc2VydmVy"
    },
    {
      "username": "gumptionthomas",
      "updated": "2013-04-13 23:06:01 CDT-0500",
      "logout_timestamp": "2013-04-13 23:06:00 CDT-0500",
      "login_timestamp": "2013-04-13 20:50:34 CDT-0500",
      "created": "2013-04-13 20:50:35 CDT-0500",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXN1chivbgw",
      "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1",
      "login_logline_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCzIGU2VydmVyIglnbG9iYWxfc2VydmVy",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIsCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSC1BsYX1lTz",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "duration": 8126,
      "logout_logline_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCzIGU2VydmVyIglnbG9iYWxfc2VydmVy"
    }
  ]
}
```

```
]
}
```

2.3.7 Chats

GET `/api/v1/servers/ (server_key) /chats`

Get a *list* of all minecraft chats on the server (*server_key*) ordered by descending timestamp.

Parameters

- **server_key** – The target server’s key. (*required*)

Query Parameters

- **q** – A search string to limit the chat results to.
- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.
- **since** – Return chats with a timestamp since the given datetime (inclusive). This parameter should be of the form `YYYY-MM-DD HH:MM:SS` and is assumed to be UTC.
- **before** – Return chats with a timestamp before this datetime (exclusive). This parameter should be of the form `YYYY-MM-DD HH:MM:SS` and is assumed to be UTC.

Status Codes

- **200 OK** – Successfully queried the chats.

Response Data

- **chats** – The list of chats.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **chats** is an object of chat information.

Chat

- **key** – The chat log line key.
- **server_key** – The chat’s server key.
- **chat** – The chat text. May be `null`.
- **username** – The minecraft username associated with the chat. May be `null`.
- **player_key** – The player key. `null` if the username is not mapped to a player.
- **user_key** – The user key. `null` if the username is not mapped to a player or the player is not mapped to a user.
- **timestamp** – The timestamp of the chat. It will be reported in the agent’s timezone.

- **line** – The complete raw chat log line text.
- **created** – The creation timestamp.
- **updated** – The updated timestamp.

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/chats HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "chats": [
    {
      "username": "gumptionthomas",
      "updated": "2013-04-19 10:33:56 CDT-0500",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpbmVjcmFmdH",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "timestamp": "2013-04-19 10:33:55 CDT-0500",
      "created": "2013-04-19 10:33:56 CDT-0500",
      "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1",
      "chat": "what's up?",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxEVXNlchivbgw",
      "line": "2013-04-19 10:33:55 [INFO] <gumptionthomas> what's up?"
    },
    {
      "username": "gumptionthomas",
      "updated": "2013-04-19 10:32:56 CDT-0500",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpbmVjcmFmdH",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "timestamp": "2013-04-19 10:32:55 CDT-0500",
      "created": "2013-04-19 10:32:56 CDT-0500",
      "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1",
      "chat": "hey guys",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxEVXNlchivbgw",
      "line": "2013-04-19 10:32:55 [INFO] [Server] <gumptionthomas> hey guys"
    }
  ]
}
```

POST /api/v1/servers/(server_key)/chats

Queue a new chat on the server (*server_key*) from the authenticated user. In game, the chat will appear as a “Server” chat with the user’s default minecraft username in angle brackets (much like a normal chat):

```
[Server] <gumptionthomas> Hello world...
```

If the API user does not have an associated minecraft username, the user’s nickname or email will be used instead:

```
[Server] <t@gmail.com> Hello world...
```

Parameters

- **server_key** – The target server’s key. (*required*)

Form Parameters

- **chat** – The chat text.

Status Codes

- **202 Accepted** – Successfully queued the chat. It will be sent to the agent on the next ping.

Example request:

```
POST /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/chats HTTP/1.1
```

```
chat=Hello+world...
```

Example response:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
```

GET /api/v1/servers/ (*server_key*) /chats/
key Get the information for the chat (*key*).

Parameters

- **server_key** – The target server’s key. (*required*)
- **key** – The requested chat’s log line key. (*required*)

Status Codes

- **200 OK** – Successfully read the chat.

Response Data See *Chat response data*

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/chats/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoC...
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "username": "gumptionthomas",
  "updated": "2013-04-19 10:32:56 CDT-0500",
  "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCxiGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpbnUY6",
  "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
  "timestamp": "2013-04-19 10:32:55 CDT-0500",
  "created": "2013-04-19 10:32:56 CDT-0500",
  "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCxiGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1BsYX",
  "chat": "hey guys",
  "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxiEVXNlchivbgw",
  "line": "2013-04-19 10:32:55 [INFO] [Server] <gumptionthomas> hey guys"
}
```

GET `/api/v1/servers/` (*server_key*) `/players/`
key_username/chats Get a *list* of a player's (*key_username*) minecraft chats on the server
(*server_key*) ordered by descending timestamp.

Parameters

- **server_key** – The target server's key. (*required*)
- **key_username** – The requested player's key or minecraft username. (*required*)

Query Parameters

- **q** – A search string to limit the chat results to.
- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.
- **since** – Return log lines with a timestamp since the given datetime (inclusive). This parameter should be of the form YYYY-MM-DD HH:MM:SS and is assumed to be UTC.
- **before** – Return log lines with a timestamp before this datetime (exclusive). This parameter should be of the form YYYY-MM-DD HH:MM:SS and is assumed to be UTC.

Status Codes

- **200 OK** – Successfully queried the chats.

Response Data

- **chats** – The list of the player's chats.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **chats** is an object of chat information. See *Chat response data*

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/players/gumptionthomas/chats HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "chats": [
    {
      "username": "gumptionthomas",
      "updated": "2013-04-19 10:33:56 CDT-0500",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIGU2VydmVyIgl1bnG9iYWxfc2VydmVyDAsSB0xvZ0xpbmVjcmFmdH",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "timestamp": "2013-04-19 10:33:55 CDT-0500",
      "created": "2013-04-19 10:33:56 CDT-0500",

```

```

    "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSBl",
    "chat": "what's up?",
    "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxEVXNlchivbgw",
    "line": "2013-04-19 10:33:55 [INFO] <gumptionthomas> what's up?"
  },
  {
    "username": "gumptionthomas",
    "updated": "2013-04-19 10:32:56 CDT-0500",
    "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpb",
    "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
    "timestamp": "2013-04-19 10:32:55 CDT-0500",
    "created": "2013-04-19 10:32:56 CDT-0500",
    "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSBl",
    "chat": "hey guys",
    "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxEVXNlchivbgw",
    "line": "2013-04-19 10:32:55 [INFO] [Server] <gumptionthomas> hey guys"
  }
]
}

```

POST /api/v1/servers/ (*server_key*) **/players/** (*key_username*) **/chats** Queue a new chat on the server (*server_key*) for the player (*key_username*) from the authenticated user. In game, the chat will appear as a “Server” chat with the username in angle brackets (much like a normal chat):

```
[Server] <gumptionthomas> Hello world...
```

Parameters

- **server_key** – The target server’s key. (*required*)
- **key_username** – The requested player’s key or minecraft username. (*required*)

Form Parameters

- **chat** – The chat text.

Status Codes

- **202 Accepted** – Successfully queued the chat. It will be sent to the agent on the next ping.
- **403 Forbidden** – The authenticated user has not claimed the requested player’s username.

Example request:

```
POST /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/players/gumptionthomas/chats HTTP/1.1
```

```
chat=Hello+world...
```

Example response:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
```

2.3.8 Deaths

GET `/api/v1/servers/ (server_key) /deaths`

Get a *list* of all minecraft deaths on the server (*server_key*) ordered by descending timestamp.

Parameters

- **server_key** – The target server’s key. (*required*)

Query Parameters

- **q** – A search string to limit the death results to.
- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.
- **since** – Return deaths with a timestamp since the given datetime (inclusive). This parameter should be of the form `YYYY-MM-DD HH:MM:SS` and is assumed to be UTC.
- **before** – Return deaths with a timestamp before this datetime (exclusive). This parameter should be of the form `YYYY-MM-DD HH:MM:SS` and is assumed to be UTC.

Status Codes

- **200 OK** – Successfully queried the deaths.

Response Data

- **deaths** – The list of deaths.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **deaths** is an object of death information.

Death

- **key** – The death log line key.
- **server_key** – The death log line’s server key.
- **message** – The death message. May be `null`.
- **username** – The minecraft username associated with the death.
- **player_key** – The player key.
- **user_key** – The user key. `null` if the player is not mapped to a user.
- **timestamp** – The timestamp of the death. It will be reported in the agent’s timezone.
- **line** – The complete raw death log line text.
- **created** – The creation timestamp.
- **updated** – The updated timestamp.

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/deaths HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
  "deaths": [
    {
      "username": "gumptionthomas",
      "updated": "2013-04-19 10:33:56 CDT-0500",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpbmVjcmFmdH",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "timestamp": "2013-04-19 10:33:55 CDT-0500",
      "created": "2013-04-19 10:33:56 CDT-0500",
      "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1",
      "message": "was squashed by a falling anvil",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCzIEVXNlchivbgw",
      "line": "2013-04-19 10:33:55 [INFO] gumptionthomas was squashed by a falling anvil"
    },
    {
      "username": "gumptionthomas",
      "updated": "2013-04-19 10:32:56 CDT-0500",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpbmVjcmFmdH",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "timestamp": "2013-04-19 10:32:55 CDT-0500",
      "created": "2013-04-19 10:32:56 CDT-0500",
      "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1",
      "message": "was shot by arrow",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCzIEVXNlchivbgw",
      "line": "2013-04-19 10:32:55 [INFO] gumptionthomas was shot by arrow"
    }
  ]
}
```

```
GET /api/v1/servers/(server_key)/deaths/
```

key Get the information for the death (*key*) on the server (*server_key*).

Parameters

- **server_key** – The target server’s key. (*required*)
- **key** – The requested death’s log line key. (*required*)

Status Codes

- **200 OK** – Successfully read the death.

Response Data See *Death response data*

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/deaths/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCzIEVXNlchivbgw
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "username": "gumptionthomas",
  "updated": "2013-04-19 10:32:56 CDT-0500",
  "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpbmUY6",
  "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
  "timestamp": "2013-04-19 10:32:55 CDT-0500",
  "created": "2013-04-19 10:32:56 CDT-0500",
  "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1BsYX",
  "message": "was shot by arrow",
  "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCzIEVXNlchivbgw",
  "line": "2013-04-19 10:32:55 [INFO] gumptionthomas was shot by arrow"
}
```

GET `/api/v1/servers/ (server_key) /players/ key_username/deaths` Get a *list* of a player's (*key_username*) minecraft deaths on the server (*server_key*) ordered by descending timestamp.

Parameters

- **server_key** – The target server's key. (*required*)
- **key_username** – The requested player's key or minecraft username. (*required*)

Query Parameters

- **q** – A search string to limit the death results to.
- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.
- **since** – Return log lines with a timestamp since the given datetime (inclusive). This parameter should be of the form `YYYY-MM-DD HH:MM:SS` and is assumed to be UTC.
- **before** – Return log lines with a timestamp before this datetime (exclusive). This parameter should be of the form `YYYY-MM-DD HH:MM:SS` and is assumed to be UTC.

Status Codes

- **200 OK** – Successfully queried the deaths.

Response Data

- **deaths** – The list of the player's deaths.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **deaths** is an object of death information. See *Death response data*

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/players/gumptionthomas/deaths HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
  "deaths": [
    {
      "username": "gumptionthomas",
      "updated": "2013-04-19 10:33:56 CDT-0500",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHioCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpbl",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "timestamp": "2013-04-19 10:33:55 CDT-0500",
      "created": "2013-04-19 10:33:56 CDT-0500",
      "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1",
      "message": "was squashed by a falling anvil",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCzIEVXNlchivbgw",
      "line": "2013-04-19 10:33:55 [INFO] gumptionthomas was squashed by a falling anvil"
    },
    {
      "username": "gumptionthomas",
      "updated": "2013-04-19 10:32:56 CDT-0500",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHioCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpbl",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "timestamp": "2013-04-19 10:32:55 CDT-0500",
      "created": "2013-04-19 10:32:56 CDT-0500",
      "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1",
      "message": "was shot by arrow",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCzIEVXNlchivbgw",
      "line": "2013-04-19 10:32:55 [INFO] gumptionthomas was shot by arrow"
    }
  ]
}
```

2.3.9 Achievements

GET /api/v1/servers/ (*server_key*) /achievements

Get a *list* of all minecraft achievements earned on the server (*server_key*) ordered by descending timestamp.

Parameters

- **server_key** – The target server’s key. (*required*)

Query Parameters

- **q** – A search string to limit the achievement results to.
- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.

- **since** – Return achievements with a timestamp since the given date-time (inclusive). This parameter should be of the form YYYY-MM-DD HH:MM:SS and is assumed to be UTC.
- **before** – Return achievements with a timestamp before this datetime (exclusive). This parameter should be of the form YYYY-MM-DD HH:MM:SS and is assumed to be UTC.

Status Codes

- **200 OK** – Successfully queried the achievements.

Response Data

- **achievements** – The list of achievements.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **achievements** is an object of achievement information.

Achievement

- **key** – The achievement log line key.
- **server_key** – The achievement log line's server key.
- **name** – The achievement name.
- **message** – The achievement message.
- **username** – The minecraft username associated with the achievement.
- **player_key** – The player key.
- **user_key** – The user key. `null` if the player is not mapped to a user.
- **timestamp** – The timestamp of the achievement. It will be reported in the agent's timezone.
- **line** – The complete raw achievement log line text.
- **created** – The creation timestamp.
- **updated** – The updated timestamp.

Example request:

```
GET /api/v1/servers/ahRzfm1bXB0aW9uLW1pbmVjcmFmdH/achievements HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "achievements": [
    {
      "username": "gumptionthomas",
      "updated": "2013-11-11 20:47:04 CDT-0500",
```

```

    "key": "agtkZXZ-bWmtY29hbHInCxIGU2VydmVyGICAgICAgIAJDAsSB0xvZ0xpbnUYgICAgICAhAkM",
    "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
    "timestamp": "2013-11-11 20:47:02 CDT-0500",
    "created": "2013-11-11 20:47:04 CDT-0500",
    "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSBl",
    "name": "Taking Inventory",
    "message": "has just earned the achievement [Taking Inventory]",
    "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxEVXNlchivbgw",
    "line": "2013-11-11 14:47:02 [INFO] gumptionthomas has just earned the achievement [T
  },
  {
    "username": "gumptionthomas",
    "updated": "2013-11-10 17:19:06 CDT-0500",
    "key": "agtkZXZ-bWmtY29hbHInCxIGU2VydmVyGICAgICAgIAJDAsSB0xvZ0xpbnUYgICAgICAtAkM",
    "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
    "timestamp": "2013-11-10 17:19:04 CDT-0500",
    "created": "2013-11-10 17:19:06 CDT-0500",
    "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSBl",
    "name": "Getting an Upgrade",
    "message": "has just earned the achievement [Getting an Upgrade]",
    "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxEVXNlchivbgw",
    "line": "2013-11-10 17:19:04 [INFO] gumptionthomas has just earned the achievement [G
  }
]
}

```

GET /api/v1/servers/ (*server_key*) /achievements/

key Get the information for the achievement (*key*) on the server (*server_key*).

Parameters

- **server_key** – The target server’s key. (*required*)
- **key** – The requested achievement’s log line key. (*required*)

Status Codes

- 200 OK – Successfully read the achievement.

Response Data See *Achievement response data*

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/achievements/bWmtY29hbHInCxIGU2VydmVyGIC
```

Example response:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```

{
  "username": "gumptionthomas",
  "updated": "2013-04-19 10:32:56 CDT-0500",
  "key": "bWmtY29hbHInCxIGU2VydmVyGICAgICAgIAJDAsSB0xvZ0xpbnUYgICAgICAhAkM",
  "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
  "timestamp": "2013-11-11 20:47:02 CDT-0500",
  "created": "2013-11-11 20:47:04 CDT-0500",
  "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSBlBsY",
  "name": "Taking Inventory",
  "message": "has just earned the achievement [Taking Inventory]",

```

```

    "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchivbgw",
    "line": "2013-11-11 14:47:02 [INFO] gumptionthomas has just earned the achievement [Takin
  }

```

GET `/api/v1/servers/` (*server_key*) `/players/` *key_username/achievements* Get a *list* of a player's (*key_username*) minecraft achievements earned on the server (*server_key*) ordered by descending timestamp.

Parameters

- **server_key** – The target server's key. (*required*)
- **key_username** – The requested player's key or minecraft username. (*required*)

Query Parameters

- **q** – A search string to limit the achievement results to.
- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.
- **since** – Return log lines with a timestamp since the given datetime (inclusive). This parameter should be of the form `YYYY-MM-DD HH:MM:SS` and is assumed to be UTC.
- **before** – Return log lines with a timestamp before this datetime (exclusive). This parameter should be of the form `YYYY-MM-DD HH:MM:SS` and is assumed to be UTC.

Status Codes

- **200 OK** – Successfully queried the achievements.

Response Data

- **achievements** – The list of the player's achievements.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **achievements** is an object of achievement information. See *Achievement response data*

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/players/gumptionthomas/achievements HTTP
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```

{
  "achievements": [
    {
      "username": "gumptionthomas",
      "updated": "2013-11-11 20:47:04 CDT-0500",

```

```

    "key": "agtkZXZ-bWMtY29hbHInCxIGU2VydmVyGICAgICAgIAJDAsSB0xvZ0xpbnUYgICAgICAhAkM",
    "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
    "timestamp": "2013-11-11 20:47:02 CDT-0500",
    "created": "2013-11-11 20:47:04 CDT-0500",
    "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1",
    "name": "Taking Inventory",
    "message": "has just earned the achievement [Taking Inventory]",
    "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchivbgw",
    "line": "2013-11-11 14:47:02 [INFO] gumptionthomas has just earned the achievement [T
  },
  {
    "username": "gumptionthomas",
    "updated": "2013-11-10 17:19:06 CDT-0500",
    "key": "agtkZXZ-bWMtY29hbHInCxIGU2VydmVyGICAgICAgIAJDAsSB0xvZ0xpbnUYgICAgICAtAkM",
    "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
    "timestamp": "2013-11-10 17:19:04 CDT-0500",
    "created": "2013-11-10 17:19:06 CDT-0500",
    "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1",
    "name": "Getting an Upgrade",
    "message": "has just earned the achievement [Getting an Upgrade]",
    "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchivbgw",
    "line": "2013-11-10 17:19:04 [INFO] gumptionthomas has just earned the achievement [G
  }
]
}

```

2.3.10 Log Lines

GET /api/v1/servers/ (*server_key*) /loglines

Get a *list* of all minecraft log lines on the server (*server_key*) ordered by descending timestamp.

Parameters

- **server_key** – The target server’s key. (*required*)

Query Parameters

- **tag** – A tag to limit the type of log line results.

Tag Options

- unknown
- timestamp
- connection
- login
- logout
- chat
- death
- server
- performance
- overloaded
- stopping

- starting
- **q** – A search string to limit the results to.
- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.
- **since** – Return log lines with a timestamp since the given datetime (inclusive). This parameter should be of the form `YYYY-MM-DD HH:MM:SS` and is assumed to be UTC.
- **before** – Return log lines with a timestamp before this datetime (exclusive). This parameter should be of the form `YYYY-MM-DD HH:MM:SS` and is assumed to be UTC.

Status Codes

- **200 OK** – Successfully queried the log lines.

Response Data

- **loglines** – The list of log lines.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **loglines** is an object of log line information.

Log Line

- **key** – The log line key.
- **server_key** – The log line's server key.
- **line** – The complete raw log line text.
- **username** – The minecraft username associated with the log line. May be `null`.
- **player_key** – The player key. `null` if the username is not mapped to a player.
- **user_key** – The user key. `null` if the username is not mapped to a player or the player is not mapped to a user.
- **timestamp** – The timestamp of the log line. It will be reported in the agent's timezone.
- **log_level** – The log level of the log line. May be `null`.
- **ip** – The ip address recorded with the log line. May be `null`.
- **port** – The port recorded with the log line. May be `null`.
- **location** – The location of the log line as an object containing `x`, `y`, and `z` keys with float values. May be `null`.
- **chat** – The chat text of the log line. May be `null`.
- **tags** – A list of the log line's tags. May be an empty list.
- **created** – The creation timestamp.

– **updated** – The updated timestamp.

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/loglines HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
  "loglines": [
    {
      "username": "gumptionthomas",
      "updated": "2013-04-19 10:32:56 CDT-0500",
      "log_level": "INFO",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpbmVjcmFmdH",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "timestamp": "2013-04-19 10:32:55 CDT-0500",
      "tags": [
        "timestamp",
        "chat"
      ],
      "ip": null,
      "created": "2013-04-19 10:32:56 CDT-0500",
      "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpbmVjcmFmdH",
      "location": null,
      "chat": "hey guys",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCzIEVXNlchivbgw",
      "line": "2013-04-19 10:32:55 [INFO] [Server] <gumptionthomas> hey guys",
      "port": null
    },
    {
      "username": "gumptionthomas",
      "updated": "2013-04-19 00:26:53 CDT-0500",
      "log_level": "INFO",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpbmVjcmFmdH",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "timestamp": "2013-04-19 00:26:53 CDT-0500",
      "tags": [
        "timestamp",
        "connection",
        "logout"
      ],
      "ip": null,
      "created": "2013-04-19 00:26:53 CDT-0500",
      "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpbmVjcmFmdH",
      "location": null,
      "chat": null,
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCzIEVXNlchivbgw",
      "line": "2013-04-19 00:26:53 [INFO] gumptionthomas lost connection: disconnect.quitting",
      "port": null
    }
  ]
}
```

GET `/api/v1/servers/` (*server_key*) `/loglines/`
key Get the information for the log line (*key*).

Parameters

- **server_key** – The target server’s key. (*required*)
- **key** – The requested log line’s key. (*required*)

Status Codes

- **200 OK** – Successfully read the log line.

Response Data See *Log line response data*

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/loglines/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH
```

Example response:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
  "username": "gumptionthomas",
  "updated": "2013-04-19 10:32:56 CDT-0500",
  "log_level": "INFO",
  "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpbmUY6",
  "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
  "timestamp": "2013-04-19 10:32:55 CDT-0500",
  "tags": [
    "timestamp",
    "chat"
  ],
  "ip": null,
  "created": "2013-04-19 10:32:56 CDT-0500",
  "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1BsYX",
  "location": null,
  "chat": "hey guys",
  "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchivbgw",
  "line": "2013-04-19 10:32:55 [INFO] [Server] <gumptionthomas> hey guys",
  "port": null
}
```

GET `/api/v1/servers/` (*server_key*) `/players/`
key_username/**loglines** Get a *list* of a player’s (*key_username*) minecraft log lines on the server
(*server_key*) ordered by descending timestamp.

Parameters

- **server_key** – The target server’s key. (*required*)
- **key_username** – The requested player’s key or minecraft username. (*required*)

Query Parameters

- **tag** – A tag to limit the type of log line results. For possible values see *Log line tag options*
- **q** – A search string to limit the results to.

- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.
- **since** – Return log lines with a timestamp since the given datetime (inclusive). This parameter should be of the form YYYY-MM-DD HH:MM:SS and is assumed to be UTC.
- **before** – Return log lines with a timestamp before this datetime (exclusive). This parameter should be of the form YYYY-MM-DD HH:MM:SS and is assumed to be UTC.

Status Codes

- **200 OK** – Successfully queried the log lines.

Response Data

- **loglines** – The list of the player's log lines.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **loglines** is an object of log line information. See *Log line response data*

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/players/gumptionthomas/loglines HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "loglines": [
    {
      "username": "gumptionthomas",
      "updated": "2013-04-19 10:32:56 CDT-0500",
      "log_level": "INFO",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIoCxIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB0xvZ0xpbmVjcmFmdH",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH",
      "timestamp": "2013-04-19 10:32:55 CDT-0500",
      "tags": [
        "timestamp",
        "chat"
      ],
      "ip": null,
      "created": "2013-04-19 10:32:56 CDT-0500",
      "player_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHIzCzIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSB1",
      "location": null,
      "chat": "hey guys",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCzIEVXNlchivbgw",
      "line": "2013-04-19 10:32:55 [INFO] [Server] <gumptionthomas> hey guys",
      "port": null
    },
  ],
}
```


Query Parameters

- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.
- **since** – Return screenshots with a create timestamp since the given date-time (inclusive). This parameter should be of the form YYYY-MM-DD HH:MM:SS and is assumed to be UTC.
- **before** – Return screenshots with a create timestamp before this date-time (exclusive). This parameter should be of the form YYYY-MM-DD HH:MM:SS and is assumed to be UTC.

Status Codes

- **200 OK** – Successfully queried the screenshot.

Response Data

- **screenshots** – The list of screenshots.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **screenshots** is an object of screenshot information.

Screenshot

- **key** – The screenshot key.
- **server_key** – The screenshot’s server key.
- **user_key** – The user’s key that uploaded the screenshot.
- **random_id** – A random float attached to the screenshot at creation time.
- **original_url** – The URL of the original screenshot.
- **blurred_url** – The URL of the blurred version of the screenshot. `null` if the blurred version isn’t ready.
- **created** – The creation timestamp.
- **updated** – The updated timestamp.

Example request:

```
GET /api/v1/servers/ahRzfm1bXB0aW9uLW1pbmVjcmFmdH/screenshots HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "screenshots": [
    {
      "updated": "2013-04-13 11:12:20 CDT-0500",
      "created": "2013-04-13 11:12:05 CDT-0500",
```

```

    "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchiBfQw",
    "original_url": "http://lh5.ggpht.com/AMWDO-e5cK153ejlWn0ExDv1DuUACRpyM0kYEgAJKqTjs8a
    "random_id": 0.23893109322623773,
    "blurred_url": "http://lh4.ggpht.com/j8qNAEjoxIubBdRNZgjj629-2vjFOzWfSgkGPOmvr8VHiIBY
    "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHlrcXIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSC1NjcmVlb
    "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH"
  },
  {
    "updated": "2013-04-07 01:52:11 CDT-0500",
    "created": "2013-04-07 01:50:57 CDT-0500",
    "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchivbgw",
    "original_url": "http://lh3.ggpht.com/IFQVCSjpctTvNkJQhqj-j7anoaApZmawMe-Qy1LVqV2GKS9
    "random_id": 0.6780209099707669,
    "blurred_url": "http://lh6.ggpht.com/x0BKS8tbI88RRkhUX6vJ7MmzjhBaZShbKf51Th5oghUYtezZ
    "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHlrcXIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSC1NjcmVlb
    "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH"
  }
]
}

```

GET `/api/v1/servers/ (server_key) /screenshots/`
 key Get the information for the screenshot (key).

Parameters

- **server_key** – The target server’s key. (*required*)
- **key** – The requested screenshot’s key. (*required*)

Status Codes

- **200 OK** – Successfully read the screenshot.

Response Data See *Screenshot response data*

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/screenshots/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```

{
  "updated": "2013-04-07 01:52:11 CDT-0500",
  "created": "2013-04-07 01:50:57 CDT-0500",
  "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchivbgw",
  "original_url": "http://lh3.ggpht.com/IFQVCSjpctTvNkJQhqj-j7anoaApZmawMe-Qy1LVqV2GKS9k_Ak
  "random_id": 0.6780209099707669,
  "blurred_url": "http://lh6.ggpht.com/x0BKS8tbI88RRkhUX6vJ7MmzjhBaZShbKf51Th5oghUYtezZbD94
  "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHlrcXIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSC1NjcmVlb1Nob
  "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH"
}

```

GET `/api/v1/servers/ (server_key) /users/`
 key/**screenshots** Get a *list* of a user (key) uploaded screenshots on the server (server_key)
 ordered by descending create timestamp.

Parameters

- **server_key** – The target server’s key. (*required*)
- **key** – The requested user’s key. (*required*) To reference the authenticated user, use `self`.

Query Parameters

- **size** – The number of results to return per call (Default: 10. Maximum: 50).
- **cursor** – The cursor string signifying where to start the results.
- **since** – Return log lines with a create timestamp since the given date-time (inclusive). This parameter should be of the form `YYYY-MM-DD HH:MM:SS` and is assumed to be UTC.
- **before** – Return log lines with a create timestamp before this date-time (exclusive). This parameter should be of the form `YYYY-MM-DD HH:MM:SS` and is assumed to be UTC.

Status Codes

- **200 OK** – Successfully queried the screenshots.

Response Data

- **screenshots** – The list of the user’s uploaded screenshots.
- **cursor** – If more results are available, this value will be the string to be passed back into this resource to query the next set of results. If no more results are available, this field will be absent.

Each entry in **screenshots** is an object of the user’s uploaded screenshot information. See *Screen shot response data*

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/users/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchivbgw/
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "screenshots": [
    {
      "updated": "2013-04-07 01:52:11 CDT-0500",
      "created": "2013-04-07 01:50:57 CDT-0500",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchivbgw",
      "original_url": "http://lh3.ggpht.com/IFQVCSjpctTvNkJQhqj-j7anoaApZmawMe-Qy1LVqV2GKS9",
      "random_id": 0.6780209099707669,
      "blurred_url": "http://lh6.ggpht.com/x0BKS8tbI88RRkhUX6vJ7MmzjhBaZShbKf51Th5oghUYtezZ",
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchivbgw",
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH"
    },
    {
      "updated": "2013-03-25 18:39:36 CDT-0500",
      "created": "2013-03-25 18:39:22 CDT-0500",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHILCxIEVXNlchivbgw",

```

```
    "original_url": "http://lh6.ggpht.com/TFqVUT4hZwgz0sImwFMI9J7rJ-AXCqwM9-K5s66v9UnXy_i
    "random_id": 0.07680268292837988,
    "blurred_url": "http://lh5.ggpht.com/B-pQmMTlp6vZ7ke48-19e7YdUclpRUE30y4L_DS45a9dUt9Q
    "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHlrcXIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSC1NjcmVlb
    "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH"
  }
]
}
```

Example request:

```
GET /api/v1/servers/ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH/users/self/screenshots HTTP/1.1
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "screenshots": [
    {
      "updated": "2013-04-07 01:52:11 CDT-0500",
      "created": "2013-04-07 01:50:57 CDT-0500",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHlrcXIEVXNlchivbgw",
      "original_url": "http://lh3.ggpht.com/IFQVCSjpctTvNkJQhqj-j7anoaApZmawMe-Qy1LVqV2GKS9
      "random_id": 0.6780209099707669,
      "blurred_url": "http://lh6.ggpht.com/x0BKS8tbI88RRkhUX6vJ7MmzjhBaZShbKf51Th5oghUYtezZ
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHlrcXIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSC1NjcmVlb
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH"
    },
    {
      "updated": "2013-03-25 18:39:36 CDT-0500",
      "created": "2013-03-25 18:39:22 CDT-0500",
      "user_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHlrcXIEVXNlchivbgw",
      "original_url": "http://lh6.ggpht.com/TFqVUT4hZwgz0sImwFMI9J7rJ-AXCqwM9-K5s66v9UnXy_i
      "random_id": 0.07680268292837988,
      "blurred_url": "http://lh5.ggpht.com/B-pQmMTlp6vZ7ke48-19e7YdUclpRUE30y4L_DS45a9dUt9Q
      "key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdHlrcXIGU2VydmVyIglnbG9iYWxfc2VydmVyDAsSC1NjcmVlb
      "server_key": "ahRzfmd1bXB0aW9uLW1pbmVjcmFmdH"
    }
  ]
}
```

/api

<p>GET /api/v1/(list_resource), 10</p> <p>GET /api/v1/(resource), 9</p> <p>GET /api/v1/servers, 23</p> <p>GET /api/v1/servers/(key), 25</p> <p>GET /api/v1/servers/(key)/properties, 27</p> <p>GET /api/v1/servers/(server_key)/achievements, 47</p> <p>GET /api/v1/servers/(server_key)/achievements/(key), 49</p> <p>GET /api/v1/servers/(server_key)/chats, 39</p> <p>GET /api/v1/servers/(server_key)/chats/(key), 41</p> <p>GET /api/v1/servers/(server_key)/deaths, 44</p> <p>GET /api/v1/servers/(server_key)/deaths/(key), 45</p> <p>GET /api/v1/servers/(server_key)/loglines, 51</p> <p>GET /api/v1/servers/(server_key)/loglines/(key), 53</p> <p>GET /api/v1/servers/(server_key)/players, 33</p> <p>GET /api/v1/servers/(server_key)/players/(key_username), 34</p> <p>GET /api/v1/servers/(server_key)/players/(key_username)/achievements, 50</p> <p>GET /api/v1/servers/(server_key)/players/(key_username)/chats, 41</p> <p>GET /api/v1/servers/(server_key)/players/(key_username)/deaths, 46</p> <p>GET /api/v1/servers/(server_key)/players/(key_username)/loglines, 54</p> <p>GET /api/v1/servers/(server_key)/players/(key_username)/sessions, 37</p> <p>GET /api/v1/servers/(server_key)/screenshots, 56</p> <p>GET /api/v1/servers/(server_key)/screenshots/(key),</p>	<p>58</p> <p>GET /api/v1/servers/(server_key)/sessions, 35</p> <p>GET /api/v1/servers/(server_key)/sessions/(key), 36</p> <p>GET /api/v1/servers/(server_key)/users/(key)/screenshots, 58</p> <p>GET /api/v1/users, 20</p> <p>GET /api/v1/users/(key), 22</p> <p>POST /api/v1/(resource), 9</p> <p>POST /api/v1/servers, 25</p> <p>POST /api/v1/servers/(key), 26</p> <p>POST /api/v1/servers/(key)/properties, 29</p> <p>POST /api/v1/servers/(key)/queue/pause, 32</p> <p>POST /api/v1/servers/(key)/queue/play, 32</p> <p>POST /api/v1/servers/(server_key)/chats, 40</p> <p>POST /api/v1/servers/(server_key)/players/(key_username)/achievements, 43</p>
	<p>/oauth</p> <p>GET /oauth/v1/auth, 17</p> <p>GET /oauth/v1/clients/(client_id), 13</p> <p>POST /oauth/v1/register, 12</p> <p>POST /oauth/v1/token, 19</p> <p>PUT /oauth/v1/clients/(client_id), 14</p> <p>DELETE /oauth/v1/clients/(client_id), 16</p>