# NNOE JSON Interface EN

**Version 0.5**

## Changelog

| Version | Date | Author | Changes |
|---|---|---|---|
| 0.5 | 23.12.2019 | Kaufmann Markus | Latest update, added some examples |
| 0.4 | 06.12.2019 | Kaufmann Markus | Input implemented |
| 0.3b | 25.10.2019 | Kaufmann Markus | english translation |
| 0.3 | 23.10.2019 | Kaufmann Markus | multiple adaptions, tables removed |
| 0.2 | 18.09.2019 | Kaufmann Markus | Markdown update, reformatting |
| 0.1b | 17.09.2019 | Hopfgartner Alexander | Markdown created |
| 0.1 | 17.09.2019 | Kaufmann Markus | first draft, based on primary talks |

## Index

## Introduction

This document describes the JSON Interface of Notruf Niederösterreich.

### Common Terms

#### *Sender*

The one who sends the initial message. If control center A sends data to control center B, then control center A is the sender.

#### *Receiver*

The endpoint which answers the primary request. If control center A sends data to control center b, then control center B is the receiver.

### Technical implementation

Every partner has a server for sending and receiving messages of other partners. The messages are pushed, pull is not implemented.

If an endpoint is unreachable, backup communication ways, for example telephones, have to be used. These should be defined while setting up the interfaces.

The datas are submitted in JSON Format.

The receiver is responsible for the translation of individual formats (Event codes, address data etc.). That means, that if Control Center A submits an event to Control Center B, Control Center B must convert the information into a format that fits their needs and vice versa.

Event numbers are also submitted in the format of the sender, a mapping has to be done by the receiver. It is good practice to save a special table for the translation.

Only one dataset is submitted per Request.

### *Authentication*

A JSON-Webtoken is used to authenticate the submission, additional security implementations can be appointed individually if needed.

For example a VPN-Tunnel could be used, but at least a concrete IP address should be used. Also OAuth could be implemented, if needed.

### *Validation*

In some fields, concrete values are used. Those have to be appointed between the Control centers. The latest valid list is added in the annex. A validation for those values should be implemented.

### *Request-ID*

To make it possible to follow up a request, every request needs a unique Request ID. This should be implemented in the UUID-Format. If UUID-Format is not possible, other unique values are also possible of course.

This Request-ID is also resent in the response, and makes a matching possible.

### *Date/Time*

If not defined in another way, always the CET/MEZ Time is used (CEST/MESZ in summertime). The time matches the time of the sender, and has to be converted to local needs.

Date and Time is submitted in DD.MM.YYYY HH24:MI:SS format.

### *GPS Coordinates*

WGS84 Latitude/Longitude are formatted in decimal, as exact as possible. A „.“ (Dot) is used for seperation. For Example: 48.19744, 15.63302

### *Telephone numbers*

Phone numbers are sent the international E.123 Format, without spaces.

For Example: +4312066012345

### *Adressen*

The following combinations are valid:

- GPSX/Y
- GPSX/Y + Adress description
- Adress description (Street, House number, Zip code usw. in any combination)
- Austrian BEV ID

An „info“ can always be added additionaly and is used for further description of the address. Street kilometers or access information could be submitted here for example.

### *Empty Values*

If no value is available for a field, it could be

- a space
- set to NULL
- Not in the request

Every one of these variants is valid.

## Interfaces

### Responses

Every Request is followed by a Response from the receiver. This is made to see if a) the submission was successful and b) if an error raised and the request could not be handled.

The response consists of 4 elements: Request-ID, status code, status text and free text. Additional fields could vary, depending on the request.

- **requestid** - ID of the request
- **statuscode** - Statuscode of the result, see Annex(OK100)
- **statustext** - explaining text, see Annex
- **freetext** - free text

***Request-ID***

The Request-ID is used to identify the response to the request.

***Statuscode***

The Statuscode is a combination out of 2 Letters and 3 Digits. Together this builds a unique code.

If the letters are „OK“, it is supposed that the submission was successful and well formatted. With „ER“ comes some kind of error, which says that the complete dataset, or parts of it, couldn‘t be computed.

The following Digits are unique through the system, and are categorized for their usage (Not used areas are reserved for later usage):

- 000-099: Systemstatus and test
- 100-199: event submission
- 300-399: resource
- 400-499: eventstatus

***freetext***

Completely free content. In the language of the sender or english.

***example***

follows

**Resources**

It is possible to interchange information about resources (vehicles, ambulance cars, helicopters etc.) A background routine checks which resources should be sent to which endpoint. A area and filter (based on the resource type for example) should be defined, so only the relevant ones are sent.

possible Values:

- **callsign** - mandatory, callsign of the resource, unique per control center (99/099)
- **status** - mandatory, the current state of the resource
  - **status** - mandatory, the state itself (create, deactivate, available_at_station)
  - **statustime** - mandatory, timestamp of the latest state change (17.09.2019 11:28:00)
- **type** - mandatory, resource type (NAH, NAW, KTW-B usw.)
- **cureventnum** - optional, currently dispatched event– individual format per Control center (190812345)
- **contact** - optional, contact information to resource
  - **tetraissi** - optional, tetra number
  - **tetragroup** - optional, current Tetra group
  - **phonenumber** - optional, phone number of the resource (+4366412345678)
- **location** - mandatory, current position of the resource
  - **gpsx** - mandatory, X-Coordinate
  - **gpsy** - mandatory, Y-Coordinate

**Creating new resources**

To ensure, that all Control Centers have knowledge about the available resource in the defined areas, resources can be submitted to them. This is done via the status "create". With "deactivate" the resource is marked as not available for the partners or out of area.

**Eventdata**

***Eventsubmission***

All relevant event information is submitted in this field. If there are updates for an event, all information is submitted again, the event number remains the same of course.

- **eventnum** - mandatory, Event number from the senders CAD System. (123456789)
- **location** - mandatory, Event location
  - **gpsx** - depending, X Coordinate in WGS84 Format
  - **gpsy** - depending, Y Coordinate in WGS84 Format
  - **quality** - optional, exactness of the Adress (street, adress, pos)
  - **info** - optional, additional information to the address
  - **zipcode** - depending, zipcode
  - **city** - depending, city
  - **street** - depending, street

    - **hnr** - depending, house number
    - **floor** - optional, floor
    - **flat** - optional, flat
    - **door** - optional, door number
    - **bevid** - depending, ID in austrian BEV System
    - **country** - optional, shortcountrycode (AT, CZ, etc.)
- **destination** - mandatory, Transportdestination
    - **gpsx** - depending, X Coordinate in WGS84 Format
    - **gpsy** - depending, Y Coordinate in WGS84 Format
    - **quality** - optional, exactness of the Adress (street, adress, pos)
    - **info** - optional, additional information to the address
    - **zipcode** - depending, zipcode
    - **city** - depending, city
    - **street** - depending, street
    - **hnr** - depending, house number
    - **floor** - optional, floor
    - **flat** - optional, flat
    - **door** - optional, door number
    - **bevid** - depending, ID in austrian BEV System
    - **country** - optional, shortcountrycode (AT, CZ, etc.)
- **patient** - optional, Data of the patient
    - **name** - optional, family name
    - **firstname** - optional, first name
    - **insurancenumber** - optional, insurancenumber (including birthdate in austria)
    - **birthdate** - optional, full date of birth
    - **age** - optional, if no birthdate is available
    - **gender** - optional, male or female
    - **notice** - optional, freetext for closer description
- **eventtype** - mandatory
    - **description** - optional, more detailed description
    - **code** - mandatory, event code itself (TEST, RD-32B1)
    - **indication** - mandatory, supply level - describes if a doctor or helicopter etc. is needed
- **caller** - mandatory, data of the caller
    - **callername** - mandatory, name of the caller
    - **callernumber** - optional, phone number of the caller
    - **callingtype** - mandatory, how was the event primarily submitted (phone, ecall, mobile, app)
- **otherinfo** - optional, free text for more details
- **time** - mandatory, multiple timestamps

- **eventcall** - mandatory, time of call
- **ontime** - optional, time at location
- **ontimetype** - optional, shows if the "ontime" is the time at the location or destination (location|destination)

- **eventtransfer** - mandatory, should the event be completely transferred or is just help needed

***Interfacestate***

A possibility to check if the interface is currently available and works correctly should be implemented. The following flags could appear:

- **Status** - (ER000 - General error, ER001 - Interface down, OK000 - Interface works and is available, OK100 - planned down)
- **description** - description of the problem

***Request***

Values:

- **Statusrequest** - Always true (true)

## Annex

### Examples

***Complete JSON with Exampledata***

` `{ "interface": { "requestid": "12cc00c4-91b6-484f-8570-a7574bcb31c3", "resource": { "callsign": "99/099", "status": { "status": "create", "statustime": "18.09.2019 10:41:13" }, "type": "NAH", "cureventnum": "19081234", "contact": { "tetraissi": "123456", "tetragroup": "NNOE-06", "phonenumber": "+4366412345678" }, "location": { "gpsx": "47.849191", "gpsy": "16.527825" } }, } "event": { "eventnum": "19081234", "location": { "gpsx": "47.849191", "gpsy": "16.527825", "quality": "adress", "info": "Testlocation", "zipcode": "9955", "city": "Testort", "street": "Teststrasse", "hnr": "1", "floor": "", "flat": "", "door": "" }, "destination": { "gpsx": "47.849191", "gpsy": "16.527825", "quality": "adress", "info": "Chirurgie Ambulanz", "zipcode": "9955", "city": "Testort", "street": "Krankenhausstrasse", "hnr": "1", "floor": "", "flat": "", "door": "" }, "patient": { "name": "Mustermann", "firstname": "Max", "insurancenumber": "1234112277" }, "caller": { "name": "Maxine Mustermann", "callingtimestamp": "18.09.2019

08:41:13", "callernumber": "+4366411223344", "callingtype": "phone" }, "otherinfo": "Testeinsatz", "time": { "eventcall": "18.09.2019 08:41:13", "ontime": "18.09.2019 08:41:13", "ontimetype": "location" }, "eventtransfer": true } } }

}
#### Response
Example Response to a request
{ "interface": { "response": { "requestid": "12cc00c4-91b6-484f-8570-a7574bcb31c3 ", "statuscode": "OK100", "statustext": "Eventsubmit successfull" } } }``

**Predefined Values**

***Resources - Types***

- **HEMS** - Ambulancehelicopter (1 doctor, 1 paramedic, 1 pilot, AIR-MED)
- **ALSB** - Ambulance, for transport (2 paramedics, big car, EN1789 Type B)
- **ALSC** - Ambulance, for transport (1 doctor, 2 paramedics, big car, EN1789 Type C)
- **MEDRRT** - Emergency pyhisician rapid response car - team - no possibility for transportation (1 doctor, 1 paramedic, small car)
- **BLSA** - Ambulance - primarily for non-emergency (2 paramedics, small car, EN1789 Type A)
- **AUXPTA** - Auxiliary Ambulance - for non-emergency only (1 paramedic, very small car)

***Location/Destination - quality***

- **adress** - Adresse
- **pos** - X/Y coordinate
- **street** - street
- **cross** - cross

***caller - Callingtype***

- **phone** - via phone
- **app** - via mobile app
- **ecall** - created with E-Call system
- **radio** - Alarm via radio
- **other**

***time - ontimetype***

- **location** - time at location
- **destination** - time at destination

***response - Statuscodes***

**Value**|**Description** ---|---

follows with implementation

**complete Statelist**

***Resources***

| Value | Description |
|---|---|
| available_at_station | available at home base |
| available_via_radio | available, on the way |
| reserved | reserved for an event, at home base |
| reserved_via_radio | reserved for event, on the way |
| to_location | on the way to the location |
| at_location | arrived at the location |
| to_destination | on the way to destination (transporting) |
| at_destination | at destination (Hospital or Residence, depending on type of transport) |
| not_available | not available... |
| on_request | available, but with extra alarm time |

*Resourcecreation states*

| Value | Description |
|---|---|
| create | a new Resource is created, starts to send state etc. |
| deactivate | the Resource is not anymore in the system from now on |

***Events***

| Value | Description |
|---|---|
| call | caller is still on the phone |
| on_disposition | available for disposition |
| reserved | reserved |
| alarmed | already alarmed |
| closed | event is regularly finished |

| | |
|---|---|
| cancelled | event was cancelled |