

ApolloService API

Table of Contents

- Introduction
- Request
- Response
- Business domain
- Operations
- Utility operations
- Search operations
 - Search detections
 - Search sequence statistics
- List operations
 - List sequence statistics
 - List signs
 - List photo detections
- Retrieve operations
 - Retrieve details of a detection
 - Retrieve detections of a sequence
 - Retrieve detections of a photo
 - Retrieve contributions of a detection
 - Retrieve overall statistics
- Edit operations
 - Update detection

Introduction

This document represents the API Specification for the new Apollo back-end component. Basic knowledge of HTTP and JSON is recommended for a better understanding of this document.

Note:

- the API is detection centered; since the OpenStreetCam API already contains methods for retrieving images
 - category and type names are not final yet
 - new functionality will be added/existing functionality might change
-

Request

Each request is a GET or POST HTTP request using UTF-8 as the character encoding. A GET request has the following syntax:

| `[LOCATION]/apolloService/[METHOD]?[ARGUMENTS]`

The POST requests are very similar with the sole exception that the arguments get transmitted in the body of the message, not in the request URL itself. For POST request the Content-Type expected by the service is application/json;charset=UTF-8.

The variables described in this section are used as place-holders for actual values. The meaning of each of these variables is as follows:

- [LOCATION]: the base URL to the service implementation, including the protocol, host and optionally port number
- [METHOD]: the name of the operation provided by the service
- [ARGUMENTS]: a list of arguments, each one in the form *name=value* and separated by the "&" sign. Both name and value must be URL encoded, as defined by the HTTP specification. Both the name and the value must be URL encoded, as defined by the HTTP specification.

An example GET request could look like this:

```
| https://apollo-api-v2.openstreetcam.org/apolloService/echo?format=fjson
```

Here <https://apollo-api-v2.openstreetcam.org/apolloService/> is the LOCATION, *echo* is the METHOD and *message=hello&locale_code=en* is the list of ARGUMENTS.

Response

The component returns JSON responses encoded in UTF-8. Each response contains a [Status](#) object which defines the success or failure of the operation along with some explanations:

```
| Status  
  
| apiCode: INTEGER  
  
| apiMessage: STRING  
  
| httpCode: INTEGER
```

All methods may return statuses 600, 611 and 690. Besides these, each method defines its specific possible statuses.

Api code	Api Message	HTTP code	Text code *	Description
600	The request has been processed without incidents	200	OK	no problems here
601	The request has produced an empty result	200	EMPTY_RESULT	no problems here but no result either
602	The request has been processed but there have been incidents	200	OK_WITH_WARNINGS	the operation has succeeded and the response has been sent as expected but there are some warnings
610	A required argument is missing	400	MISSING_ARGUMENT	at least one required argument is missing from the client request
611	An argument is invalid	400	INVALID_ARGUMENT	at least one of the arguments received from the client is invalid which means it is the wrong type (e.g. a text instead of a number); if that argument is required, the implementation must return this status; if the argument is optional, the implementation may return this status or it may ignore the argument and use its default value
612	An argument is out of range	400	OUT_OF_RANGE_ARGUMENT	at least one of the arguments received from the client is outside its allowed range of values; if that argument is required, the implementation must return this status; if the argument is optional, the implementation may return this status or it may ignore the argument and use its default value
613	Invalid request body	400	INVALID_REQUEST_BODY	the POST request body received from the client is invalid; it is either a wrong type or it has an invalid value
660	Duplicate entry	400	DUPLICATE	a duplicate entry exists

670	The identifier is incorrect	409	INCORRECT_IDENTIFIER	the supplied identifier (id or name or other) is either invalid or does not exist
671	The status is incorrect	409	INCORRECT_STATUS	the status of the desired entity is incorrect relative to the desired operation (e.g. trying to close an already closed bug)
690	An unexpected server error has occurred	500	SERVER_ERROR	this appears when an unexpected exception is thrown in our (server) code; this should actually never happen but if it does it identifies a programming mistake on our (back-end) part

Business domain

The business entities specific to this component are:

Sequence

id: *INTEGER*
device: *STRING*
author: *Author*
photos: *ARRAY of Photos*

Photo

sequenceId: *INTEGER*
sequenceIndex: *INTEGER*
creationTimestamp: *INTEGER*
point: *Point*
width: *INTEGER*
height: *INTEGER*
heading: *INTEGER*
processStatus: *STRING*
detections: *ARRAY of Detection*

Detection

id: *INTEGER*

sequenceId: INTEGER

sequenceIndex: INTEGER

creationTimestamp: INTEGER

latestChangeTimestamp: INTEGER

point: Point

sign: Sign

locationOnPhoto: Rectangle

confidenceLevel: FLOAT

validationStatus: STRING

editStatus: STRING

osmComparison: STRING

osmElement: OsmElement

mode: STRING

author: Author

algorithm: Algorithm

componentValue: STRING

parentId: INTEGER

Point

lat: FLOAT

lon: FLOAT

Rectangle

x: FLOAT

y: FLOAT

width: FLOAT

height: FLOAT

OsmElement

osmId: INTEGER

type: STRING

members: ARRAY of INTEGER

Sign

name: STRING

internalName: STRING

iconName: STRING

region: STRING

type: STRING

components: ARRAY of Sign

Algorithm

name: STRING

creationTimestamp: INTEGER

version: STRING

description: STRING

Contribution

author: Author

creationTimestamp: INTEGER

comment: STRING

edits: ARRAY of Edit

Edit

type: STRING

oldValue: STRING

newValue: STRING

Author

osclId: INTEGER

externalId: STRING

type: STRING

username: STRING

ResultMetadata

page: INTEGER

itemsPerPage: INTEGER

totalElements: INTEGER

OsmComparisonStatistic

sequenceId: INTEGER

totalDetections: INTEGER

sameDetections: INTEGER

newDetections: INTEGER

changedDetections: INTEGER

unknowDetections: INTEGER

SequenceStatistic

sequenceId: INTEGER

totalDetections: INTEGER

OverallStatistic

totalDetections: INTEGER

newDetections: INTEGER

changedDetections: INTEGER

confirmedDetections: INTEGER

removedDetections: INTEGER

automaticDetections: INTEGER

manualDetections: INTEGER

The meaning of the various entities and their attributes is given below.

Sequence

A sequence represents a GPS track containing photos and detections. In the case of this component the sequence contains a list of detections. Its attributes are:

- id: the unique identifier of the sequence
- author: the user who had submitted a track containing the restriction
- device: information regarding the device used for recording the photo
- photos: an array of photos

Photo

A photo represents a photo recorded during a GPS track. Its attributes are:

- sequenceId, sequenceIndex: the unique identifier of the photo
- creationTimestamp: the timestamp when the photo was recorded
- point: the geo location of the photo
- width: the width of the photo
- height: the height of the photo
- heading: the improved heading of the photo
- processStatus: the process status of the photo; possible values: PENDING, PROCESSED, ERROR

Detection

A detection represents an object automatically or manually recognized in a photo. Its attributes are:

- id: the unique identifier of the detection
- sequenceId, sequenceIndex pair: the unique identifier of the photo to which the detection belongs
- creationTimestamp: the timestamp of the creation of the detection, given in Unix time format (the number of seconds elapsed since 1970-01-01 00:00:00 UTC, not counting leap seconds)
- latestChangeTimestamp: the timestamp of the latest change of the detection, given in Unix time format (the number of seconds elapsed since 1970-01-01 00:00:00 UTC, not counting leap seconds)
- point: the geo location of the detection
- sign: the sign category to which the detection belongs
- locationOnPhoto: the detection location on the photo
- confidenceLevel: a measure of confidence for the validity of this restriction
- validationStatus: the validation status of the detection; possible values: CONFIRMED, REMOVED, CHANGED, TO_BE_CHECKED, TO_BE_REVIEWED_LATER
- editStatus: the edit status of the detection; possible values: OPEN, FIXED, ALREADY_FIXED, BAD_SIGN, OTHER
- osmComparison: the OSM comparison result of the detection; possible values: SAME, NEW, CHANGED, UNKNOWN (matching failed)
- osmElement: the OSM element to which the detection was matched
- mode: the detection mode; possible values: AUTOMATIC, MANUAL
- author: the user who had created the detection or the user who had submitted a track containing the restriction
- algorithm: the algorithm with which the detection was found; this field is set only for AUTOMATIC detections
- componentValue: contains extra visible information of the detection; example: 50 mph for detections of type DISTANCE
- parentId: the unique identifier of the parent detection; this field is populated only for component detections (currently we have components only for SIGN_POST detections)

Point

A point represents a geographical position. Its attributes are:

- lat: the latitude, in the WGS84 standard
- lon: the longitude, in the WGS84 standard

Rectangle

A rectangle represents a detection position in a photo. Its attributes are:

- x: the x coordinate of the upper left corner of the rectangle
- y: the y coordinate of the upper left corner of the rectangle
- width: the height of the rectangle
- height: the width of the rectangle

OsmElement

An OsmElement represents the corresponding OpenStreetMap element to which the detection was matched. Its attributes are:

- osmId: the OSM identifier
- type: the OSM element type; possible values: NODE, WAY, RELATION
- members: an array of member identifiers; in some cases this field might be null (eg. detection is mapped for a specific way/node)

Sign

A sign represents the category to which a detection belongs. Its attributes are:

- name: the name of the sign
- internal_name: the internal name of the sign; this field is used only for mapping and it is not returned by any search/retrieve methods
- region: the region to which the sign is used
- type: the sub-category of the sign
- components: the list of sign components; currently we have components only for the SIGN_POST_GENERIC sign

Algorithm

An algorithm represents information related to the algorithm that was used for detecting the detection. Its attributes are:

- name: the name of the algorithm
- creationTimestamp: the time stamp of the creation of the algorithm
- version: the version of the algorithm
- description: the description of the algorithm (human readable form)

Contribution

A contribution represents an edit related to a detection submitted by a user. Its attributes are:

- author: the user who had edited the detection
- creationTimestamp: the timestamp of the contribution, given in Unix time format (the number of seconds elapsed since 1970-01-01 00:00:00 UTC, not counting leap seconds)
- comment: the text of the comment related to the detection
- edits: an array of edits; contains details regarding the edit operation;

Edit

An edit represents the change performed by the user. Its attributes are:

- type: the edit type; possible values: SIGN_CHANGE, VALIDATION_STATUS_CHANGE, EDIT_STATUS_CHANGE, LOCATION_ON_PHOTO_CHANGE
- oldValue: the old value of the property that was changed
- newValue: the new value of the property

Author

An author represents the user of a contribution. Its attributes are:

- externalId: the external identifier of the user
- type: the user type; possible values: OSM, FACEBOOK, GOOGLE

ResultMetadata

A ResultMetadata contains meta information about a result set. Its attributes are:

- page: the number of page that is returned in the result
- itemsPerPage: the number of returned items
- total: the total number of items

OsmComparisonStatistic

An OsmComparisonStatistic contains statistical information about a sequence detections grouped by the OsmComparison attribute. Its attributes are:

- sequenceId: the unique identifier of the sequence
- totalDetections: the total number of detections that was detected for a sequence
- sameDetection: the number of detections that already exists in OSM for a sequence
- newDetections: the number of new detections
- changedDetections: the number of detections that were changed
- unknownDetections: the number of detections which state is unknown

SequenceStatistic

A SequenceStatistic contains statistical information about the sequence detections. Its attributes are:

- sequenceId: the unique identifier of the sequence
- totalDetections: the total number of detections belonging to the sequence

OverallStatistic

An OverallStatistic contains overall statistical information about the detections. Its attributes are:

- totalDetections: the total number of detections
- toBeCheckedDetections: the total number of new detections; we consider a detection new if its validation status is TO_BE_CHECKED
- changedDetections: the total number of changed detections; we consider a detection changed if its validation status is CHANGED
- confirmedDetections: the total number of confirmed detections; we consider a detection confirmed if its validation status is CONFIRMED
- removedDetections: the total number of removed detections; we consider a detection removed if its validation status is REMOVED
- automaticDetections: the total number of automatically detected detections
- manualDetections: the total number of manually detected detections

SignStatistic

A SignStatistic contains overall statistical information about the detections. Its attributes are:

- signInternalName: the internal name of the sign
- totalDetections: the total number of detections
- toBeCheckedDetections: the total number of new detections; we consider a detection new if its validation status is TO_BE_CHECKED
- changedDetections: the total number of changed detections; we consider a detection changed if its validation status is CHANGED
- confirmedDetections: the total number of confirmed detections; we consider a detection confirmed if its validation status is CONFIRMED
- removedDetections: the total number of removed detections; we consider a detection removed if its validation status is REMOVED

Operations

This section describes the methods provided by this service.

Configuration parameters

The following parameters are accepted by all methods of the API. They are configuration parameters which don't influence the content of the response just its presentation. They are optional and if an incorrect value is received for any of them, it is ignored and the default value is used instead.

Parameter name	format
Description	specifies the format of the returned document; the accepted values are: • json: one-line JSON • fjson: formatted JSON
Type	string
Allowed values	json, fjson
Default value	json
Example	format=fjson`

Parameter name	locale_code
Description	a locale code, specifying the language in which the result should be returned
Type	string
Format	[lang[_cc]][.codeset][@modifier]], where: • lang is an ISO 639 language code; • cc is an ISO 3166-1 alpha-2 country code; • codeset is a character encoding;
Default value	en
Example	locale_code=de

Utility operations

These are general purpose operations, used mostly for getting information about the service. They are mostly used for debugging and monitoring, and can safely be ignored by most clients.

Method name	echo
Request type	GET
Description	displays information about the distribution and echoes a message to the client
Required parameters	-
Optional parameters	message
Response	returns the original message sent by the client and also some version information about the distribution: <pre>status: Status echo: implVersion: String specVersion: String buildNumber: String message: String</pre> attributes of object echo explained: • implVersion: the version of the server's distribution; • specVersion: the version of the API specification to which this distribution conforms; • buildNumber: the build number of this distribution; • message: the message received as parameter, echoed back to the client;
Example (GET)	https://apollo-api-v2.openstreetcam.org/apolloService/echo?format=fjson

Parameter name	message
Description	any message
Type	string
Example	message=hello

Search operations

Search detections

Method name	searchDetections
Request type	GET
Description	returns all the detections from the specified area
Required parameters	north, south, east, west
Optional parameters	externalId, authorType, date, osmComparisons, editStatuses, validationStatuses, signRegion, signTypes, signInternalNames, detectionModes
Response	a collection of detections: <pre> status: Status detections: ARRAY of Detection</pre>
Status codes	601, 610, 612
Example (GET)	https://apollo-api-v2.openstreetcam.org/apolloService/searchDetections?format=fjson&north=42.3365656&south=42.3340912&east=-83.0486101&west=-83.0539209

Example

```
{
  "status":{
    "apiCode":600,
    "apiMessage":"The request has been processed without incidents",
    "httpCode":200,
    "httpMessage":"Success"
  },
  "detections":[
    {
      "id":1700927,
      "sequenceId":12463,
      "sequenceIndex":1243,
      "point":{
        "lat":42.335216,
        "lon":-83.049536
      },
      "sign":{
        "name":"No left turn",
        "internalName":"TURN_RESTRICTION_LEFT_US",
        "iconName":"regulatory--no-left-turn--g2.svg",
        "region":"US",
        "type":"TURN_RESTRICTION"
      },
      "validationStatus":"TO_BE_CHECKED",
      "editStatus":"OPEN",
      "mode":"MANUAL"
    },
    .....
  ]
}
```

Parameter names	north, south, east, west
Description	the northern, southern, eastern and western limits respectively, defining a rectangular search area (bounding box)
Type	geographical coordinates in WGS84 standard
Format	rational numbers using the "." as a decimal separator
Allowed values	north and south – any value in the interval [-90,90]; north>= south east and west – any value in the interval [-180,180]; east>=west
Example	north=46.8&south=46.7&west=23.52&east=23.68

Parameter name	externalId
----------------	------------

Description	the external identifier of a user; this is a special argument and should be used together with the userType argument if present the method returns the detections that were identified in the data contributed by the user
Type	string
Example	externalId=1234

Parameter name	authorType
Description	the type of user; this is a special argument and should be used together with userId argument
Type	string
Allowed values	OSM, FACEBOOK, GOOGLE
Example	authorType=OSM

Parameter name	date
Description	the creation date of the detections
Type	string
Format	YYYY-MM-DD
Example	date=2017-07-20

Parameter name	osmComparisons
Description	filters the detections based on the OSM comparison property; by using this filters one can visualize only the detections that are missing from the map
Type	string
Format	comma separated array of values
Allowed values	SAME, NEW, CHANGED, UNKNOWN
Example	osmComparisons= SAME,NEW

Parameter name	editStatuses
Description	the edit status of the returned detections
Type	string
Format	comma separated array of values
Allowed values	OPEN, FIXED, ALREADY_FIXED, BAD_SIGN, OTHER

Example	editStatuses=OPEN
----------------	-------------------

Parameter name	validationStatuses
Description	the validation status of the returned detections
Type	string
Format	comma separated array of values
Allowed values	CONFIRMED, REMOVED, CHANGED, TO_BE_CHECKED
Example	validationStatuses=CONFIRMED

Parameter name	signRegion
Description	the region of the sign
Type	string
Allowed values	TBD
Example	region=US

Parameter name	signTypes
Description	the type of the returned detections
Type	string
Format	comma separated array of values
Allowed values	INTERSECTION_LANE_CONTROL, JUGHEAD, LANE_CONTROL, ONE_WAY, OTHER, ROAD_CLOSED, ROUNDABOUT, SIGN_POST, SPEED_LIMIT, STOP, TRAFFIC_SIGNAL, TURN_RESTRICTION
Example	signTypes=SPEED_LIMIT,TURN_RESTRICTION

Parameter name	signInternalNames
Description	the detected sign internal name
Type	string
Format	comma separated array of values
Example	signInternalNames=SPEED_LIMIT_35_US, TURN_RESTRICTION_LEFT_US

Parameter name	detectionModes
-----------------------	-----------------------

Description	the detection modes
Type	string
Format	comma separated array of values
Example	detectionModes=MANUAL,AUTOMATIC

Search sequence statistics

Method name	searchSequenceStatistics
Request type	GET
Description	returns statistical information for the given list of sequences
Required parameters	sequenceIds
Optional parameters	order
Response	a collection of detections: <pre> { status: Status statistics: ARRAY of SequenceStatistics } </pre>
Status codes	601, 612
Example (GET)	https://apollo-api-v2.openstreetcam.org/apolloService/searchSequenceStatistics?format=fjson&sequenceIds=12180%2C12079

Example	<pre> { "status": { "apiCode": 600, "apiMessage": "The request has been processed without incidents", "httpCode": 200, "httpMessage": "Success" }, "statistics": [{ "sequenceId": 12180, "totalDetections": 64 }, { "sequenceId": 12079, "totalDetections": 286 }] } </pre>
---------	---

Parameter name	sequenceIds
Description	a list of sequence identifiers
Type	positive integer (value between 1 and 9223372036854775807)
Allowed values	comma separated array of values
Example	sequenceIds=1,2,3

Parameter name	order
Description	the rule by which to order the items of the result set
Type	string
Allowed values	creationDate, creationDate DESC
Example	order=creationDate

List operations

List sequence statistics

Method name	listSequenceStatistics
Request type	GET
Description	returns statistics for all the sequences that contains detections
Required parameters	page, items
Optional parameters	-
Response	a collection of sequences: <pre> status: Status metadata: ResultMetadata statistics: ARRAY of OsmComparisonStatistic </pre>
Status codes	601, 612
Example (GET)	https://apollo-api-v2.openstreetcam.org/apolloService/listSequenceStatistics?format=fjson&page=1&items=1
Example response	<pre> { "status":{ "apiCode":600, "apiMessage":"The request has been processed without incidents", "httpCode":200, "httpMessage":"Success" }, "metadata":{ "hasNextPage":true }, "statistics":[{ "sameDetections":49, "newDetections":377, "changedDetections":0, "unknownDetections":0, "sequenceId":1139095, "totalDetections":426 }] } </pre>

Parameter name	page
Description	the page number; numbering starts from 1; see also parameter items
Type	a positive integer (value between 1 and 9223372036854775807)

Default value	1
Example	page=2

Parameter name	items
Description	the number of items per page; see also parameter page
Type	a positive integer (value between 0 and 1000)
Max value	1000
Example	items=2

List signs

Method name	listSigns
Request type	GET
Description	returns the list of supported signs
Required parameters	-
Optional parameters	-
Response	a collection of sequences: <pre> status: Status signs: ARRAY of Sign </pre>
Status codes	601
Example (GET)	https://apollo-api-v2.openstreetcam.org/apolloService/listSigns?format=fjson

Example response

```
{
  "status": {
    "apiCode": 600,
    "apiMessage": "The request has been processed without incidents",
    "httpCode": 200,
    "httpMessage": "Success"
  },
  "signs": [
    {
      "name": "One way small right",
      "internalName": "ONE_WAY_SMALL_RIGHT_US",
      "iconName": "regulatory--one-way-right--g3.svg",
      "region": "US",
      "type": "ONE_WAY"
    },
    .....
  ]
}
```

List photo detections

Method name	listPhotoDetections
Request type	GET
Description	lists the detections of the photos from the database
Required parameters	page, items
Optional parameters	allPhotoDetections, validationStatuses, editStatuses, osmComparisons, detectionModes, signRegion, signTypes, signInternalNames, algorithmNames, fromDate, toDate
Response	a collection of sequences: status: Status metadata: ResultMetadata photos: ARRAY of Photo
Status codes	601, 612
Example (GET)	https://apollo-api-v2.openstreetcam.org/apolloService/listPhotoDetections?format=fjson&page=1&items=1
Example response	{ "status": {

```
"apiCode": 600,
"apiMessage": "The request has been processed without incidents",
"httpCode": 200,
"httpMessage": "Success"
},
"metadata": {
  "hasNextPage": true
},
"photos": [
  {
    "sequenceId": 1,
    "sequenceIndex": 0,
    "detections": [
      {
        "id": 1675754,
        "sequenceId": 1,
        "sequenceIndex": 0,
        "point": {
          "lat": 46.773277,
          "lon": 23.592347
        },
        "sign": {
          "name": "No turns suspended",
          "internalName": "TURN_RESTRICTION_NO_TURNS_SUSPENDED_US",
          "iconName": "regulatory--no-turns--g1.svg",
          "region": "US",
          "type": "TURN_RESTRICTION"
        },
        "locationOnPhoto": {
          "x": 0.0,
          "y": 0.0,
          "width": 0.1,
          "height": 0.1
        },
        "validationStatus": "REMOVED",
        "editStatus": "OPEN",
        "mode": "MANUAL "
      },
      {
        "id": 2998515,
        "sequenceId": 1,
        "sequenceIndex": 0,
        "point": {
          "lat": 46.773277,
          "lon": 23.592347
        },
        "sign": {
          "name": "No turns ground",
          "internalName": "TURN_RESTRICTION_NO_TURNS_GROUND_US",
          "iconName": "regulatory--no-turns--g1.svg",
          "region": "US",
          "type": "TURN_RESTRICTION"
        }
      }
    ]
  }
]
```

```
},
"locationOnPhoto": {
  "x": 0.05745394178312213,
  "y": 0.43380197467115955,
  "width": 0.2672438800312446,
  "height": 0.227198161005156
},
"validationStatus": "CONFIRMED",
"editStatus": "OPEN",
"mode": "MANUAL"
}
]
```

	<pre>}] }</pre>
--	--------------------------

Parameter name	page
Description	the page number; numbering starts from 1; see also parameter items
Type	a positive integer (value between 1 and 9223372036854775807)
Default value	1
Example	page=2

Parameter name	items
Description	the number of items per page; see also parameter page
Type	a positive integer (value between 0 and 1000)
Max value	1000
Example	items=2

Parameter name	allPhotoDetections
Description	flag indicating if all the detections of the photos are returned or only the detections that matched the given filters
Type	string
Allowed values	true, false
Example	allPhotoDetections=true

Parameter name	validationStatuses
Description	the validation status of the returned detections
Type	string
Format	comma separated array of values
Allowed values	OPEN, FIXED, ALREADY_FIXED, BAD_SIGN, OTHER
Example	validationStatuses=OPEN

Parameter name	editStatuses
----------------	--------------

Description	the edit status of the returned detections
Type	string
Format	comma separated array of values
Allowed values	OPEN, FIXED, ALREADY_FIXED, BAD_SIGN, OTHER
Example	editStatuses=OPEN

Parameter name	osmComparisons
Description	filters the detections based on the OSM comparison property; by using this filters one can visualize only the detections that are missing from the map
Type	string
Format	comma separated array of values
Allowed values	SAME, NEW, CHANGED, UNKNOWN
Example	osmComparisons= SAME,NEW

Parameter name	signRegion
Description	the region of the sign
Type	string
Allowed values	TBD
Example	region=US

Parameter name	signCategories
Description	the categories of the returned detections
Type	string
Format	comma separated array of values
Allowed values	REGULATORY, COMPLEMENTARY, INFORMATION, WARNING
Example	signCategories=REGULATORY, COMPLEMENTARY

Parameter name	signTypes
Description	the type of the returned detections
Type	string

Format	comma separated array of values
Allowed values	OTHER, SIGN_POST, SPEED_LIMIT, TURN_RESTRICTION, ONE_WAY
Example	signTypes=SPEED_LIMIT,TURN_RESTRICTION

Parameter name	signInternalNames
Description	the detected sign internal name
Type	string
Format	comma separated array of values
Example	signInternalNames=SL_US_25, TURN_RESTRICTION_US_LEFT

Parameter name	detectionModes
Description	the detection modes
Type	string
Format	comma separated array of values
Example	detectionModes=MANUAL,AUTOMATIC

Parameter name	algorithmNames
Description	the algorithm names
Type	string
Format	comma separated array of values
Example	detectionModes=0.1,caffe_voc_fcn8s_v1

Parameter name	fromDate
Description	the update date of the detection
Type	string
Format	YYYY-MM-DD
Example	fromDate=2018-01-10

Parameter name	toDate
Description	the update date of the detection

Type	string
Format	YYYY-MM-DD
Example	fromDate=2018-01-15

Retrieve operations

Retrieve details of a detection

Method name	retrieveDetection
Request type	GET
Description	retrieves the detection identified by the given id
Required parameters	id
Optional parameters	-
Response	<i>status: Status</i> <i>detection: Detection</i>
Status codes	601, 610, 612
Example (GET)	https://apollo-api-v2.openstreetcam.org/apolloService/retrieveDetection?format=fjson&id=1675754

Example response	<pre>{ "status":{ "apiCode":600, "apiMessage":"The request has been processed without incidents", "httpCode":200, "httpMessage":"Success" }, "detection":{ "id":1204350, "sequenceId":9716, "sequenceIndex":335, "creationTimestamp":1480429136, "latestChangeTimestamp":1518010477, "point":{ "lat":43.79564, "lon":-79.446429 }, "sign":{ "name":"Give way", "internalName":"YIELD_US", "iconName":"regulatory--yield--us.png", "region":"US", "type":"OTHER" }, "locationOnPhoto":{ "x":0.7347, "y":0.5685, "width":0.0358, "height":0.0483 }, "confidenceLevel":0.0, "validationStatus":"TO_BE_CHECKED", "editStatus":"OPEN", "mode":"MANUAL", "author":{ "oscId":3107, "username":"istvan_bardos", "externalId":"2740857", "type":"OSM" }, "algorithm":{ "name":"caffe_voc_fcn8s_v1", "version":"1.0.0", "description":"Caffe voc algorithm v1" } } }</pre>
------------------	--

Parameter name	id
----------------	----

Description	the identifier of the detection
Type	a positive integer (value between 1 and 9223372036854775807)
Example	id=1234

Retrieve detections of a sequence

Method name	retrieveSequenceDetections
Request type	GET
Description	retrieves the detections of the given sequence from the database, ordered by sequenceIndex
Required parameters	sequenceId
Optional parameters	osmComparisons
Response	<i>status:Status</i> <i>detection: ARRAY of Detection</i>
Status codes	601, 610, 612
Example (GET)	https://apollo-api-v2.openstreetcam.org/apolloService/retrieveSequenceDetections?format=fjson&sequenceId=12180

Example response

```
{
  "status":{
    "apiCode":600,
    "apiMessage":"The request has been processed without incidents",
    "httpCode":200,
    "httpMessage":"Success"
  },
  "detections":[
    {
      "id":5339,
      "sequenceId":12180,
      "sequenceIndex":81,
      "point":{
        "lat":42.53226,
        "lon":-83.192103
      },
      "sign":{
        "name":"Speed limit 20 mph",
        "internalName":"SPEED_LIMIT_20_US",
        "iconName":"regulatory--maximum-speed-limit-20--g3.svg",
        "region":"US",
        "type":"SPEED_LIMIT"
      },
      "locationOnPhoto":{
        "x":0.204,
        "y":0.4943,
        "width":0.0371,
        "height":0.0421
      },
      "validationStatus":"REMOVED",
      "editStatus":"OPEN",
      "mode":"AUTOMATIC"
    },
    ...
  ]
}
```

Parameter name	sequenceId
Description	the identifier of the sequence
Type	a positive integer (value between 1 and 9223372036854775807)
Example	sequenceId=123566

Parameter name	osmComparisons
Description	filters the detections based on the OSM comparison property; by using this filters one can visualize only the detections that are missing from the map

Type	string
Format	comma separated array of values
Allowed values	SAME, NEW, CHANGED, UNKNOWN
Example	osmComparisons= SAME,NEW

Retrieve detections of a photo

Method name	retrievePhotoDetections
Request type	GET
Description	retrieves the detections of the given photo
Required parameters	sequenceId, sequenceIndex
Optional parameters	-
Response	<i>status:Status</i> <i>detections: ARRAY of Detection</i>
Status codes	601, 610, 612
Example (GET)	https://apollo-api-v2.openstreetcam.org/apolloService/retrievePhotoDetections?format=fjson&sequenceId=1583&sequenceIndex=5358

Example response

```
{
  "status": {
    "apiCode": 600,
    "apiMessage": "The request has been processed without incidents",
    "httpCode": 200,
    "httpMessage": "Success"
  },
  "detections": [
    {
      "id": 1658021,
      "sequenceId": 1583,
      "sequenceIndex": 5358,
      "point": {
        "lat": 42.337975,
        "lon": -83.166907
      },
      "sign": {
        "name": "Stop",
        "internalName": "STOP_US",
        "iconName": "regulatory--stop--g1.svg",
        "region": "US",
        "category": "REGULATORY",
        "type": "STOP"
      },
      "locationOnPhoto": {
        "x": 0.625,
        "y": 0.4264,
        "width": 0.0428,
        "height": 0.0607
      },
      "validationStatus": "TO_BE_CHECKED",
      "editStatus": "OPEN",
      "mode": "MANUAL"
    }
  ]
}
```

Parameter name	sequenceId
Description	the identifier of the sequence; the sequenceId and sequenceIndex combination uniquely identifies a photo
Type	a positive integer (value between 1 and 9223372036854775807)
Example	sequenceId=123566

Parameter name	sequenceIndex
Description	the index of the photo from the given sequence; the sequenceId and sequenceIndex combination uniquely identifies a photo

Type	a positive integer (value between 0 and 9223372036854775807)
Example	sequenceIndex=123566

Retrieve contributions of a detection

Method name	retrieveContributions
Request type	GET
Description	retrieves the contributions of the given detection in descending order based on the creationTimestamp of the contribution (newest contributions are returned first)
Required parameters	detectionId
Optional parameters	-
Response	<i>status:Status</i> <i>contributions: ARRAY of Contribution</i>
Status codes	601, 610, 612
Example (GET)	https://apollo-api-v2.openstreetcam.org/apolloService/retrieveContributions?format=fjson&detectionId=3319232

Example response	<pre> { "status": { "apiCode": 600, "apiMessage": "The request has been processed without incidents", "httpCode": 200, "httpMessage": "Success" }, "contributions": [{ "author": { "oscId": 2400, "username": "arming_telenav", "externalId": "4827171", "type": "OSM" }, "creationTimestamp": 1523103384, "edits": [{ "type": "EDIT_STATUS_CHANGE", "oldValue": "OPEN", "newValue": "FIXED" }] }] } </pre>
------------------	---

Parameter name	detectionId
Description	the identifier of the detection
Type	a positive integer (value between 1 and 9223372036854775807)
Example	detectionId =123

Retrieve overall statistics

Method name	retrieveOverallStatistic
Request type	GET
Description	Retrieves an overall statistics of the detections, related to the validation status and detection modes.
Required parameters	-

Optional parameters	signInternalNames
Response	status:Status overallStatistic: OverallStatistic
Status codes	601, 612
Example (GET)	https://apollo-api-v2.openstreetcam.org/apolloService/retrieveOverallStatistic?format=fjson&signInternalNames=YIELD_US,EXCLUSION_DO_NOT_ENTER_US
Example response	<pre>{ "status":{ "apiCode":600, "apiMessage":"The request has been processed without incidents", "httpCode":200, "httpMessage":"Success" }, "overallStatistic":{ "totalDetections":15521, "toBeCheckedDetections":13826, "changedDetections":1, "confirmedDetections":1247, "removedDetection":447, "automaticDetections":15132, "manualDetections":389 } }</pre>

Parameter name	signInternalNames
Description	the detected sign internal name; for the corect values see the listSigns method
Type	string
Format	comma separated array of values
Example	signInternalNames=SPEED_LIMIT_55_US,TURN_RESTRICTION_U_TURN_LEFT_US

Edit operations

Update detection

Method name	updateDetection
Request type	POST
Description	updates an existing detection with the specified attributes
Required parameters	<i>specified in the request body</i>
Optional parameters	<i>specified in the request body</i>
Response	status:Status
Status codes	670, 671, 613

Body	
Description	all parameters, both required and optional, grouped as a single JSON object
Type	text
Content type	application/json

Schema definition

Root

contribution: Contribution (REQUIRED)

detection: Detection (REQUIRED)

Contribution

author: Author (REQUIRED)

comment: STRING (OPTIONAL)

Detection

id: INTEGER (REQUIRED)

sign: STRING (OPTIONAL)*

locationOnPhoto: Rectangle (OPTIONAL)*

validationStatus: STRING (OPTIONAL)*

editStatus: STRING (OPTIONAL)*

componentValue: STRING (OPTIONAL)

parentId: INTEGER (OPTIONAL)

Author

osclId: INTEGER (OPTIONAL)

externalId: STRING (REQUIRED)

type: STRING (REQUIRED)

userName: STRING (OPTIONAL)

Attribute descriptions	• Contribution.author: the user who is changing the detection • Contribution.comment: additional information regarding to the modification • Detection - at least one of the marked field must be present in the request • Detection.id: the identifier of the detection • Detection.sign: the category to which the detection belongs; only the signInternalName is taken into consideration if provided other sign attributes will be ignored • Detection.locationOnPhoto: the new detection location on the photo • Detection.validationStatus: the new validation status • Detection.editStatus: the new edit status • Detection.componentValue: extra visible information of the detection • Detection.parentId: the identifier of the parent detection; should be used only when a component detection is created • Author.externalId: the external identifier of the user • Author.type: the user type • Author.oscId: the OSC identifier of the author • Author.userName: the user name of the author
Example	<pre>{ "contribution":{ "author":{ "externalId":12343, "type":"OSM" }, "coment":"test comment" }, "detection":{ "id":12345, "validationStatus":"CONFIRMED", "sign":{ "internalName":"ADVANCE_INTER_LEFT_COMP_US" } } }</pre>