## Endpoint: /run

This endpoint starts an algorithm for a given camera. The algorithm is specified in the request body along with the camera URL and server URL. Here's how to use this endpoint:

### Request

* URL: /run
* Method: POST
* Request body: A JSON object that contains the following fields:
  + algorithm: The name of the algorithm to run. This is a string that corresponds to a key in the Algorithms object.
  + camera\_url: The URL of the camera to run the algorithm on. This should be a valid URL that the API can use to access the camera.
  + server\_url: The URL of the server where the API is running. This should be a valid URL that the camera can use to send data to the API.
  + extra (optional): An object containing any extra data that the algorithm may need. This field is only required for the "min\_max\_control" algorithm.

### Response

* Content-Type: application/json
* Response body: A JSON object that contains the following fields:
  + success: A boolean that indicates whether the algorithm was started successfully.
  + pid (optional): An integer that represents the process ID of the algorithm. This is only returned if the algorithm was started successfully.

## Endpoint: /stop

This endpoint stops a running algorithm. The request must contain the following parameter in the request body:

### Request

* URL: /stop
* Method: POST
* Request body: A JSON object that contains the following fields:
  + pid: An integer that represents the process ID of the algorithm.

### Response

* Content-Type: application/json
* Response body: A JSON object that contains the following fields:
  + success: A boolean that indicates whether the algorithm was stopped successfully.

## Endpoint: *api*/reports/report-with-photos/

The api/reports/report-with-photos/ endpoint is a RESTful API endpoint that accepts HTTP POST requests containing JSON data. The endpoint is used to create a report that includes photos.

### Request

* URL: /*api*/reports/report-with-photos/
* Method: POST
* Request body: A JSON object that contains the following fields:
  + algorithm: A string that indicates the type of control.
  + camera: A string that indicates the hostname of the camera.
  + start\_tracking: A string that indicates the start date and time for tracking. For operation controls, this value is provided by the control payload.
  + stop\_tracking: A string that indicates the stop date and time for tracking. For operation controls, this value is provided by the control payload.
  + photos: An array of objects that contain the photo data for the report. Each object in the array should have the following properties:

image: A string that indicates the path to the image file.

date: A string that indicates the date and time that the photo was taken.

* + violation\_found: A boolean that indicates whether a violation was found. For operation controls, this value is provided by the control payload.
  + extra: An object that contains additional data related to the report. This property is optional and can be omitted.

### Response

* Content-Type: application/json
* Response body: A JSON object that contains the following fields:
  + success: A boolean that indicates whether the report was saved successfully.