3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

# VAISense AAVA AI API

v1.6

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

Overview ..... 4
Software Components ..... 4
Architecture ..... 4
Installation ..... 5
VAISense RESTful AI API ..... 10
Basic Info ..... 10
API Testing Using Postman ..... 11
System API ..... 12
Get version ..... 12
Get UTC offset ..... 13
Get system info ..... 13
Camera API ..... 14
Get cameras ..... 14
Get camera ..... 14
Add camera ..... 16
Delete camera ..... 16
Edit camera ..... 17
Enable camera ..... 18
Disable camera ..... 18
Reconnect Camera ..... 19
Get Camera Snapshot ..... 19
AI Detections API ..... 20
Get AI Detection Types ..... 20
Get AI Detection ..... 21
Add AI Detection ..... 22
Delete AI Detection ..... 24
Edit AI Detection ..... 24

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

Camera Events ..... 25
Get Camera Event Types ..... 25
Get Camera Events ..... 26
Live Stream ..... 29
Create Live Stream ..... 29
Keep Live Stream ..... 30
Remove Live Stream ..... 31
VAISense Socket.IO API ..... 31
Basic info ..... 31
Socket API ..... 32
Receive Camera Event ..... 32
Create Live Stream ..... 34
Keep Live Stream ..... 35
Remove Live Stream ..... 36
Receive Live Stream ..... 36

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

# Overview

The VAISense AAVA API is a RESTful interface for local edge AI inference on the VAISense platform by LatticeWork. It enables developers to perform various actions such as set camera configurations, trigger AI detection, access event-based data, and create live streams.

The VAISense AAVA API Server is a lightweight, containerized solution designed to run AI inference workloads efficiently on VAISense edge devices equipped with DeGirum ORCA hardware accelerators. The server exposes a RESTful API, allowing clients to send inference requests and receive real-time predictions with minimal latency. Designed for scalability and reliability, it supports multiple AI models and includes features such as object detection, license plate recognition, line crossing, tracking and occupancy detection. Ideal for web-based applications running on Docker, this solution enhances vision AI capabilities at the edge while minimizing cloud dependencies.

# System

The AAVA service is deployed on Amber OS and interfaces with cameras and AI models. All API calls communicate over HTTP POST. Endpoints are typically accessed via local IP (e.g., 10.3.10.122) and port 9486.

# Software Components

## Architecture

The complete edge AI system on LatticeWork software ecosystem consists of the following key components, all containerized using Docker for modular deployment:

1. **VAISense AI Inference API Server (AAVA)** (this document)

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

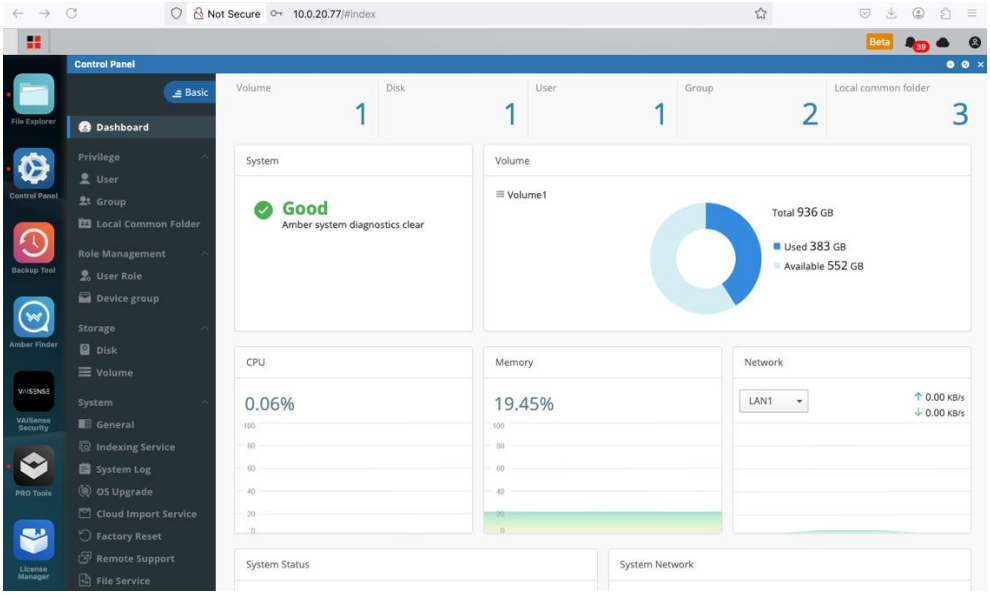
- Runs AI models on local ORCA AI hardware accelerators
- Provides RESTful APIs for Camera setup, AI inference and AI algorithms.
- Optimized for real-time multi-camera AI performance

2. **Web-Based AI Analytics Application** (provided by LatticeWork partner)
   - Web-based application running on Docker environment
   - Backend handles UI requests, data processing, and database interaction if needed
   - Interfaces with the Edge AI Inference API Server for real-time analytics.
3. **Amber OS and PROTools Docker Framework** (provided by LatticeWork)
   - Solid YOCTO Linux-based operating system with native hybrid cloud connectivity
   - PROTools Docker framework with user friendly Docker App management UI
   - Built-in data and application remote access through Amber Cloud
4. **Secure Remote Access through Amber Anywhere Services (**Provided by LatticeWork)
   - Provide external Internet accessible URL for Docker application
   - Secure remote access without DDNS or VPN needed

## Installation

After successfully setup Amber OS on VAISense hardware, you should be able to access the Amber OS web interface as shown in the following.

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com




**Note on Licensing and Deployment**

Amber OS includes built-in applications such as File Explorer, Control Panel, VAISense Security, PROTools, Backup Tool, and License Manager.

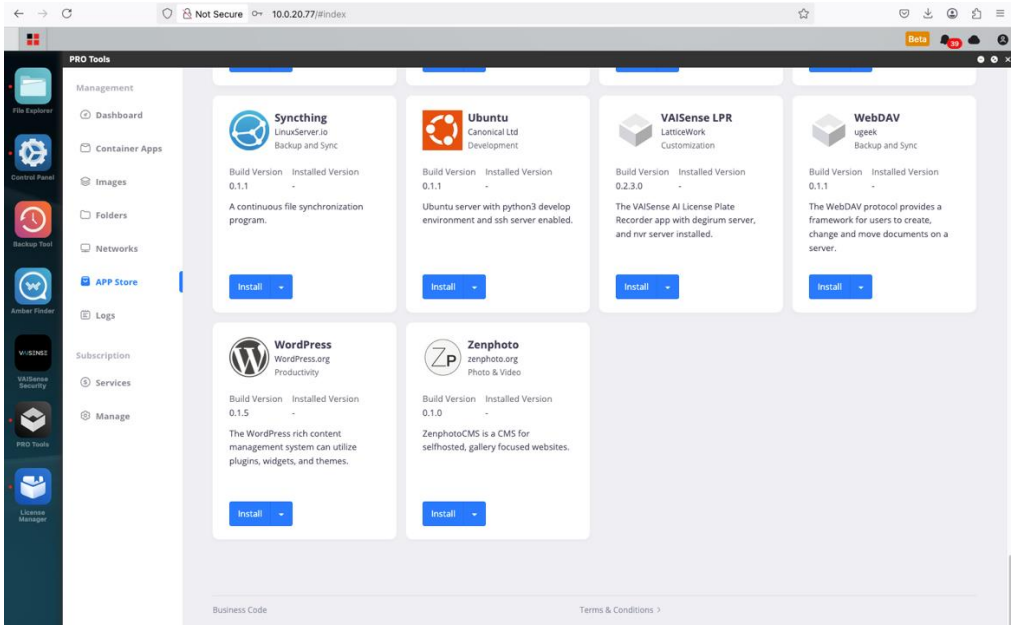
The License Manager is currently in beta and will be responsible for managing software and hardware entitlements — including Docker-based applications and camera licenses.

Please note that the AAVA API is intended for development purposes only. All production deployments will be governed and activated through the LatticeWork License Manager. Further licensing details and integration guidelines will be provided during the deployment phase.

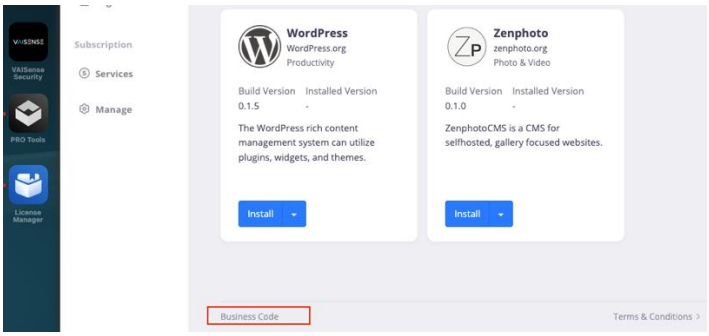
The PROTools is the Docker framework that allows you to install AAVA Docker app and your own applications. You can refer to the PROTools user guide for the details.

You will need to install a Docker App called “AAVA” in the PROTools App Store interface as shown below.

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

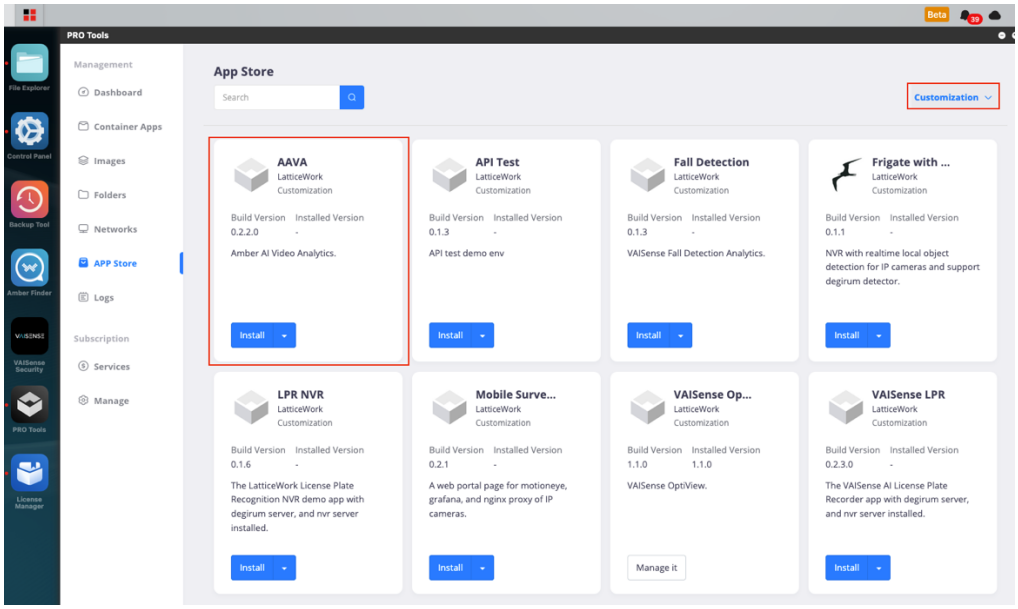



By default, the AAVA app won't be shown in the App Store. You need to install it through the "Business Code", on the bottom of the App Store interface.

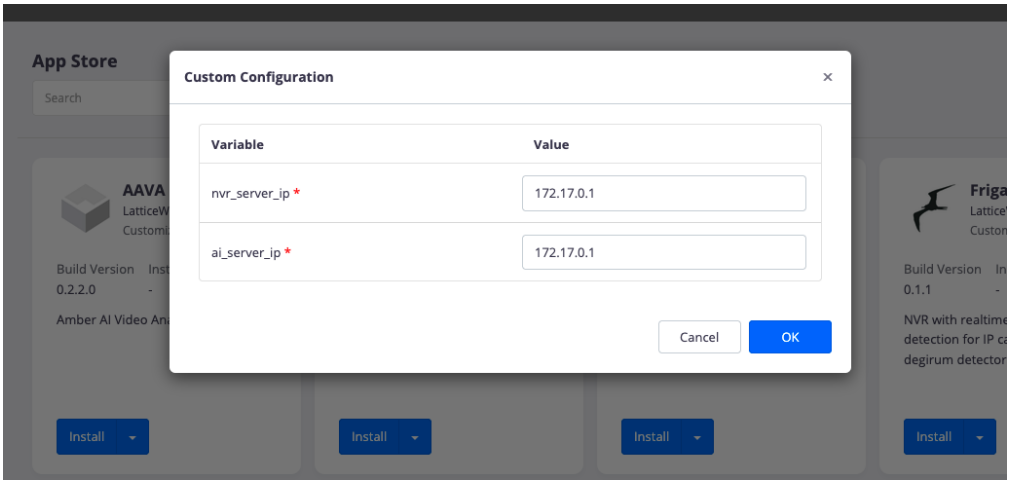



Click the "Business Code", enter "izry853726" and click Apply. (This code is Confidential and under protection by the NDA). After a while, the AAVA will be appearing in the App Store under "Customization" category as below.

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

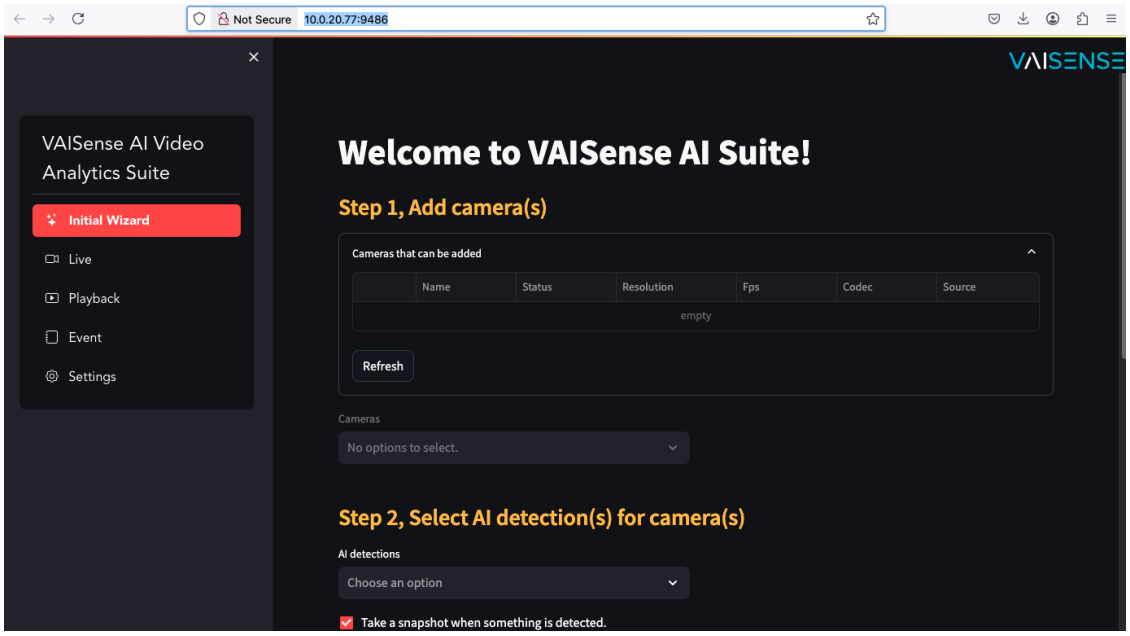


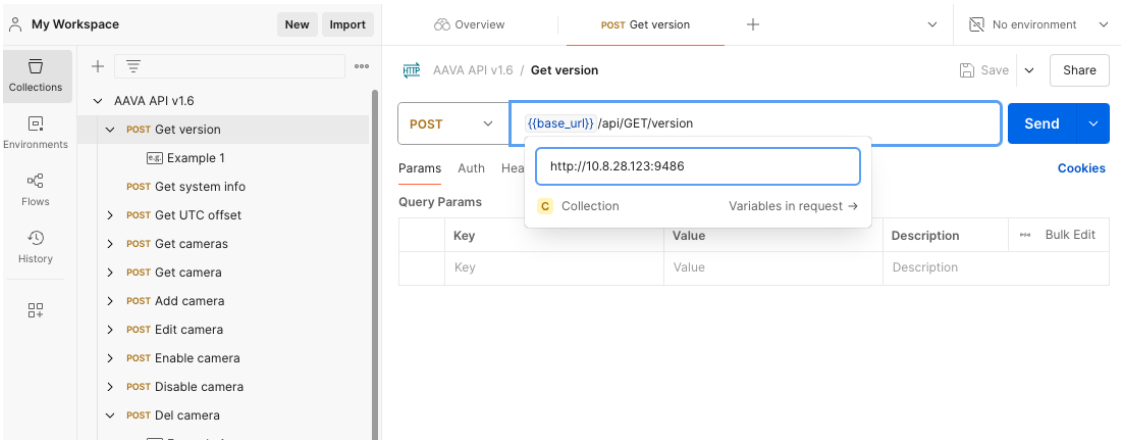

While installing AAVA, you will see the following settings. You can use the default settings with VAISense NVR and AI server on the same machine.




Once the app is running, you can open the AAVA web interface through the IP address and port number 9486. You can also use the same URL as the base_url of the Postman.

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com




## API Behavior and Tips

- All API requests must be sent via HTTP POST.
- Cameras must be added and enabled before applying AI detection.
- AI detection types must match IDs from /GET/ai-detection-types.
- Test your calls using Postman or a curl-compatible script.

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

- Common useful flows include Add → Enable → Detect → Retrieve Events.

# VAISense RESTful AI API

## Basic Info

- **Server port**: 9486
- **Protocol**: HTTP
- **Method**: POST
- **Request content type**:
  - application/json
- **Response content type**:
  - Defaults to application/json unless otherwise specified by the API
- **URI format**:
  - $PROTOCOL://$IP:$SERVER_PORT/api/$ENDPOINT
- **Common status code**:
  - 4XX/5XX will return JSON string, format is {"msg": str, "payload": object}, payload is extra info, default is null.

## Status Codes

| Code | Description |
|---|---|
| 201 | A new resource was created. |
| 204 | The request was successful, but the response has no content. |
| 400 | The request was malformed.<br>Default msg: BadRequest |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

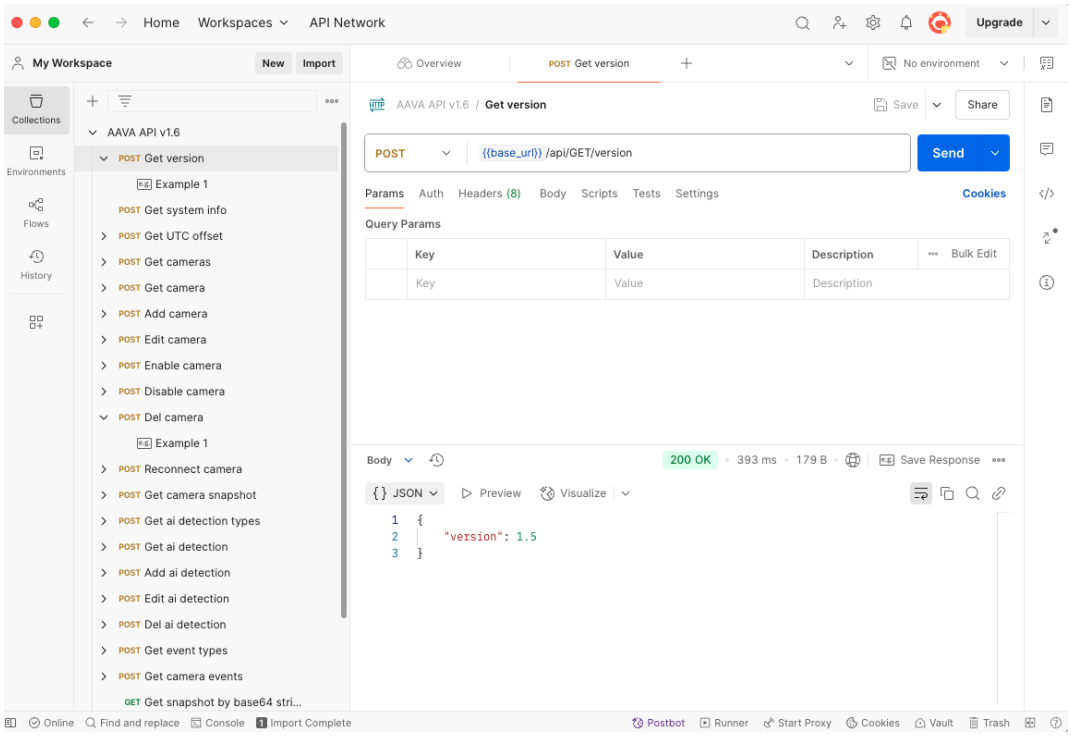
| | |
|---|---|
| | InvalidParm: XXX<br>NoParam: XXX |
| 404 | The requested resource was not found. |
| 415 | The request data format is not supported by the server.<br>msg: DataFormatNotSupported |
| 500 | The server threw an error when processing the request.<br>Default msg: UnknownError |
| 501 | Not Implemented.<br>msg: NotImplemented |
| 503 | Service Unavailable/Busy.<br>msg: ServerBusy or ServiceStopped |
| 505 | Version Not Supported.<br>msg: VersionNotSupported |

- Symbols: "*" means required.

## API Testing Using Postman

The release will come with "VAISense AAVA API v1.6.json" , which can be used with Postman (https://www.postman.com/downloads/). You can test each API on Postman directly before writing any codes.

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com




# System API

## Get version

**Description**: Get current API version.

**Endpoint**: /GET/version

**Responses**

| Code | Description |
|---|---|
| 200 | e.g.<br>{<br>"version": 1.0<br>} |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

## Get UTC offset

**Description**: Retrieves the current UTC time offset in `±HH:MM` format, based on its system timezone..

**Endpoint**: /GET/utc_offset

**Responses**

| Code | Description |
|---|---|
| 200 | Format: ±HH:MM<br>e.g.<br>{<br>"utc_offset": "+08:00"<br>} |

## Get system info

**Description**: Get system information including ai server details, cpu usage, memory usage, and version.

**Endpoint**: /GET/sys_info

**Response**

| Code | Description |
|---|---|
| 200 | e.g.<br>{<br>"ai_server": {<br>"dev_feq_mhz": 1896,<br>"dev_temp_c": 58<br>},<br>"app": {<br>"cpu_usage": 37.15,<br>"mem_usage": 849.22265625,<br>"version": "0.3.0.0"<br>}<br>} |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

# Camera API

## Get cameras

**Description**: Retrieves a list of all registered cameras, including their configurations and statuses.

**Endpoint**: /GET/cameras

**Response:**

| Code | Description |
|---|---|
| 200 | Format: [ camera1, camera2, ... etc ].<br>For camera format, please see Get camera . |

## Get camera

**Description**: Retrieves detailed information for a specific camera based on its ID.

**Endpoint**: /GET/camera

**Example Usage**: /GET/camera?id=14

**Parameters**

| Name | Type | Description |
|---|---|---|
| id* | int | |

**Response**

| Code | Description |
|---|---|
| 200 | Format:<br>{<br>"id": int, |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | |
|---|---|
| | "name": str,<br>"src": str,<br>"description": str,<br>"enabled": bool,<br>"src_info": {<br>"fps": float,<br>"width": int,<br>"height": int,<br>"codec": str,<br>"pix_fmt": str<br>},<br>"status": int,<br>"ai_detections": {<br>det_type: {"infe_fps": float},<br>...<br>},<br>"is_recording": bool,<br>"rec_start_time": str<br>}<br><br>status (int):<br>0: disconnected<br>1: connecting<br>2: connected<br>ai_detections (object): All AI detections used.<br>det_type (list): Refer to Get AI detection types.<br>infe_fps (float): AI inference frame rate.<br>rec_start_time (str): UTC datetime, format is %Y-%m-%d %H:%M:%S.%f |
| 404 | msg:IdNotExist |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

## Add camera

**Description**: Add a camera or RTSP video with name, source, and description. Added cameras are enabled by default. Note that you must use your camera''s RTSP link. If you do not know it, then refer to your camera manufacturer's manual. To add an RTSP video stream, please refer to the AddVideoFiles.docx**Endpoint**: /ADD/camera

**Parameters**

| Name | Type | Description |
|---|---|---|
| name* | str | |
| src* | str | Format: rtsp://$ID:$PW@$IP:$PORT/$ENDPOINT<br><br>E.g.<br>rtsp://admin:admin123456@10.8.32.72:8554/profile0 |
| description | str | |

Responses

| Code | Description |
|---|---|
| 200 | {<br>"id": 1234<br>} |
| 400 | msg:<br>NameExist<br>InvalidName<br>InvalidSrc |

## Delete camera

**Description**: Delete a camera by id.

**Endpoint**: /DELETE/camera

**Parameters**

| Name | Type | Description |
|---|---|---|

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | | |
|---|---|---|
| id* | int | |

**Responses**

| Code | Description |
|---|---|
| 204 | |
| 404 | msg: IdNotExist |

## Edit camera

**Description**: Updates the metadata of a camera, including its name, RTSP source URL, and description

**Endpoint**: /EDIT/camera

**Parameters**

| Name | Type | Description |
|---|---|---|
| id* | int | |
| name | str | |
| src | str | Refer to Add camera. |
| description | str | |

**Responses**

| Code | Description |
|---|---|
| 204 | |
| 400 | msg:<br>• NameExist<br>• InvalidName<br>• InvalidSrc |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | |
|---|---|
| | • NoParams |
| 404 | msg: IdNotExist |

## Enable camera

**Description**: Enables selected camera. Reflected in camera property `"enabled": true`

**Endpoint**: /ENABLE/camera

**Parameters**

| Name | Type | Description |
|---|---|---|
| id* | int | |

**Response**

| Code | Description |
|---|---|
| 204 | |
| 404 | msg: IdNotExist |

## Disable camera

**Description**: Disables selected camera. Reflected in camera property `"enabled": false`

**Endpoint**: /DISABLE/camera

**Parameters**

| Name | Type | Description |
|---|---|---|
| id* | int | |

**Response**

| Code | Description |
|---|---|

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | |
|---|---|
| 204 | |
| 404 | msg: IdNotExist |

## Reconnect Camera

**Description**: Attempt to reconnect a specified camera.

**Endpoint**: /RECONNECT/camera

**Parameters**

| Name | Type | Description |
|---|---|---|
| id* | int | |

**Responses**

| Code | Description |
|---|---|
| 204 | |
| 404 | msg:<br>IdNotExist |

## Get Camera Snapshot

**Description**: Get the latest snapshot of camera.

**Endpoint**: /GET/camera/snapshot

**Parameters**

| Name | Type | Description |
|---|---|---|
| id* | int | |
| size | list [int] | Format: [w, h]. |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | | |
|---|---|---|
| | | If W or H is -1, e.g.: -1,240, it means the height is 240 pixels and maintains the aspect ratio. |

**Response**

| Code | Description |
|---|---|
| 200 | Returns the bytes of the .jpg file.<br>Content-Type: image/jpeg |
| 204 | No snapshot. |
| 404 | msg:<br>• IdNotExist<br>• InvalidSize |

# AI Detections API

## Get AI Detection Types

**Description**: Get all available AI detection types.

**Endpoint**: /GET/ai-detection-types

**Parameters:** None

**Response:**

| Code | Description |
|---|---|
| 200 | Format: {id: name}<br>e.g.<br>{<br>"1": "Any Objects",<br>"2": "License Plate",<br>"3": "Gun/Knife",<br>"4": "Human Posture",<br>"5": "Human", |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | |
|---|---|
| | "6": "Vehicle",<br>"7": "Human/Vehicle",<br>"8": "Human Tracking",<br>"9": "Vehicle Tracking",<br>"10": "Human/Vehicle Tracking",<br>"11": "Any Objects Tracking",<br>"12": "Human Cross-Line Counting",<br>"13": "Vehicle Cross-Line Counting",<br>"14": "Human/Vehicle Cross-Line Counting",<br>"15": "Any Objects Cross-Line Counting",<br>"16": "Human Occupancy",<br>"17": "Vehicle Occupancy",<br>"18": "Human/Vehicle Occupancy",<br>"19": "Any Objects Occupancy"<br>} |

## Get AI Detection

**Description**: Get specific AI detection settings for a specified camera.

**Endpoint**: /GET/camera/ai-detection

**Parameters:**

| Name | Type | Description |
|---|---|---|
| cam_id* | int | |
| det_type* | int | Refer to Get AI detection types. |

**Response**

| Code | Description |
|---|---|
| 200 | Refer to the settings of Add AI detection. |
| 404 | msg: |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | |
|---|---|
| | • CamIdNotExist<br>• DetTypeNotExist<br>• NoSettings |

## Add AI Detection

**Description**: Add a specified ai detection model to a specified camera.

**Endpoint**: /ADD/camera/ai-detection

**Parameters:**

| Name | Type | Description |
|---|---|---|
| cam_id* | int | |
| det_type* | int | Refer to Get AI detection types.<br>Note: The same AI detection cannot be added to the camera twice. |
| settings* | object | Common parameters (required):<br>• event2IOT (bool): Sent event to IOT manager, default is false.<br>• event2DB (bool): Store event to DB, default is true.<br>• trigger_snapshot (bool): It can be enabled only when event2DB is enabled. Default is true.<br>• trigger_rec (bool): It can be enabled only when event2DB is enabled. Default is false.<br><br>**Crossline counting**:<br>• image_size* (list): [w, h]<br>• boundary* (list): [x1, y1, x2, y2]<br>• area_A* (list): (x, y). A position in area A, it is used to show the label "A".<br>• area_B* (list): (x, y). A position in area B, it is used to show the label "B".<br>• condition (int): 1: Two way, 2: Area A to Area B, 3: Area B to Area A. The default is 1.<br>e.g. |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | | |
|---|---|---|
| | | {<br>"image_size": [1280, 720],<br>"boundary": [10, 10, 100, 100] ,<br>"area_A": [10, 50],<br>"area_B": [50, 10],<br>"condition ": 3<br>}<br><br>**Occupancy detection**:<br>• image_size* (list): [w, h]<br>• zones* (object): Format is {zone_idx: zone, ...}, the starting index value is 1.<br>• zone* (object): {"center": [x, y], "path": [[x1, y1], [x2, y2], ...]}<br>e.g.<br>{<br>"image_size": [1280, 720],<br>"zones": {<br>"1": {"center": [100, 100], "path": [[0, 0], [0, 100], ...]},<br>"3": {"center": [100, 100], "path": [[0, 0], [0, 100], ...]},<br>...<br>},<br>"event2IOT": true,<br>"event2DB": true,<br>"trigger_snapshot": true<br>} |

**Response**

| Code | Description |
|---|---|
| 204 | |
| 400 | msg:<br>• DetTypeExist<br>• InvalidSettings |
| 404 | msg:<br>• CamIdNotExist<br>• DetTypeNotExist |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

## Delete AI Detection

**Description**: Delete a specified ai detection model from a specified camera.

**Endpoint**: /DELETE/camera/ai-detection

**Parameters**

| Name | Type | Description |
|---|---|---|
| cam_id* | int | |
| det_type* | int | Refer to Get AI detection types. |

**Response**

| Code | Description |
|---|---|
| 204 | |
| 404 | msg:<br>• CamIdNotExist<br>• DetTypeNotExist |

## Edit AI Detection

**Description**: Edits a selected camera's detection type and settings (limited to 1 at a time)

**Endpoint**: /EDIT/camera/ai-detection

**Parameters:**

| Name | Type | Description |
|---|---|---|
| cam_id* | int | |
| det_type* | int | Refer to Get AI detection types. |
| settings* | object | Refer to Add AI detection. |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

**Response**:

| Code | Description |
|---|---|
| 204 | |
| 404 | msg:<br>• CamIdNotExist<br>• DetTypeNotExist<br>• InvalidSettings |

## Camera Events

### Get Camera Event Types

**Description**: Get available event types.

**Endpoint**: /GET/cam-event-types

**Response:**

| Code | Description |
|---|---|
| 200 | Format: {id: name}<br><br>e.g.<br>{<br>"1": "Camera connected",<br>"2": "Camera disconnected",<br>"3": "Camera disabled",<br>"4": "Camera enabled",<br>...<br>"100": "Object found",<br>"101": "Object not found",<br>"102": "Objects continue to be found",<br>"103": "Zone occupied",<br>"104": "Zone continue to be occupied",<br>"105": "Zone not occupied" |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | |
|---|---|
| | } |

## Get Camera Events

**Description**: Get the camera's events from the DB. The results only include events where the "event2DB" attribute of the AI detection setting is true.

**Endpoint**: /GET/camera/events

**Parameters:**

| Name | Type | Description |
|---|---|---|
| cam_id* | int | |
| reverse | bool | false: ascending, true: descending.<br>The default is False. (sorted by ts) |
| page_idx | int | The default is 0. |
| items_per_page | int | The default is 50. |
| ret_snapshot | bool | Returns a base64 string that can be used to create a URL to get the snapshot.<br>The URL format is $PROTOCOL://$IP:$SERVER_PORT/snapshot/$BASE64_STR?cam_id=$CAM_ID &size=WxH (Parameter size is optional, if W or H is -1, e.g.: -1,240, it means the height is 240 pixels and maintains the aspect ratio.)<br>The default is False. |
| ret_video | bool | Returns a base64 string that can be used to create a URL to get the recorded video.<br>The URL format is $PROTOCOL://$IP:$SERVER_PORT/recording/$BASE64_STR?cam_id=$CAM_ID<br>The default is False. |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | | |
|---|---|---|
| filters | obje<br>ct | Format:<br>{<br>"det_types": [int, int, ...],<br>"time_range": [from, to]<br>}<br><br>det_types (list): Refer to Get AI detection types. 0 means this event is not related to AI detection.<br>time_range (list): The "from" and "to" are both UTC datetime string, format is %Y-%m-%d %H:%M:%S.%f.<br>The default is {}. |

**Response**:

| Code | Description |
|---|---|
| 200 | Format:<br>{<br>"total_page": int,<br>"items_per_page": int,<br>"events": [<br>{<br>"id": int,<br>"ts": str,<br>"e_type": int,<br>"det_type": int,<br>"detail": object,<br>"snapshot": str,<br>"video": str,<br>"v_fts": str<br>},<br>...<br>]<br>}<br><br>ts (str): UTC datetime, format is %Y-%m-%d %H:%M:%S.%f.<br>e_type (int): Refer to Get camera event types.<br>det_type (int): Refer to Get AI detection types. 0 means this event is not related to AI detection. |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | |
|---|---|
| | detail (object): Different events have different content.<br>{<br>"objs": {<br>"obj1": int, // The number of object.<br>"obj2": int,<br>...<br>}<br>}<br><br>OBJECT_FOUND<br>{<br>"duration": int, // Unit: second.<br>"objs": {<br>"obj1": int, // The number of object.<br>"obj2": int,<br>...<br>}<br>}<br><br>OBJECT_FOUND_CONTI<br>OBJECT_NOT_FOUND<br>{<br>"zone_idx": int,<br>"objs": {<br>"obj1": int, // The number of object.<br>"obj2": int,<br>...<br>}<br>}<br>ZONE_OCCUPIED<br>{<br>"zone_idx": int,<br>"duration": int, // Unit: second.<br>"objs": {<br>"obj1": int, // The number of object.<br>"obj2": int,<br>...<br>}<br>} |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | |
|---|---|
| | ZONE_OCCUPIED_CONTI<br>ZONE_NOT_OCCUPIED<br>{ "zone_idx": int}<br><br>**snapshot (str):** A base64 string if snapshot exists (Only present when ret_snapshot is true).<br>**video (str):** A base64 string if video exists (Only present when ret_video is true).<br>**v_fts (str):** The timestamp of first frame in video, it is UTC datetime, format is %Y-%m-%d %H:%M:%S.%f (Only present when ret_video is true). |
| 404 | **msg**: CamIdNotExist |

# Live Stream

## Create Live Stream

**Description**: Creates a live stream of the specified camera with a default duration of 1 minute. Use the "Keep Live Stream" api to extend the streams duration.

**Endpoint**: /ADD/camera/live

**Parameters:**

| Name | Type | Description |
|---|---|---|
| cam_id* | int | |
| det_type* | int | Refer to Get AI detection types.<br>0 means original video. |
| format | str | Supported format: jpg<br>The default is jpg. |

**Response**:

| Code | Description |
|---|---|

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | |
|---|---|
| 200 | E.g.<br>{<br>"stream_id": "abc123"<br>}<br><br>If the format is jpg, you can create URL using stream_id, and then put it into a html image tag <img src="$URL">. The image tag is constantly updated like a video.<br>URL format is<br>$PROTOCOL://$IP:$SERVER_PORT/live?cam_id=$CAM_ID<br>&det_type=$DET_TYPE&stream_id=$STREAM_ID<br><br>P.S. If the det_type is 0, its URL does not need "&det_type=" |
| 400 | msg: InvalidFormat |
| 404 | msg:<br>• CamIdNotExist<br>• DetTypeNotExist |

## Keep Live Stream

**Description**: Extends the duration of a specified live stream by 1 minute .

**Endpoint**: /KEEP/camera/live

**Parameters**

| Name | Type | Description |
|---|---|---|
| stream_id | str | |

**Response**:

| Code | Description |
|---|---|
| 204 | |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | |
|---|---|
| 404 | msg: StreamIdNotExist |

### Remove Live Stream

**Description**: Removes the livestream of a specified camera and frees server-side resources.

**Endpoint**: /DELETE/camera/live

**Parameters:**

| Name | Type | Description |
|---|---|---|
| stream_id | str | |

**Response**:

| Code | Description |
|---|---|
| 204 | |
| 404 | msg: StreamIdNotExist |

# VAISense Socket.IO API

## Basic info

- **Server port**: 9486
- **Protocol**: Socket.IO (ws)
- **Request content type**: application/json
- **Response content type**: application/json
- **URI format**: $PROTOCOL://$IP:$SERVER_PORT/sio
- **Common status code**: Refer to The Basic info of API

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

- **Response format**:
  - Status code is 2XX: {"code": int, "payload": object}, the default payload is null.
  - Status code is 4XX/5XX: {"code": int, "msg": str}

The client needs to receive messages "connect", "connect_error" and "disconnect". Refer to Socket.io Events

# Socket API

## Receive Camera Event

**Description**: Receives camera events using using the “event” message in the /camera namespace. Additional event details can be retrieved using the Get Camera and Get Camera Events endpoint.

**Namespace**: camera

**Message name of listening**: event

**Responses**

| Code | Description |
|---|---|
| 200 | Format:<br>{<br>'cam_id': 1,<br>'ts': '2025-01-08 06:47:10.626472',<br>'e_type': 4,<br>'detail': {'det_type': 1}<br>}<br><br>e_type:<br>1: camera added<br>2: camera deleted<br>3: info/status updated<br>4: AI detection event is created<br>detail (dict): The content varies by the e_type. |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | |
|---|---|
| | When e_type is 4, it will return the det_type, you can refer to Get AI detection types. |

Here is the sample code of using Socket IO to receive the camera event:

```
import socketio
import requests
import os

sio = socketio.Client()

@sio.event
def connect():
    print('connection established')

@sio.event
def disconnect():
    print('disconnected from server')

@sio.event
def connect_error(data):
    print(f"The connection failed! {data}")

@sio.on("event", namespace="/camera")
def handle_cam_event(data: dict):
    print(f"cam event: {data}")

def handle_response(data: dict):
    print(f"response: {data}")

# change to your server's ip address
ipAddress = "127.0.0.1"

# default namespaces is the registered event handlers are connected.
sio.connect(f'http://{ipAddress}:9486', socketio_path="sio")
```

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

```
print(f"connection namespaces: {sio.connection_namespaces}")
# say hello for testing
sio.emit("message", "hello!", namespace="/camera", callback=handle_response)
sio.wait()
```

## Create Live Stream

**Description**: Create a live stream of a specific camera. The stream lifetime is 1 minute, if you want to continue using this stream, please call Keep live stream before the stream is stopped.

Important Note: The live streaming related API in the AAVA will be obsolete after the VAISense Core API is ready. All the camera related API will be removed and controlled by VAISense Core API and License Manager.

**Namespace**: camera

**The event emitted**: start_recv_live

**Parameters:**

| Name | Type | Description |
|---|---|---|
| cam_id* | int | |
| det_type* | int | Refer to Get AI detection types.<br>0 means original video. |

**Response**:

| Code | Description |
|---|---|
| 200 | Format:<br>{ stream_id": "abc123”} |
| 400 | msg: |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

| | |
|---|---|
| | InvalidCamId |
| 404 | msg:<br>• CamIdNotExist<br>• CamDisabled<br>• DetTypeNotExist<br>• CamIdOrDetTypeNotExist |

## Keep Live Stream

**Description**: Notify the server that you want to continue using this stream for 1 minute.

Important Note: The live streaming related API in the AAVA will be obsolete after the VAISense Core API is ready. All the camera related API will be removed and controlled by VAISense Core API and License Manager.

**Namespace**: camera

**The event emitted**: keep_recv_live

**Parameters:**

| Name | Type | Description |
|---|---|---|
| stream_id | str | |

**Response**:

| Code | Description |
|---|---|
| 204 | |
| 404 | msg: StreamIdNotExist |

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

## Remove Live Stream

**Description**: Remove live stream of a specific camera. Please call this API to release resources when you don't need to display or close the video player.

Important Note: The live streaming related API in the AAVA will be obsolete after the VAISense Core API is ready. All the camera related API will be removed and controlled by VAISense Core API and License Manager.

**Namespace**: camera

**The event emitted**: stop_recv_live

**Parameters:**

| Name | Type | Description |
|---|---|---|
| stream_id | str | |

**Response**:

| Code | Description |
|---|---|
| 204 | |
| 404 | msg: StreamIdNotExist |

## Receive Live Stream

**Description**: Retrieves real-time JPEG frames streamed by the server.

Important Note: The live streaming related API in the AAVA will be obsolete after the VAISense Core API ready. All the camera related API will be removed and controlled by VAISense Core API and License Manager.

**Namespace**: camera

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

**Message name of listening**: live

**Responses:**

| Code | Description |
|---|---|
| 200 | Format:<br>{<br>'cam_id': 1,<br>'ts': '2025-01-08 06:47:10.626472',<br>'det_type': 4,<br>'frame': Arraybuffer,<br>'frame_size': [w, h],<br>'annotation': {}<br>}<br><br>frame is jpg image, type is ArrayBuffer. You can create a URL using URL.createObjectURL(blob) and assign it to a HTML image element.<br>annotation is dictionary, content is different by AI detection, the basic parameters as follows<br>str_det_type: AI detection name<br>infe_fps: The inference frame rate of AI model |

## API Endpoint List (22)

1. /GET/version
2. /GET/utc_offset
3. /GET/sys_info
4. /GET/cameras

5. /GET/camera

3211 Scott Blvd, Suite 100
Santa Clara, CA 95054
(408) 475-1471
support@latticeworkinc.com

6. /ADD/camera
7. /DELETE/camer
8. /EDIT/camera
9. /ENABLE/camera
10. /DISABLE/camera
11. /RECONNECT/camera
12. /GET/camera/snapshot
13. /GET/ai-detection-types
14. /GET/camera/ai-detection
15. /ADD/camera/ai-detection
16. /DELETE/camera/ai-detection
17. /EDIT/camera/ai-detection
18. /GET/cam-event-types
19. /GET/camera/events
20. /ADD/camera/live
21. /KEEP/camera/live
22. /DELETE/camera/live