ALERTS MANAGEMENT API

Documentation

ver. 7.0

(July 2023)

Národní organizace pro ověřování pravosti léčiv, z.s., IBC Pobřežní 620/3, Karlín, 186 00 Praha 8
IČ: 05851742, www.czmvo.cz

CONTENT

1. INTRODUCTION

The **Alert Management System** (AMS) is an extension of the **National Medicines Verification System** (NMVS). The AMS facilitates the management of alerts (suspected counterfeits). The AMS was created and is managed by the Czech Medicine Verification Organization (NOOL, z.s.).

The CZMVO.CZ API is a simple **REST API** that allows the connection of a custom MAH or end-user alert management system to the NOOL alert management system. This integration allows automation of alert handling activities, both on the MAH and end-user (pharmacy, distributor) side.
For users who do not have their own alert management system, or whose company policy does not allow integration via API, a web interface with equivalent functionality is available.

A description of the web interface is provided in separate user guides, separately for MAHs and separately for End-users. These manuals also describe in more detail the system logic and support for intercommunication between the MAH, the End user and CZMVO (NOOL).

ACCESS TO THE API AND WEB INTERFACE

Access points:

Environment	**Web interface**	**API**
Production	https://portal.czmvo.cz/	https://api.czmvo.cz/
Test	https://sandbox.czmvo.cz/	https://api.czmvo.cz/t/
Development	https://beta.czmvo.cz/	https://betaapi.czmvo.cz/

API 2.0 OAuth2 - authorization lines:

Prostředí	**Link**
Production - PROD	https://api.czmvo.cz/auth/token/
Test - IQE	https://api.czmvo.cz/t/auth/token/
Development - ITE	https://betaapi.czmvo.cz/auth/token/

A **copy of the production data** is **regularly uploaded to the test and development environment every night**. Testing can therefore be done on real user data. Changes to the test and development environment do not affect the data in the production database, so it can be changed freely. However, every night the data is overwritten, which must be taken into account in testing and development.

The test environment is used to test and develop software for the existing production environment (i.e. it is functionally identical to the production environment).

The development environment is used to test and develop application software for a forthcoming version of AMS that is yet to be deployed to the production environment.

The login details for the web interface and API 1.0 are:

a) **Login name and password** which can be generated **upon request** at email address registrace@czmvo.cz
b) One-time (only for end users) the **alert ID** (UPRC) can be used as a **login** and the **location ID** (location) as a **password**.
c) One-time (for end users only) **location ID** (location) can be used as **login** and the same location ID as **password** to verify if the product code (or batch) is in the MH exception list (it has no other authority).
d) **Development account** for end user IT software suppliers. The account allows access to test and/or development environment only (to the both API and web portal). They can select End user for which are developing software once logged in into system (if End user is not selected, any alert will not be visible). Selection of End user can be performed any time in web portal via select box beside language selection.

The API 2.0 access data are:

a) **Client ID and Secret ID** - access credentials for OAuth2. **They are generated exclusively in the web interface** and are used to obtain **an access token**.
Note: **The user must first be registered in the web interface before they can generate the access credentials.**
The "**Client ID**" and "**Secret ID**" keys **are different for different environments** (production/test), so it is **necessary to generate them in the corresponding version of the web interface**.
b) To obtain a token (for End users only), the **alert ID** (UPRC) can be used as "Client ID" and the **location ID** as "Secret ID". Such a user can **only** access **one specific Alert**, and cannot perform bulk operations and operations over groups.
c) To obtain a token (for End users only), the **location ID** can be used as "Client ID" and the **location ID** as "Secret ID". Such a user **has no access to any alerts** (or peeking). The accesses can only be used to verify that the **product code (or batch) is in the Ministry of Health exception list.**

EMAIL NOTIFICATIONS

Depending on its settings, the system uses a lot **of automatic e-mail notifications**. Automatic notifications are used during automatic status escalations, when closing alerts by the end user or NOOL (confirmation of the requested change). The system also generates an e-mail notification every time a new message/file is inserted.
The notification is sent in plain text and in **UTF-8** encoding. For easier automatic processing, a **footer** is also attached to the message, which is intended for automatic processing and is in the form:
EVENT:NEWMESSAGE*ID:20

Individual fields may be subject of change in the future but format of the message will keep the structure:
VARIABLE1:VALUE1*VARIABLE2:VALUE2* … *VARIABLEX:VALUEX
Possible variables and theirs's values are following:
EVENT – value: currently only NEWMESSAGE
ID – value: message ID for the new message (EVENT: NEWMESSAGE)

2. API DESCRIPTION

2.1. API VERSION

Starting with AMS version 6.0, API versioning is introduced. Everything before this release is considered API 1.0. **API 1.0 becomes an unsupported and unrecommended version** with the release of release 6.0 and **will be removed** from the AMS system no later than **October 2023**.

Version status headers are present in all responses from the system:

Example:

```
amscz-version: 2.1
amscz-supported-versions: 2.0,2.1
amscz-deprecated-versions: 1.0
```

amscz-version – the API version used when processing the request.
supported-versions – list of currently supported versions, versions are separated by comma
deprecated-versions – list of unsupported versions that will be removed from the system

2.1.1. API 1.0

Basic auth is used for authorization.
Simple sample request code in PHP and CURL: (for other languages it will be similar).

```
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, URL));
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_USERPWD, „LOGIN:PASSWORD");
curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, http_METHOD);
curl_setopt($ch, CURLOPT_POSTFIELDS, „REQUEST" );
curl_setopt($ch, CURLOPT_HTTPHEADER, array(,Content-
Type:application/json'));
$result = curl_exec($ch);
```

Where:

- URL is the address of a function.
- LOGIN, PASSWORD are login credentials.

Národní organizace pro ověřování pravosti léčiv, z.s., IBC Pobřežní 620/3, Karlín, 186 00 Praha 8
IČ: 05851742, www.czmvo.cz

Login credentials are either generated by the system upon a request, or for access to a specific alert (for end users only) it is possible to use an „Alert ID "– UPRC as the login and the location ID as the password.

- http_METHOD is one of the methods „GET", „POST", „DELETE", „PUT".
- REQUEST – JSON formatted request.

2.1.2. API 2.0 and higher

OAuth2 is used for authorization. Compared to API 1.0, some additional informations are required in the request header.

2.1.2.1 General API request

Example request for API 2.0:

```
GET /alerts/?list=messages&changedFrom=2021-07-06+12%3A00%3A00 HTTP/1.1
Host: api.czmvo.cz
User-Agent: Client 1.0
amscz-version: 2.0
Authorization: Bearer eyJtc2ciOiJUYWR5IGplIG7Em2tkbyB6dsSbZGF2w70i...
Accept-Language: en
Accept: application/json
```

User-Agent – mandatory - identifies the software used and its version (the name is optional, its maximum length should not exceed 100 characters)

amscz-version – mandatory - specifies which version of the API the system should use to process the request. The current value is 2.0.

Authorization – authorization token (mandatory).

Accept-Language – optional - specifies the language of response.

Accept – mandatory - the required output of the response (usually `application/json` or `application/csv`).

2.1.2.2. Obtaining an authorization token

To obtain an authorization token, you need a **Client ID** and a **Secret ID** (see section 1.1 ACCESSING THE API AND WEB INTERFACE).

Request (for production environments):

```
POST /auth/token/ http/1.1
Host: api.czmvo.cz
Content-Type: application/x-www-form-urlencoded
User-Agent: Client 1.0
Cache-Control: no-store
grant_type=client_credentials&client_id=id&client_secret=secret
```

Answer is then:

```
http/1.1 200 OK
Content-Type: application/json
{
   „access_token":"eyJtc2ciOiJUYWR5IgplIG7Em2tkbyB6dsSbZGF2w70i...",
    „expires_in":1800,
    „token_type":"Bearer"
```


Národní organizace pro ověřování pravosti léčiv, z.s., IBC Pobřežní 620/3, Karlín, 186 00 Praha 8
IČ: 05851742, www.czmvo.cz

```
}
```

where the ***access token*** is used when making requests.
Expires in - the validity of the authorization token in seconds.
If the authorization fails, then the API returns an **http/1.1 400 Bad Request** error with the content:

```
{„error“:“invalid_client“}
```

2.2. GENERAL

This is a standard **REST API**. Data is exchanged via JSON format (unless otherwise specified).

The output is JSON, the format differs in used function or method. The basic structure is as follows:

```
{„status“:“ok“,“code“:0,“message“:“OK“,“result“:{ … result …}}
```

If “*code*” is different from zero, then “*message*” contains the description of the error. The field “*status*” is either „ok“ – request was processed or „error“ – request was not processed (that usually means– if there is no internal API error, that “*code*” – number of the error – is not zero).

2.2.1 Setting-up the language of the answer

Some values of answers can be localized – e.g. API error messages, alert states in the code list. Http header “*Accept-Language*” can be used to set up the desirable language.

Supported languages are currently Czech (**cs**) and English (**en)**.

```
Accept-Language: en
```

nebo

```
Accept-Language: cs-CZ
```

2.3. PARAMETER PASSING FOR GET AND DELETE METHODS

Some systems and libraries don’t allow passing data in the body of request (POST data) for the GET and DELETE methods. In such case parameters need to be inserted directly into url. Parameters in url must correspond with parameters from the requested JSON query.

Example:
JSON request “Get all messages since 6th July 2022 12:00:00”

```
{„list“:“messages,“changedFrom“:“2022-07-06 12:00:00“}
```

Passing parameters via url (test environment) looks like:
https://api.czmvo.cz/alerts/?list=messages&changedFrom=2022-07-06+12%3A00%3A00

2.4. TYPE OF RETURN VALUES

Request parameter "*resultAs*" can be replaced equivalent http header "*Accept*". The "*resultAs*" parameter is retained for backward compatibility.

For "*resultAs*": "json":

```
Accept: application/json
```

For "*resultAs":* "csv":

```
Accept: text/csv
```

In API 2.0 and higher, the "*resultAs*" parameter is dropped and ignored, and is completely replaced by the mandatory "*Accept*" header.

2.5. CONNECTION VERIFICATION

An additional parameter "connection" can be used to verify the correctness of the connection to the API and its functionality. If the GET parameter "*connection*" with the set value "**verify**" is connected to any request to the API, then the request itself will not be performed, but only the connection verification and authentication will be performed (the request url will be eg. https://api.czmvo.cz/t/?connection=verify)
There is no need to send a request to verify the connection, just the "*connection*" parameter with the corresponding value "**verify**".

For authentication purposes (only for him), alternatively, only the **Location ID** (identical for login and password) can be used as login and password for the end user as login data.

If a request is sent:

```
{„list“:“enumState“}
```

to url https://api.czmvo.cz/alerts/?connection=verify

then answer is:

```
{„status“:“ok“,“code“:0,“message“:“OK“,“result“:{„method“:“GET“,“module“:“alerts“,“Environment“:“production“,“auth“:“Regular“,“userrole“:“Enduser“,“state“:true}}
```

where individual fields can take the following values:
method - request method - one of GET, POST, PUT, DELETE
Module - called API function - can currently take the values "*alerts*" or "*filter*"
Environment - information whether was a call to a sandbox or a production server, possible values are: "*production*", "*sandbox*"
auth - user authentication method. Possible return values are:

Possible return values are:

- "*No authorization*" - login details are invalid.

- "*Regular*" - standard login (via login and password).
- "*Enduser alert based*" - end user login using location and UPRC.
- "*Verify only*" - login only for connection verification (login and password is the site id) - this login cannot be used for requests.

Userrole - user role (if successfully logged in) - possible values: "N/A" - if no login, "Enduser" - End user, "MAH/OBP" – MAH.

State - bool value – "true" - if the connection is OK, "false" - if there is any problem with the connection.

2.6. OVERVIEW OF ERROR STATES – FIELD „*CODE*"

code (error code)	http response	Description
0	200	Request was correctly processed.
1	404	Unknown function. *Most likely an incorrect URL of a request.*
2	401	Unauthorized access. *User authentication was not successfully completed, the user cannot be logged in.*
3	401	Function not allowed. *Most likely an incorrect URL of a request, or the user does not have authorization to access the API.*
4	405	Forbidden calling method. *Unknown http method (allowed methods are GET, POST, PUT, DELETE).*
5	400	Forbidden value of a parameter. *Also returns the specific parameter.*
11	400	Parameter value not filled in. *Most likely the mandatory parameter is not filled in.*
12	404	Alert not found.
13	405	No authorization to write into and alert.
14	400	The file could not be decoded. Incorrectly coded file in JSON request.
15	400	File size exceeded MB *Returns a current maximum file size (16MB).*
16	500	Message could not be saved. *Internal API error.*
17	401	Authorization to edit the message not granted.
18	401	Message cannot be answered. *Either it no longer exists, or it is closed.*
19	401	Message cannot be deleted. *Message has an attached response and cannot be deleted.*
20	400	At least one parameter has to be entered UPRC, ID
21	404	File ID %s not found
22	401	No authorization to read this file.
23	415	File type not supported. Supported file types are *txt,pdf,csv,jpg,png,tiff.*
24	500	Internal API error.

Národní organizace pro ověřování pravosti léčiv, z.s., IBC Pobřežní 620/3, Karlín, 186 00 Praha 8
IČ: 05851742, www.czmvo.cz

		An unspecified bug in the software portion of the API. Please retry the request later.
25	405	The alert cannot be edited, it is already archived. *Alerts are archived 90 days after the alert is closed.*
26	405	Alert cannot be modified, it is assigned to another MAH. *The error occurs during mass import when trying to set the status of a foreign MAH alert.*
27	401	Unable to set status for alert. Unexpected new status. *The required alert status cannot be set because it does not match the process workflow.*
28	401	Unable to set status for alert. Not authorized. *The user is not authorized to make the requested status change.*
29	401	Unable to set status for alert. Alert is closed.
30	401	Unable to set status for alert. Not all conditions are met. *Setting the status is possible, but additional conditions were not met (e.g. "Reason for reopening" was not entered).*
31	401	The alert message cannot be sent. Alert is not in the correct state to send a message. *It is not possible to set the required alert status after sending the message, because it does not correspond to the process workflow.*
32	400	CZMVS error specification required.
33	400	An unsupported Accept request header type was specified or was not specified. *Sending an Accept header in the API request header is required.*
34	405	Alert cannot be modified, is assigned to another End User.
35	405	The alert cannot be modified, the request does not correspond to the process workflow.
36	404	Location ID not found. *The requested location was not found in the database.*
37	401	There is no permission to access the location. *The user does not have permission to access the requested location.*
38	400	Invalid or expired Authorization Token. *Authorization token is not valid. A new one needs to be generated.*
39	400	Invalid request *When requesting API v2+, one of the mandatory http headers is not filled in.*
40	401	Actions over a group cannot be performed *One or more alerts in the group are in a state where the request cannot be executed. A list of affected alerts is provided in the response.*

2.7. FUNCTION ALERTS

url function: https://api.czmvo.cz/alerts/
url function for test environments: https://api.czmvo.cz/t/alerts/
url function for development environments: https://betaapi.czmvo.cz/alerts/

2.7.1. GET method

2.7.1.1. Get alert states

Example of a request (detection of alert states):


```
{„list“:“state“,“uprc“:“CZ-0VG-ZZW-5BU-
LZ0“,“latest“:true,“createdFrom“:“2022-08-06
00:00:00“,“createdTo“:“2022-08-13 00:00:00“,“changedFrom“:“2022-08-04
00:00:00“,“state“:1,“page“:1}
```


Example of answer:


```
{„status“:“ok“,“code“:0,“message“:“OK“,“result“:{„alerts“:[{„uprc“:“
CZ-0VR-Y94-KK5-6FJ“,“created“:“2022-07-16
07:50:04“,“productcode“:“08595116521485“,“stateid“:1,“state“:“Nov\u0
0fd“,“lastmessageid“:“20“,“statedescription“:“Nov\u00fd“ }]}}
```


Example of a request (detection of number of pages in a subset):


```
{„list“:“state“,“page“:-1}
```


Example of answer:


```
{„status“:“ok“,“code“:0,“message“:“OK“,“result“:{„pages“:3,“currentPage
“:0}}
```


Expanding the answer for end users:
If the logged-in user has the "End User" role, then the response also includes *typestate* and *typestatedescription* fields that indicate the requested action.

Example of answer:


```
{„status“:“ok“,“code“:0,“message“:“OK“,“result“:{„pages“:1,“currentPage
“:1,“alerts“:[{„uprc“:“CZ-KSR-RLB-6MF-E8C-8RT“,“created“:“2022-05-05
11:07:00“,“productcode“:“08594175410327“,“stateid“:1,“state“:“01a –
Nov\u00fd – transakce
KU“,“lastmessageid“:0,“statedescription“:“Nov\u00e1 v\u00fdstraha –
v\u00fdsledek investigace NOOL p\u0159\u00edpadn\u011b v poli
„P\u0159edanal\u00fdza – automatick\u00e1“. Balen\u00ed m\u011bjte
v karant\u00e9n\u011b. Pokud se ale domn\u00edv\u00e1te, \u017ee se
jedn\u00e1 o odstranitelnou chybu na va\u0161\u00ed stran\u011b, po
odstran\u011bn\u00ed probl\u00e9mu se m\u016f\u017eete pokusit o
op\u011btovnou verifikaci. Pokud bylo n\u00e1sledn\u00e9
ov\u011b\u0159en\u00ed \u00fasp\u011b\u0161n\u00e9, lze LP vydat, a
alert m\u016f\u017eete v AMS uzav\u0159\u00edt pomoc\u00ed
p\u0159\u00edslu\u0161n\u00e9ho stavu. „,“typestate“:“Informace
```

```
MAH",“typestatedescription“:“Po\u017eadov\u00e1ny dodate\u010dn\u00e9
informace od u\u017eivatele“}]}}
```

Notes:
If a parameter is not mandatory, it does not have to be a part of the request.
resultAs (only API 1.0) can have values „*json*“ or „*csv*“. If set to “*csv*”, then the output file is a csv table.
Uprc – unique alert identifier.
List – always „*state*“.
Latest – *true/false* – sorted by the newest alert.
createdFrom – displays alerts newer than specified time only. All time values have to follow format „YYYY-MM-DD HH:MM:SS“. All times are in UTC.
createdTo – displays alerts older than specified time only.
changedFrom – displays only alerts that have been changed after the specified time.
State – displays alerts with desired state (according to the alert state code list).
Page – number – determines what page (subset) is displayed in the answer. Maximum number of alerts (which is one page) in one subset is **500** (depending on specific authorization settings, the number of alerts in any subset can be lower). If the *page* parameter is set to a negative number (e.g. -1), then the total number of subset pages will be returned in the answer.

In JSON answers the values ***pages*** and ***currentPage*** are always returned.

In the answer (apart from obvious fields):
alerts is always a field – even if the result is only one entry.
Stateid – alert state ID.
State – state ID in plain text.
Statedescription - a detailed description of the status depending on the role of the logged in user.
Lastmessageid – last entered message.
Pages – number of pages of entries (subsets).
currentPage – current returned page of results in the alerts field (if the ***page*** number is positive, then this value is returned).

2.7.1.2. Get messages

Example of a request:
Detect all messages since 7/6/2022 12:00:00

```
{„list“:“messages,“changedFrom“:“2022-07-06 12:00:00“}
```

Detect all messages regarding alert CZ-0VG-ZZW-5BU-LZP

```
{"list":"messages","uprc":"CZ-0VG-ZZW-5BU-LZP"}
```

Detect/Load message with ID 20

```
{"list":"messages","id":"20"}
```


Národní organizace pro ověřování pravosti léčiv, z.s., IBC Pobřežní 620/3, Karlín, 186 00 Praha 8
IČ: 05851742, www.czmvo.cz

Example of an answer:


```
{"status":"ok","code":0,"message":"OK","result":{"messages":[{"id":"19"
,"parent":"0","uprc":"CZ-0VR-Y94-KK5-6FJ","created":"2022-07-06
10:45:59","changed":"2022-07-06
10:45:59","subject":"info","message":"Uplne
ok","isfile":false,"public":false,"fromme":true},{"id":"20","parent":"1
9","uprc":"CZ-0VR-Y94-KK5-6FJ","created":"2022-07-06
10:59:06","changed":"2022-07-06 10:59:06","subject":"Re:
info","message":"Fajn","isfile":false,"public":true,"fromme":false}]}}
```


Notes:

uprc – unique alert identifier, it is a mandatory parameter in case that parameter **id** or **changedFrom** is not used.

list – always „*messages*“.

id – message ID.

changedFrom – time in format „YYYY-MM-DD HH:MM:SS“ – returns messages newer than specified time. If neither ***uprc*** nor ***id*** is entered, then the **changedFrom** value should not be older than 1 month.

In the answer: (apart from obvious fields)

messages – is always a field – even if the result is only one entry.

id – Message ID.

parent – Message ID, that is being responded to (requirement/request ID).

uprc – unique Alert identifier.

created – Message creation time.

changed – time of the last Message change.

subject – Message subject.

message – Message body.

isfile – *true/false* – message contains a file.

public – true/false – if “true”, then the Message is visible to all stakeholders at the Alert. If “false”, then the Message is visible only to the person who entered it (and NOOL).

fromme – true/false – identifier, that signifies if the Message was created by the current User.

id_request – ID of the request/message from the message code (see GET Retrieving the message code), or 0.

2.7.1.3. Get a File

Example of a request for API 1.0:

```
{"resultAs":"json","list":"file","id":"21"}
```

or (direct file sending) for API 1.0

```
{"resultAs":"csv","list":"file","id":"21"}
```

Example of a request for API 2.0 and higher:

```
{"list":"file","id":"21"}
```

In the request header for API 2.0 it is possible to set:

```
Accept: application/json
```

or (to get binary data)

```
Accept: application/octet-stream
```

Example of answer:


```
{"status":"ok","code":0,"message":"OK","result":{"filename":"xxx.pdf","
filedata":"JVBERi0xLjcKCjQgMCBvYmoKKElkZW50aXR5KQplbmRvYmoKNSAwIG9iagoo
QWRvYmUpCmVuZG9iago4IDAgb2JqCjw8Ci9GaWx0ZXIgL0ZsYXRlRGVjb2RlCi9MZW5ndGg
gODk1NjAKL1R5cGUgL1N0cmVhbQo+PgpzdHJlYW0KeJzsfQlgVMX9\/3fese+9Pd\/em80m
u5vNRUIOkpADAtlwiVIkAmKiRsOlYD2CcqkV8ATBA6sith7xQsWDJfEIiBWPHmqtWK2irTW
tt5Wf2FJqlez+vzO7yW7kaNJ\/aPnl9z678515M\/OdmTfzne+b77y3+4AAQBoSAdZPmHH8
cYYfjn4MhDMaAbx\/OG7CxElb6zruAe63XQB8xXEN02acdCDjb8C97QCSPu+4GSeP+\/zOs
\/4E\/ …
```


Notes:
resultAs (only API 1.0) – *json* – returns a file in a format that is indicated in the example above.
resultAs (only API 1.0) – *csv* – directly exports a file (with the corresponding mime type).
list – always „file".

2.7.1.4. Get Alert state code list

Example of a request:

```
{"list":"enumState"}
```

Example of answer:


```
{"status":"ok","code":0,"message":"OK","result":{"states":[{"id":1,"nam
e":"Nov\u00fd","externalcode":"01","finalstate":false},{"id":"5","name"
:"V
\u0159e\u0161en\u00ed","externalcode":"#","finalstate":false},{"id":"3"
,"name":"Uzav\u0159en\u00fd","externalcode":"06a,06b,06c","finalstate":
true},{"id":"6","name":"Odlo\u017een\u00fd","externalcode":"","finalsta
te":false},{"id":"7","name":"Chyba import na
callcentrum","externalcode":"CALLFAIL","finalstate":false}]}}
```


Extending the answer for End users:
If the logged-in User has the "End User" role, then the response includes extra text fields *typestate* and *typestatedescription* to indicate the action requested.

Example of answer:


```
{"status":"ok","code":0,"message":"OK","result":{"states":[{"id":1,"nam
e":"01a - Nov\u00fd - transakce
KU","externalcode":"01a","finalstate":false,"settingallowed":false,"des
cription":" ... ","typestate":"Informace
MAH","typestatedescription":"Po\u017eadov\u00e1ny dodate\u010dn\u00e9
informace od u\u017eivatele"}, ...
```

Notes:
resultAs (only for API 1.0) - may take values „*json*“ or „*csv*“ – If value csv is set, the output will be table in csv format.
states – always a field.
name – name of the State.
externalcode – code according Alert state code list.
finalstate – *true/false* – indicates whether the State is finite (i.e. whether the alert is resolved) or not.
settingallowed – *true/false* – indicates whether the State of the alert can be set through an API.
Description - detailed description of the Status depending on the Role of the logged in User.

2.7.1.5. Get code list of Messages

Example of a request:

```
{"list":"enumRequest"}
```

Example of an answer:

```
{"status":"ok","code":0,"message":"OK","result":{"requests":[{"id":1,"n
ame":"Fotka","text":"\u017d\u00e1d\u00e1me o zasl\u00e1n\u00ed fota
obalu LP, s \u010diteln\u00fdm 2D
k\u00f3dem"},{"id":2,"name":"Fotka_EAN","text":"\u017d\u00e1d\u00e1me o
zasl\u00e1n\u00ed fota obalu LP, s \u010diteln\u00fdm 2D k\u00f3dem.
Nafotte pros\u00edm i vizu\u00e1ln\u011b \u010diteln\u00e9 \u00fadaje
(EAN, \u0161ar\u017ee, SN, datum exspirace, apod.)"}]}}
```

Notes:
resultAs (only for API 1.0) can take the values "*json*" or "*csv*" - if csv is set, then the output is a table in csv format.
requests – always field, where:
id – unique identifier of the request/message, specified when sending the Message
name – name of the Message
text – predefined Message text
forStates – field of integers (int), list of IDs of Alert states (see chapter 2.7.1.4) for which the Message can be sent. (the Alert must be in one of the listed States in order to send the Message)
name – Status name.
text – text Message itself.

2.7.1.6. Get group of alerts

Only for MAH’s.
Allows loading of a list of alerts that are in the same group as the default alert. The system group alerts with respect to their similarity.
Example of a request:

```
{"list":"group","uprc":"CZ-0VR-YE5-C1N-KLM"}
```

Example of an answer:


```
{„status“:“ok“,“code“:0,“message“:“OK“,“result“:{„uprc“:[„CZ-0VR-YE5-C1N-KLM“,“CZ-0VR-YE5-VS7-BXP“]}}
```


Notes:

uprc – in query – identifier of the default (one) Alert.

In answer:

uprc - list of all alerts that belong to the same Group. If an alert does not belong to any group, the field in the answer is empty.

2.7.1.7. Loading the "Reasons for Reopening" list

When trying to change the status of an alert that is already closed, the "***Reason for its reopening***" is required (if it is possible to reopen it). The reason for opening is set by the parameter when setting the status.

Example of a request:


```
{„list“:“enumReopenReason“}
```


Example of an answer:


```
{„status“:“ok“,“code“:0,“message“:“OK“,“result“:{„reasons“:[{„id“:1,“name“:“Chybn\u011b uzav\u0159eno“}]}}
```


In answer:

reasons – current code list. When setting the alert status, the reason ID is then passed in the relevant parameter.

2.7.1.8. Loading an own Note

Used to load an existing Note and its possible parameters.

Example of a request: (loading an own Note)


```
{„list“:“note“,“uprc“:“CZ-LD8-F79-ABY-PFC-5J0“}
```


Example request: (retrieve Notes of another user who has given permission to do so)


```
{„list“:“note“,“uprc“:“CZ-LD8-F79-ABY-PFC-5J0“,“from“:“mah“}
```


Example of an answer:


```
{„status“:“ok“,“code“:0,“message“:“OK“,“result“:{„note“:“Test note“,“uprc“:“CZ-LD8-F79-ABY-PFC-5J0“,“private“:false,“changed“:“2022-01-12 11:13:42“}}
```


Notices:

uprc – in request – the number of the Alert to which the Note is assigned.

From – optional parameter, contains one of the values "**mah**", "**enduser**", "**nool**" (if a different value is specified, the API behaves as if none were specified). Specifies whose Note to retrieve. In order to load a foreign note, it must not be marked as "internal".

In answer:
uprc – alert number (if the alert for which a Note is requested is found)
Note – Note text (or an empty string if a note has not yet been entered)
Private – true/false –. if "true" then the Note is internal and not accessible to any third party within the system. If "false" then the Note can be seen by other users.
Changed – the date and time the Note was last modified. If the Note does not exist, then the value is "null".

2.7.1.9. Loading the "Status Type" list

The code list can only be loaded by the End user. This is a list of indications for Users on what to do with the packaging in a given situation. (chapter sections 2.7.1.1. and 2.7.1.4). Each alert status is assigned exactly one "*Status Type*".

Example of a request:

```
{„list“:“enumTypeState“}
```

Example of an answer:

```
{„status“:“ok“,“code“:0,“message“:“OK“,“result“:{„typestates“:[{„name“:
“N“,“description“:“Neprov\u00e1d\u011bt nic“}]}}
```

In answer:
typestates – current code list.

2.7.1.10. Loading the counter for level 3 alerts

It is used to determine the current value of the incremental counter for generating the unique identification of level 3 alerts. The feature is applicable only to the End User.

Example of a request:

```
{„list“:“a1counter“,“location“:“858d085f-324a-4938-a796-333bfac94f05“}
```

Example of an answer:

```
{„status“:“ok“,“code“:0,“message“:“OK“,“result“:{„counter“:2}}
```

Notices:
location **–** Location ID. Required field.

In response:
counter – integer – the current state of the counter for the selected Location.

Národní organizace pro ověřování pravosti léčiv, z.s., IBC Pobřežní 620/3, Karlín, 186 00 Praha 8
IČ: 05851742, www.czmvo.cz

2.7.1.11. Loading level 3 Alerts

Used to load the current list of level 3 Alerts. The feature is only applicable to the End user.

Example of a request:

```
{"list":"a1alerts","location":"858d085f-324a-4938-a796-333bfac94f05"}
```

Example of a request: (restriction by counter)

```
{"list":"a1alerts","location":"858d085f-324a-4938-a796-
333bfac94f05","from":2}
```

Example of an answer:

```
{"status":"ok","code":0,"message":"OK","result":{"a1alerts":[{"uprc":"C
Z-858d085f-324a-4938-a796-333bfac94f05-000001","created":"2022-10-19
07:48:04","productcode":"18901138057340","stateid":6},{"uprc":"CZ-
858d085f-324a-4938-a796-333bfac94f05-000002","created":"2022-10-19
07:48:38","productcode":"18901138057340","stateid":6}]}}
```

Notices:
location – location ID. Required field.
from – an integer - will limit the list of level 3 alerts only to alerts with a counter greater than or equal to the specified value. The parameter is optional (if not specified, the function returns all alerts)

In response:
a1alerts – field - list of alerts
stateid – ID of the alert State according to the Status code

Important: 2.7.1.10. and 2.7.1.11. are only valid for API 1.0, for API 2.0 the call will be replaced by 2.7.2.2.

2.7.1.12 Detection of possible actions above the alert

This function is used to get a list of possible operations over a specific alert that can be currently performed.

Example of a request:

```
{"list":"allowedActions","uprc":"CZ-LD8-F79-ABY-PFC-5J0"}
```

Example of an answer:

```
{"status":"ok","code":0,"message":"OK","result":{"sendMessage":[16,14],
"setState":[39,40,41],"group":true,"group_a":false}}
```

Notice:
uprc – number of the existing alert. If the alert does not exist, then error 12 is returned.

In the answer:

sendMessage – an array of integers (or an empty field - if no message can be sent) - a list of message IDs (see section 2.7.1.5 Loading the Message Codebook) that can currently be sent.

setState – an array of integers (or an empty field - if the alert state cannot be changed) - a list of alert state IDs (see section 2.7.1.4 Loading the Alert State code list) that can currently be set for the alert.

group – bool value - "true" - if the alert is assigned in some group (and it is expected to perform an operation over the group).

group_a – bool value - "true" - if the alert is assigned in some anonymous group (and is expected to perform an operation over the group).

2.7.2. POST method

Request - insert a message using the message dialer (the only possible method for communication between MAH and End user):

```
{"uprc":"CZ-0VR-Y94-KK5-6FJ","public":true,"id_request":1}
```

Request (simple insertion):

```
{"uprc":"CZ-0VR-Y94-KK5-
6FJ","public":true,"subject":"test","message":"test"}
```

Request (insert in response to another message):

```
{"uprc":"CZ-0VR-Y94-KK5-
6FJ","public":true,"id_parent":20,"subject":"Re: Re:
info","message":"test"}
```

Request (insert file):

```
{"uprc":"CZ-0VR-Y94-KK5-6FJ","public":false,"subject":"Re: Re:
info","message":"test","file":"iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAIAAACQ
kWg2AAAACXBIWXMAAAsTAAALEwEAmpwYAAAAIGNIUk0AAHolAACAgwAA+f8AAIDpAAB1MAA
A6mAAADqYAAAXb5JfxUYAAADjSURBVHjaYnwaJcSABBhZ2RlQwf\/fP5G5THiUwsWRpZiQV
ePSgyzFhF8dJmBB43PahAmkTEAWeRYtDLf\/\/++fjE+jhCB88cmXmXiFcRn8PEEK6iRGVn
ZOmzDJBc9eZKihGfw8QQquTnLBM4Qfvh9Z9WFOgdTSt8+iETZILX0LCVBkQUQo4dGDFiQIT
38\/sgqi6FmkgNTyD3A9EMaHOQVYQgmqZ\/mHZ9HCcKUQ1RApBgYGFohD4fYi7IHpQVb9\/
\/dPFswQhOtBVooz4pD1oKmGugU5teJJI\/A0y4QnJcMFkcUBAwCuU3b1BVKmxQAAAABJRU
5ErkJggg==","filename":"test.png"}
```

Answer:

```
{"status":"ok","code":0,"message":"OK","result":{"id":22}}
```

API 2.0 and higher:

If an operation is performed on an entire group (the "*group*" or "*group_a*" parameters are set), and it is not possible to perform the operation on one or more alerts, **then the operation**

Národní organizace pro ověřování pravosti léčiv, z.s., IBC Pobřežní 620/3, Karlín, 186 00 Praha 8
IČ: 05851742, www.czmvo.cz

will not be performed and the system **will return a list of alerts that are blocking completion operation due to their status**:

Example of an answer:

```
{"status":"error","code":40,"message":"Actions over the group cannot be performed","result":{"uprc":["CZ-0VR-Y94-KK5-6FJ","CZ-1VR-Y94-KK5-6FI"]}}
```

Notes:
The Message ID is always an integer, is unique across the system, and a newer message will always have a higher ID**.**
uprc – unique alert identifier. Mandatory if the *id_parent* field is not used.
public – *true/false* – indicates that the answer is public in a sense that both MAH and End-user has access to it (if value is set to *false*, only author and system administrator have access to the message).
id_parent – Message ID which is being replied to. If a message is being replied to, then the "*group*" and "*group_a*" parameters are ignored and the message is assigned to all alerts to which the original message was sent, for which the user has permission.
subject – mandatory – message subject
message – message text. Mandatory if file is not attached.
file – base64 encoded binary file (of an unrestricted type).
filename – name of the file (mandatory if file is attached).
id_request – request ID from the message codebook.
group – *true/false* – If "*true*", then the message/file will be sent to all alerts in the group to which the user has permission. If the parameter is not specified, then its value is taken as "*false*".
Group_a – true/false – If "true", then the message/file will be sent to all alerts in the anonymous group to which the user has permission. If the parameter is not specified, then its value is taken as "false".
Only_file – true/false – If the parameter is set to "*true*", then the system treats the message only as a separate file, and not as a regular message (alert status is not changed, notifications are not sent). If this parameter is set to "*true*", then a file must be inserted.

2.7.3. PUT method

2.7.3.1. Edit existing message

Allows editing of *public, subject* and *message* fields in an already existing message.

Note: Author only is allowed edit data. Messages for which already exist answers can’t be edited.

Example of a request:

```
{"id":22,"public":true,"subject":"TEST"}
```

Example of an answer:


```
{"status":"ok","code":0,"message":"OK","result":{"id":22,"changed":"
2019-08-06 15:36:35"}}
```


Notes:
id – message ID
only *public*, *subject*, *message* and *id_request* can be edited (see method POST).

2.7.3.2. Edit alert status

Allows state change of single alert or group of alerts.

Request: (change state of single alert)


```
{"uprc":"CZ-0VR-YE5-VS7-BXP","state":5,"group":false}
```


Request: (Bulk status setting)


```
{"uprc":"CZ-0VR-YE5-VS7-BXP","state":5,"group":true}
```


Request: (Alternate bulk status setting)


```
{"uprc":["CZ-0VR-YE5-VS7-BXP","CZ-0VR-YE5-C1N-KLM"],"state":5}
```


Answer:


```
{"status":"ok","code":0,"message":"OK","result":{"uprc":["CZ-0VR-YE5-
C1N-KLM","CZ-0VR-YE5-VS7-BXP"]}}
```


API 2.0 and higher:
If an operation is performed on an entire group (the group or group_a parameters are set) and it is not possible to perform the operation on one or more alerts, **then the operation will not be performed** and the system will return a list of alerts that are blocking due to their completion status:

Example of an answer:


```
{"status":"error","code":40,"message":"Akce nad skupinou nelze
provést","result":{"uprc":["CZ-0VR-Y94-KK5-6FJ","CZ-1VR-Y94-KK5-6FI"]}}
```


Note:
uprc – unique alert identifier - mandatory field
state – new alert state ID. For the list of all possible states see section **Loading of code list of States.** States that can be configured though API have identifier "***settingallowed***".
group – *true/false* – If "*true*", then the state is set for all alerts from the same group (see GET method, getting alerts from a group) as an input alert (parameter "***uprc***"). If set to "*false*", then the change applies to the state of selected alert only.
group_a – *true/false* – same as "***group***", but for an anonymous group.

id_request – Message ID from the message codebook - see GET Loading the Message codebook.

In answer then:
List of affected (changed) alerts is in the parameter "***uprc***".

2.7.3.3. Edit own Note

Allows to insert or edit an existing custom alert Note.

Request:

```
{"uprc":"CZ-LD8-F79-YBY-PFC-5J0","private":true,"note":"TEST NOTE"}
```

Answer:

```
{"status":"ok","code":0,"message":"OK","result":{"uprc":"CZ-LD8-F79-
YBY-PFC-5J0","changed":"2022-01-12 10:43:06"}}
```

Note:
uprc – unique alert identifier - mandatory field
note – mandatory field (if it is an empty string, then the note will be deleted), the maximum length of the note is 60,000 characters.
private – *true/false* – optional field. If not specified, then the default value of "*true*" is used. If the value of this parameter is "*true*", then this is an internal note and is not visible to anyone within the system. If the value is "*false*" - then the note is visible to all parties.

In Answer then:
changed – last modified time of the note - usually the current time (however, if no value is changed when the note is modified, then this is the time of the last actual change).

2.7.4. DELETE method

2.7.4.1. Deleting Message

Request:

```
{"id":22}
```

Answer:

```
{"status":"ok","code":0,"message":"OK","result":{"id":22}}
```

Note:
Parameter "***id***" (Message ID) is the only parameter possible. The conditions for deleting a Message are identical to conditions for its editing.

Národní organizace pro ověřování pravosti léčiv, z.s., IBC Pobřežní 620/3, Karlín, 186 00 Praha 8
IČ: 05851742, www.czmvo.cz

2.7.4.2. Deleting own Note

Request:

```
{"note":"CZ-LD8-F79-YBY-PFC-5J0"}
```

Answer:

```
{"status":"ok","code":0,"message":"OK","result":{"uprc":"CZ-LD8-F79-
YBY-PFC-5J0"}}
```

Note:
note – required field. This is the "***uprc***" for which the Message is to be deleted. In the response, the "***uprc***" value is used for checking.

2.8. FUNCTION EXCEPTIONS

Function url: https://api.czmvo.cz/filter/
Function url for the test environment: https://api.czmvo.cz/t/filter/
Function url for the development environment: https://betaapi.czmvo.cz/filter/

2.8.1. GET method

2.8.1.1. *Get the State code list*

Request:

```
{"list":"enumState"}
```

Odpověď:

```
{"status":"ok","code":0,"message":"OK","result":{"states":[{"code":"NO"
,"name":"Uzav\u0159eno - MAH - nelze
opravit"},{"code":"OP","name":"Uzav\u0159eno - MAH - opraveno"}]}}
```

Note:
resultAs (only API 1.0) can take values "json" or "csv" – it the value is set to "csv" result is table in csv format.

2.8.1.2. *Load existing Exceptions*

Request:

```
{"list":"product","productCode":"kod","batch":"","id":[1,2,3]}
```

Answer:

```
{"status":"ok","code":0,"message":"OK","result":{"products":[],"count":
0}}
```

Note:
„***id***" is array id, which are assigned to filter when inserted, parameters for filtering "id".

Národní organizace pro ověřování pravosti léčiv, z.s., IBC Pobřežní 620/3, Karlín, 186 00 Praha 8
IČ: 05851742, www.czmvo.cz

Parameters *„productCode“, „batch“* are optional.

This function will only return Exceptions that are entered by the user himself or are assigned to him (applicable to MAH).

2.8.1.3. Verify pack for the Exception

Request:

```
{"list":"verify","productCode":"08594158891136","batch":"1"}
```

Answer:

```
{"status":"ok","code":0,"message":"OK","result":{"isException":true,"in
fo":{"id":41,"productCode":"08594158891136","batch":null,"stateId":1,"s
tate":"Nov\u00fd"}}}
```

Note:
At least one of the arrays *„productCode“*, *„batch“* has to be filled.

In answer:
isException: *true/false* – the product/package is in the Exception list.
info – if isException is *“true”*, than the array contains Exception description.

2.8.2. POST method

Request (single item):

```
{"validity":"2019-04-
30","state":"OP","productCode":"0123456789","batch":"123456"}
```

Request: (bulk insert via csv file)

```
{"validity":null,"state":"NO","csv":"zCIsIsSMbMOhbmVrIDU3IGtvZGV4dSIsIk
lEIGRyxb5pdGVsZSByZWdpc3RyYWNlIChNQUggSUQpIiRyxb5pdGVsZSByZWdpc3RyYWNlI
ChNQUgpIiwiQWRyZXNhIGRyxb5pdGVsZSByZWdpc3RyYWNlIiwiTc3RvIGRyxb5pdGVsZSB
yZWdpc3RyYWNlIiwiUFPEjCBkcsW+aXRlbGU0cmFjZSIsIlplbcSbIGRyxb5gaahjfkleuH
GAFlgheghv4fdb2xxYVNS5fAS5gdrgpdGVsZSByZWdpc3RyYWNlIiwiw5pkYWplIG8gZGlz
dHJpYnV0b3JvdmkiDQo="}
```

Answer:

```
{"status":"ok","code":0,"message":"OK","result":{"products":[{"lineNo":
1,"productCode":"0123456789","batch":"123456","validity":null,"state":1
,"ID":1,"errorCode":0,"errorText":""}],"count":0}}
```

Note:
Parameter *“validity”* is date in format [YYYY]-[MM]-[DD].
Product code, batch and validity are mandatory parameters. If they are not entered, then the Exception will not be stored in the system.

Národní organizace pro ověřování pravosti léčiv, z.s., IBC Pobřežní 620/3, Karlín, 186 00 Praha 8
IČ: 05851742, www.czmvo.cz

The "*state*" parameter serves as the default value for setting the State of Alert (i.e. if it is not defined in the csv).
For bulk insertion, there are multiple rows in the answers where an ID or description of the reason for not inserting into the database is always assigned for each row.
The "*csv*" parameter is a base64 csv file in the format described in the "**Insert Data**" section of the web interface.

2.8.3. DELETE method

Request:

```
{"productCode":"0123456789","batch":"123456","id":[1,2]}
```

Answer:

```
{"status":"ok","code":0,"message":"OK","result":{"affected":1,"deleted"
:[{"id":"2","productCode":"0123456789","batch":"123456"}]}}
```

Note:
Parameters *"productCode", "batch",* and field *"id"* are optional, but at least one parameter has to be used (it is not possible to remove all items in single request).

3. OVERVIEW OF CHANGES

3.1. CHANGES COMPARED TO AMS R6.0

1. **Detecting possible actions over an alert**
   Added a function to get a list of possible operations over a particular alert that can be performed at a given time.

3.2. CHANGES COMPARED TO AMS R5.0

1. **API settings**
   The AMS API has a limit on the number of queries per given time. Client systems that exceed their quota are returned an HTTP status code (429).
   Note: the current setting is 800 requests per 5 minutes per IP address or 400 requests per 5 minutes per client id. A client can only re-authenticate and retrieve a token once every 60 minutes.
2. **OAuth 2.0 authentication API**
   OAuth 2.0 protocol is implemented for API access.
3. **API versioning and additional information**
   Introduction of API versioning. Information about supported, unsupported and terminated versions in the responses.
   a) AMS API versioning
   The version of the AMS API to be used is specified using "*amsapi-version*" as the HTTP header. (e.g. "*ams-api-version*" 2.0). Omitting the header results in requests being routed to AMS API 1.0.

b) Http header "*user-agent*"
The use of the HTTP header "*User-Agent*" is mandatory. Client systems must provide details of their client software version.

4. **Error conditions** have been added - **ID 38-40**.
5. Changed the creation of **unique identifier for level 3 alerts** (in agreement with NCA/SUKL).

3.3. CHANGES COMPARED TO AMS R4.0

1. **New "Typ_stav" code list**
   Only for the role "End user". It is a code list indicating to the user what to do with the package in a given situation. There is just one "*Type_Status*" associated with each alert state.
2. **Extension of the "Alert Status" code list**
   For the role "End user", the fields "***typestate***" and "***typestatedescription***" are added (linking to the new "Typ_stav" code list).
3. **Error states - ID 34-37** have been added.
4. **Loading counters for level 3 alerts**
   Added functionality for end users to retrieve the **value of the current counter** to generate an unambiguous identification of level 3 exceptions.
5. **Retrieve level 3 alerts**
   Added a function for end users to retrieve the current list of level 3 exceptions.

3.4. CHANGES FROM AMS R2.0

1. **Established Development Environment.**
   Used to test and develop application software for a version of AMS that has yet to be deployed to a production environment.
2. **Alert Workflow.**
   AMS uses a new **alert workflow** (modifying possible states for individual users, extending and modifying automatic escalations, confirmation e-mails).
3. Request parameter "***resultAs***" replaced by http header "***Accep***t". The "***resultAs***" parameter is retained for backward compatibility.
4. New **"*Reopen Reasons*"** code list.
5. **Added error conditions** (field "*code*") - **ID 25 - 33**.
6. **Redesigned work with groups and anonymous groups** (bulk responses, bulk state changes).
7. Possibility to **add a file to an alert** (even without MAH request).

Národní organizace pro ověřování pravosti léčiv, z.s., IBC Pobřežní 620/3, Karlín, 186 00 Praha 8
IČ: 05851742, www.czmvo.cz