

Animaze API

- [Intro](#)
- [Setup](#)
- [Overview](#)
 - [Actions](#)
 - [Triggers](#)
- [API Reference \(Actions\)](#)
 - [Load Avatar](#)
 - [Load Scene](#)
 - [Get Avatars](#)
 - [Get Current Avatar Info](#) (available only on next-version branch of Animaze App)
 - [Get Scenes](#)
 - [Get Current Scene Info](#) (available only on next-version branch of Animaze App)
 - [Get Item Icon](#)
 - [Get Emotes](#)
 - [Trigger Emote](#)
 - [Get Special Actions](#)
 - [Trigger Special Action](#)
 - [Get Poses](#)
 - [Trigger Pose](#)
 - [Get Idle Anims](#)
 - [Trigger Idle](#)
 - [Get Quicksценes](#)
 - [Load Quickscene](#)
 - [Save Quickscene](#)
 - [Calibrate Tracker](#)
 - [Broadcast](#)
 - [Set Camera Transform](#)
 - [Get Camera Transform](#)
 - [Set Camera Fov](#)
 - [Get Camera Fov](#)
 - [Set Avatar Transform](#)
 - [Get Avatar Transform](#)
 - [Set Override](#)
 - [Follow Mouse Cursor](#)
 - [Mouse Keyboard Behavior](#)
 - [Tracked Blinking](#)
 - [Auto Blink](#)
 - [Look At Camera](#)
 - [Look At Camera Head](#)
 - [Cross Eyes](#)
 - [Pupil Behavior](#)
 - [Forced Symmetry Eyebrows](#)
 - [Forced Symmetry Eyelids](#)
 - [Forced Symmetry Mouth](#)

- Enhanced Body Movement 2D
- Enhanced Body Movement 3D
- Extreme Head Angles Attenuation
- Sound to Mouth Open
- Alternate lipsync Retargeting
- Idle Intensity
- Inferred Body Yaw Movement
- Breathing Behavior
- Get Override
 - Follow Mouse Cursor
 - Mouse Keyboard Behavior
 - Tracked Blinking
 - Auto Blink
 - Look At Camera
 - Look At Camera Head
 - Cross Eyes
 - Pupil Behavior
 - Forced Symmetry Eyebrows
 - Forced Symmetry Eyelids
 - Forced Symmetry Mouth
 - Enhanced Body Movement 2D
 - Enhanced Body Movement 3D
 - Extreme Head Angles Attenuation
 - Sound to Mouth Open
 - Alternate lipsync Retargeting
 - Idle Intensity
 - Inferred Body Yaw Movement
 - Breathing Behavior
- Send Message to Animaze ChatPal
- API Reference (Triggers)
 - Animaze ChatPal Speech Start/End Events (available only on next-version branch of Animaze App)
- Sample

Intro

The Animaze API enables 3rd party apps to control certain functions (e.g. load avatar) or get certain info (e.g. list of ready to use avatars) of/from the Animaze Desktop application. The API is implemented using a **websocket** that exchanges **JSON** messages between the Animaze app and 3rd party clients.

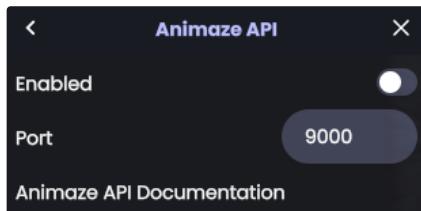
i If you are looking into making your app control the avatar's expressions and body motions, best to look into our [Programmable Tracker](#), which allows for you to directly retarget your tracking data to the avatar. Read more [here](#).

i If you are looking into making your app control the avatar's body (full body tracking or hands tracking), best to look into using the [VMC protocol](#), as Animaze supports it. Here is a user-facing guide: [Animaze VMC protocol Body Tracker Usage Guide](#)

Setup

The Animaze API websocket uses by default **ws://localhost:9000** address.

Animaze is not accepting commands by default, so you need to activate the API from Settings > Animaze API > Enabled. The Port number can also be changed from Settings > Animaze API in the Animaze Desktop app.



Present and future API settings will reside in Settings > Animaze API, so be sure to check them from time to time.

Overview

The Animaze API supports commands or actions (send a command, receive a response) and in the future will also support triggers, Animaze state changes broadcasted to all clients (e.g. the avatar changed)

Actions

Actions are messages sent to Animaze that describe a specific command that needs to be executed and provide a response.

Action request example

```
1 {  
2   "action": "LoadAvatar",           // required by all action messages  
3   "id": "0E9AA4AE-8C70-4999-BFAE-4D639A3F73E2", // optional  
4   "name": "Avatar03"               // required by the "LoadAvatar" action  
5 }
```

An action response will echo the “**action**” field and “**id**“ field if present (use this to link your original request with its response for a better management) and an optional “**error**“ message in case the command failed.

Action response example

```
1 {
2   "action": "LoadAvatar",
3   "id": "0E9AA4AE-8C70-4999-BFAE-4D639A3F73E2",
4   "error": "Avatar not found"
5 }
```

Triggers

Triggers are messages sent by Animaze to all connected clients with information about state changes initiated by the app.

Trigger example

```
1 {
2   "event": "AvatarChanged",           // changed state
3   "new_avatar": "AvatarBomb"        // state value
4 }
```

API Reference (Actions)

Load Avatar

Send to Animaze

```
1 {
2   "action": "LoadAvatar",           // required
3   "id": <id>                      // optional (string)
4   "name": "Avatar03"               // Avatar name (required by action)
5 }
```

Receive from Animaze

same command

Load Scene

```
1 {
2   "action": "LoadScene",            // required
3   "id": <id>                      // optional (string)
4   "name": "customization01_scene" // Scene name (required by action)
5 }
```

Receive from Animaze

same command

Get Avatars

Get the list of ready to use avatars (.avatar files are already on disk).

Send to Animaze

```
1 {
2   "action": "GetAvatars",           // required (string)
3   "id": <id>                   // optional (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetAvatars",           // string
3   "id": <id>,                  // string
4   "avatars": ["avatar03", "AvatarBomb"] // All available avatars (array)
5 }
```

Get Current Avatar Info (available only on next-version branch of Animaze App)

Get advanced info on a list of currently loaded avatars. The command returns a list so that it covers the Animaze Rooms scenario where there are multiple avatars spawned.

Send to Animaze

```
1 {
2   "action": "GetCurrentAvatarInfo", // required (string)
3   "id": "MyDevName"              // optional (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetCurrentAvatarInfo",
3   "avatars": [
4     {
5       "animStandard": "AnimazeStandard_1_0",
6       "description": "A spiffy and adorable red panda.",
7       "friendlyName": "Fluffo",
8       "isCustomized": true,
9       "itemName": "avatar01_BS_Full",
10      "itemType": "Avatar3D",
11      "numProps": 3,
12      "props": ["crown_king", "greenGlasses", "headphones"]
13    },
14    {
15      "id": "MyDevName",
16      "numAvatars": 1
17  }
18 }
```

Get Scenes

Get the list of ready-to-use scenes (.avatar files are already on disk).

Send to Animaze

```
1 {
2   "action": "GetScenes",           // required (string)
3   "id": <id>                   // optional (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetScenes",           // string
3   "id": <id>,                  // string
4   "scenes": ["Spaceship", "Stream"] // All available scenes (array)
5 }
```

Get Current Scene Info (available only on next-version branch of Animaze App)

Get advanced info on the currently loaded scene.

Send to Animaze

```
1 {
2   "action": "GetCurrentSceneInfo", // required (string)
3   "id": "MyDevName"             // optional (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetCurrentSceneInfo",
3   "description": "A gamer's room.",
4   "friendlyName": "The RGB Room",
5   "id": "MyDevName",
6   "itemName": "RGBRoom"
7 }
```

Get Item Icon

Get the item's icon as base64 encoded image. This function works for all Animaze items, such as: avatars, scenes, props, emotes, voice effects, etc.

Send to Animaze

```
1 {
2   "action": "GetItemIcon",          // required (string)
3   "id": <id>                   // optional (string)
4   "name": "customization01_scene" // Works for all animaze items that are present
5                                // in a gallery: avatars, scenes, props,
6                                // voice effects, etc (required by action)
7 }
```

Receive from Animaze

```
1 {
2   "action": "GetItemIcon",          // string
3   "id": <id>,                  // string
4   "image": "iVBORw0KGgoAAAANSUhE..." // base64 encoded image file (string)
5 }
```

Get Emotes

Get the list of ready to use emotes (.avatar files are already on disk).

Send to Animaze

```
1 {
2   "action": "GetEmotes"           // required (string)
3 }
```

Receive from Animaze

```
1 {
2   "action": "GetEmotes",          // string
3   "emotes": [{"friendlyName": "Alarm", "itemName": "Emote_Alarm"},  
             {"friendlyName": "Angry", "itemName": "Emote_Angry"}... ] //all available Emotes
4 }
```

Trigger Emote

Triggers an emote

Send to Animaze

```
1 {
2   "action": "TriggerEmote",      // required (string)
3   "itemName": "Emote_Alarm"    // required (string)
4 }
```

Receive from Animaze

same command

Get Special Actions

Get the special actions list of the **current** avatar

Send to Animaze

```
1 {
2   "action": "GetSpecialActions"  // required (string)
3 }
```

Receive from Animaze

```
1 {
2   "action": "GetSpecialActions", // string
3   {"action": "GetSpecialActions", "specialActions": [{"animName": "Come here", "index": 0},  
             {"animName": "Hello", "index": 1}, ...] } //list of all special actions for the current avatar
4 }
```

Trigger Special Action

Start a special action on the **current** avatar

Send to Animaze

```
1 {
2   "action": "TriggerSpecialAction", // required (string)
```

```
3   "index": <int>                                // required (int)
4 }
```

Receive from Animaze

same command

Get Poses

Get the special poses list of the **current** avatar

Send to Animaze

```
1 {
2   "action": "GetPoses"                         // required (string)
3 }
```

Receive from Animaze

```
1 {
2   "action": "GetPoses",                         // string
3   {"action":"GetPoses","poseList":[{"animName":"Puff Cheeks","index":0},{"animName":"Tongue Out","index":1}]} //list of all special poses for the current avatar
4 }
```

Trigger Pose

Set the pose for the **current** avatar

Send to Animaze

```
1 {
2   "action": "TriggerPose",                      // required (string)
3   "index" : <int>                             // required (int)
4 }
```

Receive from Animaze

same command

Get Idle Anims

Get the idle animations list of the **current** avatar

Send to Animaze

```
1 {
2   "action": "GetIdleAnims"                     // required (string)
3 }
```

Receive from Animaze

```
1 {
2   "action": "GetIdleAnims",                    // string
```

```
3     {"action": "GetIdleAnims", "idleList": [{"animName": "Default Idle", "index": 0},  
4     {"animName": "Bored", "index": 2}, {"animName": "Fight", "index": 3}]} //list of all idle animations  
for the current avatar  
4 }
```

Trigger Idle

Set the idle animations for the **current** avatar

Send to Animaze

```
1 {  
2   "action": "TriggerIdle",           // required (string)  
3   "index" : <int>                  // required (int)  
4 }
```

Receive from Animaze

same command

Get Quickscenes

Get the ready-to-use Quickscenes list

Send to Animaze

```
1 {  
2   "action": "GetQuickscenes"        // required (string)  
3 }
```

Receive from Animaze

```
1 {  
2   "action": "GetQuickscenes",       // required (string)  
3   "quickslots": [{"index": 0, "image": "iVBORw0KGgoAAAANSUhE..."}...] // list of indices and  
base64 encoded image file (string)  
4 }
```

Load Quickscene

Loads the Quickscene

Send to Animaze

```
1 {  
2   "action": "LoadQuickscene",      // required (string)  
3   "index" : <int>                // required (int)  
4 }
```

Receive from Animaze

same command

Save Quickscene

Triggers saving current setup (avatar, scene, camera placement, voice effects, etc.) to be saved as a Quickscene.

Send to Animaze

```
1 {
2   "action": "SaveQuickscene",           // required (string)
3   "index" : <int>                     // required (int)
4 }
```

Receive from Animaze

same command

Calibrate Tracker

Trigger Animaze's tracker calibration function

Send to Animaze

```
1 {
2   "action": "CalibrateTracker"        // required (string)
3 }
```

Receive from Animaze

same command

Broadcast

Enables virtual camera broadcast.

Send to Animaze

```
1 {
2   "action": "Broadcast",              // required (string)
3   "toggle": <bool>                  // required (bool)
4 }
```

Receive from Animaze

same command

Set Camera Transform

Sets a custom World Space transformation to the render camera. Camera transformation is right-handed (OpenGL).

Send to Animaze

```
1 {
2   "action": "SetCameraTransform",      // required (string)
```

```
3   "position" : [<float>, <float>, <float>],           // required ([float, float, float])
4   "rotation" : [<float>, <float>, <float>, <float>]    // required ([float, float, float,
float]) quaternion
5 }
6
```

Receive from Animaze

same command

Get Camera Transform

Gets render camera's World Space Transformation. Camera transformation is right-handed (OpenGL).

Send to Animaze

```
1 {
2   "action": "GetCameraTransform"          // required (string)
3 }
4
```

Receive from Animaze

```
1 {
2   "action": "GetCameraTransform",         // (string)
3   "position": [0.0, 1.7000000476837158, 1.0], // (float, float, float)
4   "rotation": [0.0, 0.0, 0.0, 1.0]        // (float, float, float, float) quaternion
5 }
```

Set Camera Fov

Sets a new Fov to the render camera. Fov must be given in degrees.

Send to Animaze

```
1 {
2   "action": "SetCameraFov",      // required (string)
3   "fov" : float                // required (float) in degrees
4 }
5
```

Receive from Animaze

same command

Get Camera Fov

Gets render camera's Fov

Send to Animaze

```
1 {  
2   "action": "GetCameraFov"           // required (string)  
3 }  
4
```

Receive from Animaze

```
1 {  
2   "action": "GetCameraFov",          // (string)  
3   "fov":60.000003814697266        // (float)  
4 }
```

Set Avatar Transform

Sets a custom Word Space transformation to the render camera. Camera transformation is right-handed (OpenGL).

Send to Animaze

```
1 {  
2   "action": "SetAvatarTransform",      // required (string)  
3   "position" : [<float>, <float>, <float>],    // required ([float, float, float])  
4   "rotation" : [<float>, <float>, <float>, <float>], // required ([float, float, float,  
   float]) quaternion  
5   "scale": <float>                  // required (float)  
6 }
```

Receive from Animaze

same command

Get Avatar Transform

Gets render camera's World Space Transformation. Camera transformation is right-handed (OpenGL).

Send to Animaze

```
1 {  
2   "action": "GetAvatarTransform"       // required (string)  
3 }
```

Receive from Animaze

```
1 {  
2   "action": "GetAvatarTransform",      // (string)  
3   "position": [0.0,1.7000000476837158,1.0], // (float, float, float)  
4   "rotation": [0.0,0.0,0.0,1.0],         // (float, float, float, float) quaternion  
5   "scale": 1.0                          // (float)  
6 }
```

Set Override

Activates/deactivates one of the following override behaviors on the current avatar. Parameters are optional:

[Follow Mouse Cursor](#)

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Follow Mouse Cursor", // required (string)
4   "value": <bool>                  // required (bool)
5 }
```

[Mouse Keyboard Behavior](#)

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Mouse Keyboard Behavior", // required (string)
4   "value": <bool>,                  // required (bool)
5   "keyboardScale": <float>,         // optional (float)
6   "handToKeyboardDist": <float>,    // optional (float)
7   "xAxisOffset": <float>,           // optional (float)
8   "yAxisOffset": <float>,           // optional (float)
9   "zAxisOffset": <float>            // optional (float)
10 }
```

[Tracked Blinking](#)

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Tracked Blinking",   // required (string)
4   "value": <bool>                  // required (bool)
5 }
```

[Auto Blink](#)

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Auto Blink",         // required (string)
4   "value": <bool>,                // required (bool)
5   "frequency": <float>            // optional (float)
6 }
```

Look At Camera

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Look At Camera",    // required (string)
4   "value": <bool>,                // required (bool)
5   "range": <float>                // optional (float)
6 }
```

Look At Camera Head

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Look At Camera Head", // required (string)
4   "value": <bool>                  // required (bool)
5 }
```

Cross Eyes

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Cross Eyes",        // required (string)
4   "value": <bool>                 // required (bool)
5 }
```

Pupil Behavior

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Pupil Behavior",     // required (string)
4   "value": <bool>,                // required (bool)
5   "frequency": <float>            // optional (float)
6 }
```

Forced Symmetry Eyebrows

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Forced Symmetry Eyebrows", // required (string)
4   "value": <bool>,                 // required (bool)
5   "range": <float>                // optional (float)
6 }
```

```
6 }
```

Forced Symmetry Eyelids

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Forced Symmetry Eyelids", // required (string)
4   "value": <bool>,                // required (bool)
5   "range": <float>                // optional (float)
6 }
```

Forced Symmetry Mouth

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Forced Symmetry Mouth", // required (string)
4   "value": <bool>,                // required (bool)
5   "range": <float>                // optional (float)
6 }
```

Enhanced Body Movement 2D

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Enhanced Body Movement 2D", // required (string)
4   "value": <bool>                 // required (bool)
5 }
```

Enhanced Body Movement 3D

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Enhanced Body Movement 3D", // required (string)
4   "value": <bool>                 // required (bool)
5 }
```

Extreme Head Angles Attenuation

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Extreme Head Angles Attenuation", // required (string)
```

```
4   "value": <bool>,           // required (bool)
5   "range": <float>,          // optional (float)
6   "maxSmoothingValue" : <float> // optional (float)
7 }
```

Sound to Mouth Open

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Sound to Mouth Open", // required (string)
4   "value": <bool>,                // required (bool)
5   "mouthSmoothTme": <float>,       // optional (float)
6   "soundSensitivity": <float>      // optional (float)
7 }
```

Alternate lipsync Retargeting

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Alternate lipsync Retargeting", // required (string)
4   "value": <bool>                 // required (bool)
5 }
```

Idle Intensity

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Idle Intensity",     // required (string)
4   "value": <bool>,                // required (bool)
5   "intensity": <float>            // optional (float)
6 }
```

Inferred Body Yaw Movement

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Inferred Body Yaw Movement", // required (string)
4   "value": <bool>,                // required (bool)
5   "neutralZoneSize": <float>,       // optional (float)
6   "movementSmoothing": <float>,     // optional (float)
7   "headYawWeight": <float>          // optional (float)
8 }
```

Breathing Behavior

Send to Animaze

```
1 {
2   "action": "SetOverride",           // required (string)
3   "behavior": "Breathing Behavior", // required (string)
4   "value": <bool>,                // required (bool)
5   "amplitude": <float>,          // optional (float)
6   "frequency": <float>           // optional (float)
7 }
```

Receive from Animaze

same command

Get Override

Returns the status of an override behavior (activated/deactivated) and its current parameters, applied on the current avatar.

Follow Mouse Cursor

Send to Animaze

```
1 {
2   "action": "GetOverride",          // required (string)
3   "behavior": "Follow Mouse Cursor" // required (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetOverride",          // string
3   "behavior": "Follow Mouse Cursor", // string
4   "value": true/false             // bool
5 }
6
```

Mouse Keyboard Behavior

Send to Animaze

```
1 {
2   "action": "GetOverride",          // required (string)
3   "behavior": "Mouse Keyboard Behavior" // required (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetOverride",          // string
```

```
3   "behavior": "Mouse Keyboard Behavior", // string
4   "value": true/false,                  // bool
5   "keyboardScale": <float>,           // float
6   "handToKeyboardDist": <float>,       // float
7   "xAxisOffset": <float>,              // float
8   "yAxisOffset": <float>,              // float
9   "zAxisOffset": <float>               // float
10 }
11
```

Tracked Blinking

Send to Animaze

```
1 {
2   "action": "GetOverride",             // required (string)
3   "behavior": "Tracked Blinking"     // required (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetOverride",             // string
3   "behavior": "Tracked Blinking",    // string
4   "value": true/false                // bool
5 }
6
```

Auto Blink

Send to Animaze

```
1 {
2   "action": "GetOverride",             // required (string)
3   "behavior": "Auto Blink"           // required (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetOverride",             // string
3   "behavior": "Auto Blink",          // string
4   "value": true/false,               // bool
5   "frequency": <float>              // float
6 }
7
```

Look At Camera

Send to Animaze

```
1 {
2   "action": "GetOverride",             // required (string)
```

```
3   "behavior": "Look At Camera"           // required (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetOverride",                // string
3   "behavior": "Look At Camera",          // string
4   "value": true/false,                  // bool
5   "range": <float>                   // float
6 }
7
```

Look At Camera Head

Send to Animaze

```
1 {
2   "action": "GetOverride",                // required (string)
3   "behavior": "Look At Camera Head"     // required (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetOverride",                // string
3   "behavior": "Look At Camera Head",    // string
4   "value": true/false,                  // bool
5 }
6
```

Cross Eyes

Send to Animaze

```
1 {
2   "action": "GetOverride",                // required (string)
3   "behavior": "Cross Eyes"              // required (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetOverride",                // string
3   "behavior": "Cross Eyes",             // string
4   "value": true/false,                  // bool
5 }
6
```

Pupil Behavior

Send to Animaze

```
1 {
2   "action": "GetOverride",           // required (string)
3   "behavior": "Pupil Behavior"    // required (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetOverride",           // string
3   "behavior": "Pupil Behavior",    // string
4   "value": true/false,             // bool
5   "frequency": <float>           // float
6 }
7
```

Forced Symmetry Eyebrows

Send to Animaze

```
1 {
2   "action": "GetOverride",           // required (string)
3   "behavior": "Forced Symmetry Eyebrows" // required (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetOverride",           // string
3   "behavior": "Forced Symmetry Eyebrows", // string
4   "value": true/false,              // bool
5   "range": <float>                // float
6 }
7
```

Forced Symmetry Eyelids

Send to Animaze

```
1 {
2   "action": "GetOverride",           // required (string)
3   "behavior": "Forced Symmetry Eyelids" // required (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetOverride",           // string
3   "behavior": "Forced Symmetry Eyelids", // string
4   "value": true/false,              // bool
5   "range": <float>                // float
6 }
7
```

Forced Symmetry Mouth

Send to Animaze

```
1 {  
2   "action": "GetOverride",           // required (string)  
3   "behavior": "Forced Symmetry Mouth" // required (string)  
4 }
```

Receive from Animaze

```
1 {  
2   "action": "GetOverride",           // string  
3   "behavior": "Forced Symmetry Mouth", // string  
4   "value": true/false,              // bool  
5   "range": <float>                 // float  
6 }  
7
```

Enhanced Body Movement 2D

Send to Animaze

```
1 {  
2   "action": "GetOverride",           // required (string)  
3   "behavior": "Enhanced Body Movement 2D" // required (string)  
4 }
```

Receive from Animaze

```
1 {  
2   "action": "GetOverride",           // string  
3   "behavior": "Enhanced Body Movement 2D", // string  
4   "value": true/false              // bool  
5 }  
6
```

Enhanced Body Movement 3D

Send to Animaze

```
1 {  
2   "action": "GetOverride",           // required (string)  
3   "behavior": "Enhanced Body Movement 3D" // required (string)  
4 }
```

Receive from Animaze

```
1 {  
2   "action": "GetOverride",           // string  
3   "behavior": "Enhanced Body Movement 3D", // string  
4   "value": true/false              // bool  
5 }
```

[Extreme Head Angles Attenuation](#)

Send to Animaze

```

1 {
2   "action": "GetOverride",           // required (string)
3   "behavior": "Extreme Head Angles Attenuation" // required (string)
4 }
```

Receive from Animaze

```

1 {
2   "action": "GetOverride",           // string
3   "behavior": "Extreme Head Angles Attenuation", // string
4   "value": true/false,             // bool
5   "range": <float>,              // float
6   "maxSmoothingValue" : <float> // float
7 }
8
```

[Sound to Mouth Open](#)

Send to Animaze

```

1 {
2   "action": "GetOverride",           // required (string)
3   "behavior": "Sound to Mouth Open" // required (string)
4 }
```

Receive from Animaze

```

1 {
2   "action": "GetOverride",           // string
3   "behavior": "Sound to Mouth Open", // string
4   "value": true/false,             // bool
5   "mouthSmoothTme": <float>,      // float
6   "soundSensitivity": <float> // float
7 }
8
```

[Alternate lipsync Retargeting](#)

Send to Animaze

```

1 {
2   "action": "GetOverride",           // required (string)
3   "behavior": "\"Alternate lipsync Retargeting" // required (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetOverride",           // string
3   "behavior": "Alternate lipsync Retargeting", // string
4   "value": true/false             // bool
5 }
6
```

Idle Intensity

Send to Animaze

```
1 {
2   "action": "GetOverride",           // required (string)
3   "behavior": "Idle Intensity"      // required (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetOverride",           // string
3   "behavior": "Idle Intensity",     // string
4   "value": true/false              // bool
5   "intensity": <float>            // float
6 }
7
```

Inferred Body Yaw Movement

Send to Animaze

```
1 {
2   "action": "GetOverride",           // required (string)
3   "behavior": "Inferred Body Yaw Movement" // required (string)
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetOverride",           // string
3   "behavior": "Inferred Body Yaw Movement", // string
4   "value": true/false              // bool
5   "neutralZoneSize": <float>,          // float
6   "movementSmoothing": <float>,        // float
7   "headYawWeight": <float>             // float
8 }
9
```

Breathing Behavior

Send to Animaze

```
1 {
2   "action": "GetOverride",           // required (string)
3   "behavior": "Breathing Behavior"  // required (string)
```

```
4 }
```

Receive from Animaze

```
1 {
2   "action": "GetOverride",           // string
3   "behavior": "Breathing Behavior", // string
4   "value": true/false,             // bool
5   "amplitude": <float>,          // float
6   "frequency": <float>           // float
7 }
8
```

Send Message to Animaze ChatPal

Send a prompt to Animaze ChatPal, and receive the conversational AI response, while the avatar starts speaking. You can use this command as echo (for the text-to-speech functionality) as well by adding -echo at the start of the “message” string.

Send to Animaze

```
1 {
2   "action": "ChatbotSendMessage",      // required (string)
3   "id": <id>,                      // optional (string)
4   "message": "prompt to Animaze Chatpal" // required (string)
5 }
```

Receive from Animaze

```
1 {
2   "action": "ChatbotSendMessage",      // string
3   "response": "response from Animaze ChatPal" // string
4 }
```

API Reference (Triggers)

Animaze ChatPal Speech Start/End Events (available only on next-version branch of Animaze App)

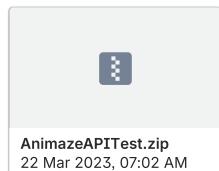
ChatbotSpeechStarted and **ChatbotSpeechEnded** events are sent by Animaze when ChatPal agent starts and stops speaking (start and end of audio). These events aren't extremely accurate and depend on the voice engine, but the error could be that the **ChatbotSpeechStarted** is sent slightly earlier and **ChatbotSpeechEnded** is sent slightly later. Error is empirically determined up to 1 second.

```
1 {
2   "event": "ChatbotSpeechStarted"
3 }
```

```
1  {
2    "event": "ChatbotSpeechEnded"
3 }
```

Sample

You can check out a sample Python script that uses all the available actions at this time. Here's the archive:



The “AnimazeAPITest.py“ script iterates through all commands in the “commands.json” file and sends them through the websocket.

Requirements:

- Python 3.11+
- Python modules: websockets