**TECHNICAL SPECIFICATION**

to adapt the functionality of automatic uploading (exporting) of frames with detected persons, vehicle registration plates, as well as additional information about detected vehicles together with information about the time and place (number of the video surveillance system of the monitoring system) of detection

# 

# **5. INTERACTION REQUIREMENTS**

Event information and event snapshots are sent separately. Three messages are sent to the system for one event:

* message with information about the event;
* message with a snapshot of the full frame;
* message with snapshot of face\license plate(vehicle).

The comparison of an event and its snapshots is performed on the receiving side.

5.1.4. The information is transmitted in the formatJSON. Message format for sending events about detection of a person's face and a vehicle

\specified in paragraphs 5.2 and 5.3 respectively.

5.1.5. Images are sent in the same size, format and quality as they are stored in the Software databases (the resolution of the original frame does not change).

5.1.6. Events are sent as events are received by the Software.

5.1.7. In case of unavailability of the receiving party's software, unsent event data and snapshots are stored on the Software side for 30 minutes with the possibility of automatic re-sending.after the specified period has expired.If the receiving party's software remains unavailable during this time, the data is no longer available for sending.

**5.2. Format of messages about detection of a person's face**

5.2.1. When generations events of detection of a person's face in the Software, the data of this event is sent in accordance with the format described in paragraphs 5.2.2 - 5.2.6 to the address that will be provided in accordance with paragraph 5.1.2. Sending of data from the Software is carried out via the HTTP 1.1 protocol over an organized closed channel with trust, by means of 3 POST requests (message with information about the event, message with a full-screen snapshot, message with a snapshot of the face), unless otherwise established by the documentation for the RSMB information security system.

5.2.2. POST request with a message with information about the event consists of a json body. Example of a json body:

**Headline**

Content-Type: application/json

**Body**

{

"id": "4319:1687267038192:75990379422576867",

"start\_time": 1687267038192,

"latitude": 13.92193690411079,

"longitude": 17.616875618696216,

"channel\_id": 1001,

"address": {

"country": "USA",

"region": "123 region",

"county": "asd district",

"city": "asd",

"district": "Beavers",

"street": "Highway street",

"place\_info": "32"

},

"snapshots": [{

"id": "79054025255fb1a26e4bc422aef54eb4",

"type": "FULLSCREEN"

},

{

"id": "79054025255fb1a26e4bc422aef54eb5",

"type": "THUMBNAIL"

}]

}

5.2.3. The description of the parameters used is given in Table 5.1.

| Parameter | Field type | Required parameter | Comments |
| --- | --- | --- | --- |
| id | string | Yes | unique event identifier and event start time in millisec format |
| startTime | integer | Yes | date and time of the event start  Value in millisec format |
| latitude | number($double) | Yes | Latitude of the camera installation location on the map.  maximum: 90  minimum: -90 |
| longitude | number($double) | Yes | Longitude of the camera installation location on the map.  maximum: 180  minimum: -180 |
| channelId | integer | Yes | camera id |
| address |  | Yes | Composite field. Address |
| country | string | Yes | Side |
| region | string | No | Region/area |
| county | string | No | Region of the region |
| city | string | No | City/village |
| district | string | No | City area |
| street | string | No | Street/dom |
| place\_info | string | No | House |
| snapshots |  | Yes | Composite field. Snapshot |
| id | string | Yes | unique snapshot identifier |
| type | string | Yes | Snapshot Type:  FULLSCREEN - full frame  THUMBNAIL - part of the frame with a face |

Table 5.1.

5.2.4. In response, the third-party system returns an HTTP response with a status code in accordance with HTTP 1.1 standards. In case of a successful response, the status code is 200 OK without HTTP BODY.

5.2.5. POST request with a message with a snapshot (fullscreen or face snapshot) consists of a json body. Example json body:

**Headline**

Content-Type: application/json

**Body**

{

"id": "79054025255fb1a26e4bc422aef54eb4",

"snapshot": ""

}

5.2.6.The description of the parameters used is given in Table 5.2.

Table 5.2.

| Parameter | Field type | Required parameter | Comments |
| --- | --- | --- | --- |
| id | string | Yes | unique snapshot identifier |
| snapshot | base64 | Yes | event snapshot |

5.2.7. In response, the third-party system returns an HTTP response with a status code in accordance with HTTP 1.1 standards. In case of a successful response, the status code is 200 OK without HTTP BODY.

**5.3. Vehicle Detection Message Format**

5.3.1 When an event about detection of a vehicle license plate or vehicle is generated in the Software, the data of this event is sent in accordance with the format described in paragraphs 5.3.2 - 5.3.6 to the address that will be provided in accordance with paragraph 5.1.2. Data is sent from the Software via the HTTP 1.1 protocol over an organized closed channel with trust, by means of 3 POST requests (message with information about the event, message with a full-screen snapshot, message with a snapshot of the license plate (vehicle), unless otherwise established by the documentation for the RSMB information security system.

5.3.2. POST request with a message with information about the event consists of a json body. Example of a json body:

**Headline**

Content-Type: application/json

**Body**

{

"id": "4319:1687267038192:75990379422576867",

"start\_time": 1687267038192,

"latitude": 13.92193690411079,

"longitude": 17.616875618696216,

"channel\_id": 1001,

"address": {

"country": "USA",

"region": "123 region",

“county”: "asd district",

"city": "asd",

"district": "Beavers",

"street": "Highway street",

"place\_info": "32"

}

"snapshots": [{

"id": "79054025255fb1a26e4bc422aef54eb4",

"type": "FULLSCREEN"

},

{

"id": "79054025255fb1a26e4bc422aef54eb5",

"type": "THUMBNAIL"

}],

"plate": {

"number": "1111OM7",

"state": "DE"

}

}

5.3.3. The description of the parameters used is given in Table 5.3.

Table 5.3.

| Parameter | Field type | Required parameter | Comments |
| --- | --- | --- | --- |
| id | string | Yes | unique event identifier and event start time in millisec format |
| startTime | integer | Yes | date and time of the event start  Value in millisec format |
| latitude | number($double) | Yes | Latitude of the camera installation location on the map.  maximum: 90  minimum: -90 |
| longitude | number($double) | Yes | Longitude of the camera installation location on the map.  maximum: 180  minimum: -180 |
| channelId | integer | Yes | camera id |
| address |  | Yes | Composite field. Address |
| country | string | Yes | Side |
| region | string | No | Region/area |
| county | string | No | Region of the region |
| city | string | No | City/village |
| district | string | No | City area |
| street | string | No | Street/dom |
| place\_info | string | No | House |
| snapshots |  | Yes | Composite field. Snapshot |
| id | string | Yes | unique snapshot identifier |
| type | string | Yes | Snapshot type:  FULLSCREEN - full frame  THUMBNAIL - part of the frame with a license plate (transport) |
| plate |  | No | Composite field.  Absent if no license plate is detected for the vehicle |
| number |  | No | Vehicle registration plate |
| state |  | No | The state to which the license plate belongs  Values:  "AT" - Austria  "AZ" - Azerbaijan  "BY" - Belarus  "BG" - Bulgaria  "UK" - Great Britain  "DE" - Germany  "IE" - Ireland  "ES" - Spain  "KZ" - Kazakhstan  "KG" - Import  "LV" - Latvia  "LT" - Lithuania  "MD" - Moldova  "MN" means Mongolia  "PL" - Poland  "RU" - Russia  "RO" - Romania  "US" - USA  "TR" - Turkey  "UZ" - Uzbekistan  "UA" - Ukraine  "UNKNOWN" -Unknown |

5.3.4. In response, the third-party system returns an HTTP response with a status code in accordance with HTTP 1.1 standards. In case of a successful response, the status code is 200 OK without HTTP BODY.

5.3.5. POST request with a message with a snapshot (fullscreen or license plate/vehicle snapshot) consists of a json body. Example of a json body:

**Headline**

Content-Type: application/json

**Body**

{

"id": "79054025255fb1a26e4bc422aef54eb4",

"snapshot": ""

}

5.3.6. The description of the parameters used is given in Table 5.4.

Table 5.4.

| Parameter | Field type | Required parameter | Comments |
| --- | --- | --- | --- |
| id | string | Yes | unique snapshot identifier |
| snapshot | base64 | Yes | event snapshot |

5.3.7. In response, the third-party system returns an HTTP response with a status code in accordance with HTTP 1.1 standards. In case of a successful response, the status code is 200 OK without HTTP BODY.