ALERTS MANAGEMENT API

Documentation

ver. 5.0

(October 2022)

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

API CZMVO.CZ is simple REST API, which allows integration of MAH's or end users' alert management system with NOOL alert management system. Benefit of the integration is automation of processes and tasks related to alerts investigation for the both parties MAHs and end users.

User without own alert management system or users unable to connect via API due to company policies can use web interface of alert management system. Web interface has the same functionalities as API. Description of alert management system web interface is in the separated user manuals different for MAHs and end users. There are also more detail description of alert system logic and support for communication between MAH and end user during investigation.

1.1. 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/

A copy of the production data is uploaded to the **test and development environment** on a regular nightly basis. So you can test on real user data. Changes in the test and development environment do not affect the data in the production database in any way, so they can be changed at will. However, the data is overwritten every night, which must be taken into account during testing and development.
The test environment is used for testing and developing software for the existing production environment (i.e. it is functionally identical to the production environment).
The development environment is used for testing and developing application software for the version of AMS that is yet to be deployed on the production environment.

Access credentials type:

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.

Access credentials allows access to the API both web GUI system.

1.2.EMAIL NOTIFICATIONS

Depending on its settings, the system uses a **number 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. GENERAL

It is equivalent in terms of function to a web interface. It is a REST API with **basic authorization**. The data exchange utilizes the JSON file format (if not specified otherwise).

A simple sample code of a request in PHP and CURL: (will be similar for other languages).

```
$ch = curl_init();
curl_setopt($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.
  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 „ID alert"– 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. Can also be generated in the web interface.

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).

All methods have their equivalents in the web portal. Everything valid in the web portal is also valid in the direct use of API. We therefore recommend obtaining additional information from the "Web portal" section.

2.1.1 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)**.

Examples:

```
Accept-Language: en
```

or

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

2.2. 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 2021 12:00:00"

```
{"resultAs":"json","list":"messages,"changedFrom":"2021-07-06 12:00:00"}
```

Passing parameters via url (test environment) looks like:

https://api.czmvo.cz/alerts/?resultAs=json&list=messages&changedFrom=2021-07-06+12%3A00%3A00

2.3. TYPE OF RETURN VALUES

Request parameter “resultAs” can be replaced equivalent http header Accept.
The "resultAs" parameter is retained for backward compatibility. It will be canceled in the future.

For "resultAs": "json":
```
Accept: application/json
```

For "resultAs": "csv":

```
Accept: text/csv
```

2.4. 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","enviroment":"production","auth":"Regular","userrole":"Endus
er","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"
enviroment - 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.

2.5. OVERVIEW OF ERROR STATES –„CODE“ FIELD

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 NMB *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.*

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

24	500	Internal API error. *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	NSOL 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.*

2.6. 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/

Generating specific queries and displaying API answers in JSON format is also available in the system's web API.

2.6.1. GET method

Is the equivalent to the operation „Insert data“ from the web portal.

2.6.1.1. Get alert states

Example of a request (detection of alert states):


```
{"resultAs":"json","list":"state","uprc":"CZ-0VG-ZZW-5BU-
LZ0","latest":true,"createdFrom":"2019-08-06 00:00:00","createdTo":"2019-
08-13 00:00:00","changedFrom":"2019-08-04 00:00:00","state":1,"page":1}
```


Example of an answer


```
{"status":"ok","code":0,"message":"OK","result":{"alerts":[{"uprc":"CZ-0VR-
Y94-KK5-6FJ","created":"2019-07-16
07:50:04","productcode":"08595116521485","stateid":1,"state":"Nov\u00fd","l
astmessageid":"20","statedescription":"Nov\u00fd" }]}}
```


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


```
{"resultAs":"json","list":"state","page":-1}
```


Example of an 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 the answer:


```
{"status":"ok","code":0,"message":"OK","result":{"pages":1,"currentPage":1,
"alerts":[{"uprc":"CZ-KSR-RLB-6MF-E8C-8RT","created":"2020-05-05
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 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).

2.6.1.2. Get messages

Example of a request:
Detect all messages since 6.7.2021 12:00:00

```
{"resultAs":"json","list":"messages,"changedFrom":"2021-07-06 12:00:00"}
```

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

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

Detect/Load message with ID 20

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

Example of an answer:

```
{"status":"ok","code":0,"message":"OK","result":{"messages":[{"id":"19","pa
rent":"0","uprc":"CZ-0VR-Y94-KK5-6FJ","created":"2021-07-06
10:45:59","changed":"2021-07-06  10:45:59","subject":"info","message":"Uplne
ok","isfile":false,"audit":false,"public":false,"fromme":true},{"id":"20","
parent":"19","uprc":"CZ-0VR-Y94-KK5-6FJ","created":"2021-07-06
10:59:06","changed":"2021-07-06                    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.
audit – true/false – returns messages that have audit requirement.

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 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.

2.6.1.3. Get file

Example of a request:

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

or (direct file export)

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

Example of an answer:


```
{"status":"ok","code":0,"message":"OK","result":{"filename":"xxx.pdf","file
data":"JVBERi0xLjcKCjQgMCBvYmoKKElkZW50aXR5KQplbmRvYmoKNSAwIG9iagooQWRvYmUp
CmVuZG9iago4IDAgb2JqCjw8Ci9GaWx0ZXIgL0ZsYXRlRGVjb2RlCi9MZW5ndGggODk1NjAKL1R
5cGUgL1N0cmVhbQo+PgpzdHJlYW0KeJzsfQlgVMX9/em80mu5vNRUIOkpADAtlw
iVIkAmKiRsOlYD2CcqkV8ATBA6sith7xQsWDJfEIiBWPHmqtWK2irTWtt5Wf2FJqlez+vzO7yW7
kaNJ\/aPnl9z678515M\/OdmTfzne+b77y3+4AAQBoSAdZPmHH8cYYfjn4MhDMaAbx\/OG7CxEl
b6zruAe63XQB8xXEN02acdCDjb8C97QCSPu+4GSeP+\/zOs\/4E\/ …
```


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

2.6.1.4. Get alert state code list

Example of a request:

```
{"resultAs":"csv","list":"enumState"}
```

Example of an answer:


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


Notes:

resultAs - 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.

2.6.1.5. Get code list of Messages

Example of a request:

```
{"resultAs":"csv","list":"enumRequest"}
```

Example of an answer:

```
{"status":"ok","code":0,"message":"OK","result":{"requests":[{"id":1,"name":"Fotka","text":"\u017d\u00e1d\u00e1me o zasl\u00e1n\u00ed fotaobalu LP, s \u010diteln\u00fdm 2Dk\u00f3dem"},{"id":2,"name":"Fotka_EAN","text":"\u017d\u00e1d\u00e1me ozasl\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 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 2.6.1.4) 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.

2.6.1.6. 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:

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

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


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

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.

2.6.1.7. 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 dial. When setting the alert status, the reason ID is then passed in the relevant parameter.

2.6.1.8. Loading an owne 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-YBY-PFC-5J0"}`

Example of a request: (retrieving a note from another user who gave permission to do so)
`{"list":"note","uprc":"CZ-LD8-F79-YBY-PFC-5J0","from":"mah"}`

Example of an answer:
`{"status":"ok","code":0,"message":"OK","result":{"note":"Test note","uprc":"CZ-LD8-F79-YBY-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".

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

2.6.1.9. 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 2.6.1.1. and 2.6.1.4). 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.

2.6.1.10. Loading the counter for A1 alerts

Notice.: A1 alerts = exceptions level 3, which have not the UPRC.

It is used to determine the current value of the incremental counter for generating the unique identification of A1 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/establishment code. Required field.

In response:

counter –integer – the current state of the counter for the selected location.

Note: these exceptions will then have the form:
„MA“-„Location ID“-„xxxxxx“, where „MA“ is transaction market (in Czech Republic „CZ“), Location ID is ID of location from CZMVS and „xxxxxx“ je vzestupná číselná řada výjimek level 3, vzniklých na dané lokaci/provozovně. is the ascending numerical series of level 3 exceptions, created on given
Example:

Market: CZ
Location ID: ca71c18a-d444-4fce-9903-92a232af2745
Counter: 123456
The level 3 exception ID will then be: CZ- ca71c18a-d444-4fce-9903-92a232af2745-123456

2.6.1.11. Loading A1 Alerts

Used to load the current list of A1 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":"CZ-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 code. Required field.
from – an integer - will limit the list of A1 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

2.6.2. POST method

Request - insert a message using the message code list (preferred variant):
{"uprc":"CZ-0VR-Y94-KK5-6FJ","public":true,"id_request":1}

Requirement (simple insertion):
{"uprc":"CZ-0VR-Y94-KK5-6FJ","public":true,"subject":"test","message":"test"}

Request (insert as an answer to another message):
{"uprc":"CZ-0VR-Y94-KK5-6FJ","public":true,"id_parent":20,"subject":"Re: Re: info","message":"test"}

Request (insertion of a file):
{"uprc":"CZ-0VR-Y94-KK5-6FJ","public":false,"audit":true,"subject":"Re: Re: info","message":"test","file":"iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAIAAACQkWg2AAAACXBIWXMAAAsTAAALEwEAmpwYAAAAIGNIUk0AAHolAACAgwAA+f8AAIDpAAB1MAAA6mAAADqYAAAXb5JfxUYAAADjSURBVHjaYnwaJcSABBhZ2RlQwf\/fP5G5THiUwsWRpZiQVePSgyzFhF8dJmBB43PahAmkTEAWeRYtDLf\/\/++fjE+jhCB88cmXmXiFcRn8PEEK6iRGVnZOmzDJBc9eZKihGfw8QQquTnLBM4Qfvh9Z9WFOgdTSt8+iETZILX0LCVBkQUQo4dGDFiQIT38\/sgqi6FmkgNTyD3A9EMaHOQVYQgmqZ\/mHZ9HCcKUQ1RApBgYGFohD4fYi7IHpQVb9\/\/dPFswQhOtBVooz4pD1oKmGugU5teJJI\/A0y4QnJcMFkcUBAwCuU3b1BVKmxQAAAABJRU5ErkJggg==","filename":"test.png"}

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

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

Notes:
The message ID is always an integer, is unique across the system, and a newer message will always have a higher ID**.**
uprc – 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.

2.6.3. PUT method

2.6.3.1. 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).

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

2.6.3.2. 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: (change state of group of alerts)

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

Request: (alternative change state of group of alerts)

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

Request: (set alert for standard processing)

```
{"uprc":"CZ-0VR-YE5-VS7-BXP","state":-1,"group":false}
```

Answer:

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

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** (chapter 2.6.1.4.). 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, chapter 2.6.1.6), 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 (chapter 2.6.1.5).

In answer:

List of affected (changed) alerts is in the **uprc**.

2.6.3.3. Edit own notice

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.

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

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).

2.6.4. DELETE method

2.6.4.1. Deleting message

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

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

Note:
Message ID is the only parameter possible. The conditions for deleting a message are identical to conditions for its editing.

2.6.4.2. 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.

2.7. FUNCTION EXCEPTIONS

It is REST API with **basic authentication**. Data exchange is via JSON format (if not stated otherwise).

Simple example code in PHP and CURL:
```
$ch = curl_init();
curl_setopt($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" );
```


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

```
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-
Type:application/json'));
$result = curl_exec($ch);
```

Where:

- URL is function address.
- LOGIN, PASSWORD are access credentials.
- HTTP_METHOD is one of the methods “GET”, “POST”, “DELETE”.
- REQUEST – request in JSON format. It can be generated in the web interface if needed.

Output is JSON. Format depends on the used method or function. Basic structure is following:
`{"status":"ok","code":0,"message":"OK","result":{ … výsledek …}}`

If code is different to “0” than message contains error description.

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/

2.7.1. GET method

2.7.1.1. Get the state code list

Request:

```
{"resultAs":"json","list":"enumState"}
```

Answer:

```
{"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 can take values “json” or “csv” – it the value is set to “csv” result is table in json format.

2.7.1.2. Load existing exceptions

Request:

```
{"resultAs":"json","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”.
Parameters *„productCode“, „batch“* are optional.

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

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

2.7.1.3. Verify pack for the exception

Request:

{"resultAs":"json","list":"verify","productCode":"08594158891136","batch":"1"}

Answer:

{"status":"ok","code":0,"message":"OK","result":{"isException":true,"info":{"id":41,"productCode":"08594158891136","batch":null,"stateId":1,"state":"Nov\u00fd"}}}

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

In answer:
isException: true/false – product or pack falls under the exception
info – if isException is “true”, than the array contains exception description

2.7.2. POST method

Request (single item):

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

Request (batch insert via csv file):

{"validity":null,"state":"NO","csv":"zCIsIsSMbMOhbmVrIDU3IGtvZGV4dSIsIklEIGRyxb5pdGVsZSByZWdpc3RyYWNlIChNQUggSUQpIiRyxb5pdGVsZSByZWdpc3RyYWNlIChNQUgpIiwiQWRyZXNhIGRyxb5pdGVsZSByZWdpc3RyYWNlIiwiTc3RvIGRyxb5pdGVsZSByZWdpc3RyYWNlIiwiUFPEjCBkcsW+aXRlbGU0cmFjZSIsIlplbcSbIGRyxb5gaahjfkleuHGAFlgheghv4fdb2xxYVNS5fAS5gdrgpdGVsZSByZWdpc3RyYWNlIiwiw5pkYWplIG8gZGlzdHJpYnV0b3JvdmkiDQo="}

Answer:

{"status":"ok","code":0,"message":"OK","result":{"products":[{"lineNo":1,"productCode":"0123456789","batch":"123456","serialNumber":"0123","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.
Parameter “state” is default value for the set alert state (if state is not defined in the csv file).
Answer can contain more rows for the batch insert, where ID or reason is described for the each not inserted row.
Parameter “csv” is csv file encoded in base64 format. Structure is described in the web interface section “Input data").

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

2.7.3. 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 COMPARED TO AMS 4.2

1. New code list "Status type"

For "End user" role only. It is a list of indications for the user what to do with the pack in a given situation. Each alert status is assigned exactly one "Status_Type".

2. Expansion of the "Alert Status" code list

Typestate and **typestatedescription** fields are added for the "End user" role (binding to the new code "Status_type").

3. Adding error states

Added error states - codes **34-37.**

4. Loading the counter for A1 alerts

Added functionality for end users to find the **value of the current counter** to generate a unique identification of level 3 exceptions.

5. Loading A1 Alerts

Added function for end users to retrieve the **current list of level 3 exceptions**.

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