

## AUTHORIZATION

BASE URL: `https://api.eyeson.team/`  
 HEADERS: `Authorization: YOUR_API_KEY`

A running meeting requires authorization with an ACCESS\_KEY

## ROOM

### Create room

Initialize room with `<api_key>`. Response contains `<access_key>` and `<guest_token>`.

```
POST /rooms
```

Authorization: `<api_key>`, required: `user[name]`,  
`options[sfu_mode]=disabled`, `options[widescreen]=true`

### Room parameters

`id`, `name`, `user[id]`, `user[name]`, `user[avatar]`,  
`options[show_names]`, `options[show_label]`, `options[exit_url]`,  
`options[recording_available]`, `options[broadcast_available]`,  
`options[reaction_available]`, `options[layout_available]`,  
`options[guest_token_available]`, `options[lock_available]`,  
`options[kick_available]`, `options[sfu_mode]`,  
`options[widescreen]`, `options[background_color]`,  
`options[audio_insert]`, `options[audio_insert_position][x]`,  
`options[audio_insert_position][y]`, `options[custom_fields][locale]`,  
`options[custom_fields][logo]`, `options[custom_fields][hide_chat]`,  
`options[custom_fields][virtual_background]`,  
`options[custom_fields][virtual_background_allow_guest]`,  
`options[custom_fields][virtual_background_image]`

### Lock room

```
POST /rooms/<access_key>/lock
```

### Get room details

```
GET /rooms/<access_key>
```

### Stop room

```
DELETE /rooms/<access_key>
```

### Get list of running rooms

```
GET /rooms
HEADERS Authorization
```

### Force stop room

```
DELETE /rooms/<room_id>
HEADERS Authorization
```

## USERS

### Get user details

```
GET /rooms/<access_key>/users/<user_id>
```

### Join

From API-Response: `links[GUI]`  
`https://app.eyeson.team/?ACCESS_KEY`

### Register user

```
POST /rooms
```

REQUIRED `id`, `user[name]`, RECOMMENDED `user[id]`

### Register guest

```
POST /guests/<guest_token>
```

### Kick

```
DELETE /rooms/<access_key>/users/<user_id>
```

## LAYERS

PNG / WEBP - 1280x960 (Default) or 1280x720 (Widescreen)

### Add layer

```
POST /rooms/<access_key>/layers
```

### Image from web url as overlay

```
curl -X POST \
  -d "url=https://www.domain.com/file.webp" \
  -d "z-index=1" \
  "https://api.eyeson.team/rooms/$ACCESS_KEY/layers"
```

### Local image in background

```
curl -X POST \
  -F "file=@path/to/local/file.png" \
  -F "z-index=-1" \
  "https://api.eyeson.team/rooms/$ACCESS_KEY/layers"
```

### Clear layer

```
DELETE /rooms/<access_key>/layers/<z-index>
```

## LAYOUT

### Apply layout

```
POST /rooms/<access_key>/layout
```

VALID names: `one`, `two`, `four`, `six`, `nine`, `present-lower-3`,  
`present-upper-6`, `present-two-upper-6`, `present-upper-right-9`,  
`present-vertical-9`

### Custom layout

User spots are filled in order and can overlap

```
POST /rooms/<access_key>/layout
map = [[x,y,width, height,(object_fit)],[...]]
```

`object_fit`: `cover`, `contain`, `auto` (narrow videos get contain)

### Position users

Their placement in the  
user list is what matters

```
users[] = ""
users[] = $USER_ID
users[] = ""
users[] = ""
```

### Freeze position

Empty spots won't get filled

```
layout = custom
```

### Voice activation

Active speakers replace inactive ones

```
voice_activation = true
```

## PLAYBACK

Only webm files can be looped. Optimal conversion via ffmpeg:

```
ffmpeg -i input.mp4 -r 25 -g 50 -c:v libvpx -b:v 5M -c:a libvorbis output.webm
```

### Start

```
POST /rooms/<access_key>/playbacks
```

### Stop

```
DELETE /rooms/<access_key>/playbacks/<play_id>
```

### Example

```
curl -X POST \
  -d "audio=true" \
  -d "play_id=demo-video" \
  -d "url=https://myapp.com/playback.webm" \
  "https://api.eyeson.team/rooms/$ACCESS_KEY/playbacks"
```

### ADD DRONE OR RTMP SOURCE

1. Download Go Host Streaming Client (Ghost)  
<https://github.com/eyeson-team/ghost/releases>
2. Set drone to stream to IP of Ghost machine (use ngrok if necessary), RTMP, port 1935

```
./rtmp-server_<OS_VERSION>.exe --user Drone  
https://app.eyeson.team/?guest=[GUEST_TOKEN]
```

3. Drone joins call



### SNAPSHOTS & RECORDINGS

You will need to have the id to retrieve a recording or a snapshot. You can list all the recordings and snapshots in regular intervals. Alternatively, use the observer.

#### Start / Stop recording

```
POST /rooms/<access_key>/recording  
DELETE /rooms/<access_key>/recording
```

#### Retrieve recording

```
GET /recordings/<recording_id>
```

#### Get list of recordings

```
GET /rooms/<room_id>/recordings
```

#### Create snapshot

```
POST /rooms/<access_key>/snapshot
```

#### Retrieve snapshot

```
GET /rooms/<access_key>/snapshots/<snapshot_id>
```

#### Get list of snapshots

```
GET /rooms/<room_id>/snapshots
```

### OBSERVER

Get call metadata via the Observer, a one-way WebSocket using Rails ActionCable (with AnyCable as its successor).

```
https://api.eyeson.team/rt?room_id=<room_id>  
(can be wss:// protocol in some cases)  
api_key=<YOUR_API_KEY>  
Subscribe to RoomChannel
```

#### Event type examples

```
room_update           broadcasts_update  
participant_update   options_update  
recording_update     playback_update  
snapshot_update      podium_update
```

### FORWARD STREAM

The API URL must contain the ROOM\_ID in contrast to ACCESS\_KEY in other API calls. forward\_id MUST be unique for each forward!

#### Forward source

```
POST /rooms/<ROOM_ID>/forward/source
```

#### Forward MCU

```
POST /rooms/<ROOM_ID>/forward/mcu
```

#### Forward playback

```
POST /rooms/<ROOM_ID>/forward/playback
```

#### End forward

```
DELETE /rooms/<ROOM_ID>/forward/<FORWARD_ID>
```

### PERMALINK

Create a persistent link for a call

#### Create permalink

```
POST /permalink  
HEADERS Authorization
```

#### Delete permalink

```
DELETE /permalink  
HEADERS Authorization
```

#### Delete host user from permalink

```
DELETE /permalink/<permalink_id>/users/<user-token>
```

#### Start meeting from permalink

```
POST /permalink/<user_token>
```

#### Add host user to meeting

Host users can start a meeting, only give away guest links.

```
POST /permalink/<permalink_id>/users
```

### NOTES

---

---

---

---

---

---

---

---

---

---

---

---